Class: II MBA Course Name: Enterprise resource planning
Course code: 17MBAPS401C Semester: IV Year: 2017-2019

UNIT 1

Introduction-Purpose of database system-View of data-Data Models-Database Languages-Transaction Management-Storage Management-Database Administrator-Database Users-System Structure

Introduction to ERP

Short for **e**nterprise **r**esource **p**lanning, a business management system that integrates all facets of the business, including planning, manufacturing, sales, and marketing.

As the ERP methodology has become more popular, software applications have emerged to help business managers implement ERP in business activities such as inventory control, order tracking, customer service, finance and human resources.

Enterprise resource planning's true ambition is to integrate all departments and functions across a company onto a single computer system that can serve all those different departments' particular needs.

Thus ERP attempts to integrate all departments and functions across a company onto a single computer system that can serve all those different departments' particular needs.

ERP systems are IT systems which are meant to serve all the IT needs of a manufacturing company. ERP stands for "Enterprise Resource Planning".

Why ERP?

1. To Enhance Profitability:

a) Increase in sales
b) /or Reduce Procurement Cost
c)

2. for Healthy Operations:

a) Integration of Systems across the Functional Departments in a Company as well as across the Enterprise as a Whole.
b) Better Customer Service.
c) Introduction of Latest Technologies as and when the are ready for the Industry acceptance
d) Expertise database
e) Avoids data redundancy

3. Competition in the Market:

a) Manufacturing Challenges.
b) Manufacturing Globally.

c) Distribution network spread.
d) New Product introduction.
e) Lower manufacturing lead time.
f) Focus on industry markets.
g) Satisfying the needs of customers.
h) Develop specific business methods and processes.
i) Integration with third party products.

4. Demands on the Industry:

a) Better products at lower costs
b) Tough competition
c) Need to analyze costs / revenues on a product or customer basis
d) Flexibility to respond to changing business requirements
e) More informed management decision making

5. Solving the Problems:

a) Unable to get accurate, timely information
b) Applications not complete for existing business practices
c) Modifications are time consuming or not feasible

Solving these Problems will the company ahead of competition

The advantages of ERP

Installing an ERP system has many advantages -both direct and indirect.

The direct advantages include improved efficiency, information integration for better decision making, faster response time to customer queries, etc. The indirect benefits include better corporate image, improved customer goodwill, customer satisfaction, and so on.

The following are some of the direct benefits of an ERP system:

1. Business Integration
2. Flexibility
3. Better Analysis and Planning Capabilities
4. Use of Latest Technology.

1. Business Integration: The first and most important advantage lies in the promotion of integration. The reason why ERP packages are considered to the integrated, is the automatic data updating (automatic data exchange among applications) that is possible among the related business components.

Since conventional company information systems were aimed at the optimization of independent business functions in business units, almost all were weak in terms of the communication and integration of information that transcended the different business functions.

In the case of large companies in particular, the timing of system construction and directives differs for each product and department/ function and sometimes, they are disconnected. For this reason, it has become an obstacle in the shift to new product and business classification.

In the case of ERP packages, the data of related business functions is also automatically updated at the time a transaction occurs. For this reason, one is able to grasp business details in real time, and carry out various types of management decisions in a timely manner, based on that information.

2. Flexibility: The second advantage of the ERP packages is their flexibility. Different languages, currencies, accounting standards and so on can be covered in one system, and functions that comprehensively manage multiple locations of a company can be packaged and implemented automatically. To cope with company globalization and system unification, this flexibility is essential and one can say that it has major advantages, not simply for development and maintenance, but also in terms of management.

3. Better Analysis and planning Capabilities: Yet another advantage is the boost to the planning functions. By enabling the comprehensive and unified management of related business and its data, it becomes possible to fully utilize many types of decision support systems and simulation functions. Furthermore, since it becomes possible to carry out, flexible and in real time, the filing and analysis of data from a variety of dimensions, one is able to give the decision-makers the information they want; thus enabling them to make better and informed decisions.

4. Use of Latest Technology: the fourth advantage is the utilization of the latest development in information Technology (IT). The ERP vendors were quick to realize that in order to grow and to sustain that growth; they had to embrace the latest developments in the field of information technology. Therefore, they quickly adapted their systems to take advantage of the latest technologies like open systems, client/ server technology, Internet/Intranet, CALS (Computer- Aided Acquisition and Logistics Support), electronic-commerce, etc.

It is this quick adaptation to the latest changes in the Information Technology that makes the flexible adaptation to changes in future business environments possible. It is this flexibility that makes the incorporation of the latest technology possible during system customization, maintenance and expansion phases.

What is that ERP enables?

a) Systematic Look into your Systems & procedures
b) Optimizing the processes
c) Enables you to adapt yourself to new technologies
d) Discipline across the functions

Problems Taken Care of by the ERP:

a) Availability check at the time of Accepting Sales order
b) On-line Material Status & Shortages
c) Productivity Enhancements
d) Material Planning

e) Customer Service
f) Cash Management
g) Inventory
h) Quality

What drives ERP?

- **Business drives**
  a) Customer Satisfaction
  b) Business Development - new areas, products, services
  c) Ability to face competition
  d) Efficient processes required to push the company to top gear
- **IT drives**
  a) Present Software does not meet business needs
  b) Legacy systems difficult to maintain
  c) Obsolete hardware/software difficult to maintain

Drivers

The market for ERP however does not sound so depressing. Companies still have growth avenues which include:

Less penetrated modules within the ERP suite, both horizontal and vertical. The new horizontal areas include E-commerce, Customer relationship management, Supply chain management, plant maintenance, field service, data warehousing, product data management, service contract management, warehousing & distribution,, transportation management etc. Among the vertical application are industries such as retail, utilities, insurance, and government organizations.

The mid market segment presents immense opportunities. However, the margins from SMEs will be far below that from the larger players.

Another problem that the SMEs present is the low transaction (order) size and the difficulty of reaching out to these players. Also they are relatively less sophisticated on the technology side.

Another major demand driver will be the e-commerce wave. As more and more company move towards e-commerce it becomes necessary to implement ERP solutions.

Inhibitors or hurdles for the growth of ERP market:

After posting significant gains for 3-4 years the ERP market may be heading towards a slowdown. Except for SAP most other players have witnessed a slowdown in revenue growth.

The main constraints to growth for the sector can be classified as:

a) Saturation of the certain horizontal applications including Finance and accounting, MRP etc which accounted for nearly 45% of the ERP revenues during 1998.
b)

c) Saturation of large customers. Most of the Fortune 500 companies and companies having revenues over $1bn have already implemented ERP.
d) Though the medium enterprises provide a good opportunity for growth, pricing for these companies will have to be highly competitive and margins may come under pressure. Thus smaller players who have a cost advantage will have an edge over the others.

ERP in India

Until recently Indian organizations were in a sellers market and operating in a regulated environment. They grew by managing the environment, rather than innovating and improving internal efficiencies. The customer was taken for granted and quality was available only at a

premium. With globalization and gradual lifting of regulation, there is a paradigm shift in running the business.

Indian companies now need to increase customer focus, improve speed of delivery, be cost competitive and provide value for money (improved quality at lower price). Indian companies therefore need to implement ERP systems for improving their business processes and becoming more competitive in the global environment. Though ERP implementation is costly and time consuming, it has several benefits which will help recover these costs in the long run.

According to NASSCOM, during the year 1998-99, the Indian ERP market has been estimated at R5200mn compared to Rs2800mn in the previous year ie a growth of 85%yoy. The growth in the export market was far higher and more than doubled during the same time period. According to the NASSCOM, by the end of FY2001-02, the total Indian ERP market is expected to multiply by nearly 4 times and reach Rs65bn compared to Rs13.4bn in 1998-99.

Purpose of Database Systems

1. To see why database management systems are necessary, let's look at a typical ``file-processing system'' supported by a conventional operating system.

   The application is a savings bank:

   - Savings account and customer records are kept in permanent system files.
   - Application programs are written to manipulate files to perform the following tasks:
     - Debit or credit an account.
     - Add a new account.
     - Find an account balance.
     - Generate monthly statements.
2. Development of the system proceeds as follows:
   - New application programs must be written as the need arises.
   - New permanent files are created as required.

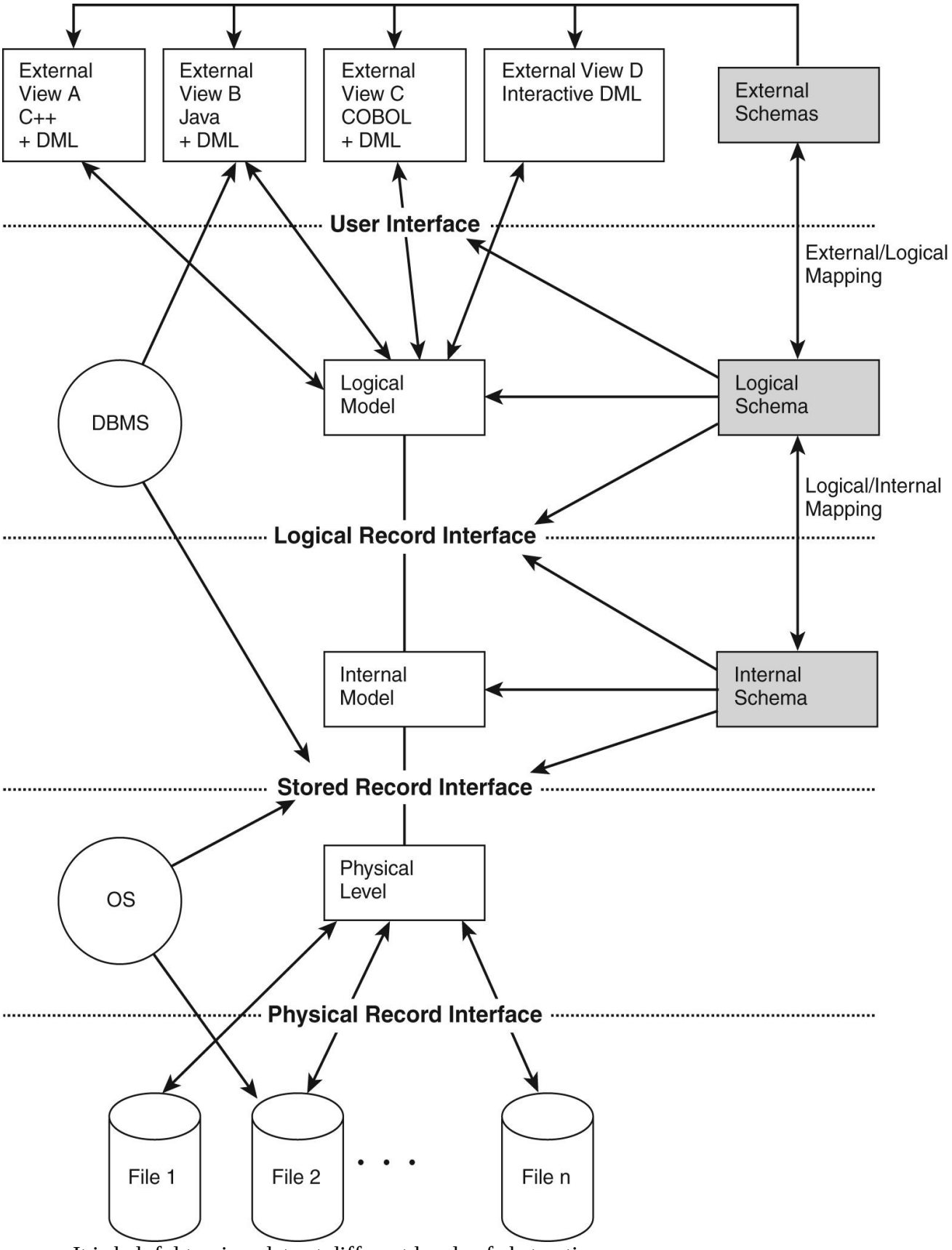
- **but** over a long period of time files may be in different formats, and
- Application programs may be in different languages.

3. So we can see there are problems with the straight file-processing approach:
   - Data redundancy and inconsistency
     - Same information may be duplicated in several places.
     - All copies may not be updated properly.
   - Difficulty in accessing data
     - May have to write a new application program to satisfy an unusual request.
     - E.g. find all customers with the same postal code.
     - Could generate this data manually, but a long job...
   - Data isolation
     - Data in different files.
     - Data in different formats.
     - Difficult to write new application programs.
   - Multiple users
     - Want concurrency for faster response time.
     - Need protection for concurrent updates.
     - E.g. two customers withdrawing funds from the same account at the same time - account has $500 in it, and they withdraw $100 and $50. The result could be $350, $400 or $450 if no protection.
   - Security problems
     - Every user of the system should be able to access only the data they are permitted to see.
     - E.g. payroll people only handle employee records, and cannot see customer accounts; tellers only access account data and cannot see payroll data.
     - Difficult to enforce this with application programs.
   - Integrity problems
     - Data may be required to satisfy constraints.
     - E.g. no account balance below $25.00.
     - Again, difficult to enforce or to change constraints with the file-processing approach.

These problems and others led to the development of **database management systems**

Data and Related Structures

Data are actually stored as bits, or numbers and strings, but it is extremely difficult to work with the variety and complexity of data at this level.



It is helpful to view data at different levels of abstraction.

Schema:

- This is the term for a description of the data organizatinon at some level. Each level has its own schema.

We will be concerned with three forms of schemas:

- **internal or physical**
- **conceptual or logical**
- **external or user view**

Internal Data Level

The **physical schema of the internal level** describes details of how data is stored: files, indices, etc. on the random access disk system. It also typically describes the record layout of files and type of files (hash, b-tree, flat).

Early applications (1960's) only worked at this level - explicitly dealt with these internal details. E.g., minimizing physical distances between related data and organizing the data structures within the file (blocked records, linked lists of blocks, etc.)

Problem:

- Routines are hardcoded to deal with physical representation.
- Changes to data structures are difficult to make.
- Application code becomes complex since it must deal with details.
- Rapid implementation of new features very difficult.

Conceptual Data Level

Also referred to as the **Logical** level when the conceptual level is implemented to a particular database architecture.

Hides storage details of the internal/physical level.

- In the relational model, the conceptual schema presents data as a set of tables.

The DBMS automatically maps data access between the logical to internal/physical schemas .

- Physical/internal schema can be changed without changing application: e.g. we may add or remove an index
- DBMS must change mapping from conceptual to physical.
- Referred to as **physical data independence**.

External Data Level

An external schema specifies a **view** of the data in terms of the conceptual level. It is tailored to the needs of a particular category of users. Portions of stored data should not be seen by some users and begins to implement a level of security and simplifies the view for these users.

In the relational model, the **external schema** also presents data as a set of relations.

Examples:

- Students should not see faculty salaries.
- Faculty should not see billing or payment data.

Information that can be derived from stored data might be viewed as if it were stored in that manner.

- GPA not stored, calculated when needed.

Applications are written in terms of an external schema. The external view is computed when accessed. It is not stored. Different external schemas can be provided to different categories of users. Translation from external level to conceptual level is done automatically by DBMS at run time. The conceptual schema can be changed without changing application:

- Mapping from external to conceptual must be changed.
- Referred to as **conceptual data independence**.
- This is a first level of security that can be imposed on the various users of the system.

Data Modeling

Schema: description of data at some level (e.g., tables, attributes, constraints, domains)

Model: tools and languages for describing:

- Conceptual/logical and external schema described by the data definition language (DDL)
- Integrity constraints, domains described by DDL
- Operations on data described by the data manipulation language (DML)
- Directives that influence the physical schema (affects performance, not semantics) are described by the storage definition language (SDL)

DBMS Database Models

A model is a representation of reality, 'real world' objects and events, associations. It is an abstraction that concentrates on the essential, inherent aspects an organization and ignores the

accidental properties. A data model represents the organization itself. It should provide the basic concepts and notations that will allow database designers and end users unambiguously and accurately to communicate their understanding of the organizational data.

Data Model can be defined as an integrated collection of concepts for describing and manipulating data, relationships between data, and constraints on the data in an organization.

A data model comprises of three components:

• A structural part, consisting of a set of rules according to which databases can be constructed.

• A manipulative part, defining the types of operation that are allowed on the data (this includes the operations that are used for updating or retrieving data from the database and for changing the structure of the database).

• Possibly a set of integrity rules, which ensures that the data is accurate.

The purpose of a data model is to represent data and to make the data understandable. There have been many data models proposed in the literature. They fall into three broad categories:

•Object Based Data Models
•Physical Data Models
• Record Based Data Models

The object based and record based data models are used to describe data at the conceptual and external levels, the physical data model is used to· describe data at the internal level.

Object Based Data Models

Object based data models use concepts such as entities, attributes, and relationships. An entity is a distinct object (a person, place, concept, and event) in the organization that is to be represented in the database. An attribute is a property that describes some aspect of the object that we wish to record, and a relationship is an association between entities.

Some of the more common types of object based data model are:

• Entity-Relationship

• Object Oriented

• Semantic

• Functional

The Entity-Relationship model has emerged as one of the main techniques for modeling database design and forms the basis for the database design methodology. The object oriented data model extends the definition of an entity to include, not only the attributes that describe the state of the object but also the actions that are associated with the object, that is, its behavior. The object is said to encapsulate both state and behavior. Entities in semantic systems represent the equivalent of a record in a relational system or an object in an OO system but they do not include behaviour (methods). They are abstractions 'used to represent real world (e.g. customer) or conceptual (e.g. bank account) objects. The functional data

model is now almost twenty years old. The original idea was to' view the database as a collection of extensionally defined functions and to use a functional language for querying the database.

Physical Data Models

Physical data models describe how data is stored in the computer, representing information such as record structures, record ordering, and access paths. There are not as many physical data models as logical data models, the most common one being the Unifying Model.

Record Based Logical Models

Record based logical models are used in describing data at the logical and view levels. In contrast to object based data models, they are used to specify the overall logical structure of the database and to provide a higher-level description of the implementation. Record based models are so named because the database is structured in fixed format records of several types. Each record type defines a fixed number of fields, or attributes, and each field is usually of a fixed length.

The three most widely accepted record based data models are:

• Hierarchical Model

• Network Model

• Relational Model

The relational model has gained favor over the other two in recent years. The network and hierarchical models are still used in a large number of older databases

A Database model defines the logical design and structure of a database and defines how data will be stored, accessed and updated in a database management system. While the **Relational Model** is the most widely used database model, there are other models too:

- Hierarchical Model
- Network Model
- Entity-relationship Model
- Relational Model

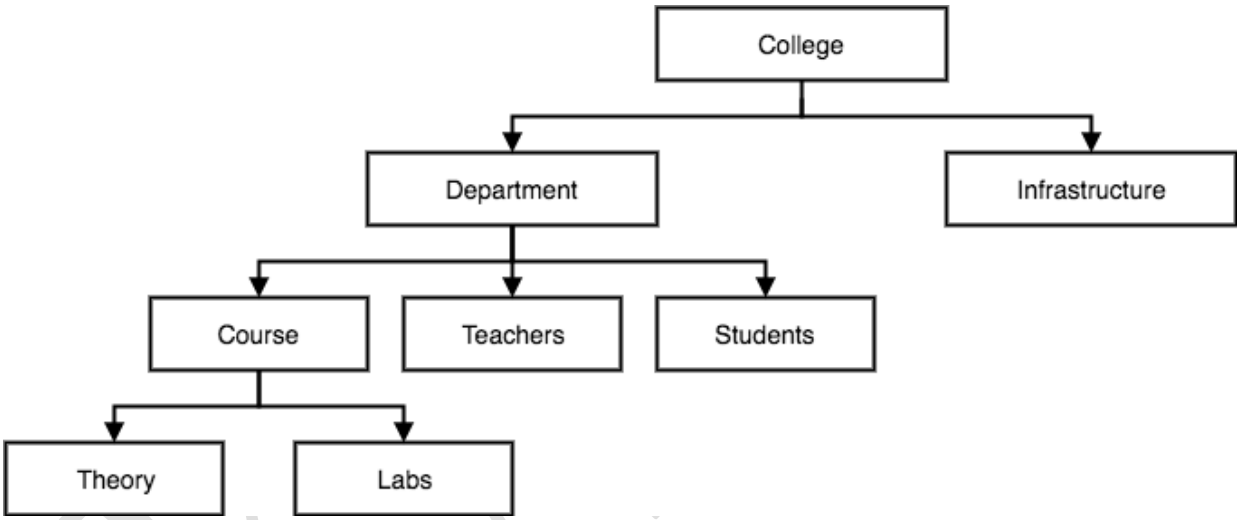
Hierarchical Model

This database model organises data into a tree-like-structure, with a single root, to which all the other data is linked. The heirarchy starts from the **Root** data, and expands like a tree, adding child nodes to the parent nodes.

In this model, a child node will only have a single parent node.

This model efficiently describes many real-world relationships like index of a book, recipes etc.

In hierarchical model, data is organised into tree-like structure with one one-to-many relationship between two different types of data, for example, one department can have many courses, many professors and of-course many students.

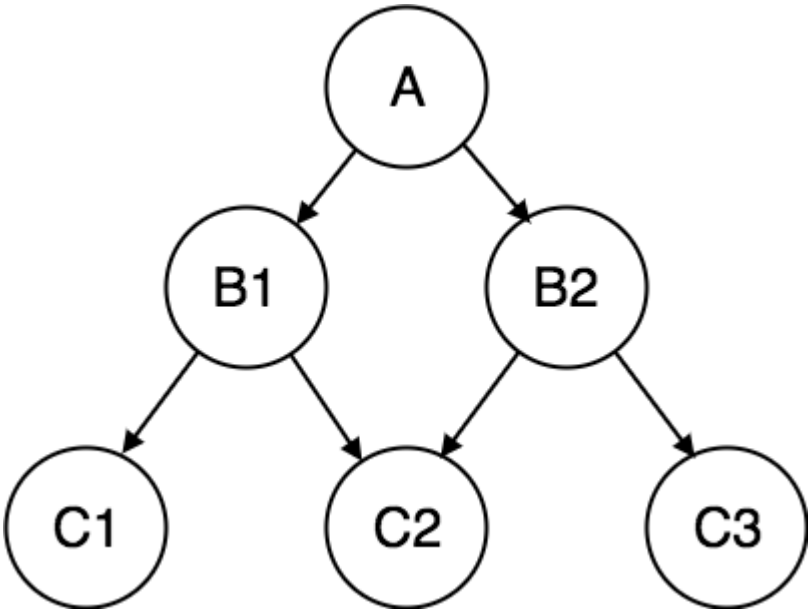


Network Model

This is an extension of the Hierarchical model. In this model data is organised more like a graph, and are allowed to have more than one parent node.

In this database model data is more related as more relationships are established in this database model. Also, as the data is more related, hence accessing the data is also easier and fast. This database model was used to map many-to-many data relationships.

This was the most widely used database model, before Relational Model was introduced.



Entity-relationship Model

In this database model, relationships are created by dividing object of interest into entity and its characteristics into attributes.

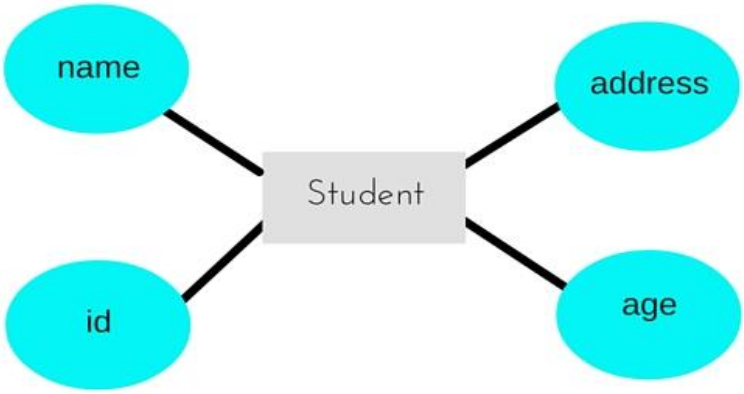
Different entities are related using relationships.

E-R Models are defined to represent the relationships into pictorial form to make it easier for different stakeholders to understand.

This model is good to design a database, which can then be turned into tables in relational model(explained below).

Let's take an example, If we have to design a School Database, then **Student** will be an **entity** with **attributes** name, age, address etc. As **Address** is generally complex, it can be another **entity** with **attributes** street name, pincode, city etc, and there will be a relationship between them.

Relationships can also be of different types. To learn about E-R Diagrams in details, click on the link.



Relational Model

In this model, data is organised in two-dimensional **tables** and the relationship is maintained by storing a common field.

This model was introduced by E.F Codd in 1970, and since then it has been the most widely used database model, infact, we can say the only database model used around the world.

The basic structure of data in the relational model is tables. All the information related to a particular type is stored in rows of that table.

Hence, tables are also known as **relations** in relational model.

In the coming tutorials we will learn how to design tables, normalize them to reduce data redundancy and how to use Structured Query language to access data from tables.

.

Class: II MBA Course Name: Enterprise resource planning
Course code: 17MBAPS401C Semester: IV Year: 2017-2019

student_id	name	age
1	Akon	17
2	Bkon	18
3	Ckon	17
4	Dkon	18

subject_id	name	teacher
1	Java	Mr. J
2	C++	Miss C
3	C#	Mr. C Hash
4	Php	Mr. P H P

student_id	subject_id	marks
1	1	98
1	2	78
2	1	76
3	2	88

DATA DEFINITION LANGUAGE (DDL)

A DBMS must provide appropriate languages and interfaces for each category of users to express database queries and updates. Database Languages are used to create and maintain database on computer. There are large numbers of database languages like Oracle, MySQL, MS Access, dBase, FoxPro etc. SQL statements commonly used in Oracle and MS Access can be categorized as data definition language (DDL), data control language (DCL) and data manipulation language (DML).

It is a language that allows the users to define data and their relationship to other types of data. It is mainly used to create files, databases, data dictionary and tables within databases.

It is also used to specify the structure of each table, set of associated values with each attribute, integrity constraints, security and authorization informationfor each table and physical storage structure of each table on disk.

The following table gives an overview about usage of DDL statements in SQL

S.No	Need and Usage	The SQL DDL statement
1	Create schema objects	CREATE
2	Alter schema objects	ALTER
3	Delete schema objects	DROP
4	Reneme schema objects	RENAME

Data Manipulation Language (DML)

It is a language that provides a set of operations to support the basic data manipulation operations on the data held in the databases. It allows users to insert, update, delete and retrieve data from the database. The part of DML that involves data retrieval is called a query language.

The following table gives an overview about the usage of DML statements in SQL:

S. No	Need and Usage	The SQL DML statement
1	Remove rows from tables or views	DELETE
2	Add new rows of data into table or view	INSERT
3	Retrieve data from one or more tables	SELECT
4	change column values In existing rows of a table or view	UPDATE

In practice, the data definition and data manipulation languages are not two separate languages. Instead they simply form parts of a single database language such as Structured Query Language (SQL). SQL represents combination of DDL and DML, as well as statements for constraints specification and schema evaluation.

Transaction management:

A transaction can be defined as a group of tasks. A single task is the minimum processing unit which cannot be divided further.

Let's take an example of a simple transaction. Suppose a bank employee transfers Rs 500 from A's account to B's account. This very simple and small transaction involves several low-level tasks.

A's Account

```
Open_Account(A)
Old_Balance = A.balance
New_Balance = Old_Balance - 500
A.balance = New_Balance
Close_Account(A)
```

B's Account

```
Open_Account(B)
Old_Balance = B.balance
New_Balance = Old_Balance + 500
B.balance = New_Balance
Close_Account(B)
```

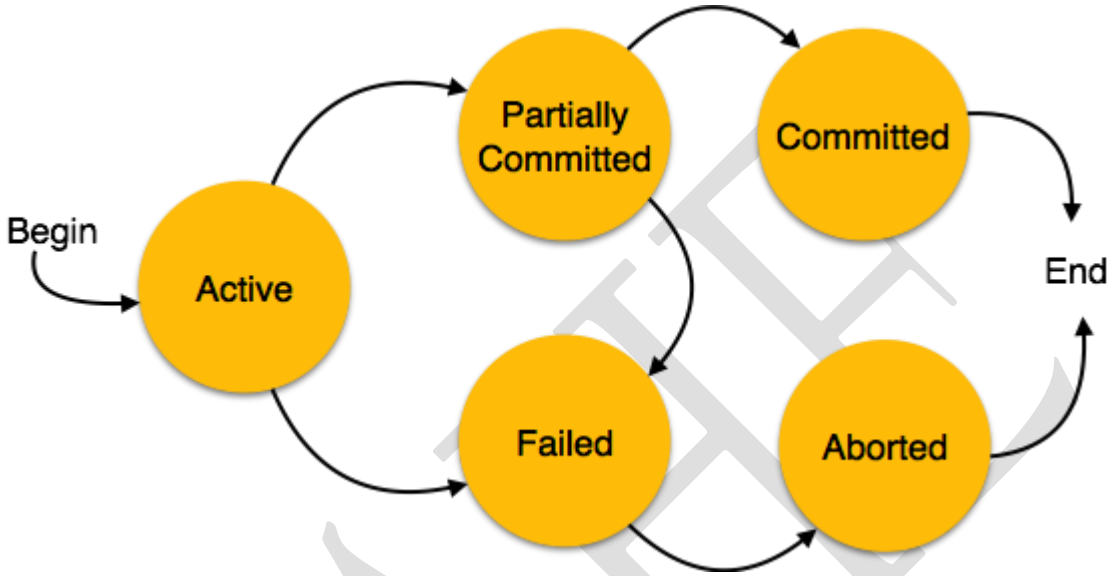
ACID Properties

A transaction is a very small unit of a program and it may contain several lowlevel tasks. A transaction in a database system must maintain **A**tomicity, **C**onsistency, **I**solation, and **D**urability − commonly known as ACID properties − in order to ensure accuracy, completeness, and data integrity.

- **Atomicity** − This property states that a transaction must be treated as an atomic unit, that is, either all of its operations are executed or none. There must be no state in a database where a transaction is left partially completed. States should be defined either before the execution of the transaction or after the execution/abortion/failure of the transaction.
- **Consistency** − The database must remain in a consistent state after any transaction. No transaction should have any adverse effect on the data residing in the database. If the database was in a consistent state before the execution of a transaction, it must remain consistent after the execution of the transaction as well.
- **Durability** − The database should be durable enough to hold all its latest updates even if the system fails or restarts. If a transaction updates a chunk of data in a database and commits, then the database will hold the modified data. If a transaction commits but the system fails before the data could be written on to the disk, then that data will be updated once the system springs back into action.
- **Isolation** − In a database system where more than one transaction are being executed simultaneously and in parallel, the property of isolation states that all the transactions will be carried out and executed as if it is the only transaction in the system. No transaction will affect the existence of any other transaction.

States of Transactions

A transaction in a database can be in one of the following states −



STORAGE MAMAGEMENT:

Storage Manager:

A storage manager is a program module which is responsible for storing, retrieving and updating data in the database.

Following are the components of the storage manager;

1. **Authorization and Integrity Manager:** It tests the integrity constraints and checks the authorization of users to access data.
2. **Transaction Manager:** It ensures that no kind of change will be brought to the database until a transaction has been completed totally.

3. **File Manager:** It manages the allocation of space on disk storage and the data structures used to represent information stored on disk.

4. **Buffer Manager:** It decides which data is in need to be cached in main memory and then fetch it up in main memory. This is very important as it defines the speed in which the database can be used.

DATABASE Administrator:

A **Database Administrator, Database Analyst or Database Developer** is the person **responsible for managing the information** within an organization. As most companies continue to experience inevitable growth of their databases, these positions are probably the most solid within the IT industry.

The DBA has many different responsibilities, but the overall goal of the DBA is to **keep the server up at all times** and to provide users with access to the required information when they need it. The DBA makes sure that the **database is protected and that any chance of data loss is minimized**.

A **DBA can be a programmer** who, by default or by volunteering, took over the responsibility of **maintaining a SQL Server** during project development and enjoyed the job so much that he switched.

A **DBA can be a system administrator** who was given the added responsibility of maintaining a SQL Server. DBAs can even come from unrelated fields, such as accounting or the help desk, and switch to Information Systems to become DBAs. To start your journey to becoming a Microsoft SQL Server DBA,

DBA Responsibilities

The following sections examine the responsibilities of the database administrator and how they translate to various Microsoft SQL Server tasks.

Installing and Upgrading an SQL Server

The DBA is responsible for installing SQL Server or upgrading an existing SQL Server. In the case of upgrading SQL Server, the DBA is responsible for ensuring that if the upgrade is not successful, the SQL Server can be rolled back to an earlier release until the upgrade issues can be resolved.

The DBA is also responsible for applying SQL Server service packs. A service pack is not a true upgrade, but an installation of the current version of software with various bug fixes and patches that have been resolved since the product's release.

Monitoring the Database Server's Health and Tuning Accordingly

Monitoring the health of the database server means making sure that the following is done:

- The server is running with optimal performance.
- The error log or event log is monitored for database errors.
- Databases have routine maintenance performed on them, and the overall system has periodic maintenance performed by the system administrator.

Using Storage Properly

SQL Server 2000 enables you to automatically grow the size of your databases and transaction logs, or you can choose to select a fixed size for the database and transaction log. Either way, maintaining the proper use of storage means monitoring space requirements and adding new storage space (disk drives) when required.

Performing Backup and Recovery Duties

Backup and recovery are the DBA's most critical tasks; they include the following aspects:

- Establishing standards and schedules for database backups
- Developing recovery procedures for each database
- Making sure that the backup schedules meet the recovery requirements

Managing Database Users and Security

With SQL Server 2000, the DBA works tightly with the Windows NT administrator to add user NT logins to the database. In non-NT domains, the DBA adds user logins. The DBA is also responsible for assigning users to databases and determining the proper security level for each user. Within each database, the DBA is responsible for assigning permissions to the various database objects such as tables, views, and stored procedures.

Working with Developers

It is important for the DBA to work closely with development teams to assist in overall database design, such as creating normalized databases, helping developers tune queries, assigning proper indexes, and aiding developers in the creation of triggers and stored procedures.

In the SQL Server 2000 environment, a good DBA will show the developers how to use and take advantage of the SQL Server Index Tuning Wizard and the SQL Server profiler.

Establishing and Enforcing Standards

The DBA should establish naming conventions and standards for the SQL Server and databases and make sure that everyone sticks to them.

Transferring Data

The DBA is responsible for importing and exporting data to and from the SQL Server. In the current trend to downsize and combine client/server systems with mainframe systems and Web technologies to create Enterprise systems, importing data from the mainframe to SQL Server is a common occurrence that is about to become more common with the SQL Server 2000 Data Transformation Services. Good DTS DBAs will be in hot demand as companies struggle to move and translate legacy system to Enterprise systems.

Replicating Data

SQL Server version 2000 has many different replication capabilities such as Merge replication (2-way disconnected replication) and queued replication. Managing and setting up replication topologies is a big undertaking for a DBA because of the complexities involved with properly setting up and maintaining replication.

Data Warehousing

SQL Server 2000 has substantial data warehousing capabilities that require the DBA to learn an additional product (the Microsoft OLAP Server) and architecture. Data warehousing provides new and interesting challenges to the DBA and in some companies a new career as a warehouse specialist.

Scheduling Events

The database administrator is responsible for setting up and scheduling various events using Windows NT and SQL Server to aid in performing many tasks such as backups and replication.

Providing 24-Hour Access

The database server must stay up, and the databases must always be protected and online. Be prepared to perform some maintenance and upgrades after hours. Also be prepared to carry that dreaded beeper. If the database server should go down, be ready to get the server up and running. After all, that's your job.

Learning Constantly

To be a good DBA, you must continue to study and practice your mission-critical procedures, such as testing your backups by recovering to a test database. In this business, technology changes very fast, so you must continue learning about SQL Server, available client/servers, and database design tools. It is a never-ending process.

The DBA should posses the following skills

(1) A good knowledge of the operating system(s)

(2) A good knowledge of physical database design

(3) Ability to perform both Oracle and also operating system performance monitoring and the necessary adjustments.

(4) Be able to provide a strategic database direction for the organization.

(5) Excellent knowledge of Oracle backup and recovery scenarios.

(6) Good skills in all Oracle tools.

(7) A good knowledge of Oracle security management.

(8) A good knowledge of how Oracle acquires and manages resources.

(9) Sound knowledge of the applications at your site.

(10) Experience and knowledge in migrating code, database changes, data and

Menus through the various stages of the development life cycle.

(11) A good knowledge of the way Oracle enforces data integrity.

(12) A sound knowledge of both database and program code performance tuning.

(13) A DBA should possess a sound understanding of the business.

(14) A DBA should have sound communication skills with management, development teams, vendors, systems administrators and other related service providers.

Application Programmers are computerprofessionals interacting with the system through DML calls embedded in a program written in a host language (e.g. C, PL/1, Pascal): These programs are called Application Programs. The DML Precompiled converts DML calls (prefaced by a special character like $, #, etc.) to normal procedure calls in a host language.

The host language compiler then generates the object code. Some special types of programming languages combine Pascal-like control structures with control structures for the manipulation of a database. These are sometimes called Fourth-Generation Languages. They often include features which to generate forms and display data.

• **Sophisticated Users** interact with the system without writing programs :They form requests by writing queries in a database query language. These are submitted to a **query processor** that breaks a DML statement down into instructions for the database manager module.
• **Specialized Users** are sophisticated users writing special database application programs. These may be CADD systems, knowledge-based and expert systems, complex data systems (audio/video), etc.

• **Naive Users** are unsophisticated users who interact with the system by using permanent application programs (e.g. automated teller machine).

SYSTEM STRUCTURE:

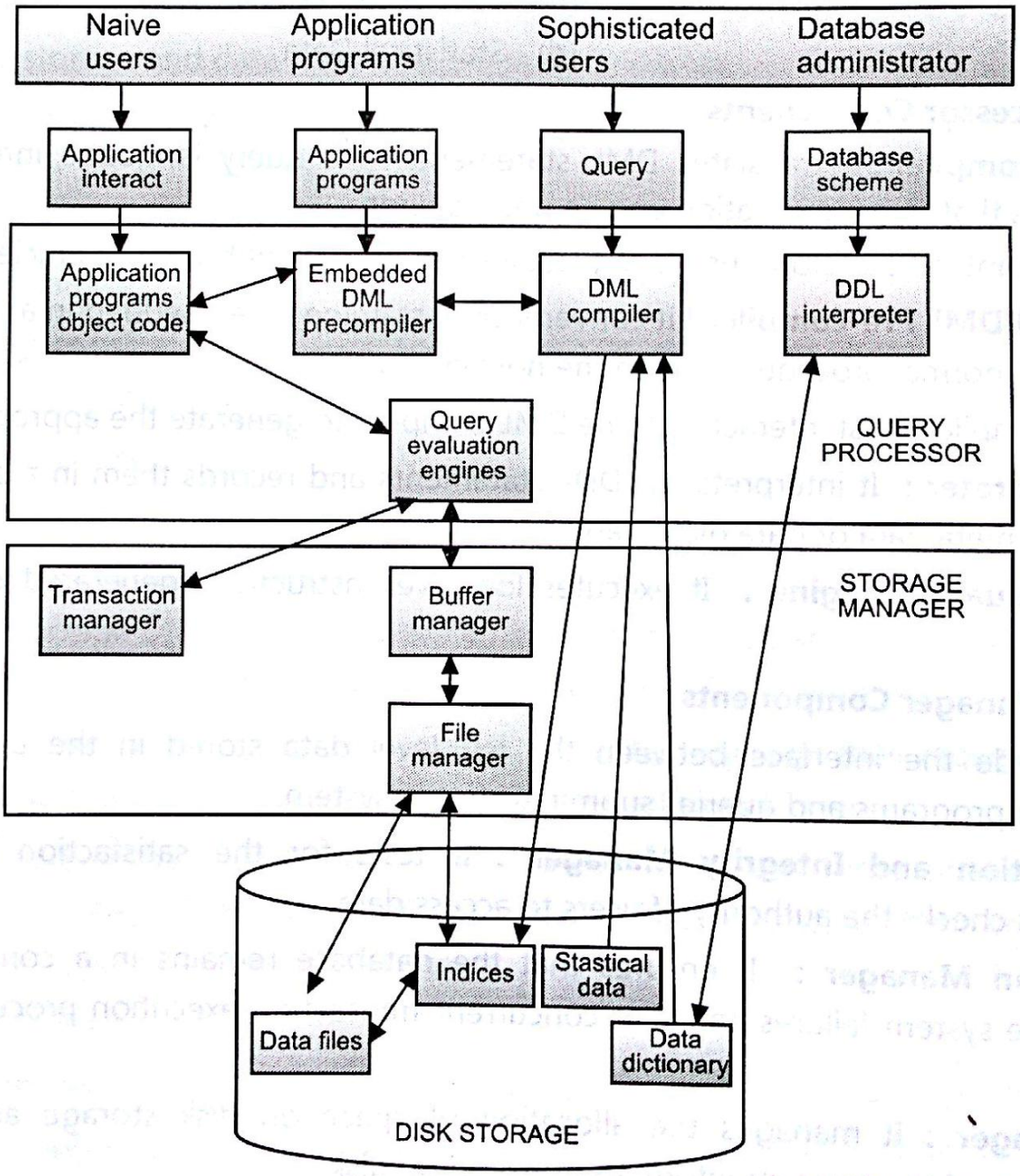
Database users are the one who really use and take the benefits of database. There will be different types of users depending on their need and way of accessing the database.

1. **Application Programmers -** They are the developers who interact with the database by means of DML queries. These DML queries are written in the application programs like C, C++, JAVA, Pascal etc. These queries are converted into object code to communicate with the database. For example, writing a C program to generate the report of employees who are working in particular department will involve a query to fetch the data from database. It will include a embedded SQL query in the C Program.
2. **Sophisticated Users -** They are database developers, who write SQL queries to select/insert/delete/update data. They do not use any application or programs to

3. request the database. They directly interact with the database by means of query language like SQL. These users will be scientists, engineers, analysts who thoroughly study SQL and DBMS to apply the concepts in their requirement. In short, we can say this category includes designers and developers of DBMS and SQL.
4. **Specialized Users -** These are also sophisticated users, but they write special database application programs. They are the developers who develop the complex programs to the requirement.
5. **Stand-alone Users -** These users will have stand –alone database for their personal use. These kinds of database will have readymade database packages which will have menus and graphical interfaces.

6. **Native Users -** these are the users who use the existing application to interact with the database. For example, online library system, ticket booking systems, ATMs etc which has existing application and users use them to interact with the database to fulfill their requests.

Class: II MBA Course Name: Enterprise resource planning
Course code: 17MBAPS401C Semester: IV Year: 2017-2019



System structure

	KARPAGAM ACADEMY OF HIGHER EDUCATION					
	Department of Management					
	Unit 1- Multiple Choice Questions- Each Question Carry ONE Mark					
S.no	**Question**	**Option 1**	**Option 2**	**Option 3**	**Option 4**	**Answer**
1	A database management system (DBMS) is_____	Collection of interrelated data	Collection of programs to access data	Collection of data describing one particular enterprise	Collection of dats in the table	**Collection of interrelated data**
2	_____ is not a level of data abstraction	Physical level	Critical level	Logical level	View level	**Critical level**
3	Disadvantages of File systems to store data is_____	Data redudancy and inconsistency	Difficulty in accessing data	Data isolation	Data abstraction	**Data isolation**
4	_____ is not a store managers component.	Transaction manager	Logical Manager	Buffer manager	File manager	**Logical Manager**
5	Data manipulation language enables users to_____	Retrievel of information stored in database	Insertion of new information into the database	Deletion of information from the database	Update the informtion	**Retrievel of information stored in database**
6	_____ is not a schema.	Database schema	Physical schema	Critical schema	Logical schema	**Critical schema**
7	_____ is called database language	DDL	Physical schema	Tables	Schema create	**DDL**
8	_____ is not the function of DBA	Network maintenance	Routine maintenance	Schema definition	Authorization of data access	**Network maintenance**
9	_____Represents a relationship among a set of values	A row	A table	A Field	A Column	**A row**
10	Column Header is refered as_____	Table	Relation	Attributes	Domain	**Attributes**
11	_____ is not the	Deletion	Insertion	Sorting	Updating	**Sorting**

	modification of the database					
12	Who Proposed the relational model?	Bill gates	E.F. Cold	Herman Hollerith	Charles babbage	**E.F. Cold**
13	Set of premitted values of each attribute is called_____ __	Domain	Tuple	Relation	Schema	**Domain**
14	Which of the following in true regarding Null Value?	Null = 0	Null 0	Null 0	Null0	**Null 0**
15	Logical design of database is called_____ ____	Database Instance	Database Snapshot	Database Schema	Database Attribute	**Database Schema**
16	Snapshot of the dta in the database at a given instant of time is called	Database Schema	Database Instance	Database Snapshot	Database Attribute	**Database Instance**
17	_____ is correct regarding Aggregate function.	it takes a list of values and return a single values as result	it takes a list of values and return a list of values as result	it takes a single value and returns a list of values as result	it takes a single value and returns a single value as result	**it takes a list of values and return a single values as result**
18	The Primary key must be_____ __	Non Null	Unique	Null 0	Null0	**Unique**
19	A command to remove a relation from an SQL database_____ _____	Delete Table < Table name>	Drop Table < Table name>	Erase table< Table name>	Alter Table < Table name>	**Drop Table < Table name>**
20	_____ _is not an Aggregate function?	Min	Max	Select	Avg	**Select**
21	The attribute that can be divided into other attributes is called	Simple Attribute	Composite Attribute	Multivalued Attribute	Derived Attribute	**Composite Attribute**

22	What is ACID properties of Transactions?	Atomicity, Consistency, Isolation, Database	Atomicity, Consistency, Isolation, Durability	Atomicity, Consistency, Inconsistent, Durability	Automatically, Concurrency, Isolation, Durability	**Atomicity, Consistency, Isolation, Durability**
23	If every nonkey attribute is functionally dependent on the primary key, the relation will be in	First Normal Form	Second Normal Form	Third Normal Form	Fourth Formal Form	**Third Normal Form**
24	Database locking concept is used to solve the problem of_____	Lost Update	Uncommitted Dependency	Inconsistent Data	Database Instance	**Uncommitted Dependency**
25	UML is stands for_____ _	Universal Modeling Language	Unified Modeling Language	United Modeling Language	Uni Modeling Language	**Unified Modeling Language**
26	Data Manipulation Language (DML) is not to_____	Create information table in the Database	Insertion of new information into the Database	Deletion of information in the Database	Modification of information in the Database	**Create information table in the Database**
27	Data values stored in database must satisfy certain types of_____	Data constraints	Consistency constraints	Type constraints	Format constraints	**Consistency constraints**
28	To allow users to manipulate information, system has a number of_____	Application programs	Application files	Manipulation program	. Manipulation files	**Application programs**
29	Data redundancy and Data Inconsistency is one of major drawbacks of_____	File processing system	File retrieval system	File storage system	File updation system	**File processing system**
30	The view of	Concept	Internal view	External	Physical	**Conceptual**

	total database Content is _____	ual view		view	view	**view**
31	DML is Provided for_____	Description of the logical base database	The Addition of the new structures in the database system	Manipulation and Processing of database	Definition of a physical structure of database system	**Manipulation and Processing of database**
32	The database schema is written in_____	HML	DML	DDL	DCL	**DDL**
33	In a Hierarchical model records are organised as_____	Graph	List	Links	Tree	**Tree**
34	The language used in application programs to request data from the DBMS is referred to as the_____ _	DML	DDL	VDL	SDL	**DML**
35	A logical schema is_____	is a standard way of organizing information into accessible parts	is the entire database.	describes how data is actually stored on disk.	both (B) and (C)	**is the entire database.**
36	Related fields in a database are grouped to form a_____	data file	data record	menu	bank	**data record**
37	An advantage of the database management approach is_____	data is dependent on programs	data redundancy increases	data is integrated and can be accessed by multiple programs	Modification of information in the Database	**data is integrated and can be accessed by multiple programs**
38	The users who use easy-to-use menu are	Sophisticated end users	Naïve users	Stand-alone users	Casual end users	**Naïve users**

	called_____					
40	_____ is the database level is closest to the users.	External	Internal	Physical	Conceptual	**External**
41	There are _____ levels of data abstraction.	2	3	4	1	**3**
42	If user doesn't know anything about the complexity of database application then that user is called as _____.	Naive User	Database Manager	Database Operator	Database Administrator	**Naive User**
43	_____is the example of Object based logical model	Entity Relationship Model	Hierarchical Model	Relational Model	Network Model	**Entity Relationship Model**
44	In Object Oriented Model values are stored into _____ variables.	Instance	Local	Static	Global	**Instance**
45	_____ is the characteristics of transactions	Atomicity	Durability	Isolation	stability	**Atomicity**
46	A _____ file system is software that enables multiple computers to share file storage while maintaining consistent space allocation and file content.	Storage	Tertiary	Secondary	Cluster	**Cluster**
47	The process of saving information onto secondary storage devices is referred to as_____	Backing up	Restoring	Writing	Reading	**Writing**

48	The storage structure which do not survive system crashes are _____	Volatile storage	Non-volatile storage	Stable storage	Dynamic storage	**Volatile storage**
49	_____ is the process of selecting the data storage and data access characteristics of the database.	Logical database design	Physical database design	Testing and performanc e tuning	Evaluation and selecting	**Physical database design**
50	The unit of storage that can store one are more records in a hash file organization are_____	Buckets	Disk pages	Blocks	Nodes	**Buckets**
51	The database administrator is responsible for_____	Data modeling	Database design	Meta data	Database storage	**Database design**
52	Monitoring jobs running on database, should be supervised by_____	Database system	Database manager	Database users	Database administra tor	**Database administrat or**
53	The DBA creates the original database schemma by_____	Writing a set of definitio ns	Storage structure	Application programms	Schema definition	**Writing a set of definitions**
54	Storage structure and defining access-method, is job done by_____	Datab ase system	Database Manager	Database administrato r	Databas e users	**Database administrat or**
55	Ensuring enough free disk space availability for normal operations, are goals to be achieved by_____	Datab ase Manager	Database system	Database users	Databas e administra tor	**Database administrat or**

56	_____ are computer professionals who interact with the system through DML Calls.	Application programmers	Database users	Naive users	Database manager	**Application programmers**
57	_____ is the component which translate DML Statements in a query language into low-level instructions	DML Precomplier	Query evaluation engine	DML Complier	File manager	**DML Complier**
58	_____ is the one who only interacts with the system without writing programs	system Manager	Sophisticated users	Speciallized User	Naive user	**Sophisticated users**
59	_____ manages the allocation of space on disk storage and the dats structure used to represent information stored on disk.	Database management system	File manager	Buffer manager	Transaction manager	**File manager**
60	_____ interprets DDL statements and records them in a set of tables containing metadata	File manager	DML Precomplier	DDL interpreter	DML Compllier	**DDL interpreter**

Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019

Unit 2: Entity Relational Model: Basic concept-Key Entity Relationship Diagram, Weak Entity Sets, E-R features, Specialization, Generalization, Relational model-Structure of Relational Database-Relational Algebra-Views

UNIT 2

The ER model defines the conceptual view of a database. It works around real-world entities and the associations among them. At view level, the ER model is considered a good option for designing databases.

ENTITY

An entity can be a real-world object, either animate or inanimate, that can be easily identifiable. For example, in a school database, students, teachers, classes, and courses offered can be considered as entities. All these entities have some attributes or properties that give them their identity.

An entity set is a collection of similar types of entities. An entity set may contain entities with attribute sharing similar values. For example, a Students set may contain all the students of a school; likewise a Teachers set may contain all the teachers of a school from all faculties. Entity sets need not be disjoint.

ENTITY–RELATIONSHIP MODEL (ER MODEL)

An **Entity–relationship model (ER model)** describes the structure of a database with the help of a diagram, which is known as **Entity Relationship Diagram (ER Diagram)**. **An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.**

What is an Entity Relationship Diagram (ER Diagram)?

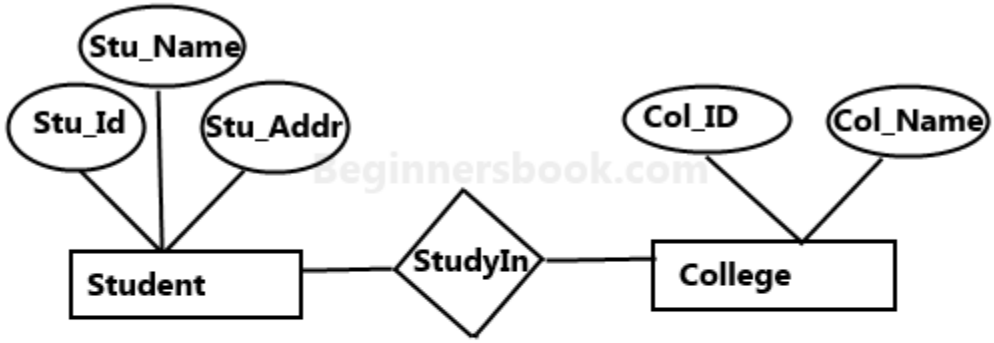
Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019

An ER diagram shows the relationship among entity sets. **An entity set is a group of similar entities and these entities can have attributes**. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes**, ER diagram shows the complete logical structure of a database**. Lets have a look at a simple ER diagram to understand this concept.

A simple ER Diagram:

In the following diagram we have two entities Student and College and their relationship. The relationship between Student and College is many to one as a college can have many students however a student cannot study in multiple colleges at the same time. Student entity has attributes such as Stu_Id, Stu_Name & Stu_Addr and College entity has attributes such as Col_ID & Col_Name.



Sample E-R Diagram

Here are the geometric shapes and their meaning in an E-R Diagram. We will discuss these terms in detail in the next section(Components of a ER Diagram) of this guide so don't worry too much about these terms now, just go through them once.

Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019

Rectangle: Represents Entity sets.

Ellipses: Attributes

Diamonds: Relationship Set

Lines: They link attributes to Entity Sets and Entity sets to Relationship Set

Double Ellipses: Multivalued Attributes

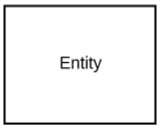
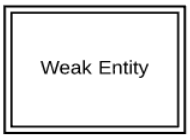
Dashed Ellipses: Derived Attributes

Double Rectangles: Weak Entity Sets

Double Lines: Total participation of an entity in a relationship set

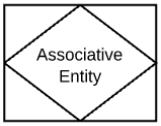
ERD entity symbols

Entities are objects or concepts that represent important data. **Entities are typically nouns such as product, customer, location, or promotion. There are three types of entities commonly used in entity relationship diagrams.**

Entity Symbol	Name	Description
	Strong entity	These shapes are independent from other entities, and are often called parent entities, since they will often have weak entities that depend on them. They will also have a primary key, distinguishing each occurrence of the entity.
	Weak entity	Weak entities depend on some other entity type. They don't have primary keys, and have no meaning in the diagram without their parent entity.

Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019

Entity Symbol	Name	Description
	Associative entity	Associative entities relate the instances of several entity types. They also contain attributes specific to the relationship between those entity instances.

ERD relationship symbols

Within entity-relationship diagrams, relationships are used to document the interaction between two entities. Relationships are usually verbs such as assign, associate, or track and provide useful information that could not be discerned with just the entity types.

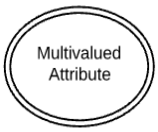
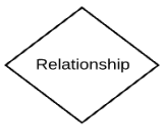
Relationship Symbol	Name	Description
	Relationship	Relationships are associations between or among entities.
	Weak relationship	Weak Relationships are connections between a weak entity and its owner.

Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019

ERD attribute symbols

ERD attributes are characteristics of the entity that help users to better understand the database. Attributes are included to include details of the various entities that are highlighted in a conceptual ER diagram.

Attribute Symbol	Name	Description
	Attribute	Attributes(features) are characteristics of an entity, a many-to-many relationship, or a one-to-one relationship.
	Multi valued attribute	Multi valued attributes are those that are can take on more than one value.
	Derived attribute	Derived attributes are attributes whose value can be calculated from related attribute values.
	Relationship	Relationships are associations between or among entities.

Class: II MBA Course Name: Enterprise resource planning

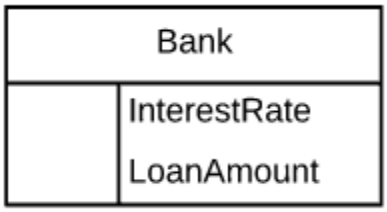
Course code: 17MBAPS401 Semester: IV Year: 2017-2019

The physical data model is the most granular level of entity-relationship diagrams, and represents the process of adding information to the database. Physical ER models show all table structures, including column name, column data type, column constraints, primary key, foreign key, and relationships between tables.

As shown below, tables are another way of representing entities. The key parts of Entity-relationship Tables are:

Fields

Fields represent the portion of a table that establish the attributes of the entity. Attributes are typically thought of as columns in the database that the ERD models.



In the image above, Interest Rate and Loan Amount are both attributes of the entity that are contained as fields.

Keys

Keys are one way to categorize attributes. ER diagrams help users to model their databases by using various tables that ensure that the database is organized, efficient, and fast. Keys are used to link various tables in a database to each other in the most efficient way possible.

Primary Keys

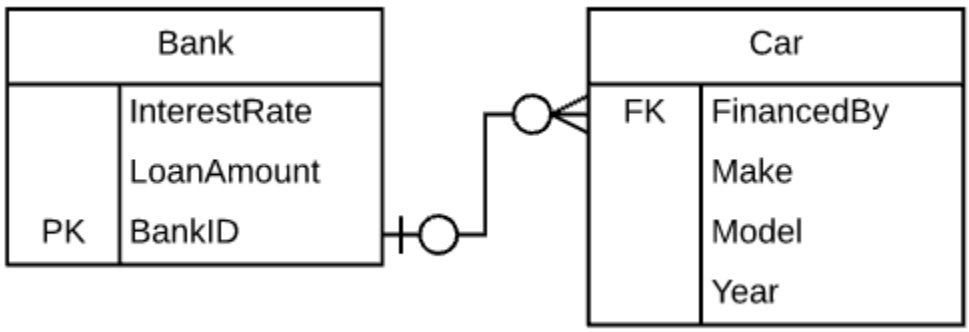
Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019

Primary keys are an attribute or combination of attributes that uniquely identifies one and only one instance of an entity.

Foreign Keys

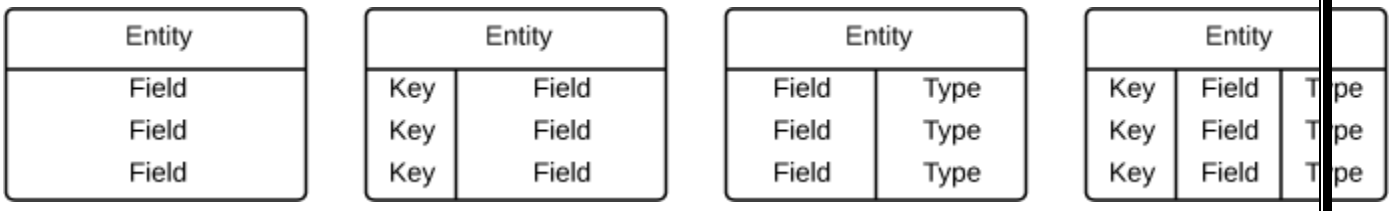
Foreign keys are created any time an attribute relates to another entity in a one-to-one or one-to-many relationship.



Each car can only be financed by one bank, therefore the primary key BankId from the Bank table is used as the foreign key Financed By in the Car table. This BankID is able to be used as the foreign key for multiple cars.

Types

Types refer to the type of data in the corresponding field in a table. Types can also refer to entity types, which describe the composition of an entity; e.g., a book's entity types are author, title, and published date.



Class: II MBA Course Name: Enterprise resource planning

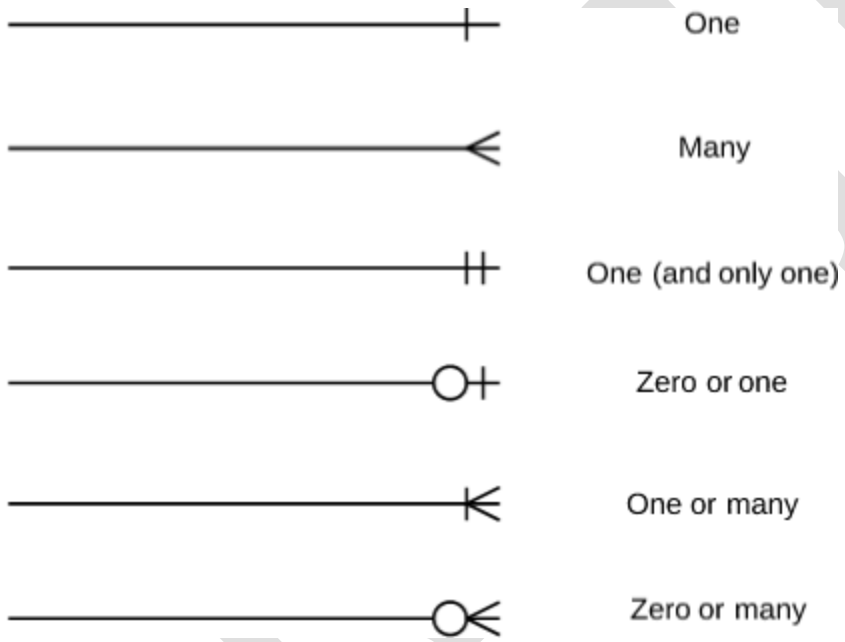
Course code: 17MBAPS401 Semester: IV Year: 2017-2019

ER diagram notation

Cardinality and ordinality

Cardinality refers to the maximum number of times an instance in one entity can relate to instances of another entity. Ordinarily, on the other hand, is the minimum number of times an instance in one entity can be associated with an instance in the related entity.

Cardinality and ordinality are shown by the styling of a line and its endpoint, according to the chosen notation style.

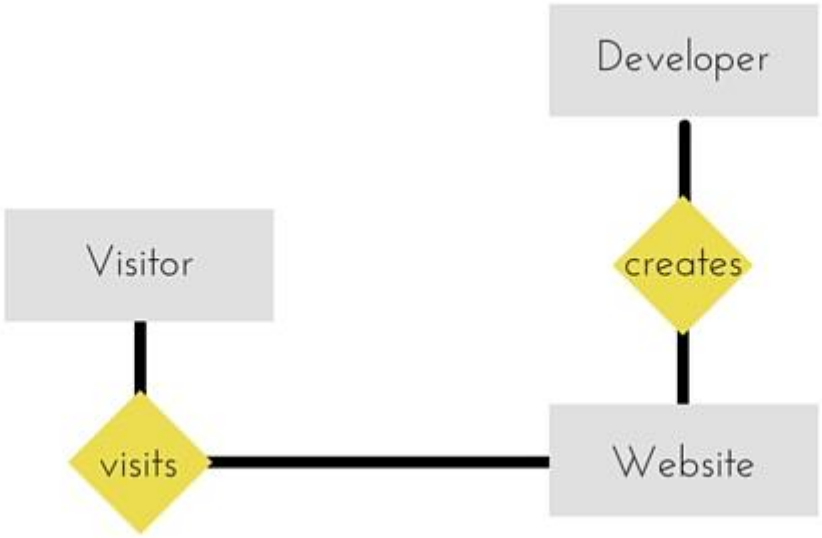


Working with ER Diagrams

ER Diagram is a visual representation of data that describes how data is related to each other. In ER Model, we disintegrate data into entities, attributes and setup relationships between entities, all this can be represented visually using the ER diagram.

Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019



Components of ER Diagram

Entitiy, Attributes, Relationships etc form the components of ER Diagram and there are defined symbols and shapes to represent each one of them.

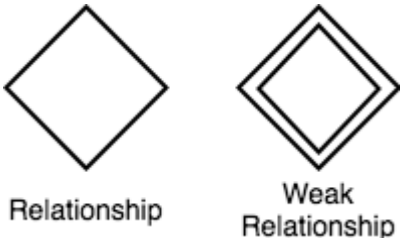
Let's see how we can represent these in our ER Diagram.

Entity

Simple rectangular box represents an Entity.

Relationships between Entities - Weak and Strong

Rhombus is used to setup relationships between two or more entities.

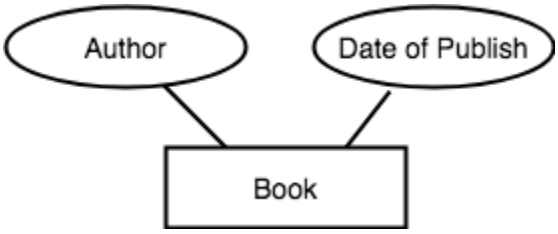


Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019

Attributes for any Entity

Ellipse is used to represent attributes of any entity. It is connected to the entity.



Weak Entity

A weak Entity is represented using double rectangular boxes. It is generally connected to another entity.

Key Attribute for any Entity

To represent a Key attribute, the attribute name inside the Ellipse is underlined.

Derived Attribute for any Entity

Derived attributes are those which are derived based on other attributes, for example, age can be derived from date of birth.

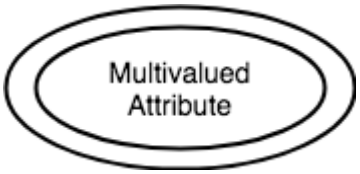
To represent a derived attribute, another dotted ellipse is created inside the main ellipse.

Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019

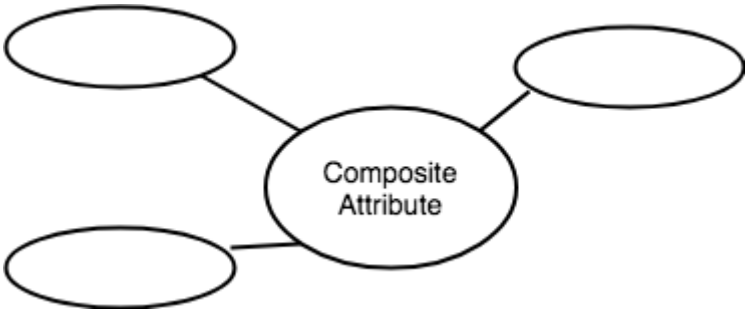
Multivalued Attribute for any Entity

Double Ellipse, one inside another, represents the attribute which can have multiple values.



Composite Attribute for any Entity

A composite attribute is the attribute, which also has attributes.



ER Diagram: Entity

An **Entity** can be any object, place, person or class. In ER Diagram, an **entity** is represented using rectangles. Consider an example of an Organisation- Employee, Manager, Department, Product and many more can be taken as entities in an Organisation.

Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019

The yellow rhombus in between represents a relationship.

ER Diagram: Weak Entity

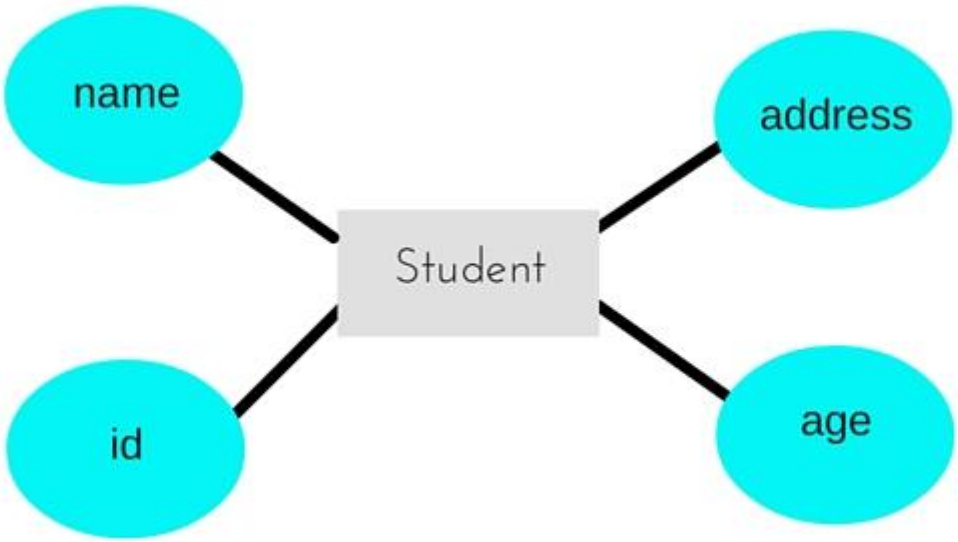
Weak entity is an entity that depends on another entity. Weak entity doesn't have anay key attribute of its own. Double rectangle is used to represent a weak entity.

Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019

ER Diagram: Attribute

An **Attribute** describes a property or characterstic of an entity. For example, **Name**, **Age**, **Address** etc can be attributes of a **Student**. An attribute is represented using eclipse.

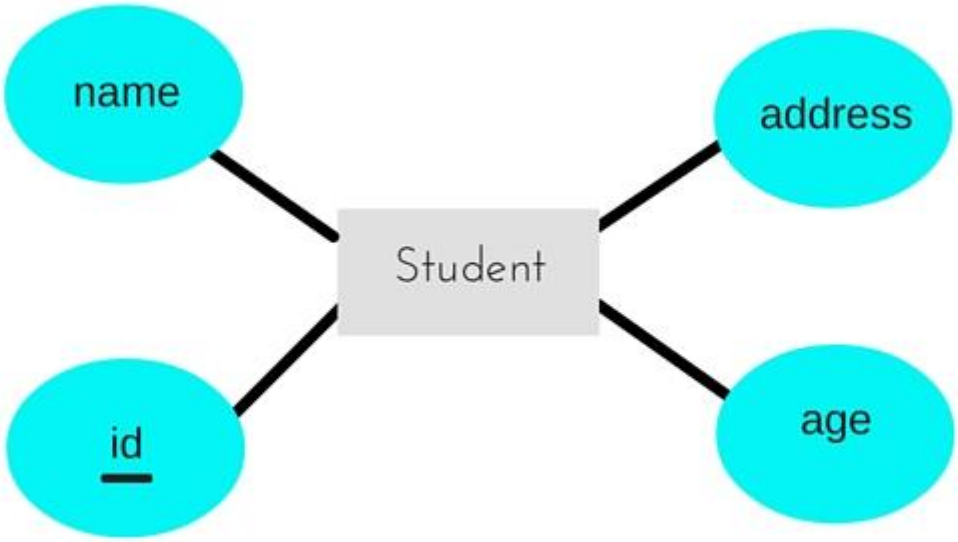


ER Diagram: Key Attribute

Key attribute represents the main characteristics of an Entity. It is used to represent a Primary key. Ellipse with the text underlined, represents Key Attribute.

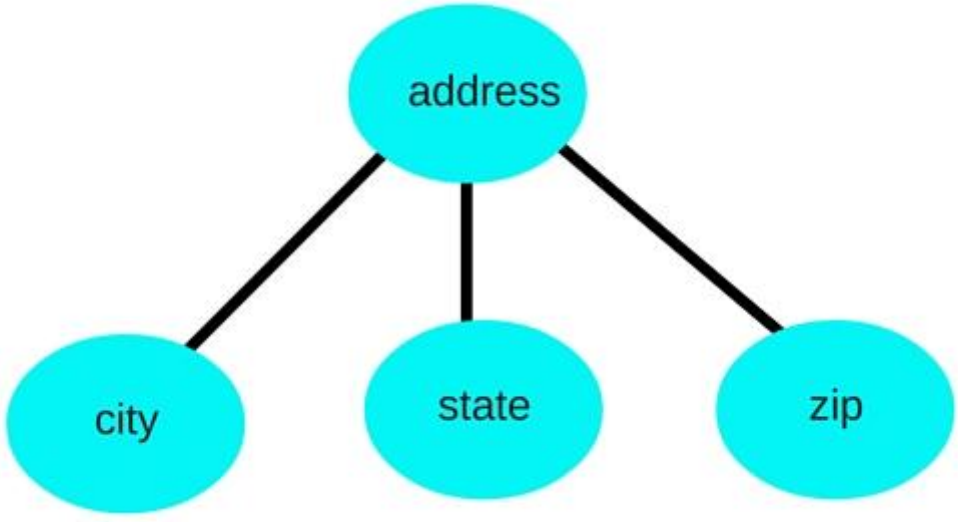
Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019



ER Diagram: Composite Attribute

An attribute can also have their own attributes. These attributes are known as **Composite** attributes.



ER Diagram: Relationship

Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019

A Relationship describes relation between **entities**. Relationship is represented using diamonds or rhombus.

There are three types of relationship that exist between Entities.

1. Binary Relationship
2. Recursive Relationship
3. Ternary Relationship

ER Diagram: Binary Relationship

Binary Relationship means relation between two Entities. This is further divided into three types.

One to One Relationship

This type of relationship is rarely seen in real world.

Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019

The above example describes that one student can enroll only for one course and a course will also have only one Student. This is not what you will usually see in real-world relationships.

One to Many Relationship

The below example showcases this relationship, which means that 1 student can opt for many courses, but a course can only have 1 student. Sounds weird! This is how it is.

Many to One Relationship

It reflects business rule that many entities can be associated with just one entity. For example, Student enrolls for only one Course but a Course can have many Students.

Many to Many Relationship

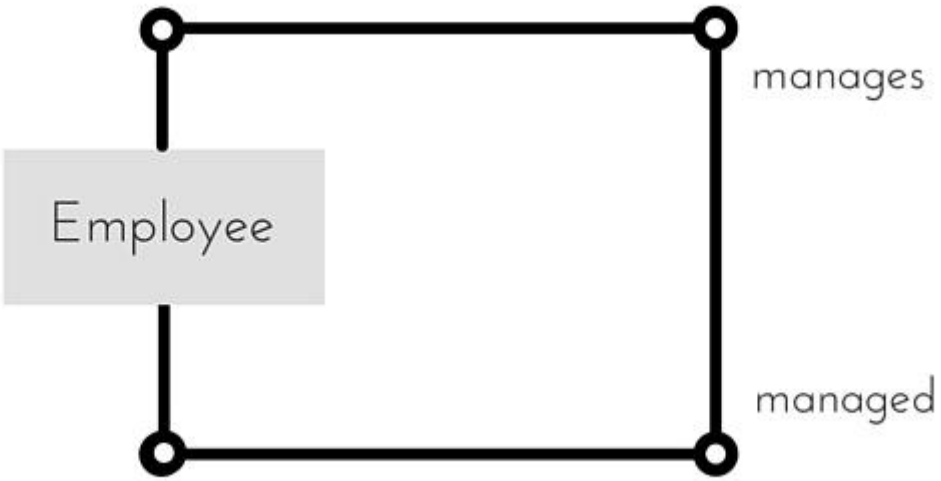
Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019

The above diagram represents that one student can enroll for more than one courses. And a course can have more than 1 student enrolled in it.

ER Diagram: Recursive Relationship

When an Entity is related with itself it is known as **Recursive** Relationship.



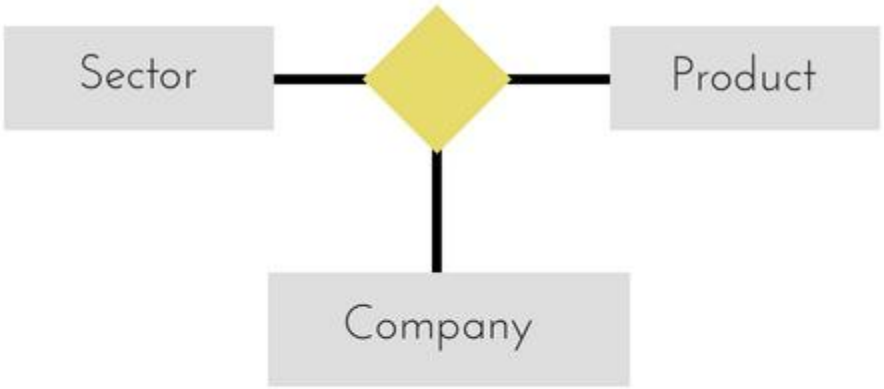
Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019

ER Diagram: Ternary Relationship

Relationship of degree three is called Ternary relationship.

A Ternary relationship involves three entities. In such relationships we always consider two entites together and then look upon the third.



- The above relationship involves 3 entities.
- Company operates in Sector, producing some Products.

For example, in the diagram above, we have three related entities, **Company**, **Product** and **Sector**. To understand the relationship better or to define rules around the model, we should relate two entities and then derive the third one.

A **Company** produces many **Products**/ each product is produced by exactly one company.

A **Company** operates in only one **Sector** / each sector has many companies operating in it.

Considering the above two rules or relationships, we see that although the complete relationship involves three entities, but we are looking at two entities at a time.

Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019

The Enhanced ER Model:E-R Features

As the complexity of data increased in the late 1980s, it became more and more **difficult to use the traditional ER Model for database modelling**. Hence some improvements or enhancements were made to the existing ER Model to make it able to handle the complex applications better.

Hence, as part of the **Enhanced ER Model**, along with other improvements, three new concepts were added to the existing ER Model, they were:

1. Generalization
2. Specialization
3. Aggregration

Let's understand what they are, and why were they added to the existing ER Model.

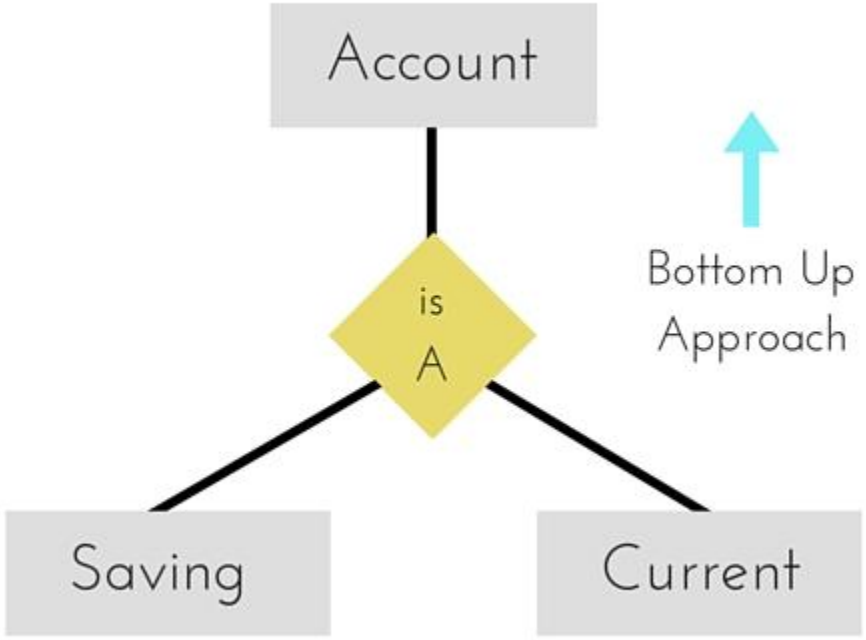
Generalization

Generalization is a bottom-up approach in which two lower level entities combine to form a higher level entity. In generalization, the higher level entity can also combine with other lower level entities to make further higher level entity.

It's more like Superclass and Subclass system, but the only difference is the approach, which is bottom-up. Hence, entities are combined to form a more generalised entity, in other words, sub-classes are combined to form a super-class.

Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019



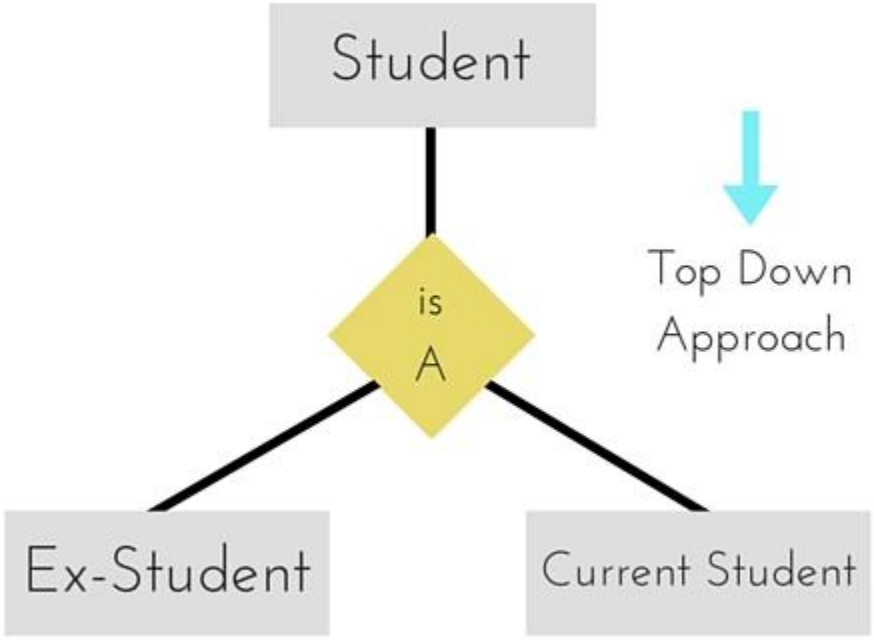
For example, **Saving** and **Current** account types entities can be generalised and an entity with name **Account** can be created, which covers both.

Specialization

Specialization is opposite to Generalization. It is a top-down approach in which one higher level entity can be broken down into two lower level entity. In specialization, a higher level entity may not have any lower-level entity sets, it's possible.

Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019

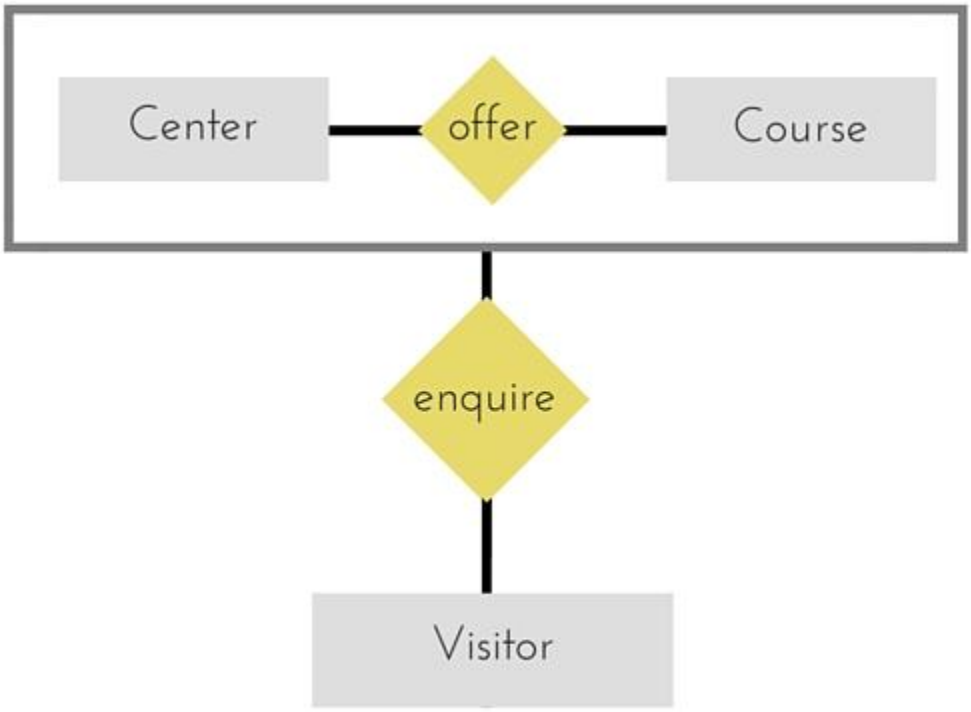


Aggregration

Aggregration is a process when relation between two entities is treated as a **single entity**.

Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019



In the diagram above, the relationship between **Center** and **Course** together, is acting as an Entity, which is in relationship with another entity **Visitor**. Now in real world, if a Visitor or a Student visits a Coaching Center, he/she will never enquire about the center only or just about the course, rather he/she will ask enquire about both.

Basic Relational DBMS Concepts

A **Relational Database management System**(RDBMS) is a database management system based on the relational model introduced by E.F Codd. In relational model, data is stored in **relations**(tables) and is represented in form of **tuples**(rows).

RDBMS is used to manage Relational database. Relational database is a collection of organized set of tables related to each other, and from which data can be accessed easily. Relational Database is the most commonly used database these days.

Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019

RDBMS: What is Table ?

In Relational database model, a table is a collection of data elements organised in terms of rows and columns. A table is also considered as a convenient representation of relations. But a table can have duplicate row of data while a true relation cannot have duplicate data. Table is the most simplest form of data storage. Below is an example of an Employee table**.**

ID	Name	Age	Salary
1	Adam	34	13000
2	Alex	28	15000
3	Stuart	20	18000
4	Ross	42	19020

RDBMS: What is a Tuple?

A single entry in a table is called a **Tuple** or **Record** or **Row**. A **tuple** in a table represents a set of related data. For example, the above **Employee** table has 4 tuples/records/rows.

Following is an example of single record or tuple.

1	Adam	34	13000

Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019

RDBMS: What is an Attribute?

A table consists of several records(row), each record can be broken down into several smaller parts of data known as **Attributes**. The above **Employee** table consist of four attributes, **ID**, **Name**, **Age** and **Salary**.

Attribute Domain

When an attribute is defined in a relation(table), it is defined to hold only a certain type of values, which is known as **Attribute Domain**.
Hence, the attribute **Name** will hold the name of employee for every tuple. If we save employee's address there, it will be violation of the Relational database model.

Name
Adam
Alex
Stuart - 9/401, OC Street, Amsterdam
Ross

What is a Relation Schema?

A relation schema describes the structure of the relation, with the name of the relation(name of table), its attributes and their names and type.

What is a Relation Key?

A relation key is an attribute which can uniquely identify a particular (row) in a relation(table).

Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019

Relational Integrity Constraints

Every relation in a relational database model should abide by or follow a few constraints to be a valid relation, these constraints are called as **Relational Integrity Constraints**.

The three main Integrity Constraints are:

1. Key Constraints
2. Domain Constraints
3. Referential integrity Constraints

Key Constraints

We store data in tables, to later access it whenever required. In every table one or more than one attributes together are used to fetch data from tables.

The **Key Constraint** specifies that there should be such an attribute(column) in a relation(table), which can be used to fetch data for any tuple(row).

The Key attribute should never be **NULL** or same for two different row of data.

For example, in the **Employee** table we can use the attribute ID to fetch data for each of the employee. No value of ID is null and it is unique for every row, hence it can be our **Key attribute**.

Domain Constraint

Domain constraints refers to the rules defined for the values that can be stored for a certain attribute.

Like we explained above, we cannot store **Address** of employee in the column for **Name**. Similarly, a mobile number cannot exceed 10 digits.

Referential Integrity Constraint

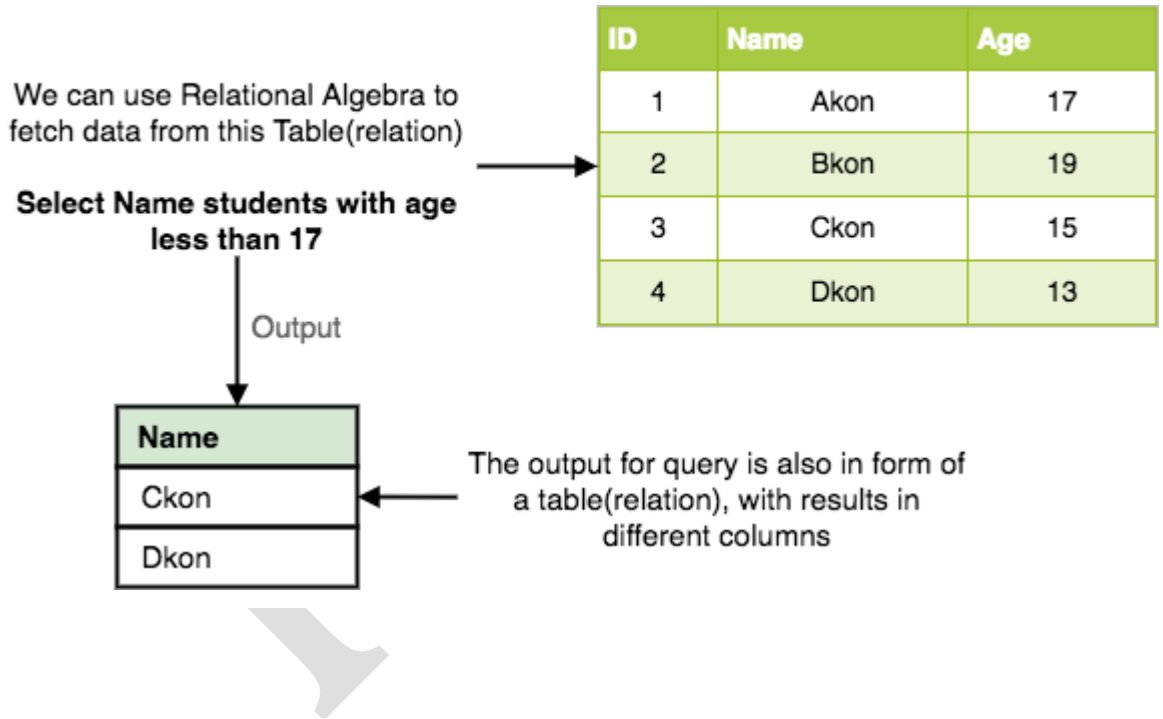
, If I say **Supriya** is my girlfriend, then a girl with name Supriya should also exist for that relationship to be present.

If a table reference to some data from another table, then that table and that data should be present for referential integrity constraint to hold true.

What is Relational Algebra?

Every database management system must define a query language to allow users to access the data stored in the database. **Relational Algebra** is a procedural query language used to query the database tables to access data in different ways.

In relational algebra, input is a relation(table from which data has to be accessed) and output is also a relation(a temporary table holding the data asked for by the user).



Relational Algebra works on the whole table at once, so we do not have to use loops etc to iterate over all the rows(tuples) of data one by one. All we have to do is specify the table name from which we need the data, and in a single line of command, relational algebra will traverse the entire given table to fetch data for you.

The primary operations that we can perform using relational algebra are:

Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019

1. Select
2. Project
3. Union
4. Set Different
5. Cartesian product
6. Rename

Select Operation (σ)

This is used to fetch rows(tuples) from table(relation) which satisfies a given condition.

Syntax: $\sigma_p(r)$

Where, σ represents the Select Predicate, r is the name of relation(table name in which you want to look for data), and p is the prepositional logic, where we specify the conditions that must be satisfied by the data. In prepositional logic, one can use **unary** and **binary** operators like =, <, > etc, to specify the conditions.

Let's take an example of the Student table we specified above in the Introduction of relational algebra, and fetch data for **students** with **age** more than 17.

$\sigma_{age > 17}$ (Student)

This will fetch the tuples(rows) from table **Student**, for which **age** will be greater than **17**. You can also use, and, or etc operators, to specify two conditions, for example,

$\sigma_{age > 17 \text{ and } gender = 'Male'}$ (Student)

This will return tuples(rows) from table **Student** with information of male students, of age more than 17.(Consider the Student table has an attribute Gender too.)

Project Operation (∏)

Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019

Project operation is used to project only a certain set of attributes of a relation. In simple words, If you want to see only the **names** all of the students in the **Student** table, then you can use Project Operation.

It will only project or show the columns or attributes asked for, and will also remove duplicate data from the columns.

Syntax: $\prod_{A1, A2...}(r)$

where A1, A2 etc are attribute names(column names).

For example,

$\prod_{Name, Age}(Student)$

Above statement will show us only the **Name** and **Age** columns for all the rows of data in **Student**table.

Union Operation (∪)

This operation is used to fetch data from two relations(tables) or temporary relation(result of another operation).

For this operation to work, the relations(tables) specified should have same number of attributes(columns) and same attribute domain. Also the duplicate tuples are autamatically eliminated from the result.

Syntax: A ∪ B

where A and B are relations.

For example, if we have two tables **RegularClass** and **ExtraClass**, both have a column **student** to save name of student, then,

$\prod_{Student}(RegularClass) \cup \prod_{Student}(ExtraClass)$

Above operation will give us name of **Students** who are attending both regular classes and extra classes, eliminating repetition.

Set Difference (-)

This operation is used to find data present in one relation and not present in the second relation. This operation is also applicable on two relations, just like Union operation.

Syntax: A - B

where A and B are relations.

For example, if we want to find name of students who attend the regular class but not the extra class, then, we can use the below operation:

$\prod_{Student}$(RegularClass) - $\prod_{Student}$(ExtraClass)

Cartesian Product (X)

This is used to combine data from two different relations(tables) into one and fetch data from the combined relation.

Syntax: A X B

For example, if we want to find the information for Regular Class and Extra Class which are conducted during morning, then, we can use the following operation:

$\sigma_{time = 'morning'}$ (RegularClass X ExtraClass)

For the above query to work, both **RegularClass** and **ExtraClass** should have the attribute **time**.

Rename Operation (ρ)

This operation is used to rename the output relation for any query operation which returns result like Select, Project etc. Or to simply rename a relation(table)

Syntax: ρ(RelationNew, RelationOld)

Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019

Converting ER Diagrams to Tables-

After we have designed an **ER Diagram**, ER diagram is converted into the tables in relational model so that it can be easily implemented by any Relational Database Management System (RDBMS) like MySQL , Oracle etc.

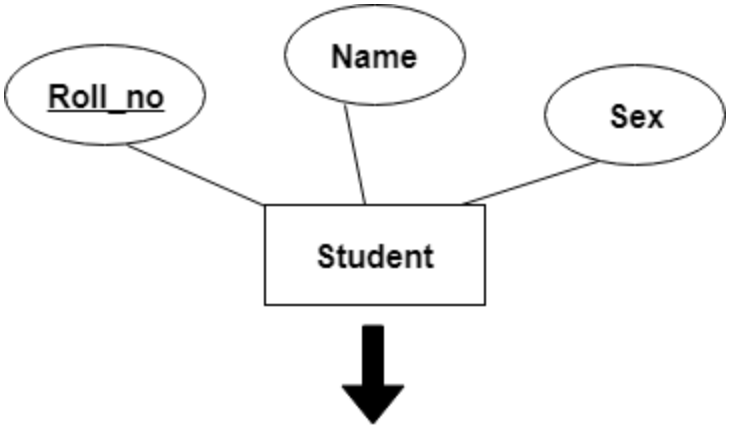
We use the following rules for converting an ER diagram into the tables in relational model-

Rule-01: For a strong entity set with only simple attributes-

A strong entity set with only simple attributes will require only one table in relational model.

- Attributes of the table will be the attributes of the entity set.
- The primary key of the table will be the key attribute of the entity set.

Example-



Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019

Roll_no	Name	Sex

Schema : Student (Roll_no , Name , Sex)

Also Read- Types of Attributes

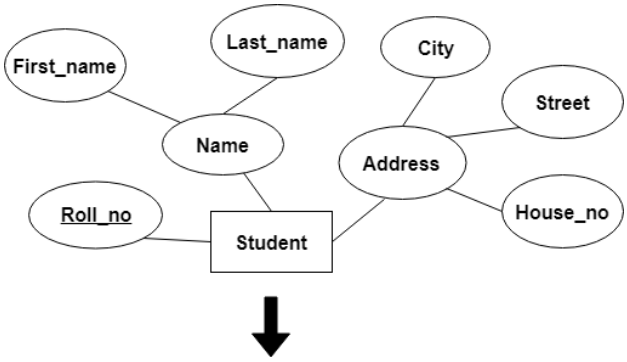
Rule-02: For a strong entity set with composite attributes-

- A strong entity set with any number of composite attributes will require only one table in relational model.
- While conversion, simple attributes of the composite attributes are taken into account and not the composite attribute itself.

Example-

Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019



Roll_no	First_name	Last_name	House_no	Street	City

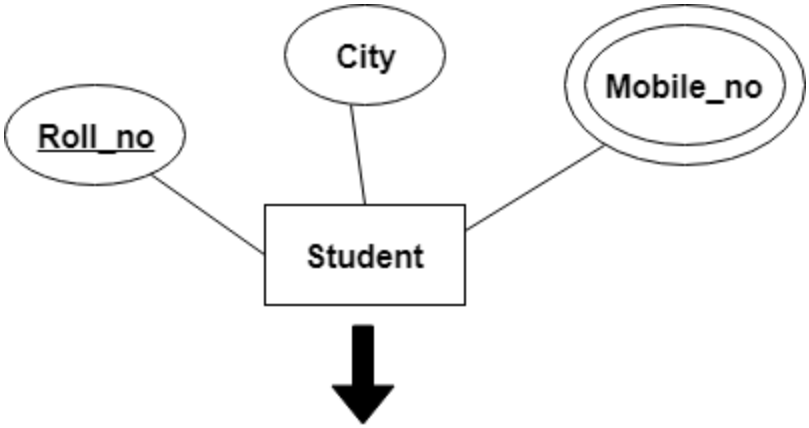
Schema : Student (Roll_no , First_name , Last_name , House_no , Street , City)

Rule-03: For a strong entity set with multi valued attributes-

A strong entity set with any number of multi valued attributes will require two tables in relational model.

- One table will contain all the simple attributes with the primary key.
- Other table will contain the primary key and all the multi valued attributes.

Example-



Roll_no	City

Roll_no	Mobile_no

Rule-04: Translating relationship set into a table-

A relationship set will require one table in the relational model.

Attributes of the table are-

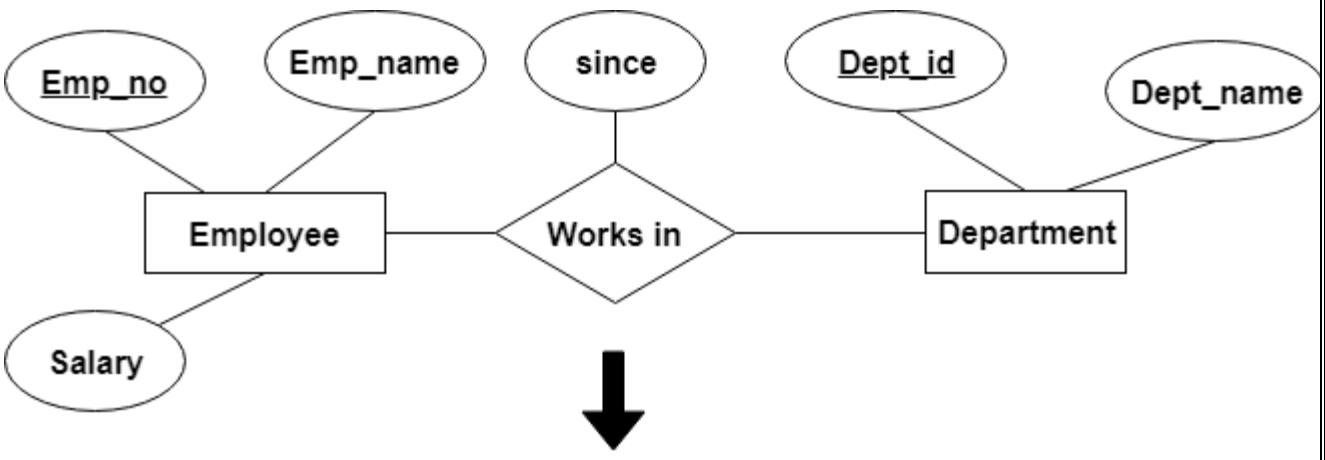
Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019

- Primary key attributes of the participating entity sets
- Its own descriptive attributes if any.

Set of non-descriptive attributes will be the primary key.

Example-



Emp_no	Dept_id	since

Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019

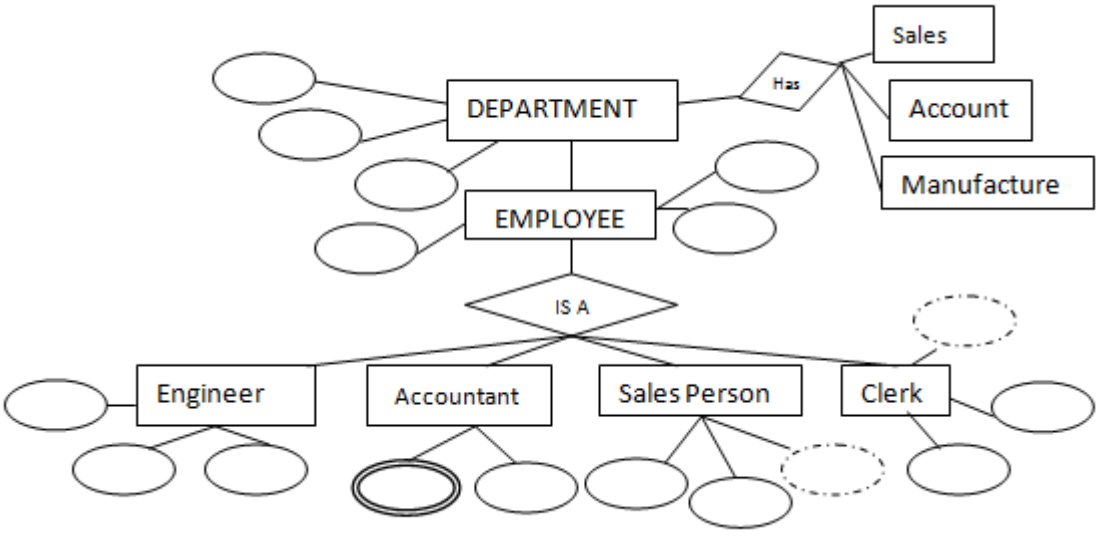
Table for relationship set

It is important to note that if we consider the overall diagram, three tables will be required-

- One table for the entity set "Employee"
- One table for the entity set "Department"
- One table for the relationship set "Works in"

GENERALIZATION:

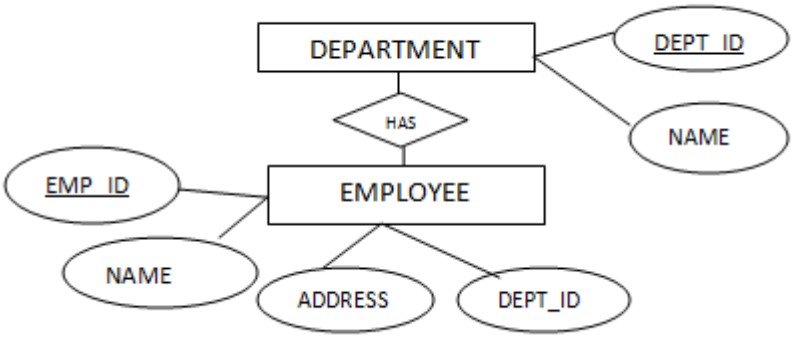
- In our Employee example, we have seen different types of employees like Engineer, Accountant, Salesperson, Clerk etc. Similarly each employee belongs to different departments. We can represent it in an ER diagram as below. When you see this diagram for the first time, you will not understand it quickly. One will take time to understand it or he might misunderstand some requirement.



Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019

- What if we group all the sub departments into one department and different employees into one employee? However sub departments and different employee types have same features in their own domain. So if we merge the child entities into their parent, it makes the diagram simpler, hence easy to understand. This method of merging the branches into one is called generalization. We can see the generalized structure of requirement to understand it quickly. So above ER diagram will be changed to as below:



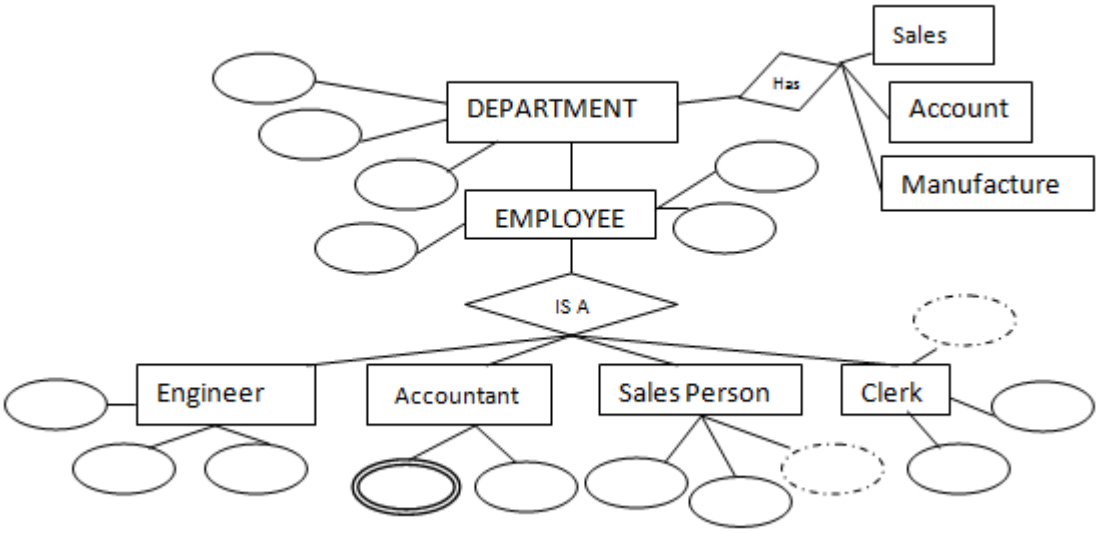
- Isn't it simpler? Generalization is the bottom up approach which helps to design the requirement at high level. Thus making one to understand quickly.

SPECIALIZATION:

- It is opposite approach of generalization. Here, each entity is further divided into sub levels to understand it deeper. In the above example, Department entity is further divided into sub departments to understand how they are scattered. This method of representation helps the developer to code correctly and quickly. It is a top down approach of breaking higher level entity to low level entity. Once the entities are understood at higher level, it makes easy to understand the requirement at low level.

Class: II MBA Course Name: Enterprise resource planning

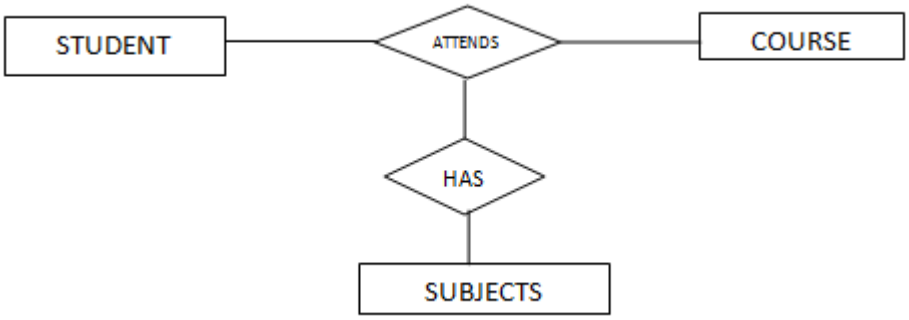
Course code: 17MBAPS401 Semester: IV Year: 2017-2019



- One more example of specialization would be Person. We can further divide person as STUDENT, TEACHER, ENGINEER, SOLDIER etc. (Merging STUDENT, TEACHER, ENGINEER etc into PERSON is an example of generalization).

AGGREGATION:

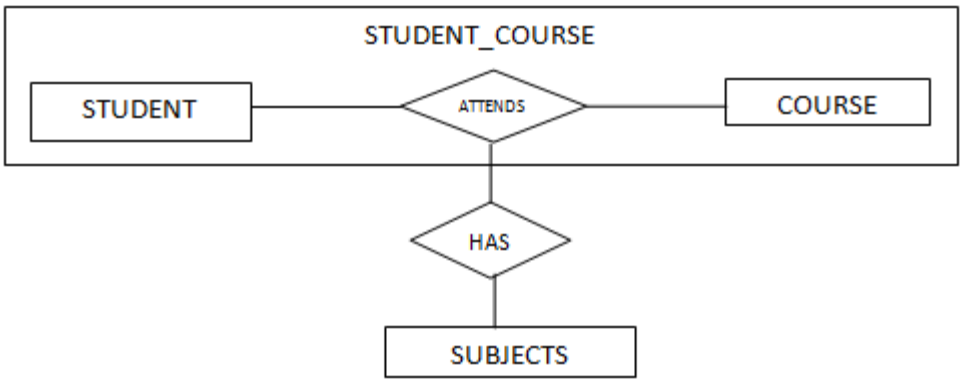
- Look at below ER diagram of STUDENT, COURSE and SUBJECTS. What does it infer? Student attends the Course, and he has some subjects to study. At the same time, Course offers some subjects. Here a relation is defined on a relation. But ER diagram does not entertain such a relation. It supports mapping between entities, not between relations. So what can we do in this case?



Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401 Semester: IV Year: 2017-2019

- If we look at STUDENT and COURSE from SUBJECT's point of view, it does not differentiate both of them. It offers it's subject to both of them. So what can we do here is, merge STUDENT and COURSE as one entity. This process of merging is called aggregation. It is completely different from generalization. In generalization, we **merge entities of same domain** into one entity. In this case we **merge related entities** into one entity.



- Here we have merged STUDENT and COURSE into one entity STUDENT_COURSE. This new entity forms the mapping with SUBJECTS. The new entity STUDENT_COURSE, in turn has two entities STUDENT and COURSE with 'Attends' relationship.

KARPAGAM ACADEMY OF HIGHER EDUCATION						
Department of Management						
Unit 2- Multiple Choice Questions- Each Question Carry ONE Mark						
S.no	**Quetions**	**Option 1**	**Option 2**	**Option 3**	**Option 4**	**Answers**
1	An _____ is a set of entities of the same type that share the same properties, or attributes.	Entity set	Attribute set	Relation set	Entity model	**Entity set**
2	Entity is a _____	Object of relation	Present working model	Thing in real world	Model of relation	**Thing in real world**
3	The descriptive property possessed by each entity set is _____	Entity	Attribute	Relation	Model	**Attribute**
4	The function that an entity plays in a relationship is called that entity's _____	Participation	Position	Role	Instance	**Role**
5	The attribute *name* could be structured as an attribute consisting of first name, middle initial, and last name. This type of attribute is called_____	Simple attribute	Composite attribute	Multivalued attribute	Derived attribute	**Composite attribute**
6	The attribute AGE is calculated from DATE_OF_BIRTH. The attribute AGE is_____	Single valued	Multi valued	Composite	Derived	**Derived**
7	Not applicable condition can be represented in relation entry as_____	NA	0	NULL	Blank Space	**NULL**
8	_____- in the following can be a multivalued attribute?	Phone_number	Name	Date_of_birth	Age	**Phone_number**
9	Which of the following is a	Register_number	Address	SUBJECT_TAKEN	Reference	**Register_number**

	single valued attribute_____					
10	In a relation between the entities the type and condition of the relation should be specified. That is called as_____attribute.	Desciptive	Derived	Recursive	Relative	**Desciptive**
11	_____ gives a logical structure of the database graphically?	Entity-relationship diagram	Entity diagram	Database diagram	Architectural representation	**Entity-relationship diagram**
12	The entity relationship set is represented in E-R diagram as_____	Double diamonds	Undivided rectangles	Dashed lines	Diamond	**Diamond**
13	The Rectangles divided into two parts represents	Entity set	Relationship set	Attributes of a relationship set	Primary key	**Entity set**
14	Consider a directed line(->) from the relationship set advisor to both entity sets instructor and student. This indicates _____ cardinality	One to many	One to one	Many to many	Many to one	**One to one**
15	We indicate roles in E-R diagrams by labeling the lines that connect _____ to _____	Diamond, diamond	Rectangle, diamond	Rectangle, rectangle	Diamond, rectangle	**Diamond, rectangle**
16	An entity set that does not have sufficient attributes to form a primary key is termed a _____	Strong entity set	Variant set	Weak entity set	Variable set	**Weak entity set**
17	For a weak entity set to be meaningful, it must be associated with another entity set,	Identifying set	Owner set	Neighbour set	Strong entity set	**Identifying set**

	called the					
18	Weak entity set is represented as_____	Underline	Double line	Double diamond	Double rectangle	**Double diamond**
19	If you were collecting and storing information about your music collection, an album would be considered a(n) _____	Relation	Entity	Instance	Attribute	**Entity**
20	What term is used to refer to a specific record in your music database; for instance; information stored about a specific album?	Relation	Instance	Table	Column	**Instance**
21	Let us consider *phone_number* ,which can take single or several values . Treating*phone_number*as an _____ permits instructors to have several phone numbers (including zero) associated with them	Entity	Attribute	Relation	Value	**Entity**
22	The total participation by entities is represented in E-R diagram as_____	Dashed line	Double line	Double rectangle	Circle	**Double line**
23	Which of the following indicates the maximum number of entities that can be involved in a relationship?	Minimum cardinality	Maximum cardinality	ERD	Greater Entity Count	**Maximum cardinality**
24	In E-R diagram generalization is	Ellipse	Dashed ellipse	Rectangle	Triangle	**Triangle**

	represented by_____					
25	What is a relationship called when it is maintained between two entities?	Unary	Binary	Ternary	Quaternary	**Binary**
26	Which of the following is a low level operator?	Insert	Update	Delete	Directory	**Directory**
27	Key to represent relationship between tables is called_____	Primary key	Secondary Key	Foreign Key	Key Entity	**Foreign Key**
28	A window into a portion of a database is_____	Schema	View	Query	Data dictionary	**View**
29	The entity set person is classified as student and employee. This process is called _____	Generalization	Specialization	Inheritance	Constraint generalization	**Specialization**
30	Which relationship is used to represent a specialization entity?	ISA	AIS	ONIS	WHOIS	**ISA**
31	The refinement from an initial entity set into successive levels of entity subgroupings represents a _____ design process in which distinctions are made explicit.	Hierarchy	Bottom-up	Top-down	Radical	**Top-down**
33	If an entity set is a lower-level entity set in more than one ISA relationship, then the entity set has	Hierarchy	Multilevel inheritance	Single inheritance	Multiple inheritance	**Multiple inheritance**
34	A _____ constraint requires that an entity belong to no more than one lower-	Disjointness	Uniqueness	Special	Relational	**Disjointness**

	level entity set					
35	The completeness constraint may be one of the following: Total generalization or specialization, Partial generalization or specialization. Which is the default?	Total	Partial	Should be specified	Cannot be determined	**Partial**
36	Functional dependencies are a generalization of_____	Key dependencies	Relation dependencies	Database dependencies	Entity model	**Key dependencies**
37	Which of the following is another name for a weak entity?	Child	Owner	Dominant	Parent	**Child**
38	Which of the following information does an SQL DDL not specify?	The schema for each relation	The integrity constraints	The operations on the tuples	The security and authorization information for each relation	**The operations on the tuples**
40	Which of the following data types does the SQL standard not support?	char(n)	String(n)	varchar(n)	float(n)	**String(n)**
41	Which of the following commands do we use to delete a relation (R) from a database?	drop table R	drop relation R	delete table R	delete from R	**drop table R**
42	A relational database consists of a collection of_____	Tables	Fields	Records	Keys	**Tables**
43	A _____ in a table represents a relationship among a set of values.	Column	Key	Row	Entry	**Row**
44	The term _____	Attribute	Tuple	Field	Instance	**Tuple**

	is used to refer to a row.					
45	The term attribute refers to a _____ of a table.	Record	Column	Tuple	Key	**Column**
46	For each attribute of a relation, there is a set of permitted values, called the _____ of that attribute.	Domain	Relation	Set	Schema	**Domain**
47	_____ which is the logical design of the database, and the database _____ which is a snapshot of the data in the database at a given instant in time	Instance, Schema	Relation, Schema	Relation, Domain	Schema, Instance	**Schema, Instance**
48	_____is represented using double rectangular boxes. It is generally connected to another entity	Weak Entity	Attribute	Derived attribute	Multivalued Attribute	**Weak Entity**
49	Database dictionary(catalog) is the structure description of the complete _____ and it must be stored online	Organisation	Database	Relational Attribute	Entity sets	**Database**
50	_____ is a collection of organized set of tables related to each other	RDBMS	DDL	SQL	DBMS	**RDBMS**
51	An entity set that does not have sufficient attributes to form a primary key is a_____	strong entity set	weak entity set	simple entity set	primary entity set	**weak entity set**
52	Key to represent relationship between tables is called_____	Primary key	Secondary Key	Foreign Key	Set	**Foreign Key**
53	Relational Algebra	Data	Procedura	Meta	Schema	**Procedura**

	is_____	Definitio n Language	l query Language	Language		**l query Language**
54	_____is a relationship called when it is maintained between two entities?	Unary	Binary	Ternary	Quaternar y	**Binary**
55	_____ produces the relation that has attributes of R1 and R2	Cartesian product	Differenc e	Intersectio n	Product	**Cartesian product**
56	An entity type whose existence depends on another entity type is called a _____ entity	Strong	Weak	Variant	Independa nt	**Weak**
57	Relational Algebra is a _____ query language that takes two relations as input and produces another relation as an output of the query.	Relationa l	Structural	Procedural	Fundame ntal	**Procedura l**
58	Updating, Deleting and Inserting in relational algebra is done using the _____ operator	Assignme nt	Modificat ion	Alteration	Inclusion	**Assignme nt**
59	_____is used to denote the selection operation in relational algebra	Pi	Sigma	lambda	Omega	**Sigma**
60	The _____ operation, denoted by -, allows us to find tuples that are in one relation but are not in anothe	Union	Set-Differenc e	Difference	Intersectio n	**Set-Difference**

Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401C Semester: IV Year: 2017-2019

SQL-Background-Basic Structure-Set operations-Aggregate Functions-Null values-Nested-Sub Queries-Derived relations-Views-Modification of the database-Joined relations-Data definition language-Embedded SQL features-Data warehousing-Concepts-Architecture-Data transformation-User Interface-Distributed database

UNIT 3

SQL:

Structured Query Language (SQL) is a standard computer language for relational database management and data manipulation. SQL is used to query, insert, update and modify data. Most relational databases support SQL, which is an added benefit for database administrators (DBAs), as they are often required to support databases across several different platforms.

First developed in the early 1970s at IBM by Raymond Boyce and Donald Chamberlin, SQL was commercially released by Relational Software Inc. (now known as Oracle Corporation) in 1979. The current standard SQL version is voluntary, vendor-compliant and monitored by the American National Standards Institute (ANSI). Most major vendors also have proprietary versions that are incorporated and built on ANSI SQL, e.g., SQL*Plus (Oracle), and Transact-SQL (T-SQL) (Microsoft).

Basic Structure

1. Basic structure of an SQL expression consists of **select, from** and **where** clauses.
   - **select** clause lists attributes to be copied - corresponds to relational algebra **project**.
   - **from** clause corresponds to Cartesian product - lists relations to be used.
   - **where** clause corresponds to selection predicate in relational algebra.
2. Typical query has the form

   select $A_1, A_2, \ldots, A_n$

   from $r_1, r_2, \ldots, r_m$

   where *P*

   where each A_i represents an attribute, each r_i a relation, and *P* is a predicate.

3. This is equivalent to the relational algebra expression

$$\Pi_{A_1, A_2, \ldots, A_n}(\sigma_P(r_1 \times r_2 \times \ldots \times r_m))$$

- If the where clause is omitted, the predicate *P* is true.
- The list of attributes can be replaced with a * to select all.
- SQL forms the Cartesian product of the relations named, performs a selection using the predicate, then projects the result onto the attributes named.
- The result of an SQL query is a relation.
- SQL may internally convert into more efficient expressions.

• SQL is based on set and relational operations with certain modifications and enhancements
•

A typical SQL query has the form:

select A1, A2, ..., An from r1, r2, ..., rm where P –

A is represent attributes –

R is represent relations –

P is a predicate.

• This query is equivalent to the relational algebra expression:

ΠA1, A2, ..., An(σP (r1 × r2 × ... × rm))

• The result of an SQL query is a relation.

The select clause corresponds to the projection operation of the relational algebra. It is used to list the attributes desired in the result of a query.

• *Find the names of all branches in the loan relation*

select branch-name

from loan

In the "pure" relational algebra syntax, this query would be:

Πbranch-name (loan)

• An asterisk in the select clause denotes "all attributes"

select ∗ from loan

SQL allows duplicates in relations as well as in query results.

• To force the elimination of duplicates, insert the keyword distinct after select.

Find the names of all branches in the loan relation, and remove duplicates

select distinct branch-name

from loan

• *The keyword all specifies that duplicates not be removed.*

select all branch-name

from loan

The select clause can contain arithmetic expressions involving the operators, +, −, ∗, and /, and operating on constants or attributes of tuples.

The query:

select branch-name, loan-number, amount ∗ 100

from loan

would return a relation which is the same as the loan relation, except that the attribute amount is multiplied by 100

The where clause corresponds to the selection predicate of the relational algebra. It consists of a predicate involving attributes of the relations that appear in the from clause.

• *Find all loan numbers for loans made at the Perryridge branch with loan amounts greater than $1200.*

select loan-number

from loan

where branch-name = "Perryridge" and amount > 1200

• SQL uses the logical connectives **and, or, and not**. It allows the use of arithmetic expressions as operands to the comparison operators

SQL includes a between comparison operator in order to simplify where clauses that specify that a value be less than or equal to some value and greater than or equal to some other value.

• *Find the loan number of those loans with loan amounts between $90,000 and $100,000 (that is, ≥ $90,000 and ≤ $100,000)*

select loan-number

from loan where amount between 90000 and 100000

THE FROM CLAUSE:

The from clause corresponds to the Cartesian product operation of the relational algebra. It lists the relations to be scanned in the evaluation of the expression.

• Find the Cartesian product borrower × loan select ∗ from borrower, loan

• Find the name and loan number of all customers having a loan at the Perryridge branch.

select distinct customer-name, borrower.loan-number

from borrower, loan

where borrower.loan-number = loan.loan-number and

branch-name = "Perryridg

THE RENAME OPERATION:

The SQL mechanism for renaming relations and attributes is accomplished through the as clause:

old-name as new-name

• *Find the name and loan number of all customers having a loan at the Perryridge branch; replace the column name loan-number with the name loan-id.*

select distinct customer-name, borrower.loan-number as loan-id

from borrower, loan

where borrower.loan-number = loan.loan-number and

branch-name = "Perryridge"

TUPLE VARIABLES:

Tuple Variables • Tuple variables are defined in the from clause via the use of the as clause.

• Find the customer names and their loan numbers for all customers having a loan at some branch.

select distinct customer-name, T.loan-number

from borrower as T, loan as S

where T.loan-number = S.loan-number

• *Find the names of all branches that have greater assets than some branch located in Brooklyn.*

select distinct T.branch-name

from branch as T, branch as S

where T.assets > S.assets and S.branch-city = "Brooklyn"

STRING OPERATIONS :

• SQL includes a string-matching operator for comparisons on character strings. Patterns are described using two special characters:

– percent (%). The % character matches any substring.

– underscore ( ). The character matches any character.

• *Find the names of all customers whose street includes the substring 'Main'.*

select customer-name

from customer

where customer-street like "%Main%"

• *Match the name "Main%" like "Main\%" escape "*

Ordering the Display of Tuples •

List in alphabetic order the names of all customers having a loan at Perryridge branch

select distinct customer-name

from borrower, loan

where borrower.loan-number = loan.loan-number and

branch-name = "Perryridge"

order by customer-name

• We may specify desc for descending order or asc for ascending order, for each attribute; ascending order is the default.

• SQL must perform a sort to fulfill an order by request. Since sorting a large number of tuples may be costly, it is desirable to sort only when necessary.

DUPLICATES:

• In relations with duplicates, SQL can define how many copies of tuples appear in the result

. • Multiset versions of some of the relational algebra operators – given multiset relations r1 and r2:

1. If there are c1 copies of tuple t1 in r1, and t1 satisfies selection σθ, then there are c1 copies of t1 in σθ(r1).

2. For each copy of tuple t1 in r1, there is a copy of tuple ΠA(t1) in ΠA(r1), where ΠA(t1) denotes the projection of the single tuple t1.

3. If there are c1 copies of tuple t1 in r1 and c2 copies of tuple t2 in r2, there are c1 × c2 copies of the tuple t1.t2 in r1 × r2.

Suppose relations r1 with schema (A, B) and r2 with schema (C) are the following multisets:

r1 = {(1, a), (2, a)} r2 = {(2), (3), (3)}

• Then ΠB(r1) would be {(a), (a)}, while ΠB(r1) × r2 would be

{(a, 2), (a, 2), (a, 3), (a, 3), (a, 3), (a, 3)}

• SQL duplicate semantics:

select A1, A2, ..., An

from r1, r2, ..., rm

where P is equivalent to the multiset version of the expression:

ΠA1, A2, ..., An(σP (r1 × r2 × ... × rm))

SET OPERATIONS:

• The set operations union, intersect, and except operate on relations and correspond to the relational algebra operations ∪, ∩, and −.

• Each of the above operations automatically eliminates duplicates; to retain all duplicates use the corresponding multiset versions union all, intersect all and except all. Suppose a tuple occurs m times in r and n times in s, then, it occurs:

– m + n times in r union all s

– min(m, n) times in r intersect all s

– max(0, m − n) times in r except all s

Find all customers who have a loan, an account, or both:

(select customer-name from depositor)

union

(select customer-name from borrower)

• *Find all customers who have both a loan and an account.*

(select customer-name from depositor)

Intersect

(select customer-name from borrower)

• *Find all customers who have an account but no loan.*

(select customer-name from depositor)

except

(select customer-name from borrower)

AGGREGATE FUNCTIONS:

These functions operate on the multiset of values of a column of a relation, and return a value

avg: average value
min: minimum value
max:maximum value
sum: sum of values
count: number of values

Find the average account balance at the Perryridge branch.

select avg (balance)
from account
where branch-name = "Perryridge"

Find the number of tuples in the customer relation

select count (*)
from customer

Find the number of depositors in the bank

select count (**distinct** customer-name)
from depositor

AGGREGATE FUNCTIONS – GROUP BY:

Find the number of depositors for each branch.

select branch-name, **count** (**distinct** customer-name)
from depositor, account
where depositor.account-number = account.account-number
group by branch-name

Note: Attributes in **select** clause outside of aggregate functions must appear in **group by** list

AGGREGATE FUNCTIONS – HAVING CLAUSE:

Find the names of all branches where the average account

balance is more than $1,200
select branch-name, **avg** (balance)
from account
group by branch-name
having avg (balance) > 1200
Note: predicates in the **having** clause are applied after the formation of groups

Null Values

_ It is possible for tuples to have a null value, denoted by null, for some of their attributes; null signifies an unknown value or that a value does not exist.
_ The result of any arithmetic expression involving null is null.
_ Roughly speaking, all comparisons involving null return false.
More precisely,

- Any comparison with null returns unknown
- (true **or** unknown) = true, (false **or** unknown) = unknown
  (unknown **or** unknown) = unknown,
  (true **and** unknown) = unknown, (false **and** unknown) = false,
  (unknown **and** unknown) = unknown
- Result of **where** clause predicate is treated as false if it evaluates to unknown
- "P **is unknown**" evaluates to true if predicate P evaluates to unknown

Find all loan numbers which appear in the loan relation with null values for amount.

select loan-number
from loan
where amount **is null**

Total all loan amounts
select sum (amount)
from loan

Above statement ignores null amounts; result is null if there is no non-null amount.
_ All aggregate operations except **count(*)** ignore tuples with null values on the aggregated attributes.

NESTED SUBQUERIES

_ SQL provides a mechanism for the nesting of subqueries.

_ A subquery is a **select-from-where** expression that is nested within another query.

_ A common use of subqueries is to perform tests for set membership, set comparisons, and set cardinality.

DERIVED RELATIONS:

Find the average account balance of those branches where the average account balance is greater than $1200.

select branch-name, avg-balance
from (**select** branch-name, **avg** (balance)
from account
group by branch-name)
as result (branch-name, avg-balance)
where avg-balance > 1200

Note that we do not need to use the **having** clause, since we compute in the **from** clause the temporary relation result, and the attributes of result can be used directly in the **where** clause.

VIEWS:

Provide a mechanism to hide certain data from the view of certain users. To create a view we use the command:

create view v **as** <query expression>

where:
– <query expression> is any legal expression
– the view name is represented by v

Example Queries:

A view consisting of branches and their customers

create view all-customer **as**
(**select** branch-name, customer-name
from depositor, account
where depositor.account-number = account.account-number)

union
(**select** branch-name, customer-name
from borrower, loan
where borrower.loan-number = loan.loan-number)

Find all customers of the Perryridge branch

select customer-name
from all-customer
where branch-name = "Perryridge"

MODIFICATION OF THE DATABASE – DELETION

Delete all account records at the Perryridge branch

delete from account
where branch-name = "Perryridge"

Delete all accounts at every branch located in Needham.

delete from account
where branch-name **in** (**select** branch-name
from branch
where branch-city = "Needham")
delete from depositor
where account-number **in** (**select** account-number
from branch, account
where branch-city = "Needham"
and branch.branch-name = account.branch-name)

Example Query:

Delete the records of all accounts with balances below the average at the bank

delete from account
where balance < (**select avg** (balance)
from account)
– Problem: as we delete tuples from deposit, the average balance changes

– Solution used in SQL:
1. First, compute **avg** balance and find all tuples to delete
2. Next, delete all tuples found above (without recomputing **avg** or retesting the tuples)

Modification of the Database – INSERTION

_ *Add a new tuple to account*

insert into account
values ("Perryridge", A-9732, 1200)
or equivalently

insert into account (branch-name, balance, account-number)
values ("Perryridge", 1200, A-9732)
_ Add a new tuple to account with balance set to null
insert into account
values ("Perryridge", A-777, null)

_ *Provide as a gift for all loan customers of the Perryridge* branch, a $200 savings account. Let the loan number serve as the account number for the new savings account

insert into account
select branch-name, loan-number, 200
from loan
where branch-name = "Perryridge"

insert into depositor
select customer-name, loan-number
from loan, borrower

where branch-name = "Perryridge"

and loan.account-number = borrower.account-number

Modification of the Database – UPDATES

_ Increase all accounts with balances over $10,000 by 6%, all other accounts receive 5%

.

– Write two **update** statements:

update account
set balance = balance _ 1.06
where balance > 10000

update account
set balance = balance _ 1.05
where balance _ 10000

UPDATE OF A VIEW:

_ Create a view of all loan data in the loan relation, hiding the amount attribute

create view branch-loan **as**

select branch-name, loan-number

from loan

_ Add a new tuple to branch-loan

insert into branch-loan

values ("Perryridge", "L-307")

This insertion must be represented by the insertion of the tuple

("Perryridge", "L-307", null)

into the loan relation.

_ Updates on more complex views are difficult or impossible to translate, and hence are disallowed.

JOINED RELATIONS:

_ Join operations take two relations and return as a result another relation.

_ These additional operations are typically used as subquery expressions in the **from** clause.

_ Join condition – defines which tuples in the two relations match, and what attributes are present in the result of the join.

_ Join type – defines how tuples in each relation that do not match any tuple in the other relation (based on the join condition) are treated.

Join Types
Inner Join
Left Outer Join
Right Outer join
Full outer join

Join Conditions
Natural
On <predicate>
Using(A1,A2,...,An)

DATA DEFINITION LANGUAGE (DDL):

Allows the specification of not only a set of relations but also information about each relation, including:

- The schema for each relation.
- The domain of values associated with each attribute.
- Integrity constraints.
- The set of indices to be maintained for each relation.
- Security and authorization information for each relation.
- The physical storage structure of each relation on disk

EMBEDDED SQL:

- The SQL standard defines embeddings of SQL in a variety of programming languages such as such as Pascal, PL/I, Fortran, C, and Cobol.
- A language in which SQL queries are embedded is referred to as a host language, and the SQL structures permitted in the host language comprise embedded SQL.
- The basic form of these languages follows that of the System R embedding of SQL into PL/I.
- EXEC SQL statement is used to identify embedded SQL requests to the preprocessor

  EXEC SQL <embedded SQL statement > END EXEC

Example Query:

From within a host language, find the names and account numbers of customers with more than the variable amount dollars in some account.

- Specify the query in SQL and declare a cursor for it

```
EXEC SQL
    declare c cursor for
    select customer-name, account-number
    from depositor, account
    where depositor.account-number = account.account-number
          and account.balance > :amount
END-EXEC
```

- The **open** statement causes the query to be evaluated

  EXEC SQL **open** c END-EXEC

- The **fetch** statement causes the values of one tuple in the query result to be placed in host language variables.

  EXEC SQL **fetch** c **into** :cn :an END-EXEC

  Repeated calls to **fetch** get successive tuples in the query result; a variable in the SQL communication area indicates

when end-of-file is reached.

- The **close** statement causes the database system to delete the temporary relation that holds the result of the query.

  EXEC SQL **close** c END-EXEC

DATA WAREHOUSING:

Data warehousing is the process of constructing and using a data warehouse. A data warehouse is constructed by integrating data from multiple heterogeneous sources that support analytical reporting, structured and/or ad hoc queries, and decision making. Data warehousing involves data cleaning, data integration, and data consolidations.

Using Data Warehouse Information

There are decision support technologies that help utilize the data available in a data warehouse. These technologies help executives to use the warehouse quickly and effectively. They can gather data, analyze it, and take decisions based on the information present in the warehouse. The information gathered in a warehouse can be used in any of the following domains −

- **Tuning Production Strategies** − The product strategies can be well tuned by repositioning the products and managing the product portfolios by comparing the sales quarterly or yearly.
- **Customer Analysis** − Customer analysis is done by analyzing the customer's buying preferences, buying time, budget cycles, etc.
- **Operations Analysis** − Data warehousing also helps in customer relationship management, and making environmental corrections. The information also allows us to analyze business operations.

Integrating Heterogeneous Databases

To integrate heterogeneous databases, we have two approaches −

- Query-driven Approach
- Update-driven Approach

Query-Driven Approach

This is the traditional approach to integrate heterogeneous databases. This approach was used to build wrappers and integrators on top of multiple heterogeneous databases. These integrators are also known as mediators.

Process of Query-Driven Approach

- When a query is issued to a client side, a metadata dictionary translates the query into an appropriate form for individual heterogeneous sites involved.
- Now these queries are mapped and sent to the local query processor.
- The results from heterogeneous sites are integrated into a global answer set.

Disadvantages

- Query-driven approach needs complex integration and filtering processes.
- This approach is very inefficient.
- It is very expensive for frequent queries.
- This approach is also very expensive for queries that require aggregations.

Update-Driven Approach

This is an alternative to the traditional approach. Today's data warehouse systems follow update-driven approach rather than the traditional approach discussed earlier. In update-driven approach, the information from multiple heterogeneous sources are integrated in advance and are stored in a warehouse. This information is available for direct querying and analysis.

Advantages

This approach has the following advantages −

- This approach provide high performance.
- The data is copied, processed, integrated, annotated, summarized and restructured in semantic data store in advance.
- Query processing does not require an interface to process data at local sources.

Functions of Data Warehouse Tools and Utilities

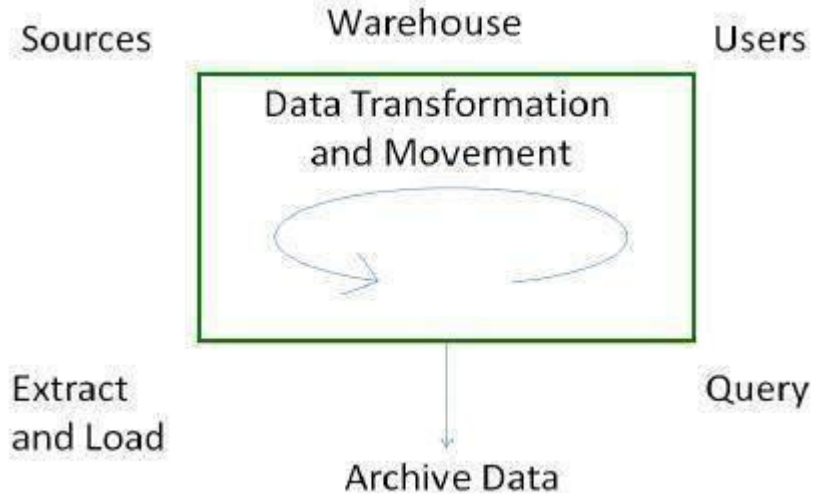
The following are the functions of data warehouse tools and utilities −

- **Data Extraction** − Involves gathering data from multiple heterogeneous sources.
- **Data Cleaning** − Involves finding and correcting the errors in data.
- **Data Transformation** − Involves converting the data from legacy format to warehouse format.
- **Data Loading** − Involves sorting, summarizing, consolidating, checking integrity, and building indices and partitions.
- **Refreshing** − Involves updating from data sources to warehouse.

PROCESS FLOW IN DATA WAREHOUSE

There are four major processes that contribute to a data warehouse −

- Extract and load the data.
- Cleaning and transforming the data.
- Backup and archive the data.
- Managing queries and directing them to the appropriate data sources.



Extract and Load Process

Data extraction takes data from the source systems. Data load takes the extracted data and loads it into the data warehouse.

Note − Before loading the data into the data warehouse, the information extracted from the external sources must be reconstructed.

Controlling the Process

Controlling the process involves determining when to start data extraction and the consistency check on data. Controlling process ensures that the tools, the logic modules, and the programs are executed in correct sequence and at correct time.

When to Initiate Extract

Data needs to be in a consistent state when it is extracted, i.e., the data warehouse should represent a single, consistent version of the information to the user.

For example, in a customer profiling data warehouse in telecommunication sector, it is illogical to merge the list of customers at 8 pm on Wednesday from a customer database with the customer subscription events up to 8 pm on Tuesday. This would mean that we are finding the customers for whom there are no associated subscriptions.

Loading the Data

After extracting the data, it is loaded into a temporary data store where it is cleaned up and made consistent.

Note − Consistency checks are executed only when all the data sources have been loaded into the temporary data store.

Clean and Transform Process

Once the data is extracted and loaded into the temporary data store, it is time to perform Cleaning and Transforming. Here is the list of steps involved in Cleaning and Transforming

- Clean and transform the loaded data into a structure
- Partition the data
- Aggregation

Clean and Transform the Loaded Data into a Structure

Cleaning and transforming the loaded data helps speed up the queries. It can be done by making the data consistent −

- within itself.
- with other data within the same data source.
- with the data in other source systems.

- with the existing data present in the warehouse.

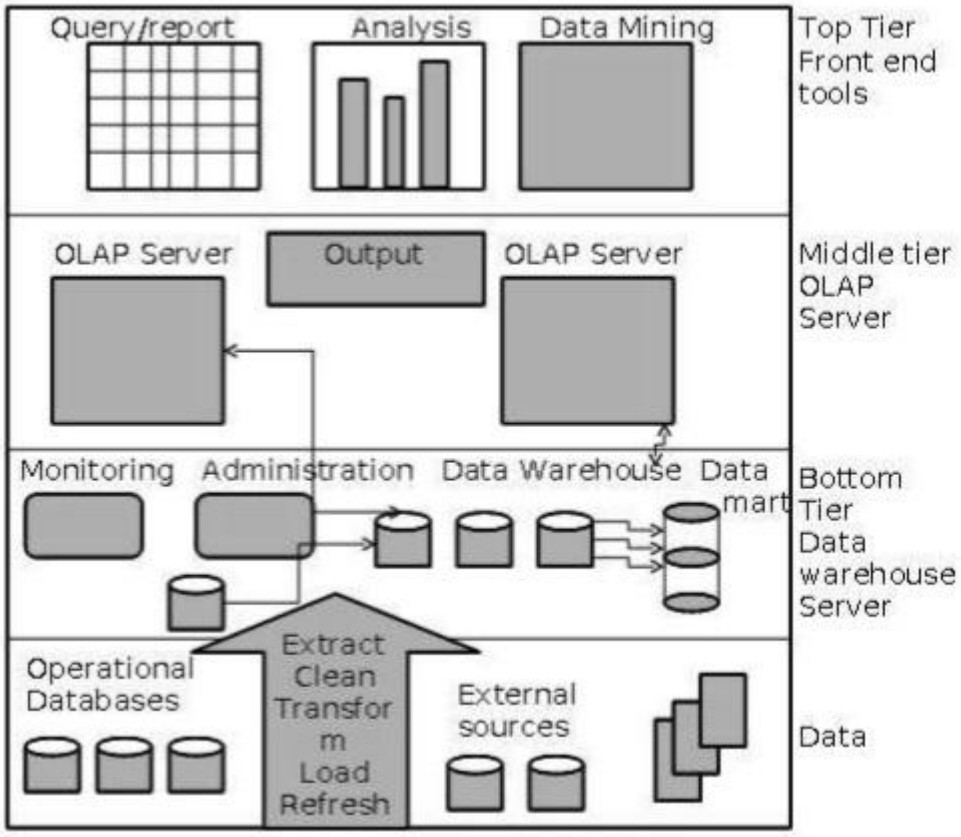
Transforming involves converting the source data into a structure. Structuring the data increases the query performance and decreases the operational cost. The data contained in a data warehouse must be transformed to support performance requirements and control the ongoing operational costs.

THREE-TIER DATA WAREHOUSE ARCHITECTURE

Generally a data warehouses adopts a three-tier architecture. Following are the three tiers of the data warehouse architecture.

- **Bottom Tier** − The bottom tier of the architecture is the data warehouse database server. It is the relational database system. We use the back end tools and utilities to feed data into the bottom tier. These back end tools and utilities perform the Extract, Clean, Load, and refresh functions.
- **Middle Tier** − In the middle tier, we have the OLAP Server that can be implemented in either of the following ways.
    - By Relational OLAP (ROLAP), which is an extended relational database management system. The ROLAP maps the operations on multidimensional data to standard relational operations.
    - By Multidimensional OLAP (MOLAP) model, which directly implements the multidimensional data and operations.
- **Top-Tier** − This tier is the front-end client layer. This layer holds the query tools and reporting tools, analysis tools and data mining tools.

The following diagram depicts the three-tier architecture of data warehouse −



This process performs the following functions −

- manages the queries.
- helps speed up the execution time of queris.
- directs the queries to their most effective data sources.
- ensures that all the system sources are used in the most effective way.
- monitors actual query profiles.

The information generated in this process is used by the warehouse management process to determine which aggregations to generate. This process does not generally operate during the regular load of information into data warehouse.

Data transformation:

Data transformation is the process of converting data or information from one format to another, usually from the format of a source system into the required format of a new destination system. The usual process involves converting documents, but data conversions sometimes involve the conversion of a program from one computer language to another to

enable the program to run on a different platform. The usual reason for this data migration is the adoption of a new system that's totally different from the previous one.

Data Transformation involves two key phases:

Data Mapping: The assignment of elements from the source base or system toward the destination to capture all transformations that occur. This is made more complicated when there are complex transformations like many-to-one or one-to-many rules for transformation.

Code Generation: The creation of the actual transformation program. The resulting data map specification is used to create an executable program to run on computer systems.
Commonly used transformational languages:

- Perl: A high-level procedural and object-oriented language capable of powerful operations
- AWK: One of the oldest languages and a popular TXT transformation language
- XSLT: An XML data transformation language
- TXL: A prototyping language mostly used for source code transformation
- Template Languages and Processors: These specialize in data-to-document transformation

USER INTERFACE:

User interface is the front-end application view to which user interacts in order to use the software. User can manipulate and control the software as well as hardware by means of user interface. Today, user interface is found at almost every place where digital technology exists, right from computers, mobile phones, cars, music players, airplanes, ships etc.

User interface is part of software and is designed such a way that it is expected to provide the user insight of the software. UI provides fundamental platform for human-computer interaction.

UI can be graphical, text-based, audio-video based, depending upon the underlying hardware and software combination. UI can be hardware or software or a combination of both.

The software becomes more popular if its user interface is:

- Attractive
- Simple to use
- Responsive in short time

- Clear to understand
- Consistent on all interfacing screens

UI is broadly divided into two categories:

- Command Line Interface
- Graphical User Interface

Command Line Interface (CLI)

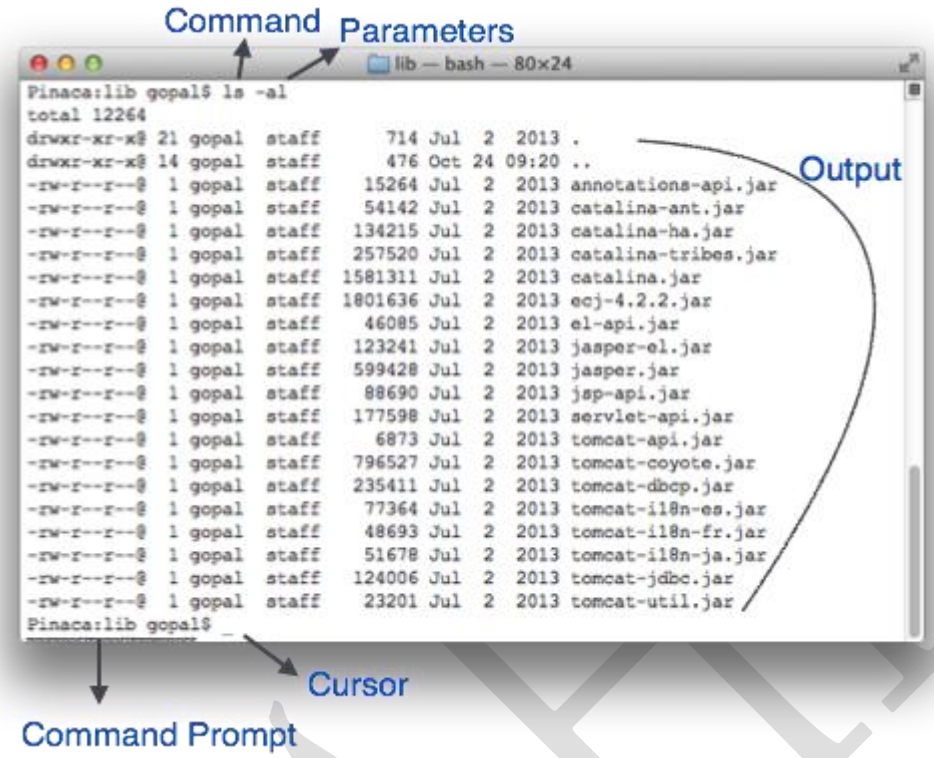
CLI has been a great tool of interaction with computers until the video display monitors came into existence. CLI is first choice of many technical users and programmers. CLI is minimum interface a software can provide to its users.

CLI provides a command prompt, the place where the user types the command and feeds to the system. The user needs to remember the syntax of command and its use. Earlier CLI were not programmed to handle the user errors effectively.

A command is a text-based reference to set of instructions, which are expected to be executed by the system. There are methods like macros, scripts that make it easy for the user to operate.

CLI uses less amount of computer resource as compared to GUI.

CLI Elements



A text-based command line interface can have the following elements:

- **Command Prompt** - It is text-based notifier that is mostly shows the context in which the user is working. It is generated by the software system.
- **Cursor** - It is a small horizontal line or a vertical bar of the height of line, to represent position of character while typing. Cursor is mostly found in blinking state. It moves as the user writes or deletes something.
- **Command** - A command is an executable instruction. It may have one or more parameters. Output on command execution is shown inline on the screen. When output is produced, command prompt is displayed on the next line.

Graphical User Interface

Graphical User Interface provides the user graphical means to interact with the system. GUI can be combination of both hardware and software. Using GUI, user interprets the software.

Typically, GUI is more resource consuming than that of CLI. With advancing technology, the programmers and designers create complex GUI designs that work with more efficiency, accuracy and speed.

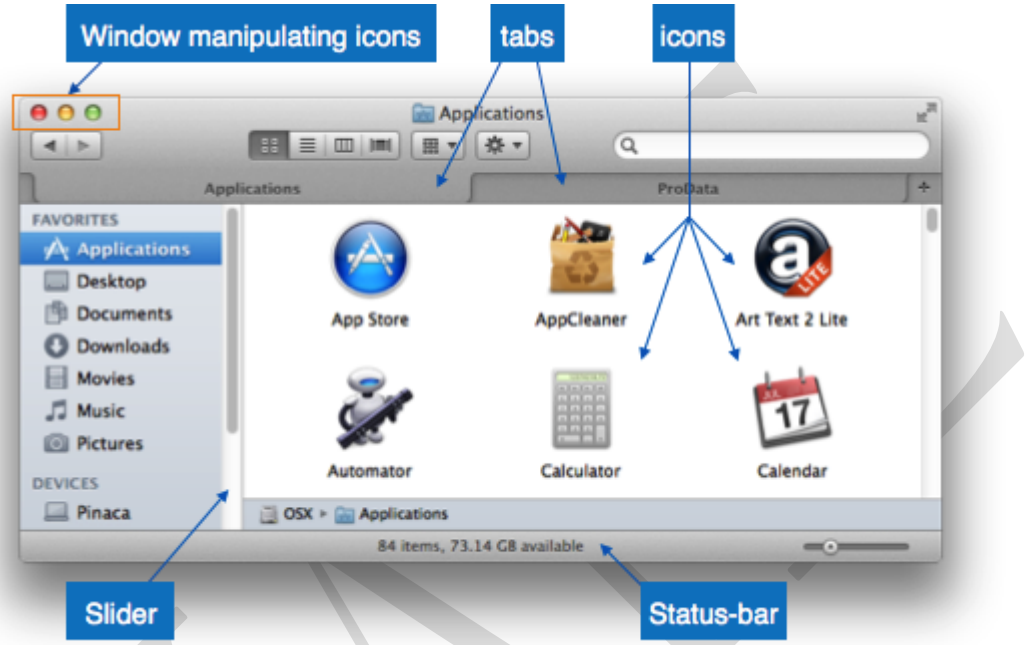
Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401C Semester: IV Year: 2017-2019

GUI Elements

GUI provides a set of components to interact with software or hardware.

Every graphical component provides a way to work with the system. A GUI system has following elements such as:



- **Window** - An area where contents of application are displayed. Contents in a window can be displayed in the form of icons or lists, if the window represents file structure. It is easier for a user to navigate in the file system in an exploring window. Windows can be minimized, resized or maximized to the size of screen. They can be moved anywhere on the screen. A window may contain another window of the same application, called child window.
- **Tabs** - If an application allows executing multiple instances of itself, they appear on the screen as separate windows. **Tabbed Document Interface** has come up to open multiple documents in the same window. This interface also helps in viewing preference panel in application. All modern web-browsers use this feature.
- **Menu** - Menu is an array of standard commands, grouped together and placed at a visible place (usually top) inside the application window. The menu can be programmed to appear or hide on mouse clicks.

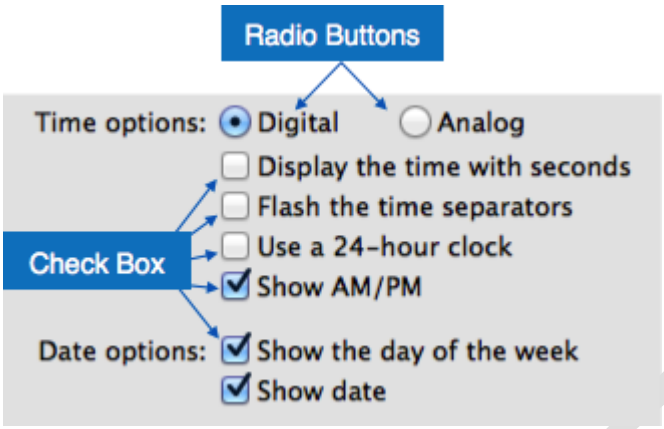
- **Icon -** An icon is small picture representing an associated application. When these icons are clicked or double clicked, the application window is opened. Icon displays application and programs installed on a system in the form of small pictures.
- **Cursor -** Interacting devices such as mouse, touch pad, digital pen are represented in GUI as cursors. On screen cursor follows the instructions from hardware in almost real-time. Cursors are also named pointers in GUI systems. They are used to select menus, windows and other application features.

Application specific GUI components

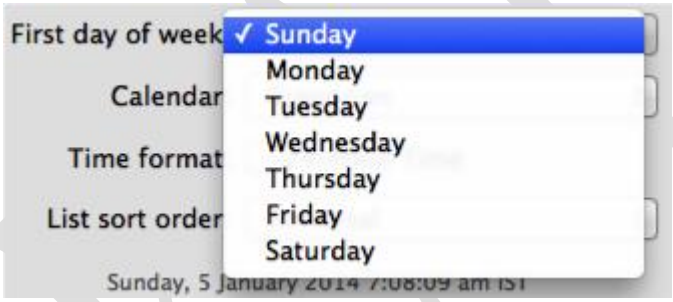
A GUI of an application contains one or more of the listed GUI elements:

- **Application Window -** Most application windows uses the constructs supplied by operating systems but many use their own customer created windows to contain the contents of application.
- **Dialogue Box -** It is a child window that contains message for the user and request for some action to be taken. For Example: Application generate a dialogue to get confirmation from user to delete a file.

- **Text-Box -** Provides an area for user to type and enter text-based data.
- **Buttons -** They imitate real life buttons and are used to submit inputs to the software.



- **Radio-button** - Displays available options for selection. Only one can be selected among all offered.
- **Check-box** - Functions similar to list-box. When an option is selected, the box is marked as checked. Multiple options represented by check boxes can be selected.
- **List-box** - Provides list of available items for selection. More than one item can be selected.



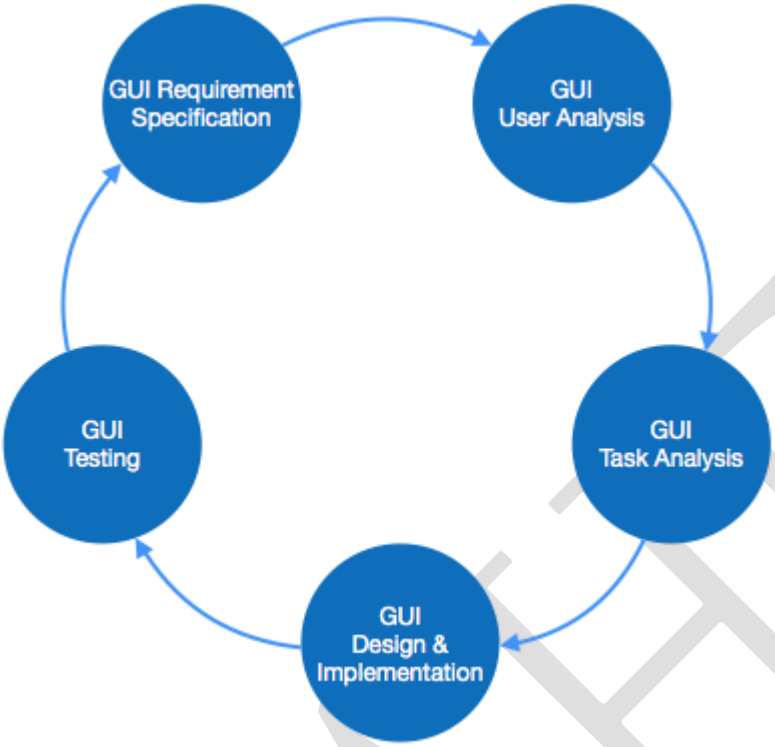
Other impressive GUI components are:

- Sliders
- Combo-box
- Data-grid
- Drop-down list

User Interface Design Activities

There are a number of activities performed for designing user interface. The process of GUI design and implementation is alike SDLC. Any model can be used for GUI implementation among Waterfall, Iterative or Spiral Model.

A model used for GUI design and development should fulfill these GUI specific steps.



- **GUI Requirement Gathering** - The designers may like to have list of all functional and non-functional requirements of GUI. This can be taken from user and their existing software solution.
- **User Analysis** - The designer studies who is going to use the software GUI. The target audience matters as the design details change according to the knowledge and competency level of the user. If user is technical savvy, advanced and complex GUI can be incorporated. For a novice user, more information is included on how-to of software.
- **Task Analysis** - Designers have to analyze what task is to be done by the software solution. Here in GUI, it does not matter how it will be done. Tasks can be represented in hierarchical manner taking one major task and dividing it further into smaller sub-tasks. Tasks provide goals for GUI presentation. Flow of information among sub-tasks determines the flow of GUI contents in the software.
- **GUI Design & implementation** - Designers after having information about requirements, tasks and user environment, design the GUI and implements into code

and embed the GUI with working or dummy software in the background. It is then self-tested by the developers.

- **Testing** - GUI testing can be done in various ways. Organization can have in-house inspection, direct involvement of users and release of beta version are few of them. Testing may include usability, compatibility, user acceptance etc.

GUI Implementation Tools

There are several tools available using which the designers can create entire GUI on a mouse click. Some tools can be embedded into the software environment (IDE).

GUI implementation tools provide powerful array of GUI controls. For software customization, designers can change the code accordingly.

There are different segments of GUI tools according to their different use and platform.

Example

Mobile GUI, Computer GUI, Touch-Screen GUI etc. Here is a list of few tools which come handy to build GUI:

- FLUID
- AppInventor (Android)
- LucidChart
- Wavemaker
- Visual Studio

User Interface Golden rules

The following rules are mentioned to be the golden rules for GUI design, described by Shneiderman and Plaisant in their book (Designing the User Interface).

- **Strive for consistency** - Consistent sequences of actions should be required in similar situations. Identical terminology should be used in prompts, menus, and help screens. Consistent commands should be employed throughout.
- **Enable frequent users to use short-cuts** - The user's desire to reduce the number of interactions increases with the frequency of use. Abbreviations, function keys, hidden commands, and macro facilities are very helpful to an expert user.

- **Offer informative feedback** - For every operator action, there should be some system feedback. For frequent and minor actions, the response must be modest, while for infrequent and major actions, the response must be more substantial.
- **Design dialog to yield closure** - Sequences of actions should be organized into groups with a beginning, middle, and end. The informative feedback at the completion of a group of actions gives the operators the satisfaction of accomplishment, a sense of relief, the signal to drop contingency plans and options from their minds, and this indicates that the way ahead is clear to prepare for the next group of actions.
- **Offer simple error handling** - As much as possible, design the system so the user will not make a serious error. If an error is made, the system should be able to detect it and offer simple, comprehensible mechanisms for handling the error.
- **Permit easy reversal of actions** - This feature relieves anxiety, since the user knows that errors can be undone. Easy reversal of actions encourages exploration of unfamiliar options. The units of reversibility may be a single action, a data entry, or a complete group of actions.
- **Support internal locus of control** - Experienced operators strongly desire the sense that they are in charge of the system and that the system responds to their actions. Design the system to make users the initiators of actions rather than the responders.
- **Reduce short-term memory load** - The limitation of human information processing in short-term memory requires the displays to be kept simple, multiple page displays be consolidated, window-motion frequency be reduced, and sufficient training time be allotted for codes, mnemonics, and sequences of actions.

A **distributed database** is a collection of multiple interconnected databases, which are spread physically across various locations that communicate via a computer network.

Features

- Databases in the collection are logically interrelated with each other. Often they represent a single logical database.
- Data is physically stored across multiple sites. Data in each site can be managed by a DBMS independent of the other sites.

- The processors in the sites are connected via a network. They do not have any multiprocessor configuration.
- A distributed database is not a loosely connected file system.
- A distributed database incorporates transaction processing, but it is not synonymous with a transaction processing system.

Advantages of Distributed Databases

Following are the advantages of distributed databases over centralized databases.

Modular Development − If the system needs to be expanded to new locations or new units, in centralized database systems, the action requires substantial efforts and disruption in the existing functioning. However, in distributed databases, the work simply requires adding new computers and local data to the new site and finally connecting them to the distributed system, with no interruption in current functions.

More Reliable − In case of database failures, the total system of centralized databases comes to a halt. However, in distributed systems, when a component fails, the functioning of the system continues may be at a reduced performance. Hence DDBMS is more reliable.

Better Response − If data is distributed in an efficient manner, then user requests can be met from local data itself, thus providing faster response. On the other hand, in centralized systems, all queries have to pass through the central computer for processing, which increases the response time.

Lower Communication Cost − In distributed database systems, if data is located locally where it is mostly used, then the communication costs for data manipulation can be minimized. This is not feasible in centralized systems.

KARPAGAM ACADEMY OF HIGHER EDUCATION						
Department of Management						
Unit 3- Multiple Choice Questions- Each Question Carry ONE Mark						
S.no	**Question**	**Option 1**	**Option 2**	**Option 3**	**Option 4**	**Answer**
1	SQL allows use of logical connectives and, or, and not in the_____	Define clause	From clause	Where clause	Select Clause	**Where clause**
2	A record-based model, can be thought of an example of_____	Logical model	Relational model	Physical model	Logical model	**Relational model**
3	AS' clause is used in SQL for_____	Selection operation	Rename Operation	Join operation	Projection Operation	**Rename Operation**
4	_____is a 'tuple	A row or record in a database table.	Another name for the key linking different tables in a database.	An attribute attached to a record	Another name for a table in an RDBMS	**A row or record in a database table.**
5	To remove duplicate rows from the results of an SQL SELECT statement, the _____ qualifier specified must be included	ONLY	UNIQUE	DISTINCT	SINGLE	**DISTINCT**
6	Query for finding department names of all instructors, is_____	select dept_name about instructor;	define dept_name from instructor;	select dept_name from instructor;	define dept_name about instructor;	**select dept_name from instructor;**
7	Basic structure of an SQL query consists of clauses of type_____	2 types	3 types	4 types	5 types	**3 types**
8	The command to remove rows from a table	REMOVE FROM CUSTOM	DROP FROM CUSTOM	DELETE FROM CUSTOM	UPDATE FROM CUSTOM	**DELETE FROM CUSTOM**

	'CUSTOMER' is_____	ER	ER	ER WHERE	ER	**ER WHERE**
9	Count function in SQL returns the number of_____	values	distinct values	groups	columns	**values**
10	The statement in SQL which allows to change the definition of a table is_____	Alter	Update	Create	Seelct	**Alter**
11						
12	Check option clause defined by SQL views, is expressed at end of_____	View definition	View schema	From clause	Where clause	**View definition**
13	User defined data types in SQL standard can be expressed as_____	List datatype	Create type	Describe type	Define datatype	**Create type**
14	In SQL, pattern must be followed by clauses, is said to be of_____	Select, where, from	From, select, where	Select, from, where	From, where, select	**Select, from, where**
15	A powerful pattern matching operation named "similar to" was introduced in_____	SQL:1997	SQL:1998	SQL:1999	SQL:2000	**SQL:1999**
16	The union operation is represented by_____	∩	U	_	*	**U**
17	The intersection operator is used to get the _____ tuples.	Different	Common	All	Repeating	**Common**
18	The number of attributes in relation is called as its_____	Cardinality	Degree	Tuples	Entity	**Degree**

19	_____ is correct syntax for applying UNION operator	SELECT column_name(s) FROM table_name1 UNION table_name2	SELECT column_name(s) FROM table_name1 SELECT column_name(s) FROM table_name2	UNION SELECT column_name(s) FROM table_name1 SELECT column_name(s) FROM table_name2	SELECT FROM table_name1 AND table_name2	**SELECT column_name(s) FROM table_name1 SELECT column_name(s) FROM table_name2**
20	Observe the given SQL query and choose the correct option. SELECT branch_name, COUNT (DISTINCT customer_name) FROM depositor, account WHERE depositor.account_number = account.account_number GROUP BY branch_id	The query is syntactically correct but gives the wrong answer	The query is syntactically wrong	The query is syntactically correct and gives the correct answer	The query contains one or more wrongly named clauses.	**The query is syntactically wrong**
21	The _____ aggregation operation adds up all the values of the attribute	add	avg	max	sum	**sum**
22	_____values does the count(*) function ignore	Repetitive values	Null values	Characters	Integers	**Null values**
23	A _____ indicates an absent value that may exist but be unknown or that may not exist at all.	Empty tuple	New value	Null value	Old value	**Null value**

24	_____is the subset of SQL commands used to manipulate Oracle Database structures, including tables	Data Definition Language(DDL)	Data Manipulation Language(DML)	DML and DDL	DBMS	**Data Definition Language(DDL)**
25	In SQL, which command(s) is(are) used to change a table's storage characteristics_____	ALTER TABLE	MODIFY TABLE	CHANGE TABLE	DELETE TABLE	**ALTER TABLE**
26	In SQL, which of the following is not a data definition language commands_____	RENAME	REVOKE	REVOKE	UPDATE	**RENAME**
27	_____clause is an additional filter that is applied to the result.	Select	Group-by	Having	Order by	**Having**
28	_____is a subquery	A subquery is a select-from-where expression that is nested within another query	A subquery is any query that is nested within another query	A subquery is a relation that is externally specified which can be used to handle data in querie	A subquery is a condition that excludes all the invalid tuples from the database	**A subquery is a select-from-where expression that is nested within another query**
29	If a set is a collection of values given by the select clause, The _____ connective tests for set membership	within	include	under	in	**in**
30	The _____ comparison checker is used to check "each and every"	all	and	every	each	**all**

	condition					
31	The _____ construct returns true if a given tuple is present in the subquery	not exists	present	not present	exists	**exists**
32	_____is a correlated sub-query	An independent query that uses the correlation name of another independent query.	A sub-query that uses the correlation name of an outer query	A sub-query that substitutes the names of the outer query	A sub-query that does not depend on its outer query's correlation names	**A sub-query that uses the correlation name of an outer query**
34	The _____ construct returns true if the argument in the sub-query is void of duplicates	not null	not unique	unique	null	**unique**
35	QL subqueries that can occur wherever a value is permitted provided the subquery gives only one tuple with a single attribute are called _____	Exact Subqueries	Vector Subqueries	Positive Subqueries	Scalar Subqueries	**Scalar Subqueries**
36	Select * from student join takes using (ID); The above query is equivalent to :	Select * from student inner join takes using (ID);	Select * from student outer join takes using (ID);	Select * from student left outer join takes using (ID);	All of the mentioned	**Select * from student inner join takes using (ID);**
37	Exist construct returns true as a value, if argument subquery is_____	Empty	Not empty	Exist	Not exist	**Not empty**
38	A subquery that uses a correlation name from an outer	Derived Subquery	Induced subQuery	Deduced Subquery	Correlated Subquery	**Correlated Subquery**

	query is called a_____					
39	SQL comparison = all is not same to operator_____	In	Like	As	Unlike	**Like**
40	Scalar subqueries can occur in_____	Select Clause	From Clause	Where Clause	As clause	**From Clause**
41	The function that an entity plays in a relationship is called that entity's _____	Participation	Position	Role	Instance	**Role**
42	The attribute *name* could be structured as an attribute consisting of first name, middle initial, and last name. This type of attribute is called_____	simple attribute	Composite attribute	Multivalued attribute	Derived attribute	**Composite attribute**
43	The attribute AGE is calculated from DATE_OF_BIRTH. The attribute AGE is_____	Single valued	Multi valued	Composite	Derived	**Derived**
44	Not applicable condition can be represented in relation entry as_____	NA	0	Null	Blank Space	**Null**
45	n a relation between the entities the type and condition of the relation should be specified. That is called as_____attribute.	Desciptive	Derived	Recursive	Relative	**Desciptive**

46	n database management systems, record which contains all data regarding tuples of database is called_____	Statement record	Environment record	Description record	Connection record	**Description record**
47	Type of iterator which is used to list types of attributes that are included in query result is called_____	positional iterator	named iterator	unnamed iterator	non-positioned iterator	**positional iterator**
48	If SQL query results in an error, database system stores an error diagnostic in SQL_____	Communication area variables	Connection area variables	SQL area variables	Programming area variables	**Communication area variables**
49	A relation is conceptually a set, tuples of result of a query are in some_____	Physical Order	Random Order	Fixed Physical order	No order	**Fixed Physical order**
50	_____type of join is needed when you wish to include rows that do not have matching values	Equi-join	Natural join	Outer join	Allthe above	**Outer join**
51	The following SQL is which type of join: SELECT CUSTOMER_T. CUSTOMER_ID, ORDER_T. CUSTOMER_ID, NAME, ORDER_ID FROM CUSTOMER_T, ORDER_T WHERE CUSTOMER_T. CUSTOMER_I	Equi-join	Natural join	Outer join	Cartesian join	**Equi-join**

	D = ORDER_T. CUSTOMER_ID					
52	Embedded SQL is_____	Hard-coded SQL statements in a program language such as Java	The process of making an application capable of generating specific SQL code on the fly	Hard-coded SQL statements in a procedure	Hard-coded SQL statements in a trigger	**Hard-coded SQL statements in a program language such as Java**
53	User defined function in SQL Server can return_____	Scalar value	Set of values	Result set	Entity	**Result set**
54	_____ is a subject-oriented, integrated, time-variant, nonvolatile collection of data in support of Management decisions	Data Mining	Data Warehousing	Web Mining	Text Mining	**Data Warehousing**
55	The data Warehouse is_____	read only	write only	read write only	Manipulate	**read only**
56	In three-tier architecture, intermediate layer between database and client servers is classified as_____	Functional Server	Transaction Server	Application server	Disk server	**Application server**
57	Types of architectures of DBMS are_____	single-tier architecture	Two-tier Architecture	Three-tier architecture	DDL	**Three-tier architecture**
58	Graphical representation of database description is	dynamic schema diagram	schema diagram	structure diagram	entity path diagram	**schema diagram**

	called_____					
59	_____incorporates data, architectural, interface, and procedural representations of the software	design model	user's model	mental image	system image	**design model**
60	_____clause is an additional filter that is applied to the result.	Select	Group-by	Having	Order by	**Having**

Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401C Semester: IV Year: 2017-2019

SQL-Background-Basic Structure-Set operations-Aggregate Functions-Null values-Nested-Sub Queries-Derived relations-Views-Modification of the database-Joined relations-Data definition language-Embedded SQL features-Data warehousing-Concepts-Architecture-Data transformation-User Interface-Distributed database

UNIT 3

SQL:

Structured Query Language (SQL) is a standard computer language for relational database management and data manipulation. SQL is used to query, insert, update and modify data. Most relational databases support SQL, which is an added benefit for database administrators (DBAs), as they are often required to support databases across several different platforms.

First developed in the early 1970s at IBM by Raymond Boyce and Donald Chamberlin, SQL was commercially released by Relational Software Inc. (now known as Oracle Corporation) in 1979. The current standard SQL version is voluntary, vendor-compliant and monitored by the American National Standards Institute (ANSI). Most major vendors also have proprietary versions that are incorporated and built on ANSI SQL, e.g., SQL*Plus (Oracle), and Transact-SQL (T-SQL) (Microsoft).

Basic Structure

1. Basic structure of an SQL expression consists of **select, from** and **where** clauses.
   - **select** clause lists attributes to be copied - corresponds to relational algebra **project**.
   - **from** clause corresponds to Cartesian product - lists relations to be used.
   - **where** clause corresponds to selection predicate in relational algebra.
2. Typical query has the form

   select $A_1, A_2, \ldots, A_n$

   from $r_1, r_2, \ldots, r_m$

   where *P*

   where each A_i represents an attribute, each r_i a relation, and *P* is a predicate.

3. This is equivalent to the relational algebra expression

$$\Pi_{A_1, A_2, \ldots, A_n}(\sigma_P(r_1 \times r_2 \times \ldots \times r_m))$$

- If the where clause is omitted, the predicate *P* is true.
- The list of attributes can be replaced with a * to select all.
- SQL forms the Cartesian product of the relations named, performs a selection using the predicate, then projects the result onto the attributes named.
- The result of an SQL query is a relation.
- SQL may internally convert into more efficient expressions.

• SQL is based on set and relational operations with certain modifications and enhancements
•

A typical SQL query has the form:

select A1, A2, ..., An from r1, r2, ..., rm where P –

A is represent attributes –

R is represent relations –

P is a predicate.

• This query is equivalent to the relational algebra expression:

ΠA1, A2, ..., An(σP (r1 × r2 × ... × rm))

• The result of an SQL query is a relation.

The select clause corresponds to the projection operation of the relational algebra. It is used to list the attributes desired in the result of a query.

• *Find the names of all branches in the loan relation*

select branch-name

from loan

In the "pure" relational algebra syntax, this query would be:

Πbranch-name (loan)

• An asterisk in the select clause denotes "all attributes"

select ∗ from loan

SQL allows duplicates in relations as well as in query results.

• To force the elimination of duplicates, insert the keyword distinct after select.

Find the names of all branches in the loan relation, and remove duplicates

select distinct branch-name

from loan

• *The keyword all specifies that duplicates not be removed.*

select all branch-name

from loan

The select clause can contain arithmetic expressions involving the operators, +, −, ∗, and /, and operating on constants or attributes of tuples.

The query:

select branch-name, loan-number, amount ∗ 100

from loan

would return a relation which is the same as the loan relation, except that the attribute amount is multiplied by 100

The where clause corresponds to the selection predicate of the relational algebra. It consists of a predicate involving attributes of the relations that appear in the from clause.

• *Find all loan numbers for loans made at the Perryridge branch with loan amounts greater than $1200.*

select loan-number

from loan

where branch-name = "Perryridge" and amount > 1200

• SQL uses the logical connectives **and, or, and not**. It allows the use of arithmetic expressions as operands to the comparison operators

SQL includes a between comparison operator in order to simplify where clauses that specify that a value be less than or equal to some value and greater than or equal to some other value.

• *Find the loan number of those loans with loan amounts between $90,000 and $100,000 (that is, ≥ $90,000 and ≤ $100,000)*

select loan-number

from loan where amount between 90000 and 100000

THE FROM CLAUSE:

The from clause corresponds to the Cartesian product operation of the relational algebra. It lists the relations to be scanned in the evaluation of the expression.

• Find the Cartesian product borrower × loan select ∗ from borrower, loan

• Find the name and loan number of all customers having a loan at the Perryridge branch.

select distinct customer-name, borrower.loan-number

from borrower, loan

where borrower.loan-number = loan.loan-number and

branch-name = "Perryridg

THE RENAME OPERATION:

The SQL mechanism for renaming relations and attributes is accomplished through the as clause:

old-name as new-name

• *Find the name and loan number of all customers having a loan at the Perryridge branch; replace the column name loan-number with the name loan-id.*

select distinct customer-name, borrower.loan-number as loan-id

from borrower, loan

where borrower.loan-number = loan.loan-number and

branch-name = "Perryridge"

TUPLE VARIABLES:

Tuple Variables • Tuple variables are defined in the from clause via the use of the as clause.

• Find the customer names and their loan numbers for all customers having a loan at some branch.

select distinct customer-name, T.loan-number

from borrower as T, loan as S

where T.loan-number = S.loan-number

• *Find the names of all branches that have greater assets than some branch located in Brooklyn.*

select distinct T.branch-name

from branch as T, branch as S

where T.assets > S.assets and S.branch-city = "Brooklyn"

STRING OPERATIONS :

• SQL includes a string-matching operator for comparisons on character strings. Patterns are described using two special characters:

– percent (%). The % character matches any substring.

– underscore ( ). The character matches any character.

• *Find the names of all customers whose street includes the substring 'Main'.*

select customer-name

from customer

where customer-street like "%Main%"

• *Match the name "Main%" like "Main\%" escape "*

Ordering the Display of Tuples •

List in alphabetic order the names of all customers having a loan at Perryridge branch

select distinct customer-name

from borrower, loan

where borrower.loan-number = loan.loan-number and

branch-name = "Perryridge"

order by customer-name

• We may specify desc for descending order or asc for ascending order, for each attribute; ascending order is the default.

• SQL must perform a sort to fulfill an order by request. Since sorting a large number of tuples may be costly, it is desirable to sort only when necessary.

DUPLICATES:

• In relations with duplicates, SQL can define how many copies of tuples appear in the result

. • Multiset versions of some of the relational algebra operators – given multiset relations r1 and r2:

1. If there are c1 copies of tuple t1 in r1, and t1 satisfies selection σθ, then there are c1 copies of t1 in σθ(r1).

2. For each copy of tuple t1 in r1, there is a copy of tuple ΠA(t1) in ΠA(r1), where ΠA(t1) denotes the projection of the single tuple t1.

3. If there are c1 copies of tuple t1 in r1 and c2 copies of tuple t2 in r2, there are c1 × c2 copies of the tuple t1.t2 in r1 × r2.

Suppose relations r1 with schema (A, B) and r2 with schema (C) are the following multisets:

r1 = {(1, a), (2, a)} r2 = {(2), (3), (3)}

• Then ΠB(r1) would be {(a), (a)}, while ΠB(r1) × r2 would be

{(a, 2), (a, 2), (a, 3), (a, 3), (a, 3), (a, 3)}

• SQL duplicate semantics:

select A1, A2, ..., An

from r1, r2, ..., rm

where P is equivalent to the multiset version of the expression:

ΠA1, A2, ..., An(σP (r1 × r2 × ... × rm))

SET OPERATIONS:

• The set operations union, intersect, and except operate on relations and correspond to the relational algebra operations ∪, ∩, and −.

• Each of the above operations automatically eliminates duplicates; to retain all duplicates use the corresponding multiset versions union all, intersect all and except all. Suppose a tuple occurs m times in r and n times in s, then, it occurs:

– m + n times in r union all s

– min(m, n) times in r intersect all s

– max(0, m − n) times in r except all s

Find all customers who have a loan, an account, or both:

(select customer-name from depositor)

union

(select customer-name from borrower)

• *Find all customers who have both a loan and an account.*

(select customer-name from depositor)

Intersect

(select customer-name from borrower)

• *Find all customers who have an account but no loan.*

(select customer-name from depositor)

except

(select customer-name from borrower)

AGGREGATE FUNCTIONS:

These functions operate on the multiset of values of a column of a relation, and return a value

avg: average value
min: minimum value
max:maximum value
sum: sum of values
count: number of values

Find the average account balance at the Perryridge branch.

select avg (balance)
from account
where branch-name = "Perryridge"

Find the number of tuples in the customer relation

select count (*)
from customer

Find the number of depositors in the bank

select count (**distinct** customer-name)
from depositor

AGGREGATE FUNCTIONS – GROUP BY:

Find the number of depositors for each branch.

select branch-name, **count** (**distinct** customer-name)
from depositor, account
where depositor.account-number = account.account-number
group by branch-name

Note: Attributes in **select** clause outside of aggregate functions must appear in **group by** list

AGGREGATE FUNCTIONS – HAVING CLAUSE:

Find the names of all branches where the average account

balance is more than $1,200
select branch-name, **avg** (balance)
from account
group by branch-name
having avg (balance) > 1200
Note: predicates in the **having** clause are applied after the formation of groups

Null Values

_ It is possible for tuples to have a null value, denoted by null, for some of their attributes; null signifies an unknown value or that a value does not exist.
_ The result of any arithmetic expression involving null is null.
_ Roughly speaking, all comparisons involving null return false.
More precisely,

- Any comparison with null returns unknown
- (true **or** unknown) = true, (false **or** unknown) = unknown
  (unknown **or** unknown) = unknown,
  (true **and** unknown) = unknown, (false **and** unknown) = false,
  (unknown **and** unknown) = unknown
- Result of **where** clause predicate is treated as false if it evaluates to unknown
- "P **is unknown**" evaluates to true if predicate P evaluates to unknown

Find all loan numbers which appear in the loan relation with null values for amount.

select loan-number
from loan
where amount **is null**

Total all loan amounts
select sum (amount)
from loan

Above statement ignores null amounts; result is null if there is no non-null amount.
_ All aggregate operations except **count(*)** ignore tuples with null values on the aggregated attributes.

NESTED SUBQUERIES

_ SQL provides a mechanism for the nesting of subqueries.

_ A subquery is a **select-from-where** expression that is nested within another query.

_ A common use of subqueries is to perform tests for set membership, set comparisons, and set cardinality.

DERIVED RELATIONS:

Find the average account balance of those branches where the average account balance is greater than $1200.

select branch-name, avg-balance
from (**select** branch-name, **avg** (balance)
from account
group by branch-name)
as result (branch-name, avg-balance)
where avg-balance > 1200

Note that we do not need to use the **having** clause, since we compute in the **from** clause the temporary relation result, and the attributes of result can be used directly in the **where** clause.

VIEWS:

Provide a mechanism to hide certain data from the view of certain users. To create a view we use the command:

create view v **as** <query expression>

where:
– <query expression> is any legal expression
– the view name is represented by v

Example Queries:

A view consisting of branches and their customers

create view all-customer **as**
(**select** branch-name, customer-name
from depositor, account
where depositor.account-number = account.account-number)

union
(**select** branch-name, customer-name
from borrower, loan
where borrower.loan-number = loan.loan-number)

Find all customers of the Perryridge branch

select customer-name
from all-customer
where branch-name = "Perryridge"

MODIFICATION OF THE DATABASE – DELETION

Delete all account records at the Perryridge branch

delete from account
where branch-name = "Perryridge"

Delete all accounts at every branch located in Needham.

delete from account
where branch-name **in** (**select** branch-name
from branch
where branch-city = "Needham")
delete from depositor
where account-number **in** (**select** account-number
from branch, account
where branch-city = "Needham"
and branch.branch-name = account.branch-name)

Example Query:

Delete the records of all accounts with balances below the average at the bank

delete from account
where balance < (**select avg** (balance)
from account)
– Problem: as we delete tuples from deposit, the average balance changes

– Solution used in SQL:
1. First, compute **avg** balance and find all tuples to delete
2. Next, delete all tuples found above (without recomputing **avg** or retesting the tuples)

Modification of the Database – INSERTION

_ *Add a new tuple to account*

insert into account
values ("Perryridge", A-9732, 1200)
or equivalently

insert into account (branch-name, balance, account-number)
values ("Perryridge", 1200, A-9732)
_ Add a new tuple to account with balance set to null
insert into account
values ("Perryridge", A-777, null)

_ *Provide as a gift for all loan customers of the Perryridge* branch, a $200 savings account. Let the loan number serve as the account number for the new savings account

insert into account
select branch-name, loan-number, 200
from loan
where branch-name = "Perryridge"

insert into depositor
select customer-name, loan-number
from loan, borrower

where branch-name = "Perryridge"

and loan.account-number = borrower.account-number

Modification of the Database – UPDATES

_ Increase all accounts with balances over $10,000 by 6%, all other accounts receive 5%

.

– Write two **update** statements:

update account
set balance = balance _ 1.06
where balance > 10000

update account
set balance = balance _ 1.05
where balance _ 10000

UPDATE OF A VIEW:

_ Create a view of all loan data in the loan relation, hiding the amount attribute

create view branch-loan **as**

select branch-name, loan-number

from loan

_ Add a new tuple to branch-loan

insert into branch-loan

values ("Perryridge", "L-307")

This insertion must be represented by the insertion of the tuple

("Perryridge", "L-307", null)

into the loan relation.

_ Updates on more complex views are difficult or impossible to translate, and hence are disallowed.

JOINED RELATIONS:

_ Join operations take two relations and return as a result another relation.

_ These additional operations are typically used as subquery expressions in the **from** clause.

_ Join condition – defines which tuples in the two relations match, and what attributes are present in the result of the join.

_ Join type – defines how tuples in each relation that do not match any tuple in the other relation (based on the join condition) are treated.

Join Types
Inner Join
Left Outer Join
Right Outer join
Full outer join

Join Conditions
Natural
On <predicate>
Using(A1,A2,...,An)

DATA DEFINITION LANGUAGE (DDL):

Allows the specification of not only a set of relations but also information about each relation, including:

- The schema for each relation.
- The domain of values associated with each attribute.
- Integrity constraints.
- The set of indices to be maintained for each relation.
- Security and authorization information for each relation.
- The physical storage structure of each relation on disk

EMBEDDED SQL:

- The SQL standard defines embeddings of SQL in a variety of programming languages such as such as Pascal, PL/I, Fortran, C, and Cobol.
- A language in which SQL queries are embedded is referred to as a host language, and the SQL structures permitted in the host language comprise embedded SQL.
- The basic form of these languages follows that of the System R embedding of SQL into PL/I.
- EXEC SQL statement is used to identify embedded SQL requests to the preprocessor

  EXEC SQL <embedded SQL statement > END EXEC

Example Query:

From within a host language, find the names and account numbers of customers with more than the variable amount dollars in some account.

- Specify the query in SQL and declare a cursor for it

```
EXEC SQL
    declare c cursor for
    select customer-name, account-number
    from depositor, account
    where depositor.account-number = account.account-number
          and account.balance > :amount
END-EXEC
```

- The **open** statement causes the query to be evaluated

  EXEC SQL **open** c END-EXEC

- The **fetch** statement causes the values of one tuple in the query result to be placed in host language variables.

  EXEC SQL **fetch** c **into** :cn :an END-EXEC

  Repeated calls to **fetch** get successive tuples in the query result; a variable in the SQL communication area indicates

when end-of-file is reached.

- The **close** statement causes the database system to delete the temporary relation that holds the result of the query.

  EXEC SQL **close** c END-EXEC

DATA WAREHOUSING:

Data warehousing is the process of constructing and using a data warehouse. A data warehouse is constructed by integrating data from multiple heterogeneous sources that support analytical reporting, structured and/or ad hoc queries, and decision making. Data warehousing involves data cleaning, data integration, and data consolidations.

Using Data Warehouse Information

There are decision support technologies that help utilize the data available in a data warehouse. These technologies help executives to use the warehouse quickly and effectively. They can gather data, analyze it, and take decisions based on the information present in the warehouse. The information gathered in a warehouse can be used in any of the following domains −

- **Tuning Production Strategies** − The product strategies can be well tuned by repositioning the products and managing the product portfolios by comparing the sales quarterly or yearly.
- **Customer Analysis** − Customer analysis is done by analyzing the customer's buying preferences, buying time, budget cycles, etc.
- **Operations Analysis** − Data warehousing also helps in customer relationship management, and making environmental corrections. The information also allows us to analyze business operations.

Integrating Heterogeneous Databases

To integrate heterogeneous databases, we have two approaches −

- Query-driven Approach
- Update-driven Approach

Query-Driven Approach

This is the traditional approach to integrate heterogeneous databases. This approach was used to build wrappers and integrators on top of multiple heterogeneous databases. These integrators are also known as mediators.

Process of Query-Driven Approach

- When a query is issued to a client side, a metadata dictionary translates the query into an appropriate form for individual heterogeneous sites involved.
- Now these queries are mapped and sent to the local query processor.
- The results from heterogeneous sites are integrated into a global answer set.

Disadvantages

- Query-driven approach needs complex integration and filtering processes.
- This approach is very inefficient.
- It is very expensive for frequent queries.
- This approach is also very expensive for queries that require aggregations.

Update-Driven Approach

This is an alternative to the traditional approach. Today's data warehouse systems follow update-driven approach rather than the traditional approach discussed earlier. In update-driven approach, the information from multiple heterogeneous sources are integrated in advance and are stored in a warehouse. This information is available for direct querying and analysis.

Advantages

This approach has the following advantages −

- This approach provide high performance.
- The data is copied, processed, integrated, annotated, summarized and restructured in semantic data store in advance.
- Query processing does not require an interface to process data at local sources.

Functions of Data Warehouse Tools and Utilities

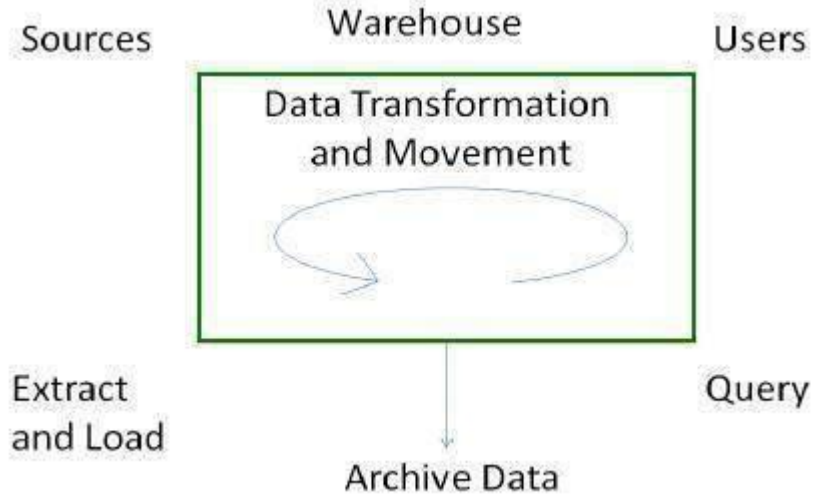
The following are the functions of data warehouse tools and utilities −

- **Data Extraction** − Involves gathering data from multiple heterogeneous sources.
- **Data Cleaning** − Involves finding and correcting the errors in data.
- **Data Transformation** − Involves converting the data from legacy format to warehouse format.
- **Data Loading** − Involves sorting, summarizing, consolidating, checking integrity, and building indices and partitions.
- **Refreshing** − Involves updating from data sources to warehouse.

PROCESS FLOW IN DATA WAREHOUSE

There are four major processes that contribute to a data warehouse −

- Extract and load the data.
- Cleaning and transforming the data.
- Backup and archive the data.
- Managing queries and directing them to the appropriate data sources.



Extract and Load Process

Data extraction takes data from the source systems. Data load takes the extracted data and loads it into the data warehouse.

Note − Before loading the data into the data warehouse, the information extracted from the external sources must be reconstructed.

Controlling the Process

Controlling the process involves determining when to start data extraction and the consistency check on data. Controlling process ensures that the tools, the logic modules, and the programs are executed in correct sequence and at correct time.

When to Initiate Extract

Data needs to be in a consistent state when it is extracted, i.e., the data warehouse should represent a single, consistent version of the information to the user.

For example, in a customer profiling data warehouse in telecommunication sector, it is illogical to merge the list of customers at 8 pm on Wednesday from a customer database with the customer subscription events up to 8 pm on Tuesday. This would mean that we are finding the customers for whom there are no associated subscriptions.

Loading the Data

After extracting the data, it is loaded into a temporary data store where it is cleaned up and made consistent.

Note − Consistency checks are executed only when all the data sources have been loaded into the temporary data store.

Clean and Transform Process

Once the data is extracted and loaded into the temporary data store, it is time to perform Cleaning and Transforming. Here is the list of steps involved in Cleaning and Transforming

- Clean and transform the loaded data into a structure
- Partition the data
- Aggregation

Clean and Transform the Loaded Data into a Structure

Cleaning and transforming the loaded data helps speed up the queries. It can be done by making the data consistent −

- within itself.
- with other data within the same data source.
- with the data in other source systems.

- with the existing data present in the warehouse.

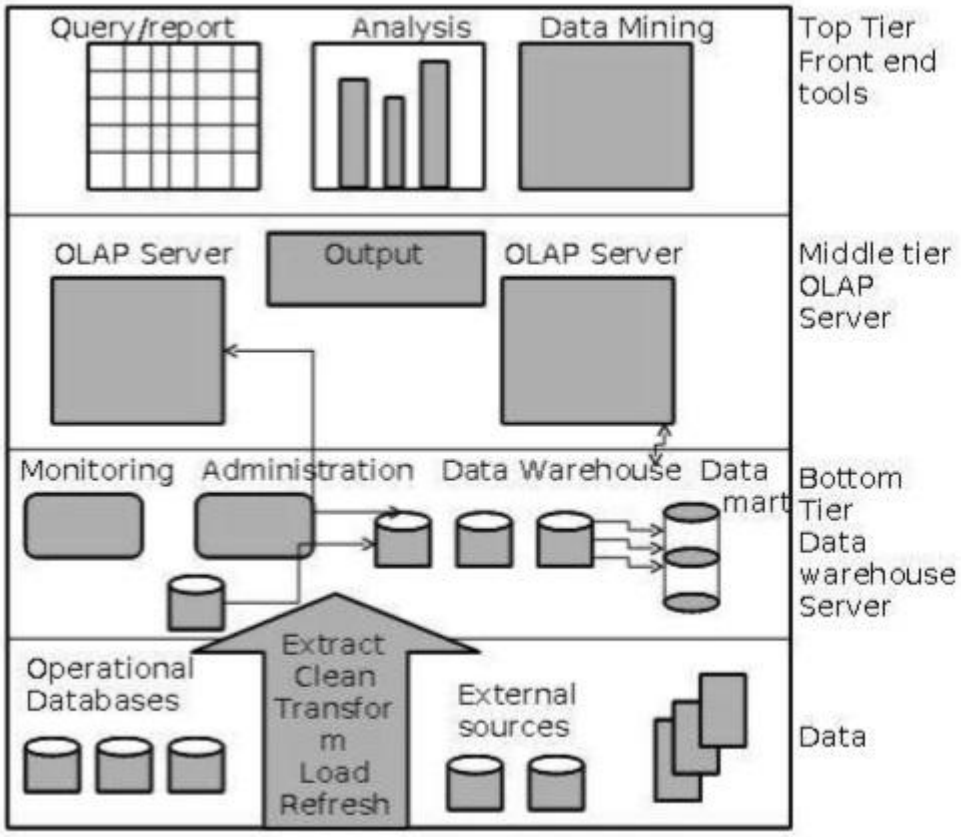
Transforming involves converting the source data into a structure. Structuring the data increases the query performance and decreases the operational cost. The data contained in a data warehouse must be transformed to support performance requirements and control the ongoing operational costs.

THREE-TIER DATA WAREHOUSE ARCHITECTURE

Generally a data warehouses adopts a three-tier architecture. Following are the three tiers of the data warehouse architecture.

- **Bottom Tier** − The bottom tier of the architecture is the data warehouse database server. It is the relational database system. We use the back end tools and utilities to feed data into the bottom tier. These back end tools and utilities perform the Extract, Clean, Load, and refresh functions.
- **Middle Tier** − In the middle tier, we have the OLAP Server that can be implemented in either of the following ways.
    - By Relational OLAP (ROLAP), which is an extended relational database management system. The ROLAP maps the operations on multidimensional data to standard relational operations.
    - By Multidimensional OLAP (MOLAP) model, which directly implements the multidimensional data and operations.
- **Top-Tier** − This tier is the front-end client layer. This layer holds the query tools and reporting tools, analysis tools and data mining tools.

The following diagram depicts the three-tier architecture of data warehouse −



This process performs the following functions −

- manages the queries.
- helps speed up the execution time of queris.
- directs the queries to their most effective data sources.
- ensures that all the system sources are used in the most effective way.
- monitors actual query profiles.

The information generated in this process is used by the warehouse management process to determine which aggregations to generate. This process does not generally operate during the regular load of information into data warehouse.

Data transformation:

Data transformation is the process of converting data or information from one format to another, usually from the format of a source system into the required format of a new destination system. The usual process involves converting documents, but data conversions sometimes involve the conversion of a program from one computer language to another to

enable the program to run on a different platform. The usual reason for this data migration is the adoption of a new system that's totally different from the previous one.

Data Transformation involves two key phases:

Data Mapping: The assignment of elements from the source base or system toward the destination to capture all transformations that occur. This is made more complicated when there are complex transformations like many-to-one or one-to-many rules for transformation.

Code Generation: The creation of the actual transformation program. The resulting data map specification is used to create an executable program to run on computer systems.
Commonly used transformational languages:

- Perl: A high-level procedural and object-oriented language capable of powerful operations
- AWK: One of the oldest languages and a popular TXT transformation language
- XSLT: An XML data transformation language
- TXL: A prototyping language mostly used for source code transformation
- Template Languages and Processors: These specialize in data-to-document transformation

USER INTERFACE:

User interface is the front-end application view to which user interacts in order to use the software. User can manipulate and control the software as well as hardware by means of user interface. Today, user interface is found at almost every place where digital technology exists, right from computers, mobile phones, cars, music players, airplanes, ships etc.

User interface is part of software and is designed such a way that it is expected to provide the user insight of the software. UI provides fundamental platform for human-computer interaction.

UI can be graphical, text-based, audio-video based, depending upon the underlying hardware and software combination. UI can be hardware or software or a combination of both.

The software becomes more popular if its user interface is:

- Attractive
- Simple to use
- Responsive in short time

- Clear to understand
- Consistent on all interfacing screens

UI is broadly divided into two categories:

- Command Line Interface
- Graphical User Interface

Command Line Interface (CLI)

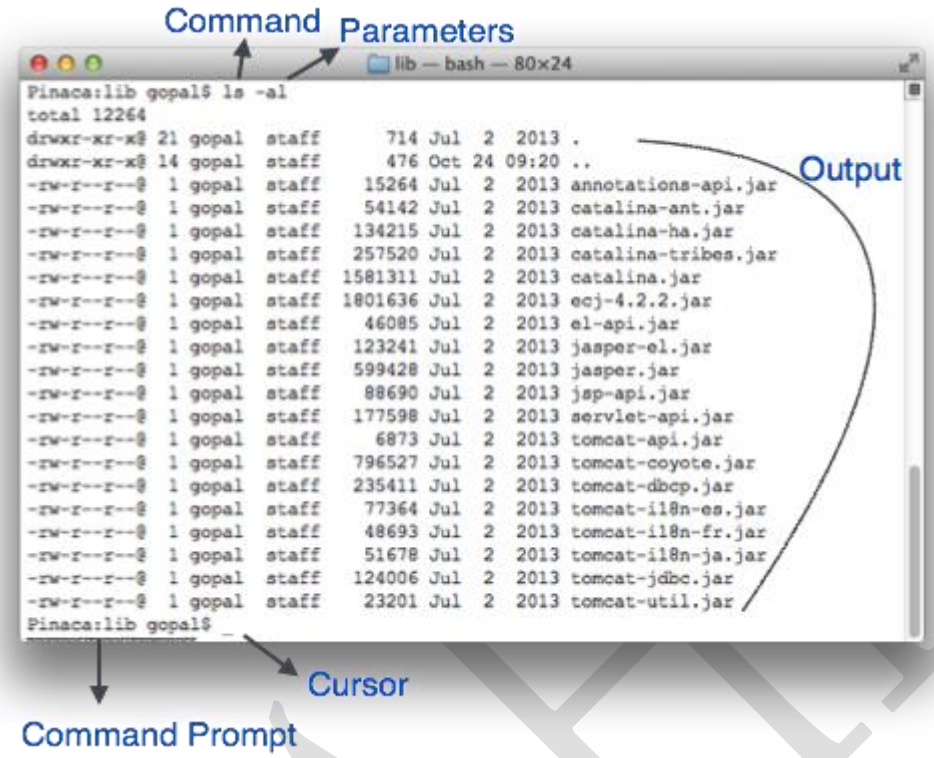
CLI has been a great tool of interaction with computers until the video display monitors came into existence. CLI is first choice of many technical users and programmers. CLI is minimum interface a software can provide to its users.

CLI provides a command prompt, the place where the user types the command and feeds to the system. The user needs to remember the syntax of command and its use. Earlier CLI were not programmed to handle the user errors effectively.

A command is a text-based reference to set of instructions, which are expected to be executed by the system. There are methods like macros, scripts that make it easy for the user to operate.

CLI uses less amount of computer resource as compared to GUI.

CLI Elements



A text-based command line interface can have the following elements:

- **Command Prompt** - It is text-based notifier that is mostly shows the context in which the user is working. It is generated by the software system.
- **Cursor** - It is a small horizontal line or a vertical bar of the height of line, to represent position of character while typing. Cursor is mostly found in blinking state. It moves as the user writes or deletes something.
- **Command** - A command is an executable instruction. It may have one or more parameters. Output on command execution is shown inline on the screen. When output is produced, command prompt is displayed on the next line.

Graphical User Interface

Graphical User Interface provides the user graphical means to interact with the system. GUI can be combination of both hardware and software. Using GUI, user interprets the software.

Typically, GUI is more resource consuming than that of CLI. With advancing technology, the programmers and designers create complex GUI designs that work with more efficiency, accuracy and speed.

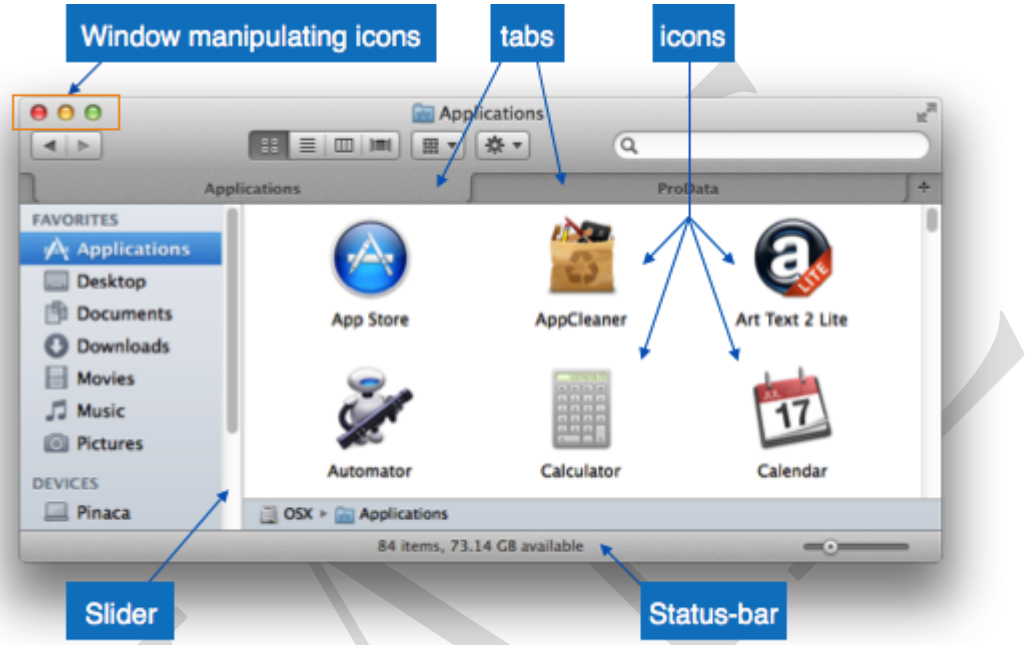
Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401C Semester: IV Year: 2017-2019

GUI Elements

GUI provides a set of components to interact with software or hardware.

Every graphical component provides a way to work with the system. A GUI system has following elements such as:



- **Window** - An area where contents of application are displayed. Contents in a window can be displayed in the form of icons or lists, if the window represents file structure. It is easier for a user to navigate in the file system in an exploring window. Windows can be minimized, resized or maximized to the size of screen. They can be moved anywhere on the screen. A window may contain another window of the same application, called child window.
- **Tabs** - If an application allows executing multiple instances of itself, they appear on the screen as separate windows. **Tabbed Document Interface** has come up to open multiple documents in the same window. This interface also helps in viewing preference panel in application. All modern web-browsers use this feature.
- **Menu** - Menu is an array of standard commands, grouped together and placed at a visible place (usually top) inside the application window. The menu can be programmed to appear or hide on mouse clicks.

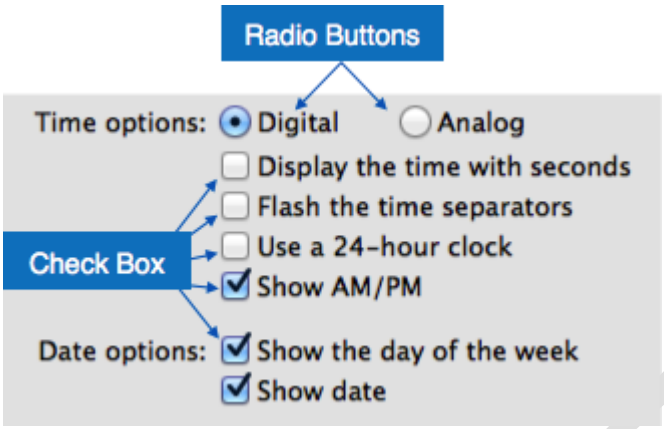
- **Icon -** An icon is small picture representing an associated application. When these icons are clicked or double clicked, the application window is opened. Icon displays application and programs installed on a system in the form of small pictures.
- **Cursor -** Interacting devices such as mouse, touch pad, digital pen are represented in GUI as cursors. On screen cursor follows the instructions from hardware in almost real-time. Cursors are also named pointers in GUI systems. They are used to select menus, windows and other application features.

Application specific GUI components

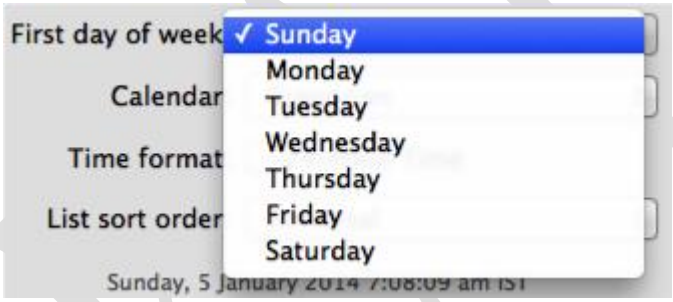
A GUI of an application contains one or more of the listed GUI elements:

- **Application Window -** Most application windows uses the constructs supplied by operating systems but many use their own customer created windows to contain the contents of application.
- **Dialogue Box -** It is a child window that contains message for the user and request for some action to be taken. For Example: Application generate a dialogue to get confirmation from user to delete a file.

- **Text-Box -** Provides an area for user to type and enter text-based data.
- **Buttons -** They imitate real life buttons and are used to submit inputs to the software.



- **Radio-button** - Displays available options for selection. Only one can be selected among all offered.
- **Check-box** - Functions similar to list-box. When an option is selected, the box is marked as checked. Multiple options represented by check boxes can be selected.
- **List-box** - Provides list of available items for selection. More than one item can be selected.



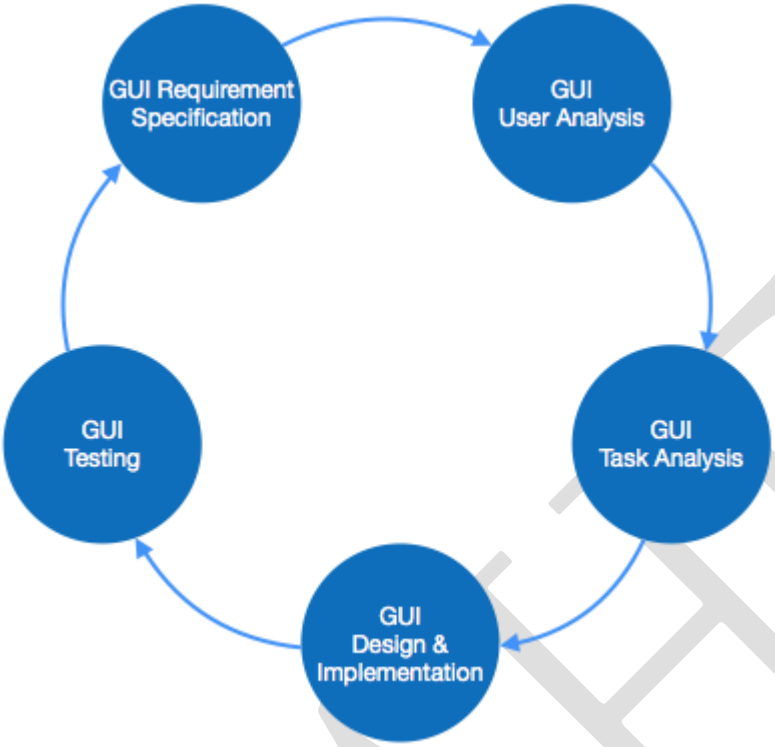
Other impressive GUI components are:

- Sliders
- Combo-box
- Data-grid
- Drop-down list

User Interface Design Activities

There are a number of activities performed for designing user interface. The process of GUI design and implementation is alike SDLC. Any model can be used for GUI implementation among Waterfall, Iterative or Spiral Model.

A model used for GUI design and development should fulfill these GUI specific steps.



- **GUI Requirement Gathering** - The designers may like to have list of all functional and non-functional requirements of GUI. This can be taken from user and their existing software solution.
- **User Analysis** - The designer studies who is going to use the software GUI. The target audience matters as the design details change according to the knowledge and competency level of the user. If user is technical savvy, advanced and complex GUI can be incorporated. For a novice user, more information is included on how-to of software.
- **Task Analysis** - Designers have to analyze what task is to be done by the software solution. Here in GUI, it does not matter how it will be done. Tasks can be represented in hierarchical manner taking one major task and dividing it further into smaller sub-tasks. Tasks provide goals for GUI presentation. Flow of information among sub-tasks determines the flow of GUI contents in the software.
- **GUI Design & implementation** - Designers after having information about requirements, tasks and user environment, design the GUI and implements into code

and embed the GUI with working or dummy software in the background. It is then self-tested by the developers.

- **Testing** - GUI testing can be done in various ways. Organization can have in-house inspection, direct involvement of users and release of beta version are few of them. Testing may include usability, compatibility, user acceptance etc.

GUI Implementation Tools

There are several tools available using which the designers can create entire GUI on a mouse click. Some tools can be embedded into the software environment (IDE).

GUI implementation tools provide powerful array of GUI controls. For software customization, designers can change the code accordingly.

There are different segments of GUI tools according to their different use and platform.

Example

Mobile GUI, Computer GUI, Touch-Screen GUI etc. Here is a list of few tools which come handy to build GUI:

- FLUID
- AppInventor (Android)
- LucidChart
- Wavemaker
- Visual Studio

User Interface Golden rules

The following rules are mentioned to be the golden rules for GUI design, described by Shneiderman and Plaisant in their book (Designing the User Interface).

- **Strive for consistency** - Consistent sequences of actions should be required in similar situations. Identical terminology should be used in prompts, menus, and help screens. Consistent commands should be employed throughout.
- **Enable frequent users to use short-cuts** - The user's desire to reduce the number of interactions increases with the frequency of use. Abbreviations, function keys, hidden commands, and macro facilities are very helpful to an expert user.

- **Offer informative feedback** - For every operator action, there should be some system feedback. For frequent and minor actions, the response must be modest, while for infrequent and major actions, the response must be more substantial.
- **Design dialog to yield closure** - Sequences of actions should be organized into groups with a beginning, middle, and end. The informative feedback at the completion of a group of actions gives the operators the satisfaction of accomplishment, a sense of relief, the signal to drop contingency plans and options from their minds, and this indicates that the way ahead is clear to prepare for the next group of actions.
- **Offer simple error handling** - As much as possible, design the system so the user will not make a serious error. If an error is made, the system should be able to detect it and offer simple, comprehensible mechanisms for handling the error.
- **Permit easy reversal of actions** - This feature relieves anxiety, since the user knows that errors can be undone. Easy reversal of actions encourages exploration of unfamiliar options. The units of reversibility may be a single action, a data entry, or a complete group of actions.
- **Support internal locus of control** - Experienced operators strongly desire the sense that they are in charge of the system and that the system responds to their actions. Design the system to make users the initiators of actions rather than the responders.
- **Reduce short-term memory load** - The limitation of human information processing in short-term memory requires the displays to be kept simple, multiple page displays be consolidated, window-motion frequency be reduced, and sufficient training time be allotted for codes, mnemonics, and sequences of actions.

A **distributed database** is a collection of multiple interconnected databases, which are spread physically across various locations that communicate via a computer network.

Features

- Databases in the collection are logically interrelated with each other. Often they represent a single logical database.
- Data is physically stored across multiple sites. Data in each site can be managed by a DBMS independent of the other sites.

- The processors in the sites are connected via a network. They do not have any multiprocessor configuration.
- A distributed database is not a loosely connected file system.
- A distributed database incorporates transaction processing, but it is not synonymous with a transaction processing system.

Advantages of Distributed Databases

Following are the advantages of distributed databases over centralized databases.

Modular Development − If the system needs to be expanded to new locations or new units, in centralized database systems, the action requires substantial efforts and disruption in the existing functioning. However, in distributed databases, the work simply requires adding new computers and local data to the new site and finally connecting them to the distributed system, with no interruption in current functions.

More Reliable − In case of database failures, the total system of centralized databases comes to a halt. However, in distributed systems, when a component fails, the functioning of the system continues may be at a reduced performance. Hence DDBMS is more reliable.

Better Response − If data is distributed in an efficient manner, then user requests can be met from local data itself, thus providing faster response. On the other hand, in centralized systems, all queries have to pass through the central computer for processing, which increases the response time.

Lower Communication Cost − In distributed database systems, if data is located locally where it is mostly used, then the communication costs for data manipulation can be minimized. This is not feasible in centralized systems.

KARPAGAM ACADEMY OF HIGHER EDUCATION						
Department of Management						
Unit 4- Multiple Choice Questions- Each Question Carry ONE Mark						
S.no	**Question**	**Option 1**	**Option 2**	**Option 3**	**Option 4**	**Answer**
1	ERP supports _____ currency value.	Multiple	Single	Three	Five	**Multiple**
2	The most important step of ERP implementation is _____ phase	Installing	training	Gap analysis	testing	**Gap analysis**
3	An enterprise is a group of people with _____	Common goal	Seperate goal for each departmen t	Multiple Goals	Two or more goals	**Common goal**
4	In _____, entire organization is considered as a system and the departments are its subsystem	business wa y	general	enterpris e way	planning	**enterprise w ay**
5	_____is at the heart of any ERP system	Informations	Employee s	Customer s	Database	**Database**
6	An information system produces information using the _____ cycle	Data Analysis	Input-Process-output	Input-output	process-input-output	**Input-Process-output**
7	_____ used to support the old process to become useful in the new process, involves reducing somerequiremen ts while expanding others	Transitionin g the inform ation	software managem ent	front-office sof tware	informati on syste m	**Transitionin g the inform ation**
8	The traditional business system is called as	modern method	advanced method	effective method.	legacy method	**legacy method**

	_____					
9	_____ is the following method is used to produce reports about data.	Decision Support Systems	Executive Information Systems	Query/Report Writing Tool.	front-office software	**Query/Report Writing Tool.**
10	_____ approaches reduces data redundancy and provide update information	Legacy system	Information system	Integrated data model	Data base	**Integrated data model**
11	The elapsed time between placing an order and it receiving it is also known as _____	elapsed time	waiting time	time interval	expected time	**elapsed time**
12	In the case of make-to-order items, the ERP systems save time by integrating with _____ systems	engineering change control	engineering change order	CAD and CAM	CAD	**CAD and CAM**
13	The _____ planning features of most ERP systems offer, both rough cut and detailed capacity planning	capacity	production	marketing	test	**capacity**
14	_____ is a key issue in the formation of strategic plans in companies.	Computerized	Quantity	Quality	Flexibility	**Flexibility**
15	The business information has _____ fundamental characteristics	1	2	3	many	**3**
16	The first step in business	planning	develop blueprint	marketing	assessment	**assessment**

	strategy is _____					
17	A _____ consists of a sequence of query and/or update statemen	Transaction	Commit	Rollback	Flashback	**Transaction**
18	_____ makes the transaction permanent in the database?	View	Commit	Rollback	Flashback	**Commit**
19	A transaction is said to be a unit of program's_____ ___	Evaluation	Execution	Computation	Controlling	**Execution**
20	A single transaction failure may result into a set of transaction rollbacks, is known to be_____	Iterated Rollback	Cascadeless Rollback	Cascading Rollback	Serial Rollback	**Cascading Rollback**
21	For committing a transaction, the DBMS might discard all the records_____ ___	After Image	Before Image	log	Redo Log	**Before Image**
22	A Transaction ends_____ _	only when it is Committed	only when it is Rolled-back	when it is Committed or Rolled-back	only when it is initialized	**when it is Committed or Rolled-back**
23	In_____, each transaction there is a first phase during which new lock is acquired	Shrinking Phase	Release phase	Commit phase	Growing Phase	**Growing Phase**
24	A transaction processing system is also called as_____	processing monitor	transaction monitor	TP monitor	monitor	**TP monitor**

25	_____servers which is widely used in relational database systems	Data servers	Transaction servers	Query servers	Client servers	**Transaction servers**
26	In categories of data modeling, low level data models are also called_____	conceptual data models	physical data models	triggered data models	logical data models	**physical data models**
27	Type of diagram in which operations are specified on objects is considered as_____	Functional Diagrams	Class Diagrams	Attribute Diagrams	Entity Diagrams	**Class Diagrams**
28	Attributes that can be arranged into hierarchy are called_____	Composite attributes	Atomic Attributes	Derived Attributes	Simple Attributes	**Composite attributes**
29	Set of all entities having same attributes is classified as_____	Entity Type	Attribute Type	Fuction Attribute	Hierarchy type	**Entity Type**
30	Set of values which specifies which values are to be assigned to individual entities is considered as_____	Domain of values	Composition Of values	Attribution of values	derivation of values	**Domain of values**
31	Most popular high level conceptual model in database management system is_____	close end relation model	query relation model	entity relationship model	attribute relationship model	**entity relationship model**
32	If in database of employees, Age attribute is derived attribute then date of birth is	Logical Attribute	Physical Attribute	Conceptual Attribute	Stored Attribute	**Stored Attribute**

	classified as_____					
33	In binary relationship, participation cardinality is also known as_____	intensive cardinality constraint	recursive cardinality constraint	minimum cardinality constraint	maximum cardinality constraint	**minimum cardinality constraint**
34	Snapshot of the dta in the database at a given instant of time is called_____	Database Schema	Database Instance	Database Snapshot	Database management system	**Database Instance**
35	A logical description of some portion of database that is required by a user to perform task is called as_____	System View	User View	Logical View	Data View	**User View**
36	The companies can use supply chain management for _____	critical business process	effective planning.	inventory control	material management	**critical business process**
37	Most organizations are implementing or planning to implement _____.	packaged software solution	separate software solution	simple software	complex software	**packaged software solution**
38	ERP system is for _____	major departments only	entire organization	marketing and sales	production department	**entire organization**
39	The company should conduct a _____ about the implementation and implementation strategies	feasibility study	survey	case study	test	**feasibility study**
40	ERP implementation process has	one	two	Five	Eleven	**Eleven**

	_____ phases					
41	PeopleSoft is strong in _____	Manufacturing	HR	Plant and Maintenance	Finanace	**HR**
42	Normally the ERP package needs _____ of company's functional requirements	50%	60%	80%	100%	**80%**
43	_____is the most critical phase in ERP implementation	Understanding the problem	Defining solutions	Getting down to work	Going live	**Defining solutions**
44	_____ is the first phase in ERP implementation	Understanding the problem.	Defining solutions	Getting down to work	Going live	**Understanding the problem.**
45	_____ is the last phase in ERP implementation life cycle	Testing	End user training	Post implementation	Going live	**Post implementation**
46	. In _____ phase roles are identified and responsibilities assigned.	project planning	gap analysis.	reengineering.	configuration.	**project planning**
47	The project planning phase will be headed by _____	management	vendor.	ERP in-charge	consultant	**ERP in-charge**
48	All basic definitions of information and data modeling will be made in _____ phase	pre-implementation	post implementation	defining solutions	going live	**defining solutions**
49	. Users with heavy data analysis needs should include the cost of _____ in the ERP budget	software	analysis tool	data warehouse	hardware	**data warehouse**
50	. _____ provides planning, scheduling and control of	HR module	Plant maintenance control.	Sales and distribution	Finance	**Plant maintenance control.**

	facilities and equipment..					
51	The _____ team will contain people from the company's IT/EDP departments.	in-house	executive	administrative	technical support	**in-house**
52	_____ team is responsible for making available work space	Technical support	Executive	Administrative support	Work	**Administrative support**
53	_____ are the people who have developed the ERP packages	Consultants	End users	Administrative team	Vendors	**Vendors**
54	_____ are professionals who specialise in developing techniques.	Software engineers	Vendors	Technical support team members	Business consultants	**Business consultants**
55	_____ analyzes the profitability of internal responsibility centers	Profit center accounting	Funds management	Market risk management	Budgetting	**Profit center accounting**
56	During ERP implementation _____ should be converted into tasks and should be allocated to the right people	rules	procedures	methodologies	functionality	**methodologies**
57	_____ sub-system allows the company to analyze financial transactions for a given period.	Cash management	Treasury management	Investment management	Treasury module	**Cash management**
58	_____ __ is the people who will be using the ERP system once it is in place	Management	End-user	Consultant	Vendor	**End-user**
59	The _____ include package vendor, hardware and peripheral vendor, networking people, consultants, and employees	testing	training	ERP implementation	ERP planning	**ERP implementation**

60	_____ is the tool used to prepare a work plan with quality	CAD	CAM	PERT	Software	**PERT**

Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401C Semester: IV Year: 2017-2019

UNIT – V

SYLLABUS

UNIT- V:- Sales and Distribution-Basic functions-Billing-Electronic Data Interchange-Transportation-SAP R/3,Oracle Financials-Development Tools-Administration tools-Reporting and Analysis Tools and Integration Tools-BPR Implementation-Project Management-Meaning-Business Benefit of ES

Sales and Distribution module:

Sales and Distribution Module to keep pace with rapid changes in the business world, companies need an integrated and flexible enterprise system that supports all aspects of their business with state-of-the-art functionality. This innovative solution should upgrade effortlessly and interface easily with third-party applications as well as have the ability to incorporate existing systems while extending its reach to the Internet and e-commerce.

In today's competitive business environment, companies are increasingly being forced to streamline business processes. In a world where it is no longer enough to simply have the best product, companies are focusing on core competencies and closer partnerships over the whole supply chain.

Here, increased efficiency in sales and distribution is a key factor to ensure that companies retain a competitive edge and improve both profit margins and customer service. In helping business to 'beat them on delivery', the Sales and Distribution module of eresource ERP systems offers a comprehensive set of best-of-bred component for both order and logistics management.

eresource ERP system is tightly integrated with the Sales and Distribution module. This integration enables the mapping and supply of single-site or multi-site organizations. Developing precise logistics planning for just-in-time deliveries, this system can also generate replenishment orders by using defined warehouse requirements.

The following are the sales related business transactions:

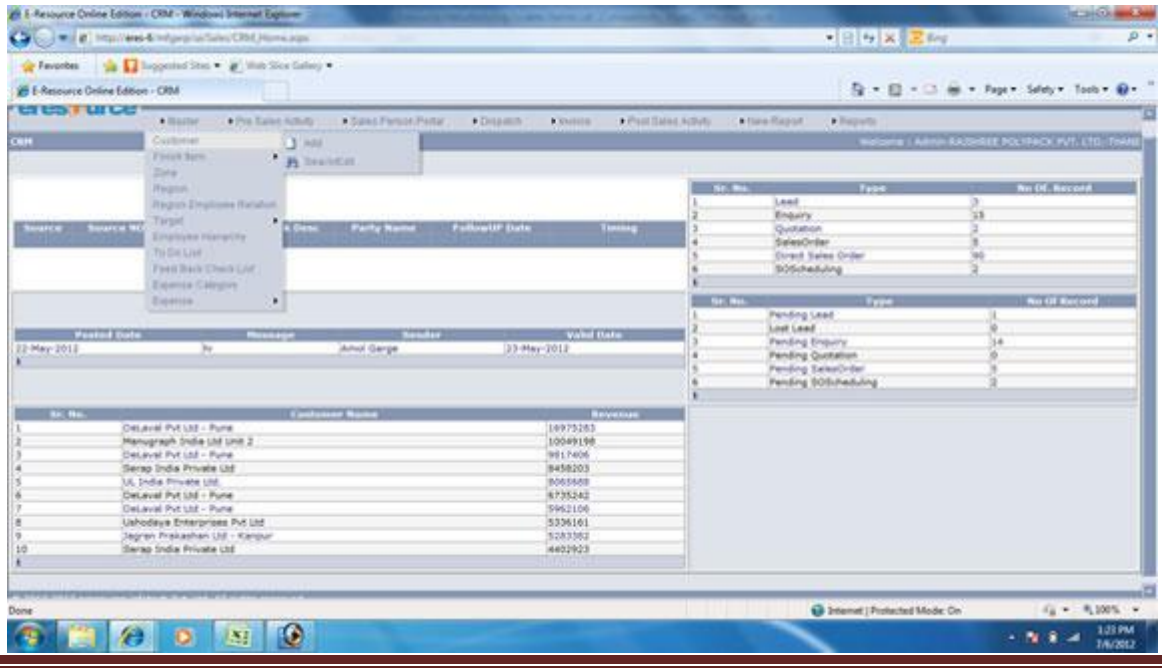
- Sales queries, such as inquiries and quotations
- Sales orders
- Outline agreements, such as contracts and scheduling agreements
- Delivery/Shipment
- Invoicing/billing
- After sales support

During sales order-processing the following basic functions are carried out:

- Inquiry handling
- Quotation preparation and processing
- Contracts and contact management (order management)
- Monitoring the sales transaction
- Checking for availability
- Transferring requirements to materials planning (MRP)
- Scheduling the delivery
- Calculating pricing and taxes
- Checking credit limits
- Invoicing/billing
- Creating printed or electronically transmitted documents

Depending on how your particular system is configured, these functions may be completely automated. The data that results from these basic functions is stored in the system where it can be displayed. eresource ERP 's Sales and Distribution module very actively interacts with the material management and financial accounting module for delivery and billing his Module comprises of Contact Management and Sales Order Processing Management. It is an integrated solution comprising of marketing and sales activities. Organization can act immediately to improve sales, service and marketing effectiveness by using this Module. eresource Sales Management module is all about retaining customers, improving customer loyalty and gaining customer insight. This module also help to you getting your customer order management easier day by day. Whether it is maintaining customer information, quickly creating a quote or migrating them to sales order or being responsive to your customers, we provide you right solution and efficient flow of information.

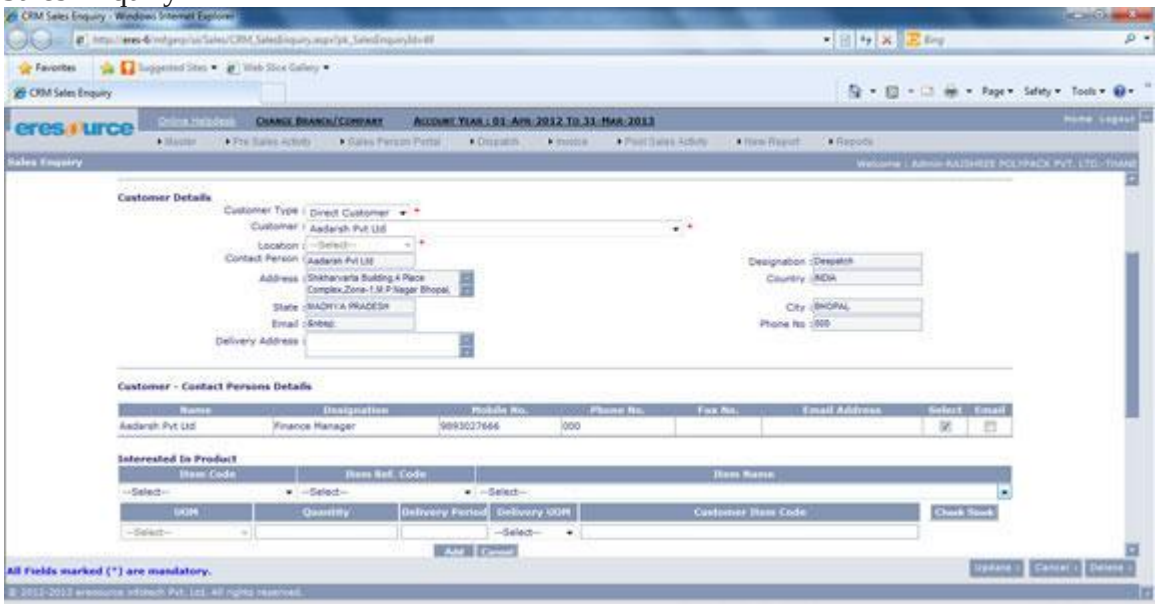
Customer List



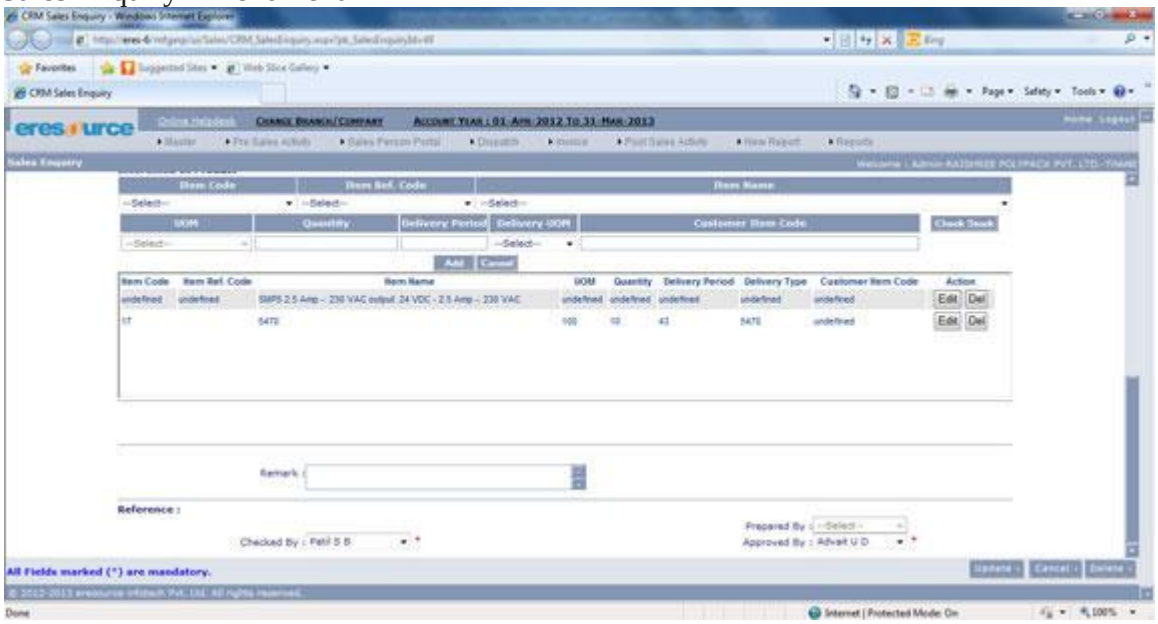
Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401C Semester: IV Year: 2017-2019

Sales Enquiry



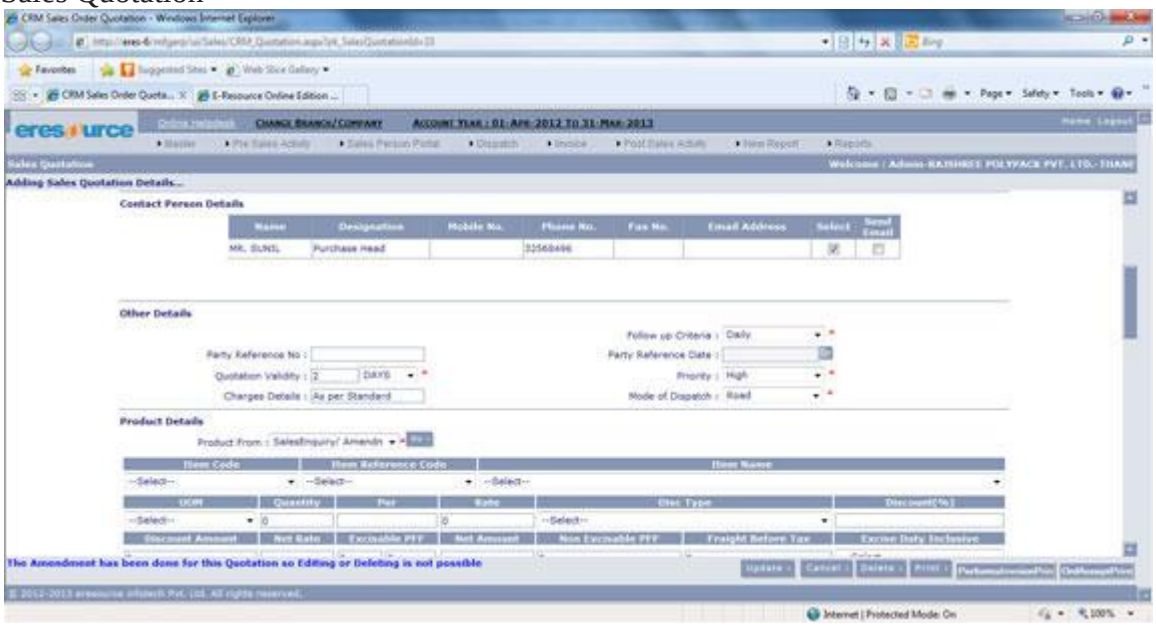
Sales Enquiry Amendment



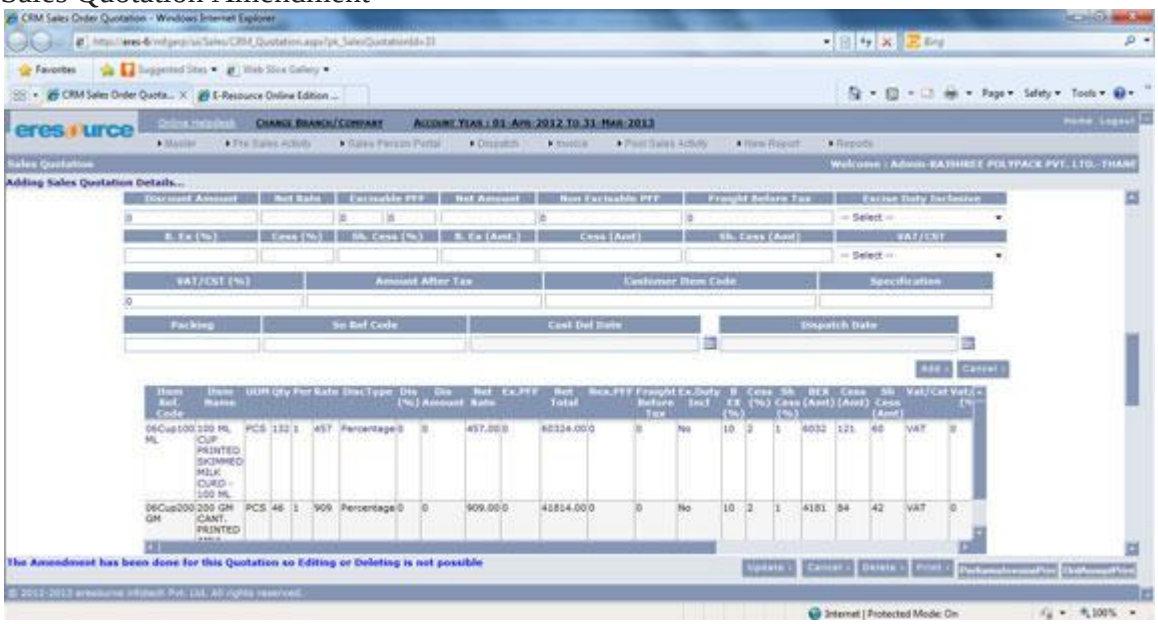
Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401C Semester: IV Year: 2017-2019

Sales Quotation



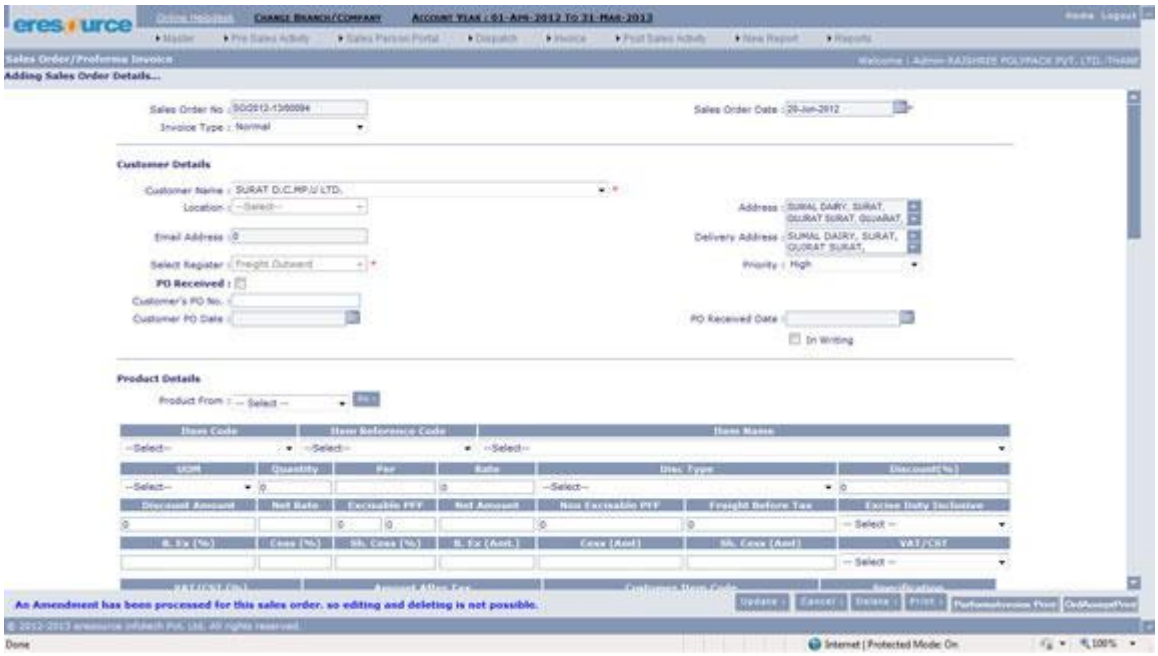
Sales Quotation Amendment



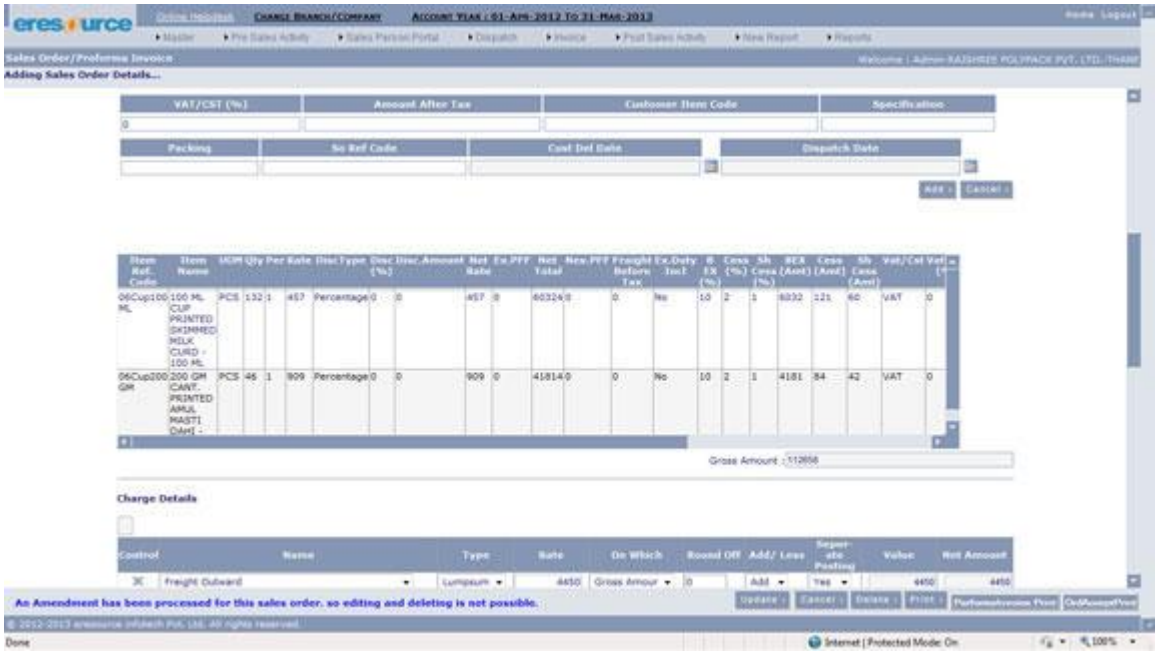
Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401C Semester: IV Year: 2017-2019

Sales Order



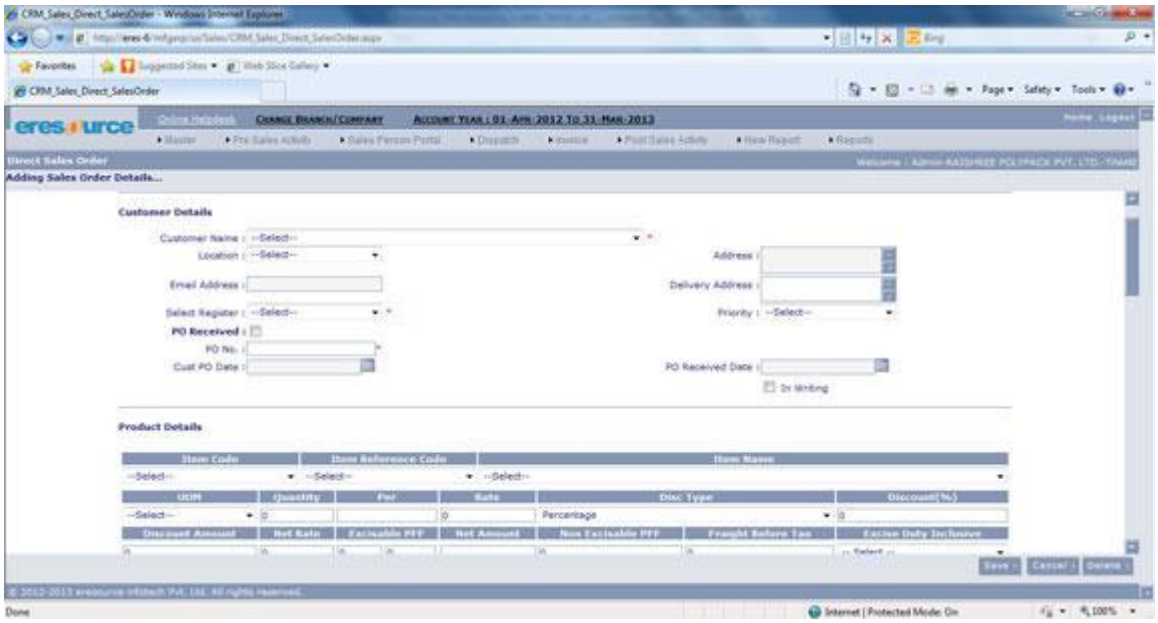
Sales Order Amendment



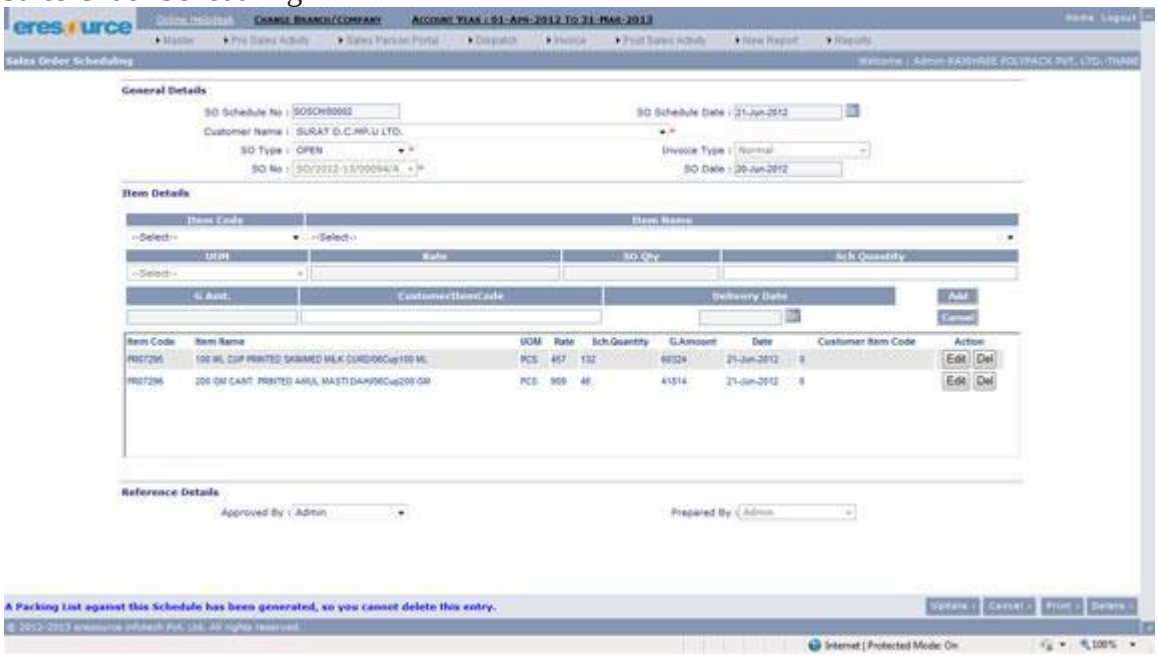
Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401C Semester: IV Year: 2017-2019

Direct Sales Order



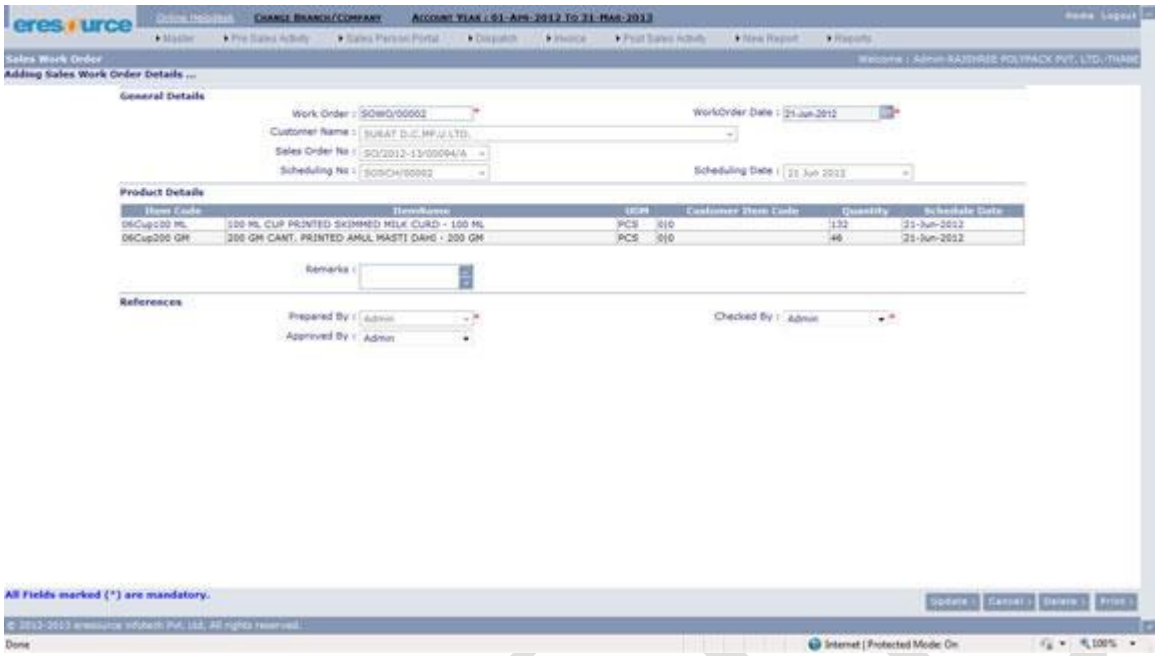
Sales Order Scheduling



Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401C Semester: IV Year: 2017-2019

Sales Work Order



ERP Software for Trading & Distribution Company

ERP eTrade helps predictable business performance by enabling actionable information to every decision maker in the company. It comprehensively covers CRM, Order Management, supply chain and business accounting for small, medium, and large trading enterprises. It creates greater room to grow your business rather than just manage it.

eresource ERP for Trading business is a 100% Browser based (Web based) ERP solution which promises you to connect all your offices, branches under a central system. eresource eTrade is compatible to all browsers and technologies, likewise, Android, Microsoft and IOS. You can run the entire trading business even on your android handheld device. Deployment of the ERP can be done either on-the-premises or also on-the-cloud.

Trading Model View

>> Customer Relationship (CRM)
>> Purchase Management
>> Inventory Management
>> Human Resource Management
>> Finance Resource Management
>> Warehouse Management

Class: II MBA Course Name: Enterprise resource planning

Course code: 17MBAPS401C Semester: IV Year: 2017-2019

Electronic Data Interchange (EDI) is the computer-to-computer exchange of business documents in a basic electronic format in between business partners. EDI was first developed with the goal of speeding the movement of shipping and transport files. Its application has broadened from allowing the electronic exchange of purchase orders, acknowledgments and invoices to include international procurement and sourcing.

Being the force that helps with the motion of goods throughout our planet, the transport market is possibly the most crucial to our economies. Not unexpected, the first market to obtain involved with EDI was the transport market.

ELECTRONIC DATA INTERCHANGE (EDI) FUNCTION

Here is the details of the definition of EDI and how does it work:

Computer-to-computer – EDI changes postal mail, fax and email. While email is also an electronic technique, the documents exchanged via email needs to still be handled by people instead of computers. Having actually individuals involved slows down the processing of the documents as well as introduces mistakes. Instead, EDI documents can flow directly through to the proper application on the receiver's computer system (e.g., the Order Management System) and processing can start right away.

Business files – These are any of the documents that are usually exchanged in between businesses. The most typical files exchanged through EDI are purchase orders, invoices and advance ship notifications. However there are lots of, numerous others such as bill of lading, customs files, inventory files, delivering status documents and payment files.

Standard format – Because EDI files must be processed by computer systems rather than humans, a basic format must be made use of so that the computer system will be able to read and comprehend the files. A conventional format explains what each piece of information is and in what format (e.g., integer, decimal, mmddyy). Without a standard format, each company would send files using its company-specific format and, much as an English-speaking individual most likely does not understand Japanese, the receiver's computer system does not comprehend the company-specific format of the sender's format. The American National Standards Institute (ANSI X12) and the United Nations (EDI for Administration, Commerce and Trade, EDIFACT) maintain the most commonly used standards.

Business partners – The exchange of EDI files is normally between 2 various companies, referred to as business partners or trading partners. For instance, Company A may buy items from Company B. Company A sends orders to Company B. Company A and Company B are business partners.

EDI IN TRANSPORTATION

EDI has actually ended up being incredibly popular in the haulage and transport market as transportation companies and load brokers are adjusting their systems to cater for EDI document exchange.

The advantages for those working in the transport industry is that regular high volume communications can be automated allowing dispatchers and accounts receivables staff more time to concentrate on more productive/profitable tasks and supply clients with much better customer care. One of the major benefits of EDI is that it gets rid of a dispatcher from having to by hand crucial information into the dispatch functional and billing system. This leads to saving money and time while getting rid of any pricey data entry mistakes. The other benefit is that transportation companies who are EDI certified can interact seamlessly and digitally with all celebrations in the supply chain process.

All data is interacted between the provider, carrier, and the consignee in electronic EDI documents known as ANSI X12 "Transaction Sets". The most frequently used EDI Transaction Sets in the Transportation Industry are:

204 – Motor Carrier Load Tender: The carrier or 3PL via a TMS sends this transaction set to YRC the carrier to ask for a delivery pick-up.

990 – Response to the Load Tender: Used by motor providers to suggest whether it will certainly pick up a certain shipment formerly provided by the shipper. This deal is generated in response to a 204 deal. This is made use of by the shipper or 3rd party making– or "tender"– the offer of the shipment. The 990 is made use of to react to that offer. It may also be made use of to accept or turn down a Spot Bid Request from a carrier.

211 – Bill of Lading: The carrier or 3PL through a TMS sends this transaction set to the carrier to supply us detailed Bill of Lading information relevant to a shipment.

212 – Delivery Trailer Manifest: The Carrier sends this deal set to the consignee or other interested celebrations, noting the contents of a trailer that contains numerous shipments that have actually hurt for shipment.

214 – Shipment Status Message: The provider sends this deal set to the shipper and/or consignee to offer updated information on your shipments. Data consists of dates, times, places, path and reference numbers.

210 – Freight Details and Invoice: The carrier sends this deal set to the customer or 3rd party as an invoice to demand payment for services rendered. It offers comprehensive information of charges.

820 – Payment Order/Remittance Advice: The payer (carrier or your third-party payer) sends this deal set to the carrier to supply the carrier remittance/payment information.

997 – Functional Acknowledgement: This deal set is sent in response to each transaction set received to suggest acceptance by shipper, provider or payee. It is an acknowledgement by any celebration to another party of data received.

SAP R/3 System

SAP R/3 pursues a flexible and modular structure of individual components. In previous page you have reviewed the general module groups such financials. If we would like to see them altogether in terms of sub modules, these individual components are as follows:

- Basic System:
  - Basic components (BC)
  - Advanced business application programming (ABAP4)
- Accounting System
  - Financial accounting (FI)
  - Controlling (CO)
  - Asset Management (AM)
- Production and logistics
  - Sales and Distribution (SD)
  - Materials Management (MM)
  - Production Planning (PP)
  - Quality Management (QM)
  - Plant Management (PM)
- Others
  - Project System (PS)
  - Human Resources (HR)
  - Workflow (WF)
  - Industry Solutions (IS)

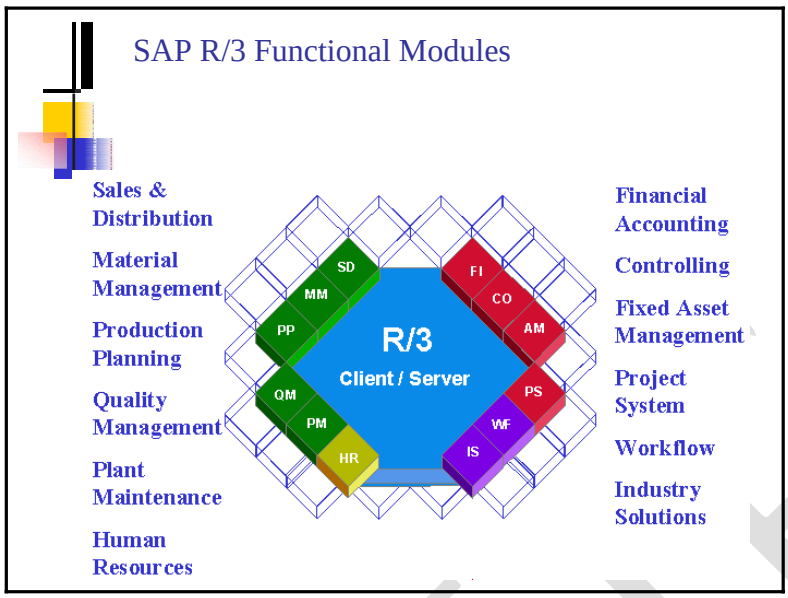


Figure 1.4. SAP R/3 Modules

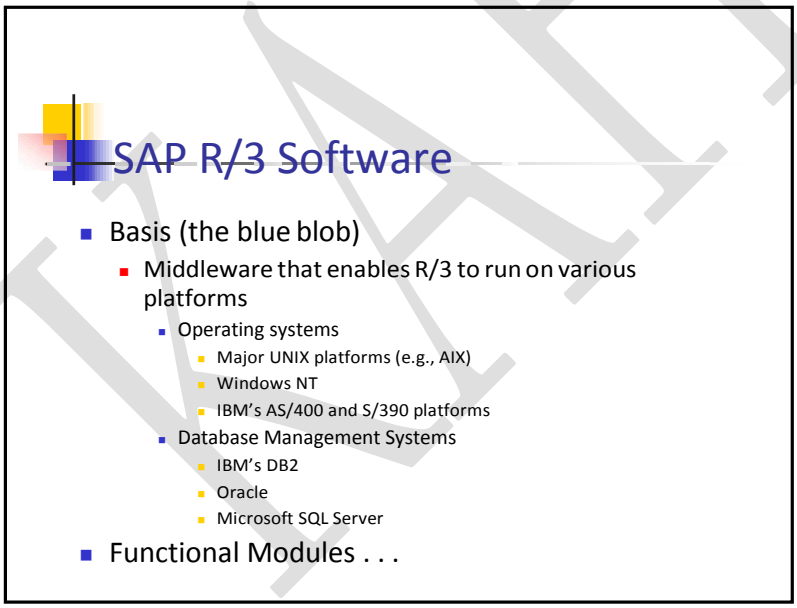


Figure :SAP Basis

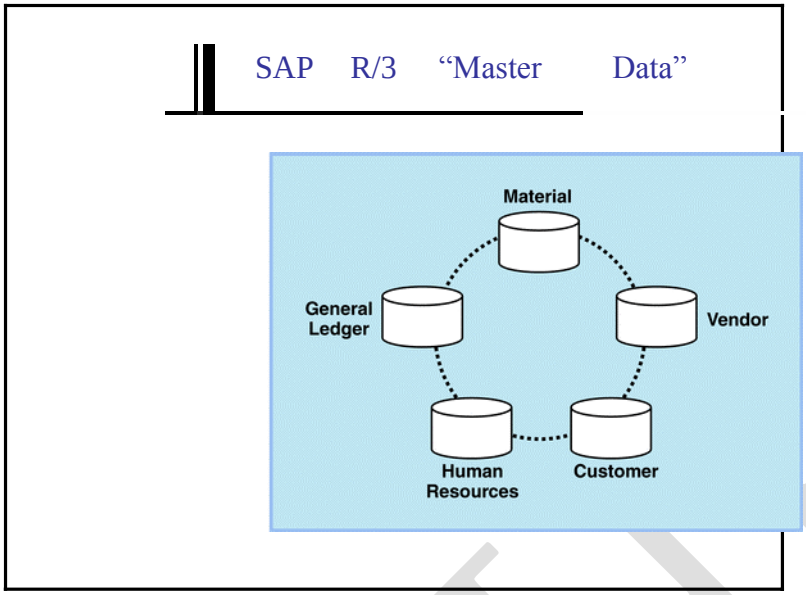


Figure: SAP Master Data

BPR Implementation:

When British Telecom had announced their Business Plan, all competitors were eager to find out who would be the new CEO of the organization. To the surprise of all the new CEO it was the customer. The company had decided to transform all the operations of the organization the way customers wanted them to operate. The most important action in applying BPR is the company's strategic goal to provide customer oriented services. BPR is a technique used to implement this type of organizational structure. Having the management commitment for change, another very important factor for implementing BPR, is the enabling role of Information Technology. The way that businesses are organized around departments is very logical since, for instance, there were physical barriers in the communication of the accounting department with production department. (The warehouse could be in another location in the other part of the city). So it wasn't possible for a crossfunctional team to communicate efficiently. In the 90s when telecommunication technologies were becoming abundant and low costing BPR was becoming a world-wide applicable managing technique for business upgrade, enabled by the technology. Employees can easily operate as a team using intranets/extranets, workflow and groupware applications, eliminating distances. We can work together even though we are located in different places.

Objectives of BPR: When applying the BPR management technique to a business organization the implementation team effort is focused on the following objectives: Customer focus: Customer service oriented processes aiming to eliminate customer complaints. In service sector and production department customer is the main source of revenue.

Speed: Dramatic compression of the time it takes to complete a task for key business processes. For instance, if process before BPR had an average cycle time 5 hours, after BPR the average cycle time should be cut down to half an hour. Compression: Cutting major tasks of cost and capital, throughout the value chain. Organizing the processes a company develops transparency throughout the operational level reducing cost. For instance the decision to buy a large amount of raw material at 50% discount is connected to eleven cross checking in the organizational structure from cash flow, inventory, to production planning and marketing. These checking become easily implemented within the cross-functional teams, optimizing the decision making and cutting operational cost. Flexibility: Adaptive processes and structures to changing conditions and competition. Being closer to the customer the company can develop the awareness mechanisms to rapidly spot the weak points and adapt to new requirements of the market.

Innovation: Leadership through imaginative change providing to organization competitive advantage.

Productivity: Improve drastically effectiveness and efficiency.

Introduction into Business Reengineering The first step in reengineering is to prepare and communicate the "case for action" and the "vision statement". The "case for action" is a description of the organization"s business problem and current situation; it presents justification for the need for change. The "vision statement" describes how the organization is going to operate and outlines the kind of results it must achieve. This qualitative and quantitative statement can be used during a BPR effort, as a reminder of reengineering objectives, as a metric for measuring the progress of the project, and as a prod to keep reengineering action going. The articulation and the communication of the case for action and the vision statement is the leader"s (CEO) responsibility, who should inform firstly the senior management team and secondly the rest of the organization.

2. Identification of Business Processes During this phase, the most important business processes are identified and are described from a global perspective using a set of process maps. Process maps give a picture of the work flows through the company. They show high-level processes, which can be decomposed into sub-processes on separate sub-process maps. Process maps are also used as a means of communication to help people discuss reengineering. The output of this phase is a number of process maps reflecting how these high-level processes interact within the company and in relation to the outside world.

3. Selection of Business Processes It is unrealistic to reengineer all the high level processes of an organization at the same time. Therefore, it has to be decided which are the

processes to be redesigned. This is a very important part of a BPR effort. Candidate for reengineering are the most problematic processes those with great impact to customers or processes with more chances to be successfully reengineered, processes that contribute to organization‟s objectives and so on. According to an organization‟s strategic objectives more criteria could be defined for selecting processes for redesign, such as whether a process contributes to the organization‟s strategic direction, has an impact on customer‟s satisfaction e.t.c.

4. Understanding of Selected Business Processes Before proceeding to redesign, the reengineering team needs to gain a better understanding of the existing selected processes, concerning what they do, how well or how poorly they perform, and the critical issues that govern their performance. Detailed analysis and documentation of current processes is not within the scope of this phase. The objective is the provision of a high level view of the process under consideration, in order the team members to have the intuition and insight required to create a totally new and superior design.

5. Redesign of the Selected Business Processes This is the most creative phase of the methodology, because new rules and new ways of work should be invented. Iimagination and inductive thinking should characterize this phase. Redesigning a process is not algorithmic or routine and therefore Hammer and Champy suggest three kinds of techniques that can help reengineering teams to generate new ideas: As redesign proceeds teams can consider these techniques again to stimulate additional thought. 6. Implementation of Redesigned Business Processes The last phase covers the implementation phase of the BPR project. Hammer/Champy do not talk about implementation as much about project planning. They believe that the success of the implementation depends on whether the five preliminary phases have been properly performed.

Design and Build a Prototype of the Process

The final step in a redesign effort is the design of the new process. The actual design of the new process should be viewed as a prototype and successive iterations should be expected. Three key factors and tactics are considered in process design and prototype: • using IT as a Design Tool • understanding generic design criteria • creating organizational prototypes

Consolidated BPR implementation Methodology:

There are various BPR methodologies are available like apart from Hammer/Champy methodology and Davenport Methodology like Manganelli/Klein Methodology, Kodak Methodology, Process Analysis and Design Method (PADM) and some more are available comparison between them shows a similar way of approach which is consolidated

In consolidated methodology here are 5 steps and they are explained below i. Activity #1: Preparing for Reengineering: Planning and preparation are vital factors for any activity or

event to be successful and reengineering is no exception. There should be a significant need for the process to be reengineered. This activity begins with the development of executive consensus, on the importance of reengineering and the link between breakthrough business goals and reengineering projects. A mandate for change is produced, and a cross-functional team is established with a plan for the process of reengineering. While forming cross functional team, steps should be taken to ensure that the organization continues to function in the absence of several key players. As typical BPR projects involve cross-functional cooperation and significant changes to the status quo, the planning for organizational changes is difficult to conduct without strategic direction from the top. The impact of the environmental changes that serve as the impetus for the reengineering effort must also be considered in establishing guidelines for the reengineering project. Another important factor to be considered while establishing the strategic goals for the reengineering effort is to make it your first priority to understand the expectations of your customers and where your existing process falls short of meeting those requirements. Having identified the customer driven objectives, the mission or vision statement is formulated. The vision is what a company believes it wants to achieve when it is done, and a well-defined vision will sustain a company‟s resolve through the stress of the reengineering process. It can act as the flag around which to rally the troops when the morale begins to sag and it provides the yard stick for measuring the company‟s progress. ii. Activity #2: Map and Analyze As-Is Process: Before the reengineering team can proceed to redesign the process, they should understand the existing process. Although some BPR proponents (in particular Hammer and Champy) argue against analyzing the current enterprise, saying that i Business Process Reengineering for Retail Bank IDRBT Page 17 which are in dire straits might go the Hammer and Champy way (attempt a new process design while totally ignoring the existing processes) most organizations need to map the existing processes first, analyze and improve on it to design new processes. The important aspect of BPR (what makes BPR, BPR) is that the improvement should provide dramatic results. Many people do not understand the value of an As-Is analysis and rather prefer to spend a larger chunk of their valuable time on designing the To-Be model directly. What follows is an illustration that illustrates this fallacy. A large manufacturer spent six million dollars over a period of one year in a bid to develop a parts-tracking system and was all set to go online. Only then did he realize that he had totally overlooked a small piece of information – „the mode of transmission of information between the scheduling staff and the shop floor was through a phone call.‟ But just because this small yet vital information had not been documented all his efforts added up to naught and the whole system that he had so painstakingly developed had to be scrapped. Alas! He had recognized the need for an as-Is analysis, way too late. The main objective of this phase is to identify disconnects (anything that prevents the process from achieving desired results and in particular information transfer between organizations or people) and value adding processes]. This is initiated by first creation and documentation of Activity and Process models making use of the various modeling methods available. Then, the amount of time that each activity takes and the cost that each activity requires in terms of resources is calculated through simulation and activity based costing (ABC). All the groundwork required having been completed; the processes that need to be reengineered are identified. iii. Activity #3: Design To-Be process: The objective of this phase is to produce one or

more alternatives to the current situation, which satisfy the strategic goals of the enterprise. The first step in this phase is benchmarking. "Benchmarking is the comparing of both the performance of the organization‟s processes and the way those processes is conducted with those relevant peer organizations to obtain ideas for improvement." The peer organizations need not be competitors or even from the same industry. Innovative practices can be adopted from anywhere, no matter what their source. Business Process Reengineering for Retail Bank IDRBT Page 18 Having identified the potential improvements to the existing processes, the development of the To-Be models is done using the various modeling methods available, bearing in mind the principles of process design. Then, similar to the As-Is model, we perform simulation and ABC to analyze factors like the time and cost involved. It should be noted that this activity is an iterative process and cannot be done overnight. The several To-Be models that are finally arrived at are validated. By performing Trade off Analysis the best possible To-Be scenarios are selected for implementation. iv. Activity #4: Implement Reengineered Process: The implementation stage is where reengineering efforts meet the most resistance and hence it is by far the most difficult one. If we expect that the environment would be conducive to the reengineering effort we are sadly mistaken. The question that confronts us would be,‟ If BPR promises such breath taking results then why wasn‟t it adopted much earlier?‟ We could expect to face all kinds of opposition - from blatantly hostile antagonists to passive adversaries: all of them determined to kill the effort. When so much time and effort is spent on analyzing the current processes, redesigning them and planning the migration, it would indeed be prudent to run a culture change program simultaneously with all the planning and preparation. This would enable the organization to undergo a much more facile transition. But whatever may be the juncture in time that the culture change program may be initiated, it should be rooted in our minds that „winning the hearts and minds of everyone involved in the BPR effort is most vital for the success of the effort. Once this has been done, the next step is to develop a transition plan from the As-Is to the redesigned process. This plan must align the organizational structure, information systems, and the business policies and procedures with the redesigned processes. "Rapid implementation of the information system that is required to support a reengineered business process is critical to the success of the BPR project. The IDEF models that were created in the As-Is can be mapped to those created during the To-Be and an initial list of change requirements generated. Additional requirements for the construction of the To-Be components can be added and the result organized into a Work Breakdown Structure (WBS). Recent developments in BPR software technologies enable automatic migration of these WBS activity/relationships into a process modeling environment. The benefit here is that we can now define the causal and time sequential relationships between the activities planned." Using prototyping and simulation techniques, the transition plan is validated and its pilot versions are Business Process Reengineering for Retail Bank IDRBT Page 19 designed and demonstrated. Training programs for the workers are initiated and the plan is executed in full scale. V Activity #5: Improve Process Continuously: A process cannot be reengineered overnight. A very vital part in the success of every reengineering effort lies in improving the reengineered process continuously. The first step in this activity is monitoring. Two things have to be monitored – the progress of action and the results. The progress of action is measured by seeing how much more informed the people feel, how

much more commitment the management shows and how well the change teams are accepted in the broader perspective of the organization. This can be achieved by conducting attitude surveys and discrete „fireside chats" with those initially not directly involved with the change. As for monitoring the results, the monitoring should include such measures as employee attitudes, customer perceptions, supplier responsiveness etc. Communication is strengthened throughout the organization, ongoing measurement is initiated, team reviewing of performance against clearly defined targets is done and a feedback loop is set up wherein the process is remapped, reanalyzed and redesigned. Thereby continuous improvement of performance is ensured through a performance tracking system and application of problem solving skills. Continuous improvement (TQM) and BPR have always been considered mutually exclusive to each other. But on the contrary, if performed simultaneously they would complement each other wonderfully well. In fact TQM can be used as a tool to handle the various problems encountered during the BPR effort and to continuously improve the process. In corporations that have not adopted the TQM culture as yet, application of TQM to the newly designed processes should be undertaken as a part of the reengineering effort.

Key Success Factors for BPR: 1.Top Management Sponsorship: Major business process change typically affects processes, technology, job roles and work place culture. Significant changes to even one of these areas require resources, money, and leadership. Changing them simultaneously is an extraordinary task. If top management does not Business Process Reengineering for Retail Bank IDRBT Page 20 provide strong and consistent support , most likely one of these three elements will be present over the life of the project, severely crippling the chance for success.

2. Effective communication Effective communication is considered a major key to successful BPR-related change efforts. Communication is needed throughout the change process at all levels and for all audiences even with those not involved directly in the re-engineering project. Effective communication between stakeholders inside and outside the organisation is necessary to market a BPR program and to ensure patience and understanding of the structural and cultural changes needed as well as the organisation's competitive situation. Communication should take place frequently and in both directions between those in charge of the change initiatives and those affected by them. Communication should be open, honest, and clear, especially when discussing sensitive issues related to change such as personnel reductions

3. An adequate job integration approach Several researchers emphasize that designing and implementing an adequate organisational human resources infrastructure is important to a BPR project's success. Job and labor integration (case worker) is the most appropriate approach of human resources design that supports the process-based organisational structure rather than a function-based one. When individuals within a process perform a series of tasks efficiently, product quality, processing time, and cost are all going to improve. However, the move to integrate human resources architecture necessitates a careful consideration of all related organisational changes.

4. Adequate resources Adequate resources and sufficient budget allocated properly are important for a successful BPR project.

5. Appropriate use of methodology Establishing a disciplined approach for BPR and using a sound methodology are prerequisites for BPR success. A BPR methodology should be designed or selected creatively to satisfy the Business Process Reengineering for Retail Bank IDRBT Page 21 current needs of the organization. Adequate customization of available BPR methodologies determines the level of comprehensiveness and effectiveness that a new customized BPR methodology can reach.

6. Effective BPR teams Cross-functional BPR teams are a critical component of successful BPR implementation. Teams should be adequately composed. Team members should be experienced in variety of techniques. Teams should be made up of people from both inside and outside the organisation. The determinants of an effective BPR team are as follows: competency of team members, their credibility within the organisation and their creativity, team empowerment, motivation, effective team leadership, the training of members in process mapping and brainstorming techniques, proper organisation of the team, complementary skills among team members, adequate size, interchangeable accountability, clarity of work approach, and specificity of goals.

7. Appropriate job definitions and allocation of responsibilities As BPR results in a major structural change in the form of new jobs and responsibilities, it becomes a prerequisite for successful implementation to have formal and clear descriptions of all jobs and responsibilities that the new designed processes bring along with them.

8.Factors related to BPR project management Successful BPR implementation is highly dependent on an effective BPR programme management which includes adequate strategic alignment, effective planning and project management techniques, identification of performance measures, adequate resources, appropriate use of methodology, external orientation and learning, effective use of consultants , building process vision, effective process redesign, integrating BPR with other improvement techniques, and adequate identification of the BPR value.

Expert System Technology

There are several levels of ES technologies available. Two important things to keep in mind when selecting ES tools include:

1. The tool selected for the project has to match the capability and sophistication of the projected ES, in particular, the need to integrate it with other subsystems such as databases and other components of a larger information system.

2. The tool also has to match the qualifications of the project team.

Expert systems technologies include:

1. ***Specific expert systems***

- These expert systems actually provide recommendations in a specific task domain.

2. ***Expert system shells***

- are the most common vehicle for the development of specific ESs. A shell is an expert system without a knowledge base. A shell furnishes the ES developer with the inference engine, user interface, and the explanation and knowledge acquisition facilities.

Domain-specific shells are actually incomplete specific expert systems, which require much less effort in order to field an actual system.

3. ***Expert system development environments***

- these systems expand the capabilities of shells in various directions. They run on engineering workstations, minicomputers, or mainframes; offer tight integration with large databases; and support the building of large expert systems.

4. ***High-level programming languages***

Several ES development environments have been rewritten from LISP into a procedural language more commonly found in the commercial environment, such as C or C++. ESs are now rarely developed in a programming language.

Roles in Expert System Development

Three fundamental roles in building expert systems are:

1. ***Expert*** - Successful ES systems depend on the experience and application of knowledge that the people can bring to it during its development. Large systems generally require multiple experts.

2. ***Knowledge engineer*** **-** The knowledge engineer has a dual task. This person should be able to elicit knowledge from the expert, gradually gaining an understanding of an area of expertise. Intelligence, tact, empathy, and proficiency in specific techniques of knowledge acquisition are all required of a knowledge engineer. Knowledge-acquisition techniques include conducting interviews with varying degrees of structure, protocol analysis, observation of experts at work, and analysis of cases.

On the other hand, the knowledge engineer must also select a tool appropriate for the project and use it to represent the knowledge with the application of the ***knowledge acquisition facility***.

3. ***User*** - A system developed by an end user with a simple shell, is built rather quickly an inexpensively. Larger systems are built in an organized development effort. A prototype-oriented iterative development strategy is commonly used. ESs lends themselves particularly well to prototyping.

Development and Maintenance of Expert Systems

Steps in the methodology for the iterative process of ES development and maintenance include:

1. Problem Identification and Feasibility Analysis:

- the problem must be suitable for an expert system to solve it.

- must find an expert for the project

- cost-effectiveness of the system has to be established (feasibility)

2. System Design and ES Technology Identification:

- the system is being designed. The needed degree of integration with other subsystems and databases is established

- concepts that best represent the domain knowledge are worked out

- the best way to represent the knowledge and to perform inferencing should be established with sample cases

3. Development of Prototype:

- knowledge engineer works with the expert to place the initial kernel of knowledge in the knowledge base.

- knowledge needs to be expressed in the language of the specific tool chosen for the project

4. Testing and Refinement of Prototype:

- using sample cases, the prototype is tested, and deficiencies in performance are noted. End users test the prototypes of the ES.

5. Complete and Field the ES:

- the interaction of the ES with all elements of its environment, including users and other information systems, is ensured and tested.

- ES is documented and user training is conducted

6. Maintain the System:

- the system is keep current primarily by updating its knowledge base.

- interfaces with other information systems have to be maintained as well, as those systems evolve.

Expert Systems in Organizations: Benefits and Limitations

Expert systems offer both tangible and important intangible benefits to owner companies. These benefits should be weighted against the development and exploitation costs of an ES, which are high for large, organizationally important ESs.

Benefits of Expert Systems

An ES is no substitute for a knowledge worker's overall performance of the problem-solving task. But these systems can dramatically reduce the amount of work the individual must do to solve a problem, and they do leave people with the creative and innovative aspects of problem solving.

Some of the possible organizational benefits of expert systems are:

1. An Es can complete its part of the tasks much faster than a human expert.

2. The error rate of successful systems is low, sometimes much lower than the human error rate for the same task.

3. ESs make consistent recommendations

4. ESs are a convenient vehicle for bringing to the point of application difficult-to-use sources of knowledge.

5. ESs can capture the scarce expertise of a uniquely qualified expert.

6. ESs can become a vehicle for building up organizational knowledge, as opposed to the knowledge of individuals in the organization.

7. When use as training vehicles, ESs result in a faster learning curve for novices.

8. The company can operate an ES in environments hazardous for humans.

Limitations of Expert Systems

No technology offers an easy and total solution. Large systems are costly and require significant development time and computer resources. ESs also have their limitations which include:

1. Limitations of the technology

2. Problems with knowledge acquisition

3. Operational domains as the principal area of ES application

4. Maintaining human expertise in organizations

		Un
S.no	**Question**	**Option 1**
1	The most important objectives of _____ is to convince customers to make a purchase	Direct marketing
2	_____is not a component in SAP Sales and Distribution	Sales Support
3	While creating a Sales order, if you don't mention Sales area, where does the system take the sales area_____	Product Master
4	The _____activity comes under presales activities in Sales and distribution	Inquiries
5	In SAP system, when customer master is changed, these changes are reflected in_____	Yes, all the fields to existing orders
6	_____control is used for copying from billing doc to billing doc	VTAA
7	In _____ sales order type, customer place an order, pick up the order and pay for the goods	Cash Sales
8	_____ is the correct definition for a match code in billing function	It represent a Transaction Code
9	_____is not relevant in determining pricing procedure	Ship to party
10	Electronic Data Interchange is necessary in_____	B2C e-Commerce
11	EDI requires_____	representation of common business documents in computer readable forms
12	EDI standards are_____	not universally available
13	For secure EDI transmission on internet_____	MIME is used
14	PM will also be involved in making choices that require balancing in_____	Goals of the project
15	Projects provide an excellent growth environment for future executives and for developing_____	Sales skills
16	_____ are knowledge based system to which present rules are applied to solve a particular problem.	ES
17	_____is a component of an expert system.	explanation module
18	_____is an application of the computer where the computer makes decisions or judgments that appear to require human intuition, reasoning and intelligence.	AI

19	_____ attempt to provide the same judgmental advice that human experts such as doctors provide.	ES
20	Expert systems are part of the general area of research known as _____	AI
21	The field that investigates the mechanics of human intelligence is_____	history
22	_____is the name of the computer program that simulates the thought processes of human beings	Human logic
23	_____is the name of the computer program that contains the distilled knowledge of an expert	Data base management system
24	The explanation facility of an expert system may be used to_____	construct a diagnostic model
25	A bidirectional feedback loop links computer modelling with_____	artificial science
26	What are the problems with re-structuring?	loss of comments
27	_____ became the fundamental concept of production management and control.	BOM
28	In the _____, ERP packages were targeted at the manufacturing industry.	1970's
29	The _____ should plan well and execute perfectly the implementation of ERP.	Organisation
30	Which is one of the most critical steps in the ERP implementation?	Creation of Organisational Model
31	What is one of the key resources of every organisation, in today's competitive business environment?	Employee
32	When a customer needs to check the performance of the company before he places an order, which department has to be in a position to provide the necessary information?	Production
33	In an integrated data model, what gives a snapshot of the organisation at any given time	ERP
34	OLAP stands for_____	On-Line Analytical Processing
35	Data mining is the process of identifying valid, new, potentially useful, and ultimately clear _____ from databases.	Decision
36	DSS is quite _____ and is available on request	Structured
37	ERP system is built on a _____ utilising a common computing platform	Centralised database

38	ERP vendors usually design their systems around standard business processes, based upon _____	Business standards
39	The _____ module's internal functions do not directly interact with the data or processes of other modules	Finance
40	Which among the following systems can be assigned to a cost centre directly which illustrates the interface to the cost accounting system?	FAPA
41	The processes described in the quality manual can be implemented and automated in the EDP system. Here what does EDP stand for?	Electronic Data Processing
42	During the 80s, tools were developed to assist the planning of_____	Production levels
43	ERP is restricted to being a _____operations system in the country	Finance
44	_____ allow companies to enter requirements for various types of items	Purchase order
45	What is EDI?	Electronic Data Interface
46	_____ module supports the entire sales and purchase processes from start to finish	Order management
47	_____evaluates whether or not customer contract agreement are being met	Purchase order management
48	_____ maintains reports of warehouse supplies	ERP financial module
49	Web ERP is suitable for _____	Both small and big organisations
50	Installing ERP Inventory systems is _____	Simple
51	In recent years, CRM has become a primary component of_____	ERP vendors
52	_____is the most common operating system for running CRM software	UNIX
53	_____ module keeps track of liquidation process	Market management module
54	_____ system maintains employees information in graphical format	Employee master data
56	The vendor should have a _____, who must constantly interact with the implementation team	Liaison officer
57	ERP use software appication to_____the processes of an Organisation	Automate
58	Which one of the following is not a myth about the ERP_____	ERP means more work and procedure

59	ERP system is built on a _____ utilising a common computing platform	Centralised database
60	While creating a Sales order, if you don't mention Sales area, where does the system take the sales area_____	Product Master

KARPAGAM ACADEMY OF HIGHER EDUCATIO

Department of Management

ıit 5- Multiple Choice Questions- Each Question Carry ON

Option 2	Option 3
Person selling	Person to person communication
Production Planning	Related
Customer Info Record	Customer Master
Quotation	Sales Order
Yes but only address is copied to existing orders	Changes only application to new orders
VTLA	VTFL
Rush Order	Scheduling Agreement
A match code is used as an alternative of company code	It represents a comparison key to search
Sales document type	Delivery document type
C2C e-Commerce	B2B e-Commerce
data entry operators by receivers	special value added networks
essential for B2B commerce	not required for B2B commerce
S/MIME is used	PGP is used
Goals of the firm	Goals of the resources
Manufacturing skills	Managerial skills
AI	KBS
knowledge base	natural language interface for the user
ES	KBS

AI	KBS
ES	STUDENT
cognitive science	psychology
Expert reason	Expert system
Management information System	Expert system
expedite the debugging process	explain the system's reasoning process
heuristic processing	human intelligence
Loss of documentation	Heavy computational demands
MRP	ERP
1990's	1980’s
System developers	Vendors
Creation of Integrated Data Model	Creation of Business Model
Information	ERP
Quality	Marketing
Database	Management
On-Line Account Processing	On-Line Arithmetic Processing
Strategies	Information
Non-structured	Semi-structured
Individual databases	Modular databases

Global standards	Best business practices
Quality Management	Sales and Distribution
Purchasing	Sales and Distribution
Electronic Dictionary Project	Electrical Data Processing
Priority planning system	Capacity requirements
Budgeting modules	Manufacturing
Purchase requisitions	Invoice
Exchange Data Interchange	Exchange Data Interface
Sales management	Purchase management
Sales order management	Master data management
ERP Inventory software module	ERP Resource module
Only big organisations	Medium scale organisations
Trouble free	Complex
ERP software solutions	CRM software
Windows NT	Windows Vista
Liquidation module	Treasury module
Personnel administration	Payroll accounting
Project Supervisor	Project Manager
Speed	Growth
ERP Make's many employees redudant	ERP Integrate and Automote organization process

Individual databases	Modular databases
Customer Info Record	Customer Master

N

E Mark

Option 4		Answer
Integrating programs		Person selling
Foreign Trade		Production Planning
Vendor Master		Customer Master
Purchase Order		Quotation
No changes will be made		Yes but only address is copied to existing orders
VTFF		VTFF
Transaction code		Cash Sales
Customer Info Record		It represents a comparison key to search
Sold to party		Ship to party
Commerce using internet		B2B e-Commerce
special hardware at co-operating Business premises		representation of common business documents in
still being evolved		not universally available
TCP/IP is used		S/MIME is used
Goals of the Management		Goals of the firm
Execution skills		Managerial skills
Base rule 0		ES
Expert system		natural language interface for the user
Base-rule		AI

RAND		ES
RAND		AI
sociology		cognitive science
Personal information		Expert system
Artificial intelligence		Expert system
natural language interface for the user		explain the system's reasoning process
cognitive science		cognitive science
		Heavy computational demands
MRP II		MRP
1960’s		1990's
Top Management		System developers
Creation of Data Model		Creation of Integrated Data Model
Database		ERP
Finance		Marketing
MIS		Database
On-Line Application Processing		On-Line Analytical Processing
Account		Information
Unstructured		Semi-structured
Centralised layout		Centralised layout

Best profitable standards		Business standards
Plant Maintenance		Quality Management
CASO		Sales and Distribution
Employee Development Plan		Electrical Data Processing
Planning functions		Priority planning system
Transaction-oriented		Transaction-oriented
General ledger		Invoice
Electronic Data Interchange		Electronic Data Interchange
Master Data Management		Sales management
Warehouse management		Warehouse management
ERP verification module		ERP Inventory software module
Only small organisations		Both small and big organisations
Cost effective		Simple
CRM solutions		ERP software solutions
Windows XP		Windows XP
Enterprise management module		Treasury module
Benefits administration		Employee master data
Implementation co-ordinator		Implementation co-ordinator
Develop		Automate
ERP is the sole rsponsibility of management		ERP Make's many employees redudant

Centralised layout		Centralised layout
Vendor Master		Customer Master