17BECS401 DATABASE MANAGEMENT SYSTEMS 4H-5C

Instruction Hours/week: L:3 T:0 P:4

COURSE OBJECTIVES:

- To understand the different issues involved in the design and implementation of a database system.
- To study the physical and logical database designs, database modeling, relational, hierarchical, and network models
- To understand and use data manipulation language to query, update, and manage a database
- To develop an understanding of essential DBMS concepts such as: database security, integrity, concurrency, distributed database, and intelligent database, Client/Server (Database Server), Data Warehousing.
- To design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.

COURSE OUTCOMES:

- For a given query write relational algebra expressions for that query and optimize the developed expressions
- For a given specification of the requirement design the databases using E R method and normalization.
- For a given specification construct the SQL queries for Open source and Commercial DBMS -MYSQL, ORACLE, and DB2.
- For a given query optimize its execution using Query optimization algorithms
- For a given transaction-processing system, determine the transaction atomicity, consistency, isolation, and durability.
- Implement the isolation property, including locking, time stamping based on concurrency control and Serializability of scheduling.

UNIT 1

Database system architecture: Data Abstraction, Data Independence, Data Definition Language (DDL), Data Manipulation Language (DML).

Data models: Entity-relationship model, network model, relational and object oriented data models, integrity constraints, data manipulation operations.

UNIT 2:

Relational query languages: Relational algebra, Tuple and domain relational calculus, SQL3, DDL and DML constructs, Open source and Commercial DBMS - MYSQL, ORACLE, DB2, SQL server.

Relational database design: Domain and data dependency, Armstrong's axioms, Normal forms, Dependency preservation, Lossless design.

Query processing and optimization: Evaluation of relational algebra expressions, Query equivalence, Join strategies, Query optimization algorithms.

UNIT 3:
Storage strategies: Indices, B-trees, hashing.
Transaction processing: Concurrency control, ACID property, Serializability of
scheduling, Locking and timestamp based schedulers, Multi-version and optimistic
Concurrency Control schemes, Database recovery.

UNIT 4:
Database Security: Authentication, Authorization and access control, DAC, MAC and
RBAC models, Intrusion detection, SQL injection.
UNIT 5:
Advanced topics: Object oriented and object relational databases, Logical databases, Web
databases, Distributed databases, Data warehousing and data mining.

TEXT BOOKS:
1. "Database System Concepts", 6th Edition by Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw-Hill.

REFERENCES:
1 "Principles of Database and Knowledge – Base Systems", Vol 1 by J. D. Ullman, Computer Science Press.
2 "Fundamentals of Database Systems", 5th Edition by R. Elmasri and S. Navathe, Pearson Education
3 "Foundations of Databases", Reprint by Serge Abiteboul, Richard Hull, Victor Vianu, Addison-Wesley

KARPAGAM ACADEMY OF HIGHER EDUCATION

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

17BECS401 **Database Management Systems** **Lecture plan**

	UNIT I			
1	**Database system architecture:** Data Abstraction,	1	T(1) Page no.3	
2	Data Independence	1	T(1) Page no.28-29	
3	Data Definition Language (DDL)	1	T(1) Page no.26-28	
4	Data Manipulation Language (DML).	1		
5	**Data models:** Entity-relationship model,	1	T(1) Page no.43-47	BB & PPT
6	network model	1	T(1) Page no.55-56	
7	relational and object oriented data model		T(1) Page no.60-65	
8	**Tutorial Hour** - Entity-relationship model	1		
9	integrity constraints, data manipulation operations.	1	T(1) Page no.65-67	
	UNIT-II			
10	**Relational query languages:** Relational algebra, Tuple and domain relational calculus	1	T(1) Page no 68-70	

11	SQL3, DDL and DML constructs, Open source and Commercial	1	T(1) Page no.76-78	BB & PPT
12	DBMS - MYSQL, ORACLE, DB2, SQL server.	1	T(1) Page no.158-168	
13	**Tutorial Hour** - Queries in SQL- Updates- Views	1		
14	**Relational database design:** Domain and data dependency, Armstrong's axioms, Normal forms	1	R(1) page no 3.25-3.32	
15	Dependency preservation,Lossless design.	1	T(1) Page no.193-199	
16	**Query processing and optimization:** Evaluation of relational algebra expressions	1	R(1) page no 4.13-4.18	
17	Query equivalence, Join strategies, Query optimization algorithms.	1	T(1) Page no.204	

	UNIT III			
18	**Storage strategies**: Indices, B-trees	1	T(1) Page no.245-251 R(1) page no 5.2-5.7	**9**
19	Hashing.	1	T(1) Page no.252-255 R(1) page no 5.8-5.10	
20	**Transaction processing:** Concurrency control,	1	T(1) Page no.256-261 R(1) page no 5.10-5.20	
21	**Tutorial Hour** - Secondary storage Devices- Operations on File	1		
22	ACID property	1	R(1) page no 5.21-5.23, T(1) Page no.266-267	
23	Locking and timestamp based schedulers,	1	R(1) page no 6.2-6.8	
24	Multi-version Control schemes	1	R(1) page no 6.8-6.10	

			T(1) Page no.282-283	
25	**Tutorial Hour** - B-Tree - B+Tree Query Processing	1		BB & PPT
26	optimistic Concurrency Control schemes	1	R(1) page no 6.21-6.29 T(1) Page no.288-296	
27	Database recovery.	1	R(1) page no 6.31-6.33	
28	Serializability of scheduling	1	R(1) page no 6.33-6.34	
29	**Tutorial Hour** - optimistic Concurrency Control schemes	1		
	UNIT IV			
30	**Database Security:** Authentication	1	R(1) page no 7.2-7.9 T(1) Page no.319-324	BB & PPT
31	Authorization and access control	1		
32	DAC	1	R(1) page no 7.11-7.19 T(1) Page no.327-339	
33	**Tutorial Hour** - Authorization and access control			
34	MAC	1	R(1) page no 7.20-7.23	
35	RBAC models	1	R(1) page no 8.2-8.7 T(1) Page no.373-381	
36	Intrusion detection, SQL injection. .	1	R(1) page no 8.8-8.11	
37	**Tutorial Hour** - Concepts- Immediate Update- Deferred Update - Shadow Paging.	1		
	UNIT-V			

53	**Advanced topics:** Object oriented database Model	1	R(1) page no 8.24-8.27	BB & PPT
54	object relational database Model	1	R(1) page no 8.29-8.35 T(1) Page no.419-121	
55	Logical database	1	T(1) Page no.429-431 R(1) page no 8.35,	
56	**Tutorial Hour** - Logical database	1		
57	Web databases	1	R(1) page no 9.12-9.15 R(1) page no 9.21-9.22	BB & PPT
58	Distributed databases	1	R(1) page no 9.24-9.26 T(1) Page no.456-461	
59	Data warehousing	1	R(1) page no 9.27-9.35	
60	**Tutorial Hour** - Distributed databases	1		BB & PPT
61	data mining	1	R(1) page no 9.36-9.37	

TEXT BOOKS:

1. "Database System Concepts", 6th Edition by Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw-Hill.

REFERENCES:

1 "Principles of Database and Knowledge – Base Systems", Vol 1 by J. D. Ullman, Computer Science Press.
2 "Fundamentals of Database Systems", 5th Edition by R. Elmasri and S. Navathe, Pearson Education
3 "Foundations of Databases", Reprint by Serge Abiteboul, Richard Hull, Victor Vianu, Addison-Wesley

UNIT I

INTRODUCTION AND CONCEPTUAL MODELING

Introduction to File and Database systems- Database system structure –Introduction and concept Modeling-Database user Data Models – Introduction to Network and Hierarchical Models – ER model – Relational Model – Relational Algebra and Calculus

1. INTRODUCTION

Data: Known facts that can be recorded that have implicit meaning.
E.g. Student roll no, names, address etc

Database: collection of inter-related data organized meaningfully for a specific purpose**.**

DBMS: DBMS is a collection of interrelated data and a set of program to access those data. The primary goal of a DBMS is to provide a way to store and retrieve database information that is both *convenient* and *efficient*.

Database System: Database and DBMS collectively known as database system.
INTRODUCTION TO FILE AND DATABASE SYSTEMS:

Database Applications

- Banking: all transactions
- Airlines: reservations, schedules
- Universities: registration, grades
- Sales: customers, products, purchases
- Online retailers: order tracking, customized recommendations
- Manufacturing: production, inventory, orders, supply chain
- Human resources: employee records, salaries, tax deductions
- Credit card transactions
- Telecommunications & Finance

2. PURPOSE OF DATABASE SYSTEMS

Drawbacks of Conventional File Processing System

i. **Data redundancy and inconsistency**

- Since the files and application programs are created by different programmers over a long period of time, the files have different formats and the programs may be written in several programming language. The same piece of information may be duplicated in several files.
- **For Example:** The address and phone number of particular customer may appear in a file that consists of personal information and in saving account records file also. This redundancy leads to data consistency that is, the various copies of the same data may no longer agree.
- **For example:** a changed customer address may be reflected in personal information file, but not in saving account records file.

ii. **Difficulty in accessing data**

- Conventional file processing environments do not allow needed data to be retrieved in a convenient and efficient manner.
- **For Example:** Suppose that bank officer needs to find out the names of all customers who live within the city's 411027 zip code. The bank officer has now two choices: Either get the list of customers and extract the needed information manually, or ask the data processing department to have a system programmer write the necessary application program. Both alternatives are unsatisfactory.

iii. **Data isolation**

- Since, data is scattered in various files, and files may be in different formats, it is difficult to write new application programs to retrieve appropriate data.

iv. **Concurrent access anomalies**

- In order to improve the overall performance of the system and obtain a faster response time many systems allow multiple users to update the data simultaneously. In such environment, interaction of concurrent updates may results in inconsistent data.
- **For Example:** Consider bank account A, with \$500.If two customers with draw funds (say \$50 and \$100 resp) from account A at the same time, the result of the concurrent executions \$400, rather than \$350. In order to guard against this possibility, some form of supervision must be maintained in the system.

v. **Atomicity Problem**

- System failure will lead to atomicity problem.

- **For Example:** Failure during transfer of fund from system A to A. It will be debited from A but not credited to B leading to wrong transaction.

vi. **Concurrent Access Anomalies**

- In order to improve the overall performance of the system and obtain a faster response time many systems allow multiple users to update the data simultaneously. In such environment, interaction of concurrent updates may result in inconsistent data.
- **For Example:** Consider bank account *A*, containing $500. If two customers withdraw funds say $50 and $100 respectively) from account *A* at about the same time, the result of the concurrent executions may leave the account in an incorrect (or inconsistent) state. Balance will be $400 instead of $350. To protect against this possibility, the system must maintain some form of supervision.

vii. **Security problems**

- Not every user of the database system should be able to access all the data. System should be protected using proper security.
- **For Example:** In a banking system, pay roll personnel should be only given authority to see the part of the database that has information about the various bank employees. They do not need access to information about customer accounts.
- Since application programs added to the system in an ad-hoc manner, it is difficult to enforce such security constraints.

viii. **Integrity problems**

- The data values stored in the database must satisfy certain types of consistency constrains.
- **For Example:** The balance of a bank account may never fall below a prescribed amount (say $100).These constraints are enforced in the system by adding appropriate code in the various application programs.

Advantages of Database

Data base is a way to consolidate and control the operational data centrally. It is a better way to control the operational data. The advantages of having a centralized control of data are:

i. **Redundancy can be reduced**

In non-database systems, each application or department has its own private files resulting in considerable amount of redundancy of the stored data. Thus storage space is wasted. By having a centralized database most of this can be avoided.

ii. **Inconsistency can be avoided**

When the same data is duplicated and changes are made at one side, which is not propagated to the other site, it gives rise to inconsistency. Then the two entries regarding the same data will not agree. So, if the redundancy is removed, chances of having inconsistent data are also removed.

iii. **The data can be shared**

The data stored from one application, can be used for another application. Thus, the data of database stored for one application can be shared with new applications.

iv. **Standards can be enforced**

With central control of the database, the DBA can ensure that all applicable standards are observed in the representation of the data.

v. **Security can be enforced**

DBA can define the access paths for accessing the data stored in database and he can define authorization checks whenever access to sensitive data is attempted.

vi. **Integrity can be maintained**

Integrity means that the data in the database is accurate. Centralized control of the data helps in permitting the administrator to define integrity constraints to the data in the database.

3. VIEW OF DATA

A major purpose of a database system is to provide users with an abstract view of the data. That is, the system hides certain details of how the data are stored and maintained.

Data abstraction

The Complexity is hidden from the users through several level of abstraction. There are three levels of data abstraction:

i. **Physical level:** It is the lowest level of abstraction that describes how the data are actually stored. The physical level describes complex low-level data structures in details.

ii. **Logical level:** It is the next higher level of abstraction that describes what data are stored in the database and what relationships exist among those data.

iii. **View level:** It is the highest level of abstraction that describes only part of the entire database.

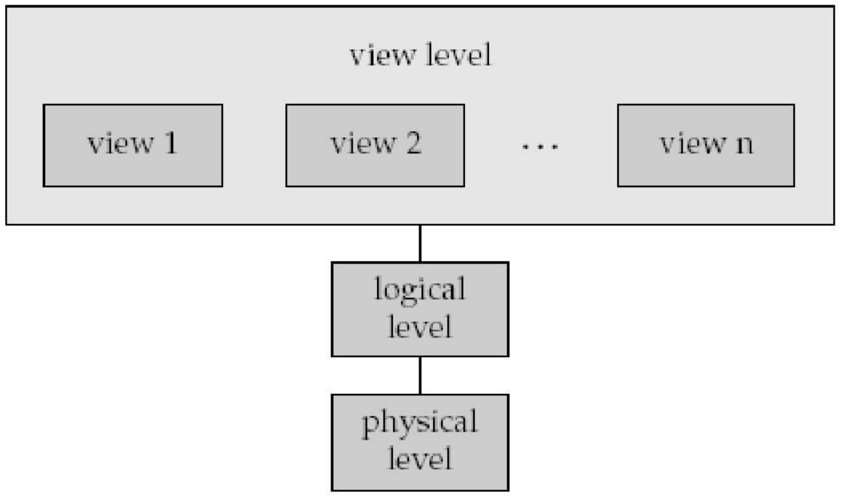


Figure The three levels of data abstraction

Data Independence

The ability to modify a scheme definition in one level without affecting a scheme definition in the next higher level is called data independence. There are two levels of data independence:

1. **Physical data independence** is the ability to modify the physical scheme without causing application programs to be rewritten. Modifications at the physical level are occasionally necessary in order to improve performance.

2. **Logical data independence** is the ability to modify the conceptual scheme without causing application programs to be rewritten. Modifications at the conceptual level are necessary whenever the logical structure of the database is altered.

Logical data independence is more difficult to achieve than physical data independence since application programs are heavily dependent on the logical structure of the data they access.

Instances and schemas

Database change over times as information is inserted and deleted. The collection of information stored in the database at a particular moment is called an **instance** of the database.

The overall design of the database is called the database **schema**.

Types of database schemas

i. **Physical schema**: It describes the database design at the physical level.
ii. **Logical schema:** It describes the database design at the physical level.

iii. **Subschema:** A database may also have several subschemas at the view level called as subschemas that describe different views of the database.

□. DATA MODE□S

- Underlying structure of the database is called as data models.
- It is a collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints.
- It is a way to describe the design of the database at physical, logical and view level.

Different types of data models are:

- Entity relationship model
- Relational model
- Hierarchical model
- Network model
- Object Based model
- Object Relational model
- Semi Structured Data model

<u>Entity relationship model</u>

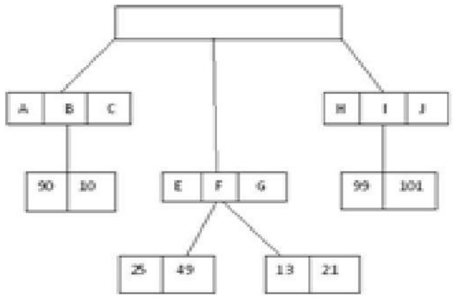
- It is based on a collection of real world things or objects called entities and the relationship among these objects.
- The Entity relationship model is widely used in database design.

<u>Relational Model</u>

- The relational model uses a collection of tables to represent both data and the relationship among those data.
- Each table has multiple columns and each column has a unique name.
- Software such as Oracle, Microsoft SQL Server and Sybase are based on the relational model.
- E.g. Record Based model. It is based on fixed format records of several types.

<u>Hierarchical Model</u>

- Hierarchical database organize data in to a tree data structure such that each record type has only one owner
- Hierarchical structures were widely used in the first main frame database management systems.
- Links are possible vertically but not horizontally or diagonally.



Advantages

- High speed of access to large
- datasets. Ease of updates.
- Simplicity: the design of a hierarchical database is simple.
- Data security: Hierarchical model was the first database model that offered the data security that is provided and enforced by the DBMS.
- Efficiency: The hierarchical database model is a very efficient one when the database contains a large number of transactions, using data whose relationships are fixed.

Disadvantages

- Implementation complexity
- Database management problems
- Lack of structural independence

<u>Network Model</u>

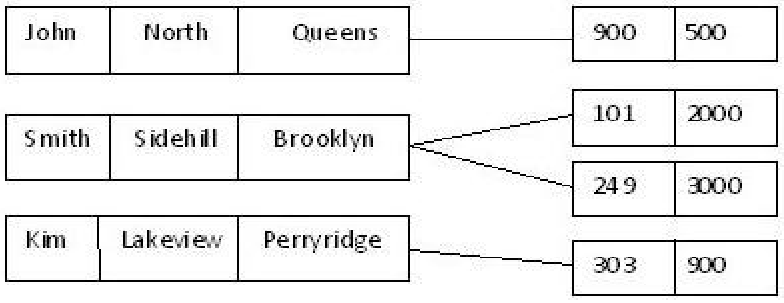
➢ The model is based on directed graph theory.

➢ The network model replaces the hierarchical tree with a graph thus allowing more general connections among the nodes.

➢ The main difference of the network model from the hierarchical model is its ability to handle many-to-many (n: n) relationship or in other words, it allows a record to have more than one parent.

➢ **Example** is, an employee working for two departments.

Sample network model



Advantages:

- Conceptual simplicity :

- Capability to handle more relationship types:
- Data independence:

Disadvantages:

- Detailed structural knowledge is required.
- Lack of structural independence.

<u>Object-Based Data model</u>

- The object- oriented model is an extension of E-R model.
- The object- oriented model is based on a collection of objects.
- An object contains values stored in instance variables within the object.
- An object also contains bodies of code that operate in the object these bodies of code are called methods.
- Objects that contain the same types of values and methods are grouped together into classes.

Advantages:

- Applications require less code.
- Applications use more natural data
- model. Code is easier to maintain.
- It provides higher performance management of objects and complex interrelationships between objects.
- Object-oriented features improve productivity.Data access is easy.

<u>Object Relational Model</u>

- Object-relational data model combines the feature of modern object-oriented programming languages with relational database features.
- Some of the object-relational systems available in the market are IBM DB2 universal server, Oracle Corporation‘s oracle 8, Microsoft Corporations SQL server 7 and so on.

<u>Semi Structured Data Model</u>

- This data model allows the individual data items of same type to have different sets of attributes.
- Other data model allows a particular type of data item to have same set of attributes.
- Extensible Markup Language (XML) is used to represent structured data.

5. *DATABASE ▯ AN▯ UA▯ ES*

- A database system provides
  - A **Data Definition Language** to specify the database schema (DDL)
  - A **Data Manipulation Language** to express database queries and updates.

- Data definition and data manipulation languages are not two separate languages but part of a single database language such as SQL language.

Data definition language

- DDL specifies the database schema and some additional properties to data.
- The storage structure and access methods are specified using specified using special type of DDL called s **data storage and data definition langu**age.
- The data values stored in the database must satisfy certain **consistency constraints**. For example, suppose the balance on an account should not fall below $100.
- Database system concentrates on constraints that have less overload.

1. **Domain Constraints:**
   - Domain of possible value should be associated with every attributes.
     E.g. integer type, character type, date/time type
   - Declaring attributes to a particular domain will act as a constraint on that value.
   - They are tested as and when values are entered in to database.
2. **Referential Constraints:**
   - In some cases there will be value that appears in one relation for a given set of attributes also appears for a certain set of attributes in some other relation. Such constraint is called Referential Constraints.
   - If any modification violates the constraints then the action that caused the violation should be rejected.
3. **Assertions**
   - It is a condition that database should always satisfy.
   - Domains and referential integrity are special form of assertion.
     - E.g. Every loan should have a customer whose account balance is minimum of $1000.00
   - Modifications to database should not cause violation to assertion.
4. **Authorization**
   - The users are differentiated as per the access permit given to them on the different data of the database. This is known as authorization.
   - The most common authorizations are

   i. Read authorization

   Allows reading but no modification of data.

   ii. Insert authorization

   Allows insertion of new data but no modification of existing data.

iii. Update authorization

Allows modification but not deletion.

iv. Delete authorization

Allows deletion of data.

- The users may be assigned with all, none or combination of these
- types. The DDL gets some input and generates some output.
- This output is placed in data dictionary which contains Meta
- data. Meta data is data about data.
- Data Dictionary is a special type of table which can only be accessed and updated by database system.
- Database system consults the Data Dictionary before reading or modifying actual data.

Data Manipulation Language

- A **data-manipulation language (DML)** is a language that helps users to access or manipulate data.
- A query is a statement requesting the retrieval of information.
- The portion of DML that involves information retrieval is called as query
- language. There are basically two types of DML:
  - **Procedural DMLs**

    User should specify *what* data are needed and *how* to get those data.
  - **Declarative DMLs** (**nonprocedural** DMLs)

    User should specify *what* data are needed *without* specifying how to get those data. This is easier to learn and user than procedural DML.
- Data manipulation that can be performed using DML are
  - The retrieval of information stored in the database
  - The insertion of new information into the database
  - The deletion of information from the database
  - The modification of information stored in the database

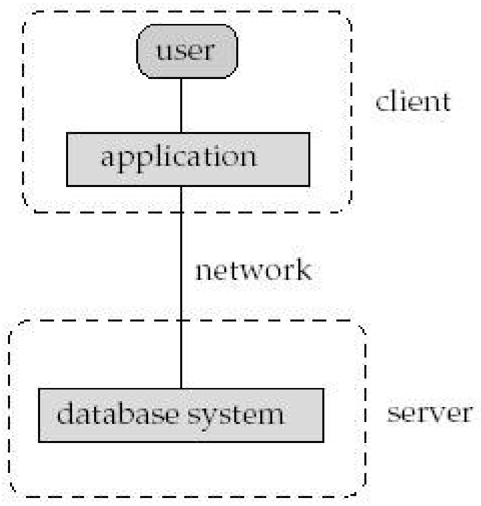
□. DATABASE SYSTEM ARC□ ITECTURE

A database system is partitioned into modules that deal with each of the responsibilities of the overall system. The functional components of a database system can be broadly divided into

- Storage Manager
- Query Processor

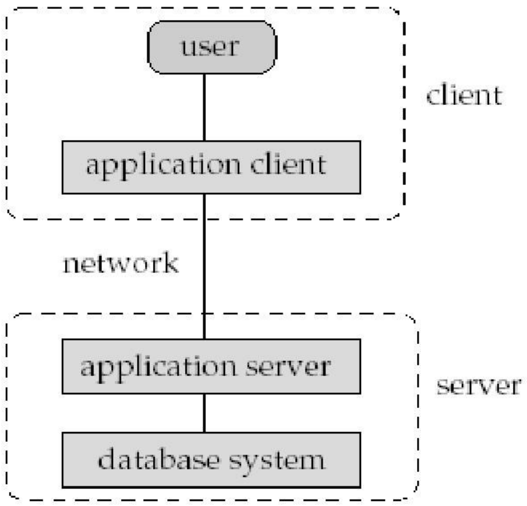
- The database system architecture is influenced by the underlying computer architecture. The database system can be centralized or client server.
- Database systems are partitioned into two or three parts.
- In **two tier architecture**, the application is partitioned into a component that resides at the client machine and invokes database functionality at the server machine through query language.
- Application program interface standards like ODBC and JDBC are used for interaction between the client and the server.

Two tier architecture



- In **three tier architecture**, the client machines act as a front end and do not contain any direct database calls.
- The client end communicates with the application servers through interface.
- The application server interacts with database system to access data.
- The business logic of application says what actions to be carried out under what condition.
- Three tier is more appropriate for large applications.

Three tier architecture



Storage Manager

- A storage manager is a program module that provides the interface between the low level data stored in the database and the application programs and queries submitted to the system.
- The storage manager is responsible for the interaction with the file manager.
- The storage manager translates the various DML statements into low-level file system commands. Thus, the storage manager is responsible for storing, retrieving, and updating data in the database.

Components of the storage manager are:

1. **Authorization and integrity manager**: It tests for satisfaction of various integrity constraints and checks the authority of users accessing the data.
2. **Transaction manager**: It ensures that the database remains in a consistent state despite system failures, and concurrent executions proceed without conflicting.
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 is responsible for fetching data from disk storage into main memory and to decide what data to cache in main memory. It enables the database to handle data sizes that are much larger than the size of the main memory. The storage manager implements several data structures as part of physical system implementation.
    i. **Data files**: which store the database itself.
    ii. **Data dictionary**: It contains metadata that is data about data. The schema of a table is an example of metadata. A database system consults the data dictionary before reading and modifying actual data.
    iii. **Indices**: Which provide fast access to data items that hold particular values.

The Query Processor

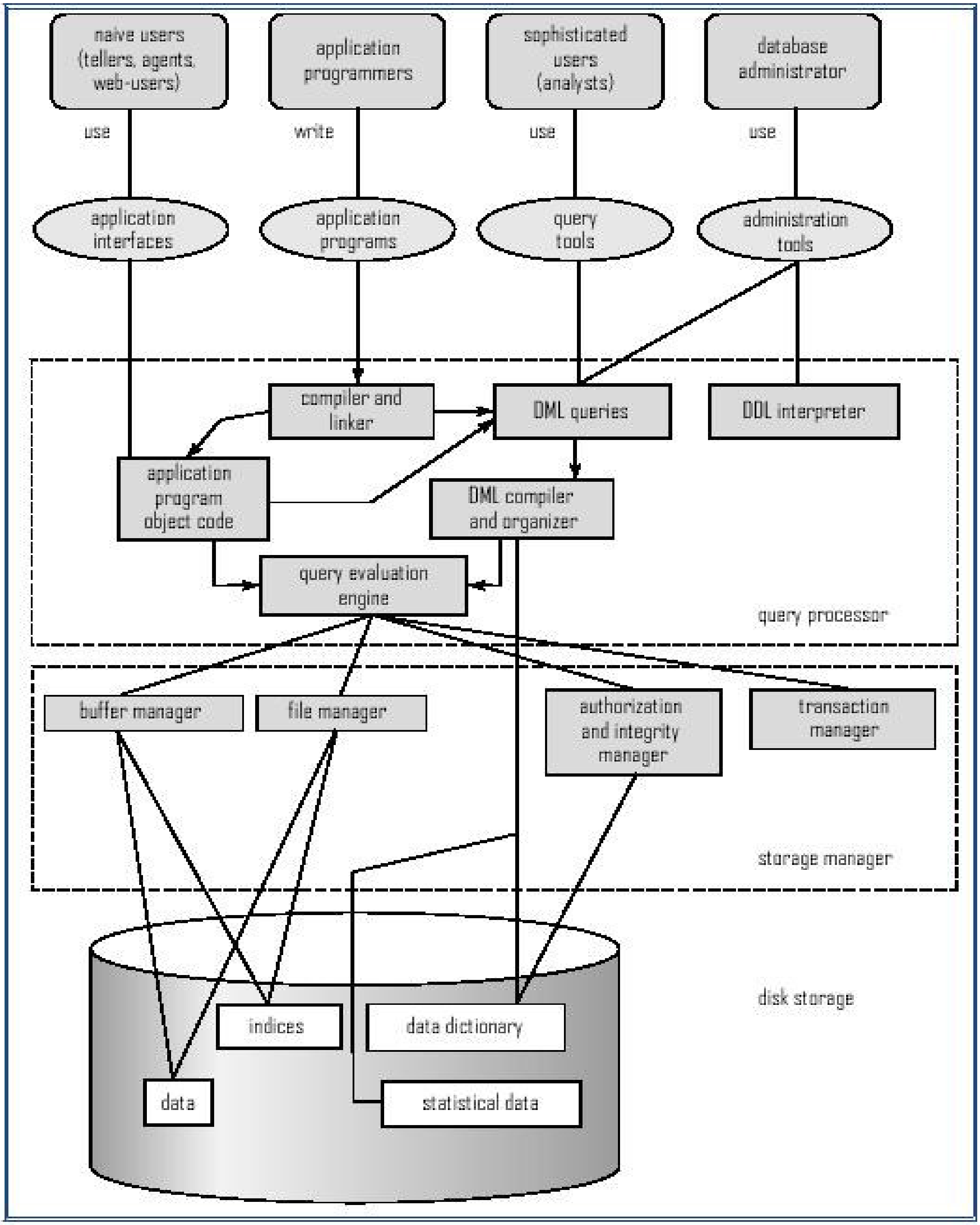
■ The query processor is an important part of the database system. It helps the database system to simplify and facilitate access to data. The query processor components include:

1. **DDL interpreter**, which interprets DDL statements and records the definitions in the data dictionary.
2. **DML compiler**, which translates DML statements in a query language into an evaluation plan consisting of low-level instructions that the query evaluation engine understands.

   A query can be translates into any number of evaluations plans that all give the same result.

   The DML compiler also performs query optimization, that is, it picks up the lowest cost evaluation plan from among the alternatives.
3. **Query evaluation engine**, which executes low-level instructions generated by the DML compiler.

Database System Structure:



. DATABASE USERS AND ADMINISTRATOR

People who work with a database can be categorized as:

- Database Users
- Database administrators

7.1. DATABASE USERS

There are four types of database users, differentiated by the way they interact with the system.

1. **Naive users**
   - ➢ Naive users interact with the system by invoking one of the application programs that have been written previously.
   - ➢ Naive users are typical users of form interface, where the user can fill in appropriate fields of the form.
   - ➢ Naive users may also simply read *reports* generated from the database.
2. **Application Programmers**
   - ➢ Application programmers are computer professionals who write application programs.
   - ➢ Rapid application development (RAD) tools enable the application programmer to construct forms and reports without writing a program.
   - ➢ Special types of programming languages that combine control structures with data manipulation language. These languages, sometimes called *fourth-generation languages.*
3. **Sophisticated users**
   - ➢ Sophisticated users interact with the system without writing programs. Instead, they form their requests in a database query language.
   - ➢ They submit each such query to a **query processor** that the storage manager understands.
   - ➢ **Online analytical processing (OLAP)** tools simplify analysis and **data mining** tools specify certain kinds of patterns in data.
4. **Specialized users**
   - ➢ Specialized users are sophisticated users who write specialized database applications that do not fit into the traditional data-processing framework.
   - ➢ The applications are computer-aided design systems, knowledge base and expert systems, systems that store data with complex data types

7.2. DATABASE ADMINISTRATORS

A person who has such central control over the system is called a **database administrator** (**DBA**).

The functions of a DBA include:

• **Schema definition**. The DBA creates the original database schema by executing a set of data definition statements in the DDL.

• **Storage structure and access-method definition**.

• **Schema and physical-organization modification**. The DBA carries out changes to the schema and physical organization to reflect the changing needs of the organization.

• **Granting of authorization for data access**. By granting different types of authorization, the database administrator can regulate which parts of the database various users can access.

Authorization information is kept in a special system structure that the database system consults whenever someone attempts to access the data in the system.

• **Routine maintenance**. Examples of the database administrator‘s routine maintenance activities are:

1. periodically backing up the database
2. Ensuring that enough free disk space
3. Monitoring jobs running on the database and ensuring that performance is not degraded by very expensive tasks submitted by some users.
4. Ensuring that performance is not degraded by very expensive tasks submitted by some users.

□. *ENTITY RE□ ATIONS□ IP MODE□ □ER MODE□□*

The E-R data model considers the real world consisting of a set of basic objects, called entities, and relationships among these objects.

■ The E-R data model employs three basic notions:

1. Entity sets
2. Relationship sets
3. Attributes

1. Entity Sets

■ An ***entity*** is _thing‘ or _object in the real world that is distinguishable from all other

■ objects. For example, each person is an entity.

■ An entity has a set of properties, and the values for some set of properties may uniquely identify an entity.

■ For example, a customer with customer-id property with value C101 uniquely identifies that person.

■ An entity may be concrete, such as person or a book, or it may be abstract, such as a loan, or a holiday.

■ An ***entity set*** is a set of entities of the same type that share the same properties, or attributes.

- For example all persons who are customers at a given bank can be defined as entity set *customer*.
- The properties that describe an entity are called **attributes**.

2. Relationships and Relationships sets

- **Relationship** is an association among several entities.
- **Relationship set** is a set of relationships of the same type.
- The association between entity set is referred to as **participation.** That is, the entity sets *E*1, *E*2, . . .,*En* **participate** in relationship set *R*.
- **Recursive relationship set:** Same entity set participating in a relationship more than once in a different role is called Recursive relationship set.
- The attributes of entities in Recursive relationship set is called **descriptive attributes**.

Types of relationships

i) Unary relationship: A unary relationship exists when an association is maintained within a single entity.

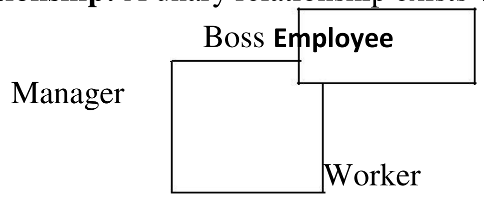
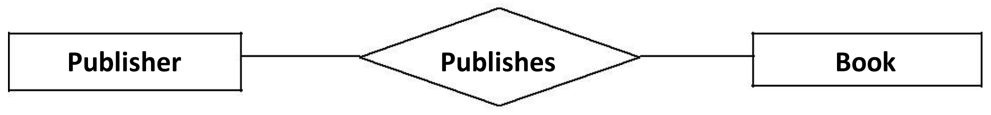
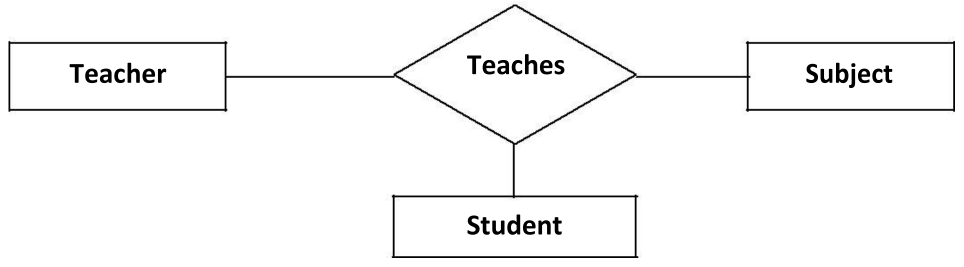


Figure → Association between two objects of the same entity set.

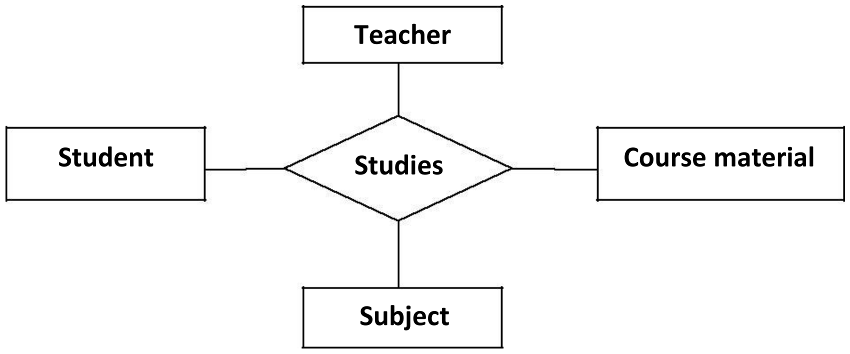
ii) Binary relationship: A binary relationship exists when two entities are associated.



iii) Ternary relationship: A ternary relationship exists when there are three entities associated.

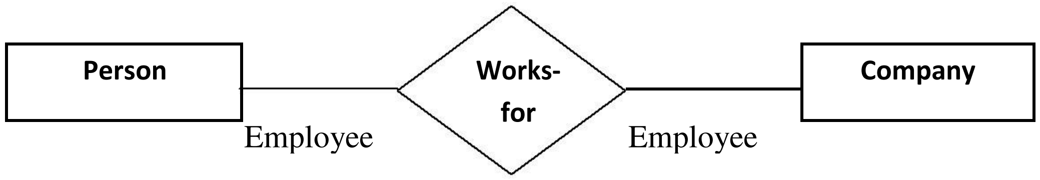


iv) Quaternary relationship: A quaternary relationship exists when there are four entities associated.



- The number of entity set participating in a relationship is called **degree of the relationship set.**
- Binary relationship set is of degree 2; a tertiary relationship set is of degree 3.

Entity role: The function that an entity plays in a relationship is called that entity's role. A role is one end of an association.



Here Entity role is Employee

3. Attributes

- The properties that describes an entity is called attributes.
- The attributes of customer entity set are customer_id, customer_name and city.
- Each attributes has a set of permitted values called the domain or value set.
- Each entity will have value for its attributes.
- Example:

Customer Name	John
Customer Id	321

Attributes are classified as

- Simple
- Composite
- Single- valued
- Multi-valued
- Derived

1) Simple attribute:

This type of attributes cannot be divided into sub parts.

Example: Age, sex, GPA

2) Composite attribute:

This type of attributes Can be subdivided.

Example: Address: street, city, state, zip

3) Single-valued attribute:

This type of attributes can have only a single value

Example: Social security number

4) Multi-valued attribute:

Multi-valued attribute Can have many values.

Example: Person may have several college degrees, phone numbers

5) Derived attribute:

Derived attribute Can be calculated or derived from other related attributes or entities.

Example: Age can be derived from D.O.B.

6) Stored attributes:

The attributes stored in a data base are called stored attributes.

- An attribute takes a null value when an entity does not have a value for it.
- Null values indicate the value for the particular attribute does not exists or unknown.

E.g. : 1. Middle name may not be present for a person (non existence case)

2. Apartment number may be missing or unknown.

CONSTRAINTS

An E-R enterprise schema may define certain constraints to which the contents of a database system must conform.

Three types of constraints are

1. Mapping cardinalities
2. Key constraints
3. Participation constraints

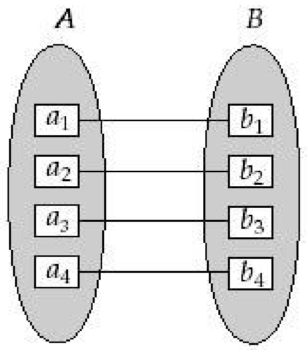
1. Mapping cardinalities

- Mapping cardinalities express the number of entities to which another entity can be associated via a relationship set.
- Cardinality in E-R diagram that is represented by two ways:

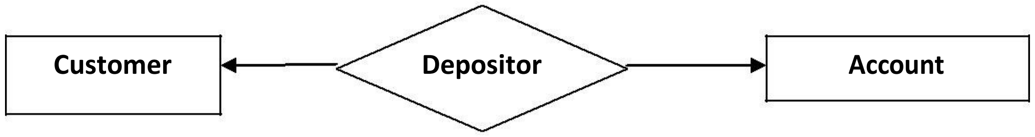
i) Directed line ($\longrightarrow$ ii) Undirected line ( )——

There are 4 categories of cardinality.

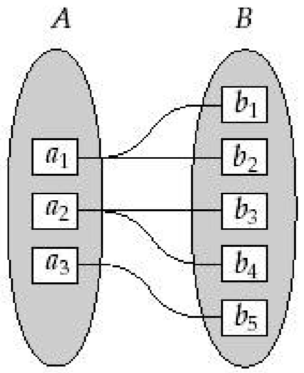
i) One to one: An entity in A is associated with at most one entity in B, and an entity in B is associated with at most one entity in A.



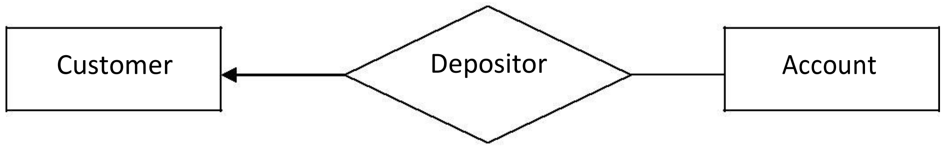
Example: A customer with single account at given branch is shown by one-to-one relationship as given below



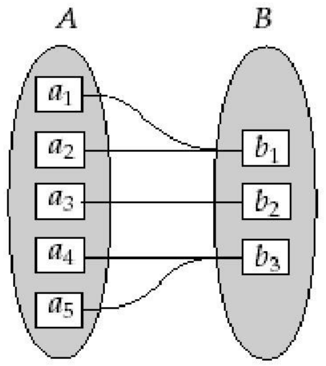
ii) One-to-many: An entity in A is associated with any number of entities (zero or more) in B. An entity in B, however, can be associated with at most one entity in A.



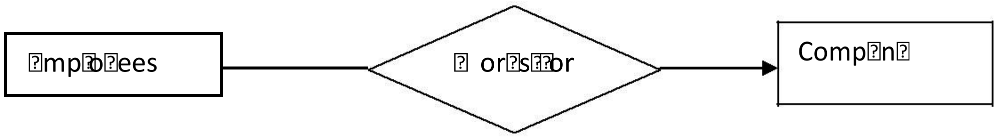
Example: A customer having two accounts at a given branch is shown by one-to-many relationship as given below.



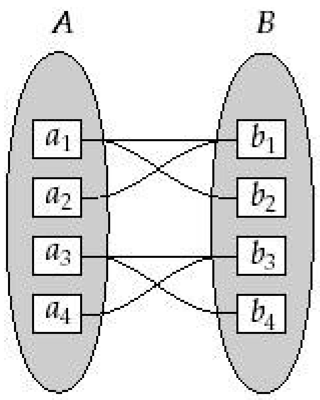
iii) Many-to-one: An entity in A is associated with at most one entity in B. An entity in B, however, can be associated with any number (zero or more) of entities in A.



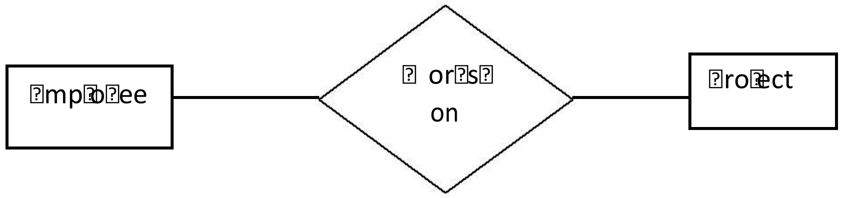
Example: Many employees works for a company. This relationship is shown by many-to-one as given below.



iv) Many-to-many: An entity in A is associated with any number (zero or more) of entities in B, and an entity in B is associated with any number (zero or more) of entities in A.



Example: Employee works on number of projects and project is handled by number of employees. Therefore, the relationship between employee and project is many-to-many as shown below.



2. Keys

- A key allows us to identify a set of attributes and thus distinguishes entities from each other.
- Keys also help uniquely identify relationships, and thus distinguish relationships from each other.

Key Type	Definition

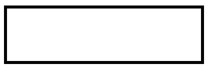
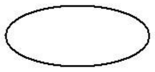
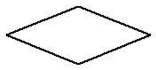
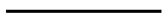
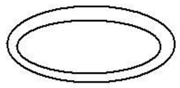
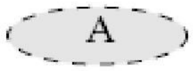
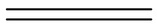
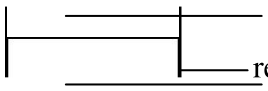
Superkey	Any attribute or combination of attributes that uniquely identifies a row in the table. **Example:** Roll_No attribute of the entity set _student‛ distinguishes one student entity from another. Customer_name, Customer_id together is a Super key
Candiate Key	Minimal Superkey. A superkey that does not contain a subset of attributes that is itself a superkey. **Example:** Student_name and Student_street,are sufficient to uniquely identify one particular student.
Primary Key	The candidate key selected to uniquely identify all rows. It should be rarely changed and cannot contain null values. **Example:** Roll_No is a primary set of _student‛ entity set.
Foreign Key	An attribute (or combination of attributes) in one table that must either match the primary key of another table or be null **Example:** Consider in the staff relation the branch_no attribute exists to match staff to the branch office they work in. In the staff relation, branch_no is foreign key.
Secondary Key	An attribute or combination of attributes used to make data retrieval more efficient.

3. Participation Constraint

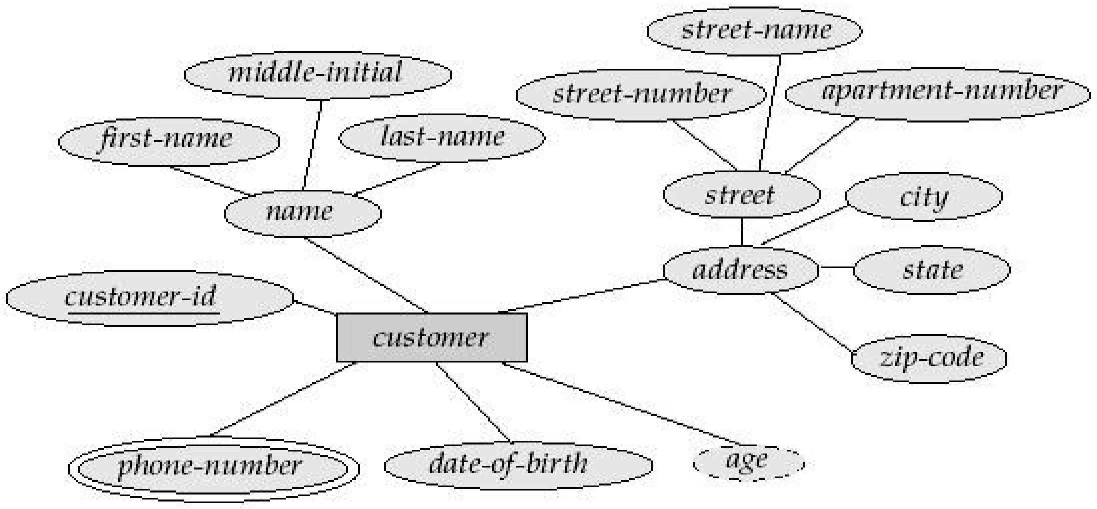
- Participation can be divided into two types**.**
  1. Total 2. Partial
- If every entity in the entity set E participates in at least one relationship in R. Then participation is called Total Participation
- If only some entities in the entity set E participate in relationships in R. Then the participation is called Partial Participation.

9. ENTITY-RELATIONSHIP(E-R) DIAGRAMS

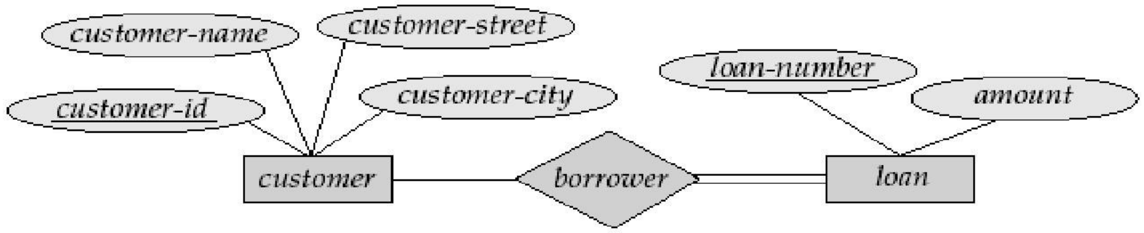
- **E-R diagram** can express the overall logical structure of a database graphically.
- **E-R diagram** consists of the following major components:

Component name	*Symbol*	*Description*
Rectangles		represent entity sets
Ellipses		represent attributes
Diamonds		represent relationship sets
Lines		link attributes to entity sets and entity sets to relationship sets
Double ellipses		represent multivalued attributes
Dashed ellipses		represent derived attributes
Double lines		Represent total participation of an entity in a relationship set
Double rectangles		represent weak entity sets

E-R diagram with composite, multivalued, and derived attributes.



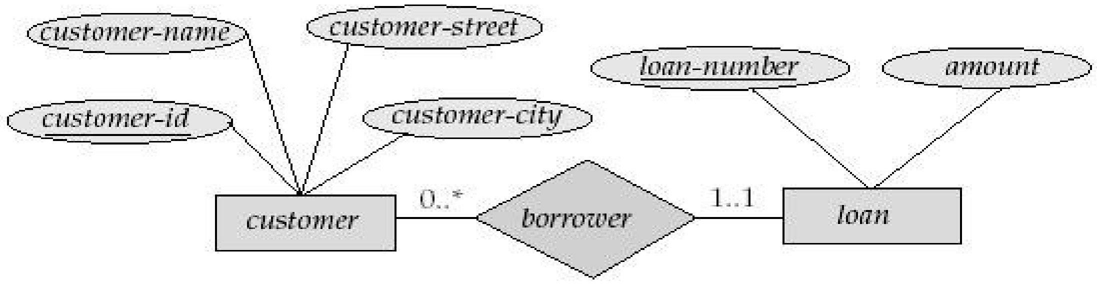
- **Double lines** are used in an E-R diagram to indicate that the participation of an entity set in a relationship set is total; that is, each entity in the entity set occurs in at least one relationship in that relationship set.



- The number of time an entity participates in a relationship can be specified using complex **cardinalities.**
- An edge between an entity set and binary relationship set can have an associated minimum and maximum cardinality assigned in the form of l..h.

  l **-** Minimum cardinality
  h **-** Maximum cardinality

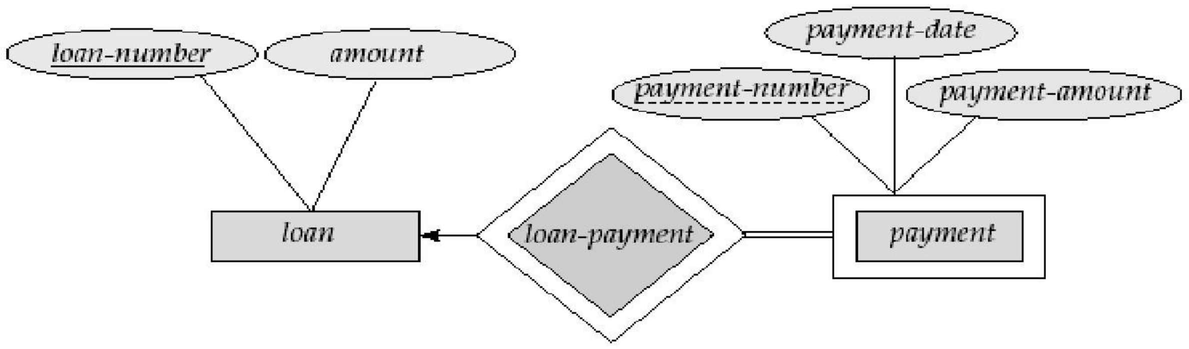
- A minimum value of 1 indicates total participation of the entity set in the relationship
- set. A maximum value of 1 indicates that the entity participates in at most one
- relationship. A maximum value * indicates no limit.
- A label 1... on an edge is equivalent to a double line.



0..* indicates a customer can have 0 or more loan
1..1 indicates a loan must have one associated customer

Strong and Weak entity sets

- An entity set may not have sufficient attributes to form a primary key. Such an entity set is termed a **weak entity set**.
- An entity set that has a primary key is termed a **strong entity set**.
- Weak entity set is associated with another entity set called the **identifying** or **owner entity set**. ie, weak entity set is said to be existence dependent on the identifying entity set.
- Identifying entity set is said to own the weak entity set.
- The relationship among the weak and identifying entity set is called the **identifying relationship**.



■

Discriminator in a weak entity set is a set of attributes that distinguishes the different entities among the weak entity also called as partial key.

Extended E-R Features

■ ER model that is supported with the additional semantic concepts is called the extended entity relationship model or EER model.

■ EER model deals with

1. Specialization
2. Generalization
3. Aggregation

1. Specialization:

■ The process of designating subgroupings within an entity set is called

■ **Specialization Specialization** is a top-down process.

■ Consider an entity set *person.* A person may be further classified as one of the following:

• *Customer*

• *Employee*

■ All person has a set of attributes in common with some additional

■ attributes. Specialization is depicted by a *triangle* component labeled **ISA**.

■ The label ISA stands for is a for example, that a customer is a person.

■ The ISA relationship may also be referred to as a **super class-subclass** relationship.

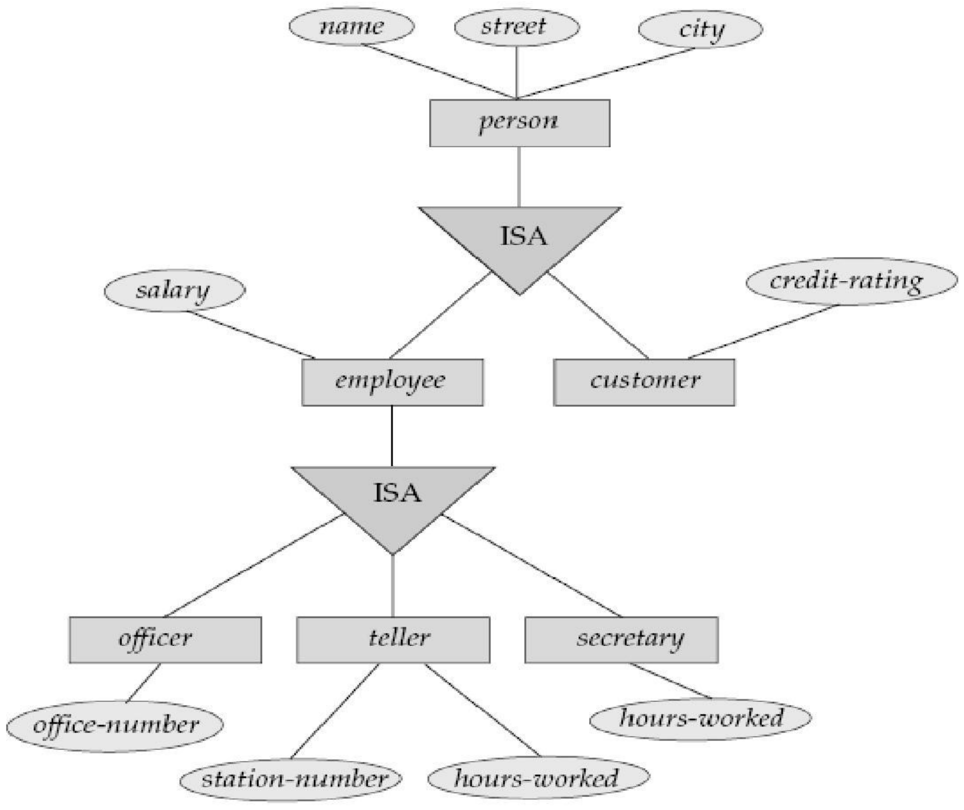
2. Generalization:

■ Generalization is a simple inversion of specialization.

■ Generalization is the process of defining a more general entity type from a set of more specialized entity types.

■ **Generalization** is a bottom-up approach.

■ Generalization results in the identification of a generalized super class from the original subclasses.



Specialization and generalization.

- *Person* is the higher-level entity set
- *Customer* and *employee* are lower-level entity sets.
- The *person* entity set is the superclass of the *customer* and *employee* subclasses.

Attribute Inheritance

- A property of the higher- and lower-level entities created by specialization and generalization is **attribute inheritance**.
- The attributes of the higher-level entity sets are said to be **inherited** by the lower-level entity sets.
- For example, *customer* and *employee* inherit the attributes of
- *person*. The outcome of attribute inheritance is
  1. A higher-level entity set with attributes and relationships that apply to all of its lower-level entity sets.
  2. Lower-level entity sets with distinctive features that apply only within a particular lower-level entity set
- If an entity set is involved as a lower-level entity set in only one ISA relationship, then the entity set has **single inheritance**

- If an entity set is involved as a lower-level entity set in more than one ISA relationship, then the entity set has **multiple inheritance** and the resulting structure is said to be a *lattice*.

Constraints on Generalizations

1. One type of constraint determining which entities can be members of a lower-level entity set. Such membership may be one of the following:

   • **Condition-defined**. In condition-defined the members of lower-level entity set is evaluated on the basis of whether or not an entity satisfies an explicit condition.

   • **User-defined**. User defined constraints are defined by user.

2. A second type of constraint relates to whether or not entities may belong to more than one lower-level entity set within a single generalization. The lower-level entity sets may be one of the following:

   • **Disjoint**. A *disjointness constraint* requires that an entity belong to no more than one lower-level entity set.

   • **Overlapping**. Same entity may belong to more than one lower-level entity set within a single generalization.

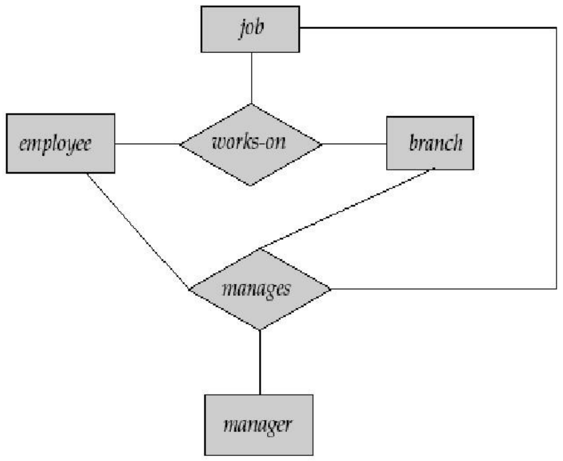
3. A final constraint, the **completeness constraint** specifies whether or not an entity in the higher-level entity set must belong to at least one of the lower-level entity sets .This constraint may be one of the following:

   • **Total generalization** or **specialization**. Each higher-level entity must belong to a lower-level entity set. It is represented by double line.

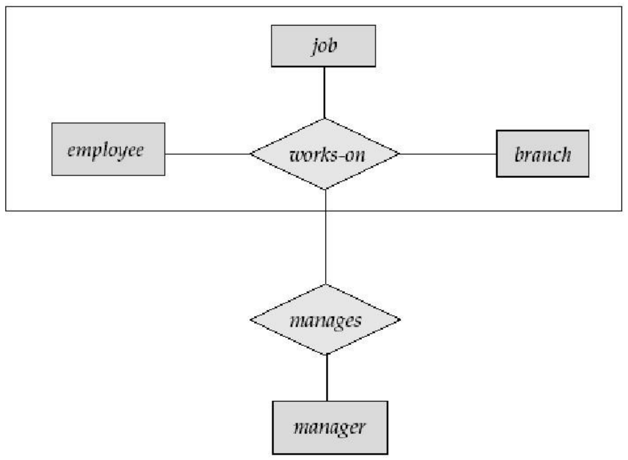
   • **Partial generalization** or **specialization**. Some higher-level entities may not belong to any lower-level entity set.

3. Aggregation

- One limitation of the E-R model is that it cannot express relationships among relationships.
- Consider the ternary relationship *works-on*, between a *employee*, *branch*, and *job*. Now, suppose we want to record managers for tasks performed by an employee at a branch. There another entity set *manager* is created.
- The best way to model such a situation is to use aggregation.
- **Aggregation** is an abstraction through which relationships are treated as higherlevel entities.
- In our example works-on act as high level entity.

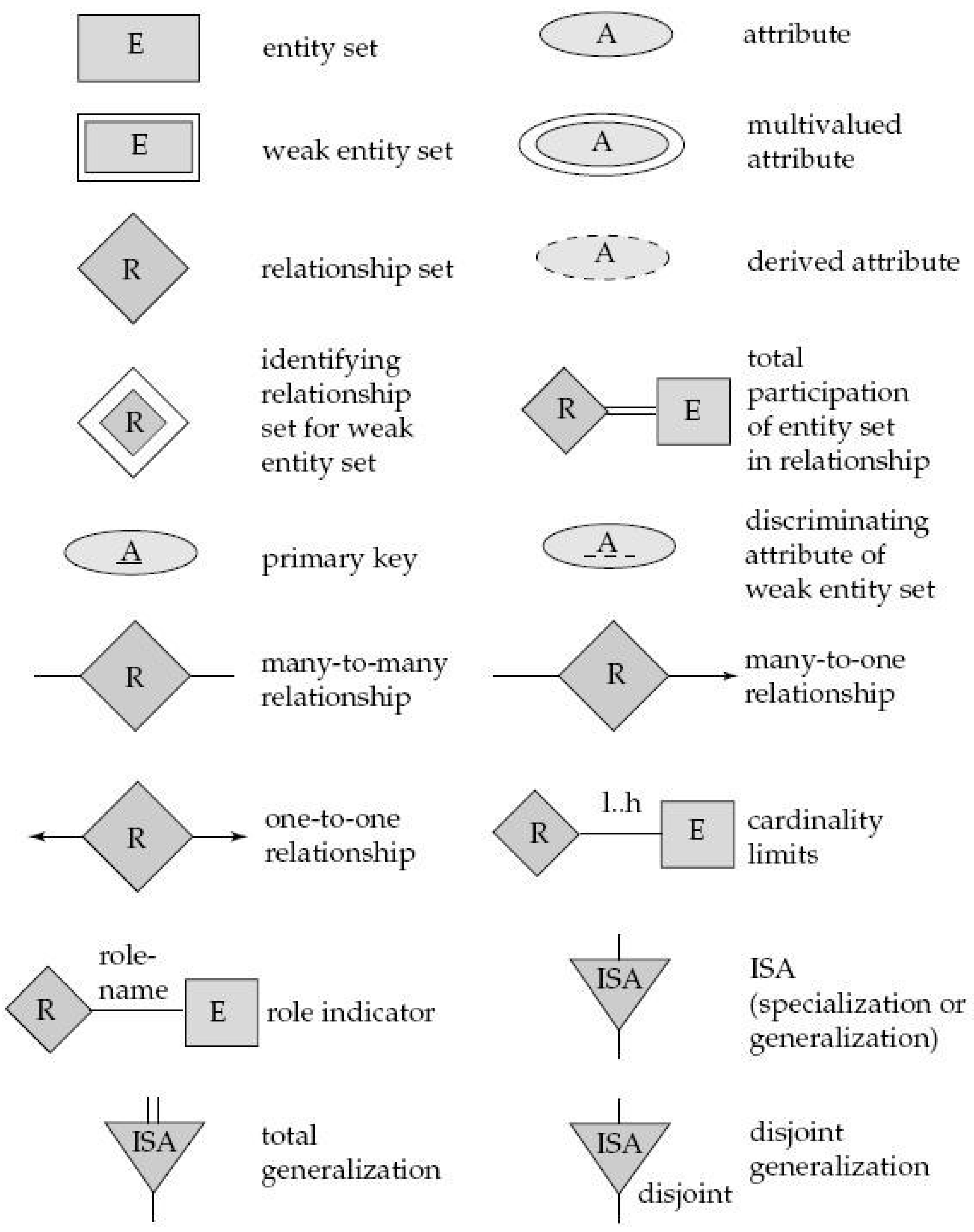


E-R diagram with redundant relationships.



E-R diagram with aggregation.

Summary of ER diagram



Symbols used in the E-R notation.

▯▯. *INTROD▯ ▯TION TO RELATIONALDATA▯ ASES*

- A relational database is based on the relational model and uses a collection of tables to represent both data and the relationship among those data.
- It includes DML and DDL languages.

Tables:

- Each table has multiple columns and each column has unique name.

Account number	Balance
A-101	500
A-215	700
A-102	400

- A relational model is an example of a record based model.
- Record based model are structured in fixed format record of several types.
- Each table contains record of particular type. Each record type defines a fixed number of fields or attributes.
- The columns of the table correspond to the attribute of record type.

Data Manipulation Language (DML)

DML includes following commands

1. INSERT – To insert one or more number of Rows.
2. SELECT – To display one or more rows.
3. UPDATE **–** Used to alter the column values in a table.
4. DELETE **–** Used to delete one or more rows.

Data Definition Language (DDL)

DDL includes following commands

1. CREATE – Command used for creating tables.
2. DESC – Command used to view the table structure.
3. ALTER – Command used for modifying table structure.
4. RENAME – used to change the name of the table.
5. DROP – Command used for removing an existing table.

2 Mark Questions

1. Define data, database, database management system, database system?
2. List any eight applications of DBMS.
3. What are the disadvantages of keeping organization information in a file processing system?
4. What are the advantages of using a DBMS (Centralized control of data)?
5. With the block diagram, discuss briefly the various levels of data abstraction?
6. Define instance and schema?
7. Define the terms 1) physical schema 2) logical schema 3)Subschema
8. What is conceptual schema?
9. Define data model?
10. What is storage manager?
11. What are the components of storage manager?
12. What is the purpose of storage manager?
13. List the data structures implemented by the storage manager.
14. What is a data dictionary?
15. What is an entity relationship model?
16. What are attributes? Give examples.
17. What are the types of Attributes?
18. What is relationship? Give examples
19. Define the terms
20. Define null values.
21. Define the terms
22. What is meant by the degree of relationship set?
23. Define the terms
24. Define weak and strong entity sets?
25. What does the cardinality ratio specify?
26. Explain the two types of participation constraint.
27. Define the terms
28. Write short notes on relational model
29. Define the term Domain.
30. Specify with suitable examples, the different types of keys used in database management systems.
31. Define Data model.

16 Mark Questions

1. Explain DBMS System Architecture.
2. Explain E-R Model in detail with suitable example.
3. Explain about various data models.

4 Draw an E – R Diagram for Banking, University, Company, Airlines, ATM, Hospital, Library, Super market, Insurance Company.

5. Explain in details about the various database languages.
6. Discuss about various operations in Relational Databases.
7. Discuss about database users and administrators.

UNIT II

RELATIONAL MODEL

SQL – Data definition- Queries in SQL- Updates- Views – Integrity and Security – Relational Database design-Relational Models-Design issues – Functional dependences and Normalization for Relational Databases (up to BCNF).

1. THE RELATIONAL MODEL

STRUCTURE OF RELATIONAL DATABASES:

- A relational database consists of a collection of **tables**, each of which is assigned a unique name.
- A row in a table represents a *relationship* among a set of values.

BASIC STRUCTURE

- Each column header is attributes. Each attribute allows a set of permitted values called domain of that attribute.
- A table of n-attributes must be a subset of

$$D1 \times D2 \times \cdot \quad \cdot \quad \cdot \times Dn.1 \times Dn$$

- A relation is a cartesian product of list of domains.

- Mathematically table is called as a **relation** and rows in a table are called as **tuples.**
- The tuples in a relation can be either **sorted or unsorted**.
- Several attributes can have **same domain**. **E.g.:** customer_name,
- employee_name. Attributes can also be **distinct. E.g.:** balance, branch_name
- Attributes can have null values incase if the value is **unknown or does not exist.**
- Database schema begins with upper case and database relation begins with lower

  case. Account-schema = (account-number, branch-name, balance)

  account (Account-schema)

account-number	*branch-name*	*balance*
A-101	Downtown	500
A-215	Mianus	700
A-102	Perryridge	400
A-305	Round Hill	350
A-201	Brighton	900
A-222	Redwood	700
A-217	Brighton	750

Account Table

□. THE □ATALO□

- The catalog is a place where all the schemas and the corresponding mappings are kept.
- The catalog contains detailed information also called as descriptor information or meta data.
- Descriptor information is essential for the system to perform its job properly.
- For example the authorization subsystem uses catalog information about users and security constraints to grant or deny access to a particular user.
- The catalog should be self describing.

□. RELATIONAL AL□ E□ RA

- The relational algebra is a *procedural* query language.
- It consists of a set of operations that take one or two relations as input and produce a new relation as their result.

Formal Definition

A basic expression in the relational algebra consists of either one of the following:

- A relation in the database
- A constant relation

Let E_1 and E_2 be relational-algebra expressions; the following are all relational-algebra

exp■sions: E_1 E_2

$E_1 - E_2$

$E_1 \times E_2$

■(E$_1$), P is a predicate on attributes in E$_1$

■(E$_1$), S is a list consisting of some of the attributes in E$_1$

■ (E$_1$), x is the new name for the result of E$_1$

Operations can be divided into

- **Basic operations or Fundamental Operations -**Select, Project, Union, rename, set difference & Cartesian product.
- **Additional operations** that can be expressed in terms of basic operations-Set intersection, Natural join Division and Assignment.
- **Extended operations-**Generalized projection, Aggregate operations and Outer join.

3.1. □A□J□ O□ERATION□ OR □□ NDAMENTAL O□ERATION□

- The select, project, and rename operations are called ***unary* operations**, because they operate on one relation.
- The other three operations (union, set difference, cartesian product) operate on pairs of relations and are, therefore, called ***binary* operations**.
- The basic or fundamental operations are as follows
  1. Select
  2. Project
  3. Union
  4. Rename
  5. Set difference
  6. Cartesian product.

1. Select Operation (σ)

Te select operation selects tuples that satisfy a given predicate.

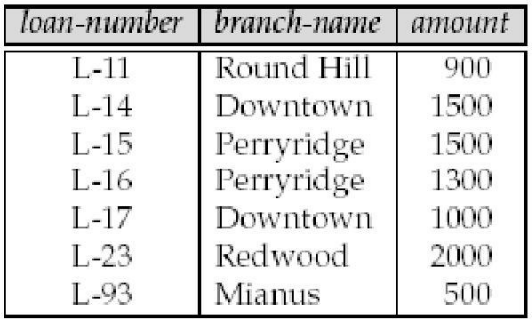
Syntax

$$\boxed{\sigma_{\text{<select condition>}}(R)}$$

- Symbol σ is used to denote the select operator.
- Predicate appears as a subscript to σ and argument relation in paranthesis.

Example

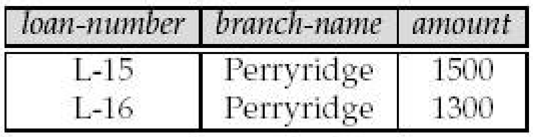
Consider the loan relation



loan-number	branch-name	amount
L-11	Round Hill	900
L-14	Downtown	1500
L-15	Perryridge	1500
L-16	Perryridge	1300
L-17	Downtown	1000
L-23	Redwood	2000
L-93	Mianus	500

Query: $\sigma_{\text{branch-name =—Perryridge□}}$ (loan)

Output relation is



loan-number	branch-name	amount
L-15	Perryridge	1500
L-16	Perryridge	1300

- Select operation allows all comparisons using =, _=, <, ., >,.
- It allows combination of server predicates using connectives like and (∧), or (∨), and not (¬).

E.g.: 1. σ amount>1200 (loan)
2. $\sigma_{\text{branch-name =—Perryridge□ ∧amount>1200}}$ (loan)

Other Examples

Consider following Book relation.

Book_Id	Title	Author	Publisher	Year	Price
B001	DBMS	Korth	McGraw_Hill	2000	250
B002	Compiler	Ulman		2004	350
B003	OOMD	Rambaugh		2003	450
B004	PPL	Sabista		2000	500

Following are the some examples of the select operation.

Example 1: Display books published in the 2000.

Query 1: $\sigma_{year=2000}$(Book)

The output of query 1 is shown below.

Book_Id	Title	Author	Publisher	Year	Price
B001	DBMS	Korth	McGraw_Hill	2000	250
B004	PPL	Sabista		2000	500

Example 2: Display all books having price greater than 300.

Query 2: $\sigma_{price>300}$(Book)

The output of query 2 is shown below.

Book_Id	Title	Author	Publisher	Year	Price
B002	Compiler	Ulman		2004	350
B003	OOMD	Rambaugh		2003	450
B004	PPL	Sabista		2000	500

Example 3: Select the tuples for all books whose publishing year is 2000 or price is greater than 300.

Query 3: $\sigma_{(year=2000)\ OR\ (price>300)}(Book)$

The output of query 3 is shown below.

Book_Id	Title	Author	Publisher	Year	Price
B001	DBMS	Korth	McGraw_Hill	2000	250
B002	Compiler	Ulman		2004	350
B003	OOMD	Rambaugh		2003	450
B004	PPL	Sabista		2000	500

Example 4: Select the tuples for all books whose publishing year is 2000 and price is greater than 300.

Query 3: $\sigma_{(year=2000)\ AND\ (price>300)}(Book)$

The output of query 4 is shown below.

Book_Id	Title	Author	Publisher	Year	Price
B004	PPL	Sabista		2000	500

2. Project operation (Π)

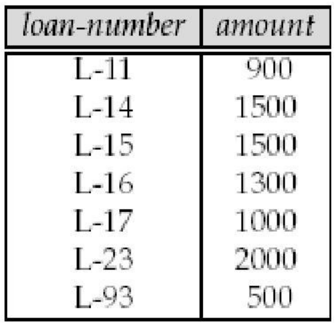
The project operation selects certain columns from a table while discarding others. It removes any duplicate tuples from the result relation.

Syntax

$$\boxed{\Pi_{<attributelist>} \ (R)}$$

- The symbol □ (pi) is used to denote the project operation
- Attribute list to be projected is specified as subscript of □ and R denotes the relation.

E.g.: $□_{loan\text{-}number,\ amount}\ (loan)$



loan-number	*amount*
L-11	900
L-14	1500
L-15	1500
L-16	1300
L-17	1000
L-23	2000
L-93	500

Example: The following are the examples of project operation on Book relation.

Example 1: Display all titles with author name.

Query 1: □$_{\text{Title, Author}}$ (Book)

The output of query 1 is shown below.

Title	**Author**
DBMS	Korth
Compiler	Ulman
OOMD	Rambaugh
PPL	Sabista

Example 2: Display all book titles with authors and price.

Query 2: □$_{\text{Title, Author, Price}}$ (Book)

The output of query 2 is shown below.

Title	**Author**	**Price**
DBMS	Korth	250
Compiler	Ulman	350
OOMD	Rambaugh	450
PPL	Sabista	500

Composition of select and project operations

The relational operations select and project can be combined to form a complicated query.

□$_{\text{customer-name}}$ ($\sigma_{\text{customer-city =□ Harrison□}}$ (customer))

Input Table: customer

customer-name	*customer-street*	*customer-city*
Adams	Spring	Pittsfield
Brooks	Senator	Brooklyn
Curry	North	Rye
Glenn	Sand Hill	Woodside
Green	Walnut	Stamford
Hayes	Main	Harrison
Johnson	Alma	Palo Alto
Jones	Main	Harrison
Lindsay	Park	Pittsfield
Smith	North	Rye
Turner	Putnam	Stamford
Williams	Nassau	Princeton

DATA BASE MANAGEMENT SYSTEMS

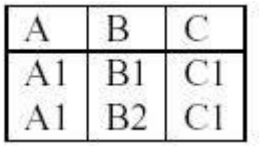
QUESTION BANK

UNIT III

PART A

1. With an example explain a Functional Dependency.

Let *R* be a relation schema, a subset of .*R*.þ *is also subset of R*. The functional dependency a->þholds on *R* if and only if for any legal relations *r*(R), whenever any two tuples *t*1 and *t*2 of *r* agree on the attributes a, they also agree on the attributes þ . That is, *t*1[a] = *t*2 [a] =>*t*1[þ.] = *t*2 [þ.] eg.



A	B	C
A1	B1	C1
A1	B2	C1

Here A->C.

2. Define Normalization?

Normalization is to generate a set of relation schemas that allows to store information with out redundancy and to retrieve information easily

3. Justify the need for normalization.

A bad relational database have the following pit falls: inability to represent some data and redundancy. To overcome this pit falls, normalization is to be done. Good database design should be Lossless join and Dependency preservation. Decomposition of databases are done to make the database consistent and efficient.

4. Why it is necessary to decompose a relation?

A relation is to be decomposed in order to avoid repetition of information and inability to represent certain information.

5. What is decomposition and how does it address redundancy?

Decomposition is reducing a large database to set of smaller database. When splitting database, some of the fields will be repeated in databases to maintain association with records.

6. **Give the comparison between BCNF and 3NF.**

3NF can be achieved without sacrificing losslessness or dependency preservation. Disadvantage: Use null values to represent some of the meaningful relation and repetition of information. If it is difficult to get a dependency-preserving BCNF algorithm, it is preferable to opt for BCNF and use techniques such as materialized views.

7. **Explain trivial dependency?**

Functional dependency of the form a->þ is trivial if þ is subset of a. Trivial functional dependencies are satisfied by all the relations.

8. **Define canonical cover?**

A canonical cover Fc for F is a set of dependencies such that F logically implies all dependencies in FC and Fc logically implies all dependencies in F. Fc must have the following properties.

9. **List the properties of canonical cover.**
Fc must have the following properties.

_ No functional dependency in Fc contains an extraneous attribute. _ Each
left side of a functional dependency in Fc is unique.

10. Explain the desirable properties of decomposition.

_ Lossless-join decomposition _
Dependency preservation
_ Repetition of information

11. **What is 2NF?**

A relation schema R is in 2NF if it is in 1NF and every non-prime attribute A in R is fully functionally dependent on primary key.

12. **Define Boyce codd normal form**

A relation schema R is in BCNF with respect to a set F of functional
dependencies if, for all functional dependencies in F+ of the form a->þ.

13. **What is first normal form?**
The domain of attribute must include only atomic (simple, indivisible) values.

14. **What is meant by computing the closure of a set of functional dependency?**

The closure of F denoted by F+ is the set of functional dependencies logically implied by F.

15. **What are axioms?**

Axioms or rules of inference provide a simpler technique for reasoning about functional dependencies.

16. What are the uses of functional dependencies?

_ To test relations to see whether they are legal under a given set of functional dependencies.

_ To specify constraints on the set of legal relations.

17. What is meant by normalization of data?

It is a process of analyzing the given relation schemas based on their Functional Dependencies (FDs) and primary key to achieve the properties

_ Minimizing redundancy
_ Minimizing insertion, deletion and updating anomalies.

18. Why certain Functional dependencies are called trivial dependency?.
The functional dependency a->þis trivial if þ*C* a

19. What is multi valued dependency?

If A->B then, two tuples with same A value have different B value. Eg.
Customer name -> customer street, customer city.

20. What is fourth normal form?.

A relation schema *R* is in 4NF with respect to a set *D* of functional and multivalued dependencies if for all multivalued dependencies in *D*+ of the form

□.□□.□, where □.□ *R* and □.□ *R,* at least one of the following hold: 1.□□□□
is trivial (i.e., □.□.□ or □.□.□ = *R)*

2.□ is a superkey for schema *R* If a
relation is in 4NF it is in BCNF

21. What are the closure set of functional dependency?.

-Given a set *F* of functional dependencies, there are certain other functional dependencies that are logically implied by *F*.

25. For example: If *A* □ *B* and *B* □ *C*, then we can infer that *A* □ *C*

-The set of all functional dependencies logically implied by *F* is the *closure* of *F*. -We denote the *closure* of *F* by F+.

-F+ is a superset of *F*.

22. What are the goals of normalization?.

Let *R* be a relation scheme with a set *F* of functional dependencies. Decide whether a relation scheme *R* is in "good" form.

In the case that a relation scheme *R* is not in "good" form, decompose it into a set of relation scheme {*R*1, *R*2, ..., *Rn*} such that

27. each relation scheme is in good form
28. the decomposition is a lossless-join decomposition
29. Preferably, the decomposition should be dependency preserving.

23. How would you find the extraneous attribute?.

Consider a set *F* of functional dependencies and the functional dependency □.□.□ in

F.

- Attribute A is extraneous in □ if *A* □.□

 and *F* logically implies (*F* – {□.□.□}) □ {(□ – *A*) □.□}. -
Attribute *A* is extraneous in □ if *A* □.□
 and the set of functional dependencies

 (*F* – {□.□.□}) □ {□.□(□ – *A*)} logically implies *F*.

24. What do you mean by lossless join decomposition?.

A decomposition of *R* into *R*1 and *R*2 is lossless join if and only if at least one of the following dependencies is in *F*+:

o *R*1 □ *R*2 □ *R*1 o *R*1 □
*R*2 □ *R*2

25. Give the uses of Multivalued dependency?.

We use multivalued dependencies in two ways:

- To test relations to determine whether they are legal under a given set of functional and multivalued dependencies

- To specify constraints on the set of legal relations. We shall thus concern ourselves *only* with relations that satisfy a given set of functional and multivalued dependencies.

PART B

1. Explain the following with relevant examples.
   i. B-Tree ii. B+Tree iii.Static and dynamic hashing.

2. Write a relevant example discuss the steps involved in processing a query.

3. Construct a B+ tree to insert the following key elements (order of the tree is 3)
   5, 3, 4, 9, 7, 15, 14,21,22, 23.

4. Describe in detail about how records are represented in a file and how to
   Organize them in a file.

5. Write about the file organization technique and type.

6. Explain the following
   i) B+- tree
   ii) Dynamic hashing & static hashing

7. Mention the purpose of indexing. How this can be done by B+ tree? Explain.

8. Explain following with relevant examples:
   (i) B tree
   (ii) B+Tree
   (iii) Static and dynamic hashing

9. With a relevant example discuss the steps involved in processing a query.

DATA BASE MANAGEMENT SYSTEMS

QUESTION BANK

UNIT IV

PART A

1. State the atomicity property of a transaction.

Atomicity: Either all operations of the transaction are properly reflected in the database or none.

Correctness: Execution of a transaction in isolation preserves the consistency **Isolation.**: Each transaction must be unaware of other concurrently executing transactions. **Durability:** After a transaction completes successfully, the changes it has made to the database persist, even if there are system failures.

2. Define deadlock.

A transaction waits for another transaction to release lock on data item. Mean while, transaction also waits for some other data item to be released by first transaction.

3. When are two schedules conflict equivalent?

If a schedule *S* can be transformed into a schedule *S´* by a series of swaps of nonconflicting instructions, we say that *S* and *S´* are conflict equivalent.

4. State the benefits of strict two-phase locking.

Cascading rollbacks can be avoided by a modification of two-phase locking. Any data written by an uncommitted transaction are locked in exclusive mode until the transaction commits, preventing any other transactions from reading the data.

5. What benefit is provided by strict-two-phase locking? What are the disadvantages results?

The transactions can be serialized in the order of their lock points**.** In strict two-phase locking, transaction must hold all its exclusive locks till it commits/aborts.

6. What is concurrency control ?
Concurrency control is maintaining consistency

7. What is transaction?

Collections of operations that form a single logical unit of work are called transactions.

8. What are the two statements regarding transaction?

The two statements regarding transaction of the form: _
Begin transaction
End transaction

9. When is a transaction rolled back?

Any changes that the aborted transaction made to the database must be undone. Once the changes caused by an aborted transaction have been undone, then the transaction has been rolled back.

10. What are the states of transaction?
The states of transaction are

- Active
- Partially committed
- Failed
- Aborted
- Commited
- Terminated

11. What is a shadow copy scheme?

It is simple, but efficient, scheme called the shadow copy schemes. It is based on making copies of the database called shadow copies that one transaction is active at a time. The scheme also assumes that the database is simply a file on disk.

12. Give the reasons for allowing concurrency?

The reasons for allowing concurrency is if the transactions run serially, a short transaction may have to wait for a preceding long transaction to complete, which can lead to unpredictable delays in running a transaction.
So concurrent execution reduces the unpredictable delays in running transactions.

13. What is average response time?

The average response time is that the average time for a transaction to be completed after it has been submitted.

14. What are the two types of serializability?

The two types of serializability is _

Conflict serializability

_ View serializability

15. Define lock?

Lock is the most common used to implement the requirement is to allow a transaction to access a data item only if it is currently holding a lock on that item.

16. What are the different modes of lock?

The modes of lock are: _
Shared

_ Exclusive

17. Differentiate strict two phase locking protocol and rigorous two phase locking protocol.

In **strict two phase locking protocol** all exclusive mode locks taken by a transaction is held until that transaction commits.

Rigorous two phase locking protocol requires that all locks be held until the transaction commits.

18. How the time stamps are implemented

- Use the value of the system clock as the time stamp. That is a transaction's time stamp is equal to the value of the clock when the transaction enters the system.

-Use a logical counter that is incremented after a new timestamp has been assigned; that is the time stamp is equal to the value of the counter.

19. What are the time stamps associated with each data item?

-W-timestamp (Q) denotes the largest time stamp if any transaction that executed WRITE (Q) successfully.

- R-timestamp (Q) denotes the largest time stamp if any transaction that executed READ (Q) successfully.

20. What is recovery management component?

Ensuring durability is the responsibility of a software component of the base system called the recovery management component.

21. Define the phases of two phase locking protocol
_ Growing phase: a transaction may obtain locks but not release any lock.

_ Shrinking phase: a transaction may release locks but may not obtain any new locks.

22. What is meant by log-based recovery?
The most widely used structures for recording database modifications is the log.

The log is a sequence of log records, recording all the update activities in the database. There are several types of log records.

23. What are the two methods for dealing deadlock problem?

The two methods for dealing deadlock problem is deadlock detection and
deadlock recovery.

24. What are the two types of errors?

The two types of errors are: _
Logical error

_ System error

25. What are the storage types?

The storage types are: _
Volatile storage
_ Nonvolatile storage

PART B
1. Explain testing for serializability with respect to concurrency control schemes.How will you determine, whether a schedule is serialiazble or not.

2. Explain the following protocols for concurrency control.

    i.Lock based protocols

    ii.Time stamp based protocols.

3. Discuss in detail about Transaction Recovery, System Recovery and Media Recovery

4.Write down in detail about Deadlock and Serializability.

5. Explain about

    i) Concurrent execution

    ii)Recoverability

6. Explain various recovery techniques during transaction

7.(i) Explain about immediate update and deferred update recovery techniques.

(ii) Explain the concepts of serializability.

8.(i) Explain Two-phase locking protocol.

(ii) Describe about the deadlock prevention schemes

9.(i)How can you implement atomicity in transactions? Explain.

(ii)Describe the concept of serilalizability with suitable example.

10. How concurrency is performed? Explain the protocol that is used to maintain the concurrency concept.

11. Explain testing for serializability with respect to concurrency control schemes. How will you determine, whether a schedule is serializable or not.

12. Explain the following concurrency control:

(i) Lock based protocol

(ii) Time stamp based protocol

13.Explain briefly about the Deadlock.

14. Explain in details about Database Recovery techniques

15. What is Transaction Processing? Explain in detail about concept, properties and State?

16. Explain ACID in detail.

17.Explain serializability.

18.Explain in details about Log-Based Recovery.

DATA BASE MANAGEMENT SYSTEMS

QUESTION BANK

UNIT V

PART A

1) What are ordered Indices?

In an **ordered index,** index entries are stored sorted on the search key value. E.g., author catalog in library.

Two types: Sparse Index and Dense index.

2) Distinguish between sparse index and dense index.

An index record appears for every search key value in the file is dense index. An index record appears for only some of the search key values is sparse index. Each index record contains a search key value and a pointer to the first data record with that search value.

3. What is a heap file? How are pages organized in a heap file?

Heap file:- Any record can be placed anywhere in the file where there is space for the record. There is no ordering of records.

4. How does a B-tree differ from B+ - tree? Why is a B+ - tree usually preferred as an access structure to a data file?
B-tree allows search-key values to appear only once; eliminates redundant

storage of search keys. Search keys in nonleaf nodes appear nowhere else in the B-tree; All paths from root to leaf are of the same length. Each node that is not a root or a

leaf has between □□̃.and *n* children. A leaf node has between (*n*-1)/2.and *n*-1 values.

5. Define Static hashing. Static Hashing:

Hash function *h* is a function from the set of all search-key values *K* to the set of all bucket addresses *B*. Hash function is used to locate records for access, insertion as well as deletion. In static hashing, function *h* maps search-key values to a fixed set of *B* of bucket addresses.

6. Define Dynamic hashing.
Dynamic Hashing: Good for database that grows and shrinks in size, allows the hash function to be modified dynamically. The number of buckets also changes dynamically due to

coalescing and splitting of buckets.

7. How does a DBMS represent a relational query plan?

Parsing and translation: Translate the query into its internal form. This is then translated into relational algebra. Parser checks syntax, verifies relations **Evaluation**: The query-execution engine takes a query-evaluation plan, executes that plan, and returns the answers to the query.

8. Define RAID:
Redundant Arrays of Independent Disks.

9. Compare sequential access devices versus random access devices with an example.

Records are stored in sequential order according to the value of a search key. If n th record has to be accessed then all n-1 records has to be passed eg. Tape. In random access device, n th record can be pointed directly eg. Disk.

10. What can be done to reduce the occurrences of bucket overflows in a hash file organization?

The bucket overflows can be reduced by using overflow buckets. Overflow handling can be done using linked list. This is called overflow chaining.

11. Give the measures of quality of a disk.
_ Capacity

_ Access time _ Seek
time

_ Data transfer rate _
Reliability

_ Rotational latency time.

12. Compare sequential access devices versus random access devices with an example sequential access devices random access devices

Must be accessed from the beginning It is possible to read data from any location Eg:-
tape storage Eg:-disk storage

Access to data is much slower Access to data is faster Cheaper
than disk Expensive when compared with disk

13. What are the types of storage devices?
_ Primary storage

_ Secondary storage _

Tertiary storage

14. Define access time.

Access time is the time from when a read or write request is issued to when data transfer begins.

15. Define seek time.

The time for repositioning the arm is called the seek time and it increases with the distance that the arm is called the seek time.

16. Define average seek time.

The average seek time is the average of the seek times, measured over a sequence of random requests.

17. Define rotational latency time.

The time spent waiting for the sector to be accessed to appear under the head is called the rotational latency time.

18. Define average latency time.

The average latency time of the disk is one-half the time for a full rotation of the disk.

19. What is meant by data-transfer rate?

The data-transfer rate is the rate at which data can be retrieved from or stored to the disk.

20. What is meant by mean time to failure?

The mean time to failure is the amount of time that the system could run continuously without failure.

21. What are a block and a block number?
A block is a contiguous sequence of sectors from a single track of one platter.

Each request specifies the address on the disk to be referenced. That address is in the form of a block number.

22. What is the use of RAID?

A variety of disk-organization techniques, collectively called redundant arrays of independent disks are used to improve the performance and reliability.

23. Explain how reliability can be improved through redundancy?

The simplest approach to introducing redundancy is to duplicate every disk. This technique is called mirroring or shadowing. A logical disk then consists of two physical disks, and write is carried out on both the disk. If one of the disks fails the data can be read from the other. Data will be lost if the second disk fails before the first fail ed disk is repaired.

24. What is called bit-level striping?

Data striping consists of splitting the bits of each byte across multiple disks. This is called bit-level striping.

25. What is called block-level striping?

Block level striping stripes blocks across multiple disks. It treats the array of disks as a large disk, and gives blocks logical numbers.

26. Distinguish between fixed length records and variable length records? Fixed length records
Every record has the same fields and field lengths are fixed.

Variable length records
File records are of same type but one or more of the fields are of varying size.

27. What are the ways in which the variable-length records arise in database systems?
_ Storage of multiple record types in a file.

_ Record types that allow variable lengths for one or more fields. _
Record types that allow repeating fields.

28. Explain the use of variable length records.
_ They are used for Storing of multiple record types in a file.

_ Used for storing records that has varying lengths for one or more fields. _ Used for storing records that allow repeating fields

29. What is known as heap file organization?

In the heap file organization, any record can be placed anywhere in the file where there is space for the record. There is no ordering of records. There is a single file for each relation.

30. What is known as sequential file organization?

In the sequential file organization, the records are stored in sequential order, according to the value of a “search key” of each record.

31. What is hashing file organization?
In the hashing file organization, a hash function is computed on some attribute of

each record. The result of the hash function specifies in which block of the file the record should be placed.

32. What is known as clustering file organization?

In the clustering file organization, records of several different relations are stored in the same file.

33. What is an index?

An index is a structure that helps to locate desired records of a relation quickly, without examining all records.

34. What are the two types of ordered indices?
_ Primary index
_ Secondary index

35. What are the types of indices?

_ Ordered indices _ Hash
indices

36. What are the techniques to be evaluated for both ordered indexing and hashing?

_ Access types _
Access time _ Insertion
time _ Deletion time

_ Space overhead

37. What is linear probing?

Linear probing is a type of open hashing. If a bucket is full the system inserts records in to the next bucket that has space. This is known as linear probing.

38. What is called query processing?

Query processing refers to the range of activities involved in extracting data from a database.

39. What are the steps involved in query processing?
The basic steps are:

_ parsing and translation _
optimization_ evaluation
40. What is called an evaluation primitive?

A relational algebra operation annotated with instructions on how to evaluate is called an evaluation primitive.

41. What is called a query evaluation plan?

A sequence of primitive operations that can be used to evaluate ba query is a query evaluation plan or a query execution plan.

42. What is called a query –execution engine?

The query execution engine takes a query evaluation plan, executes that plan, and returns the answers to the query.

43. How do you measure the cost of query evaluation?

The cost of a query evaluation is measured in terms of a number of different resources including disk accesses, CPU time to execute a query, and in a distributed database system the cost of communication

44. List out the operations involved in query processing

Selection operation Join
operations. Sorting.

Projection Set
operations
Aggregation

45. Define query optimization.

Query optimization refers to the process of finding the lowest –cost method of evaluating a given query.

46. Which level of RAID is best? Why?
RAID level 1 is the RAID level of choice for many applications with moderate

storage requirements and high I/O requirements. RAID 1 follows mirroring and provides best write performance.

PART B

1. State and explain the features of object oriented data model. Use banking application as an example.
2. Write detailed note on the following:
   i. Distributed data base.
   ii. Data mining.
3. Write detailed on the following
   i) Distributed database

ii)Data mining

4. Discuss in detail about the object relational database and its advantage
5. State and explain the object oriented data model. Use banking application as an example. (16)

6. Write detail notes on following:
   (i) Distributed Databases
   (ii) Data Mining
7. Explain briefly about Inheritance and Multiple Inheritance.
8. Discuss in detail about Steps for Data Mining.
9. Discuss in detail about Data Warehousing.
10. Explain briefly about the Querying and Transformation.

Question	Option 1	Option 2	Option3	Option 4	Answer
The DBMS acts as an interface between what two components of an enterprise-class database system?	Database application and the database	Data and the database	The user and the database application	Database application and SQL	Database application and the database
Which of the following products was an early implementation of the relational model developed by E.F. Codd of IBM?	Nothing	DB2	dBase	R:base	DB2
The following are components of a database except _________	user data	metadata	reports	indexes	reports
An application where only one user accesses the database at a given time is an example of a(n) _________.	single-user database application	multiuser database application	e-commerce database application	data mining database application	single-user database application
An on-line commercial site such as Amazon.com is an example of a(n)_________.	single-user database application	multiuser database application	e-commerce database application	data mining database application	e-commerce database application
SQL stands for_________.	Structured Query Language	Sequential Query Language	Structured Question Language	Sequential Question Language	Structured Query Language
Because it contains a description of its own structure, a database is considered to be _________.	described	metadata compatible	self-describing	an application program	self-describing
The following are functions of a DBMS except _________	creating and processing forms	creating databases	processing data	administrating databases	creating and processing forms
Helping people keep track of things is the purpose of a(n) _________.	database	table	instance	relationship	database
An Enterprise Resource Planning application is an example of a(n) _________.	single-user database application	multiuser database application	e-commerce database application	data mining database application	multiuser database application
A DBMS that combines a DBMS and an application generator is _________.	Microsoft's SQL Server	Microsoft's Access	IBM's DB2	Oracle Corporation's Oracle	Microsoft's Access

You have run an SQL statement that asked the DBMS to display data in a table named USER_TABLES. The results include columns of data labeled "TableName," "NumberOfColumns" and "PrimaryKey." You are looking at _________.	user dat	metadata	A report	indexes	metadata
Which of the following is not considered to be a basic element of an enterprise-class database system?	Users	Database applications	DBMS	COBOL programs	COBOL programs
The DBMS that is most difficult to use is ________	Microsoft's SQL Server	Microsoft's Access	IBM's DB2	Oracle Corporation's Oracle	Oracle Corporation 's Oracle
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 (GEC)	Maximum cardinality
Which type of entity cannot exist in the database unless another type of entity also exists in the database, but does not require that the identifier of that other entity be included as part of its own identifier?	Weak entity	Strong entity	ID-dependent entity	ID-independent entity	Weak entity
In a one-to-many relationship, the entity that is on the one side of the relationship is called a(n) _________entity.	parent	child	instance	subtype	parent

Which type of entity represents an actual occurrence of an associated generalized entity?	Supertype entity	Subtype entity	Archetype entity	Instance entity	Instance entity
A recursive relationship is a relationship between an entity and_________.	itself	a subtype entity	an archetype entity	an instance entity	itself
Which of the following indicates the minimum number of entities that must be involved in a relationship?	Minimum cardinality	Maximum cardinality	ERD	Greater Entity Count (GEC)	Minimum cardinality
Which of the following refers to something that can be identified in the users' work environment, something that the users want to track?	Entity	Attribute	Identifier	Relationship	Entity
In which of the following is a single-entity instance of one type related to many entity instances of another type?	One-to-One Relationship	One-to-Many Relationshi p	Many-to-Many Relationship	Composite Relationship	One-to-Many Relationship
Which of the following refers to an entity in which the identifier of one entity includes the identifier of another entity?	Weak entity	Strong entity	ID-dependent entity	ID-independent entity	ID-dependent entity
Which type of entity is related to two or more associated entities that each contain	Supertype entity	Subtype entity	Archetype entity	Instance entity	Supertype entity

specialized attributes that apply to some but not all of the instances of the entity?					
An attribute that names or identifies entity instances is a(n):	entity.	attribute.	identifier.	relationship.	identifier.
Properties that describe the characteristics of entities are called:	entities.	attributes.	identifiers.	relationships.	attributes.
In which of the following can many entity instances of one type be related to many entity instances of another type?	One-to-One Relationship	One-to-Many Relationship	Many-to-Many Relationship	Composite Relationship	Many-to-Many Relationship
Entities of a given type are grouped into a(n):	database.	entity class.	attribute.	ER	entity class.
Which of the following is NOT a basic element of all versions of the E-R model?	Entities	Attributes	Relationships	Primary keys	Primary keys
In which of the following is a single-entity instance of one type of related to a single-entity instance of another type?	One-to-One Relationship	One-to-Many Relationship	Many-to-Many Relationship	Composite Relationship	One-to-One Relationship
Entities can be associated with one another in which of the following?	Entities	Attributes	Identifiers	Relationships	Relationships
Which type of entity has its relationship to another entity determined by an attribute in that other entity called a discriminator?	Supertype entity	Subtype entity	Archetype entity	Instance entity	Subtype entity
Which type of entity represents a logical generalization whose actual occurrence is represented by a second, associated entity?	Supertype entity	Subtype entity	Archetype entity	Instance entity	Archetype entity
In a one-to-many relationship, the entity that is on the many side of the relationship is called a(n)_________entity.	parent	child	instance	subtype	child

Which of the following data constraints would be used to specify that the value of cells in a column must be one of a specific set of possible values?	A domain constraint	A range constraint	An intrarelation constraint	An interrelation constraint	A domain constraint
In a 1:N relationship, the foreign key is placed in:	either table without specifying parent and child tables.	the parent table.	the child table.	either the parent table or the child table.	the child table.
Which of the following column properties specifies whether or not cells in a column must contain a data value?	Null status	Data type	Default value	Data constraints	Null status
A primary key should be defined as:	NULL.	NOT NULL.	Either of the above can be use	None of the above are correct.	NOT NULL.
Which of the following column properties would be used to specify that cells in a column must contain a monetary value?	Null status	Data type	Default value	Data constraints	Data type
Which of the following situation requires the use of ID-dependent entities?	Association relationships only	Multivalued attributes only	Archetype/instance relationships only	All of the above use ID dependent entities	All of the above use ID dependent entities
A foreign key is:	a column containing the primary key of another table.	used to define data types.	used to define null status.	all of the above are above correct.	a column containing the primary key of another table.
Which of the following columns is(are) are required in a table?	A foreign key	An alternate key	A primary key	A surrogate key.	A primary key
In a 1:1 relationship, the foreign key is placed in:	either table without specifying parent and child tables.	the parent table.	the child table.	either the parent table or the child table.	either table without specifying parent and child tables.
Which of the following column properties would be used to specify that cells in a column must be immediately filled with a monetary value of $10,000?	Null status	Data type	Default value	Data constraints	Default value
The identifier of an entity will become the_________of the new table.	foreign key	main attribute	primary key	identity key	primary key

Which of the following data constraints would be used to specify that the value of a cell in one column must be less than the value of a cell in another column in the same row of the same table?	A domain constraint	A range constraint	An intrarelation constraint	An interrelation constraint	An intrarelatio n constraint
A unique, DBMS-supplied identifier used as the primary key of a relation is called a(n):	primary key.	foreign key.	composite key.	surrogate key.	surrogate key.
Which is not true about surrogate keys?	They are short.	They are fixe	They have meaning to the user.	They are numeri	They have meaning to the user.
For every relationship, how many possible types of actions are there when enforcing minimum cardinalities?	Two	Three	Four	Six	Six
Which constraint requires that the binary relationship indicate all combinations that must appear in the ternary relationship?	MUST COVER	MUST NOT	Both of the above.	None of the above is correct.	MUST COVER
Each entity is represented as a(n):	tuple.	table.	attribute.	file.	table.
For every relationship, how many possible sets of minimum cardinalities are there?	Two	Three	Four	Six	Four
If a relationship has a cascade updates constraint, then if _________in the parent table is changed, then the same change will automatically be made to any corresponding foreign key value.	the primary key	any alternate key	a surrogate key	a foreign key	the primary key
Which of the following column properties would be used to specify that cells in a column must contain a monetary value that is less than another monetary value in the same row?	Null status	Data type	Default value	Data constraints	Data constraints
Poor data administration can lead to which of the following?	A single definition of the same data entity	Familiarity with existing data	Missing data elements	All of the above.	Missing data elements

A traditional data administrator performs which of the following roles?	Tune database performance	Establish backup and recovery procedures	Resolve data ownership issues	Protect the security of the database.	Resolve data ownership issues
If both data and database administration exist in an organization, the database administrator is responsible for which of the following?	Data modeling	Database design	Metadata	All of the above.	Database design
Which of the following is part of an administrative policy to secure a database?	Authentication policies	Limiting particular areas within a building to only authorized people	Ensure appropriate responses rates are in external maintenance agreements	All of the above.	All of the above.
The fact that the same operation may apply to two or more classes is called what?	Inheritance	Polymorphi sm	Encapsulation	Multiple classification	Polymorphi sm
The transaction log includes which of the following?	The before-image of a record	The after-image of a record	The before and after-image of a record	The essential data of the record	The essential data of the record