**13BECS7E07 ADVANCED DATABASES**

**INTENDED OUTCOMES:**

- Advanced database aims at developing computer application with different kinds of data models. It is also deals with the Transaction management of these different databases.
- To study the needs of different databases.
- To understand about different data models that can be used for these databases.
- To make the students to get familiarized with transaction management of the database
- To develop in-depth knowledge about web and intelligent database.
- To provide an introductory concept about the way in which data can be stored in geographical information systems etc.,

**UNIT- I DISTRIBUTED DATABASES**

Distributed DBMS Concepts and Design – Introduction – Functions and Architecture of DDBMS – Distributed Relational Database Design – Transparency in DDBMS – Distributed Transaction Management – Concurrency control – Deadlock Management – Database recovery – The X/Open Distributed Transaction Processing Model – Replication servers – Distributed Query Optimisation - Distribution and Replication in Oracle.

**UNIT- II OBJECT ORIENTED DATABASES** Object Oriented Databases – Introduction – Weakness of RDBMS – Object Oriented Concepts Storing Objects in Relational Databases – Next Generation Database Systems – Object Oriented Data models – OODBMS Perspectives – Persistence – Issues in OODBMS – Object Oriented Database Management System Manifesto – Advantages and Disadvantages of OODBMS – Object Oriented Database Design – OODBMS Standards and Systems – Object Management Group – Object Database Standard ODMG – Object Relational DBMS –Postgres - Comparison of ORDBMS and OODBMS.

**UNIT -III WEB DATABASES**

Web Technology And DBMS – Introduction – The Web – The Web as a Database Application Platform – Scripting languages – Common Gateway Interface – HTTP Cookies – Extending the Web Server – Java – Microsoft's Web Solution Platform – Oracle Internet Platform – Semi structured Data and XML – XML Related Technologies – XML Query Languages

**UNIT- IV INTELLIGENT DATABASES**

Enhanced Data Models For Advanced Applications – Active Database Concepts And Triggers – Temporal Database Concepts – Deductive databases – Knowledge Databases.

**UNIT- V CURRENT TRENDS**

Mobile Database – Geographic Information Systems – Genome Data Management – Multimedia Database – Parallel Database – Spatial Databases - Database administration – Data Warehousing and Data Mining.

**TEXT BOOKS:**

| S.NO | Author(s) Name | Title of the book | Publisher | Year of publication |
|---|---|---|---|---|
| **1** | Thomas M. Connolly, Carolyn E. Begg | Database Systems - A Practical Approach to Design , Implementation , and Management (3rd Edition) | Pearson Education | 2003 |

**REFERENCE BOOKS:**

| S.NO | Author(s) Name | Title of the book | Publisher | Year of publication |
|---|---|---|---|---|
| **1** | Ramez Elmasri & Shamkant B.Navathe | Fundamentals of Database Systems (4th Edition) | Pearson Education | 2004 |
| **2** | M.Tamer Ozsu , Patrick Ualduriel | Principles of Distributed Database Systems (Second Edition) | PearsonEducation | 2003 |
| **3** | C.S.R.Prabhu | Object Oriented Database Systems | PHI | 2003 |
| **4** | Peter Rob and Corlos Coronel | Database Systems – Design, Implementation and Managemen (5th Edition) | Thompson Learning, Course Technology | 2003 |

1) **Distributed Databases**

# An Introduction to Distributed Databases

A *distributed database* appears to a user as a single database but is, in fact, a set of databases stored on multiple computers. The data on several computers can be simultaneously accessed and modified using a network. Each database server in the distributed database is controlled by its local DBMS, and each cooperates to maintain the consistency of the global database. Figure 21 - 1 illustrates a representative distributed database system.

The following sections outline some of the general terminology and concepts used to discuss distributed database systems.

## Clients, Servers, and Nodes

A database *server* is the software managing a database, and a *client* is an application that requests information from a server. Each computer in a system is a *node*. A node in a distributed database system can be a client, a server, or both. For example, in Figure 21 - 1, the computer that manages the HQ database is acting as a database server when a statement is issued against its own data (for example, the second statement in each transaction issues a query against the local DEPT table), and is acting as a client when it issues a statement against remote data (for example, the first statement in each transaction is issued against the remote table EMP in the SALES database).

Oracle supports heterogeneous client/server environments where clients and servers use different character sets. The character set used by a client is defined by the value of the NLS_LANG parameter for the client session. The character set used by a server is its database character set. Data conversion is done automatically between these character sets if they are different. For more information about National Language Support features, refer to *Oracle7 Server Reference*.

## Direct and Indirect Connections

A client can connect directly or indirectly to a database server. In Figure 21 - 1, when the client application issues the first and third statements for each transaction, the client is connected directly to the intermediate HQ database and indirectly to the SALES database that contains the remote data.

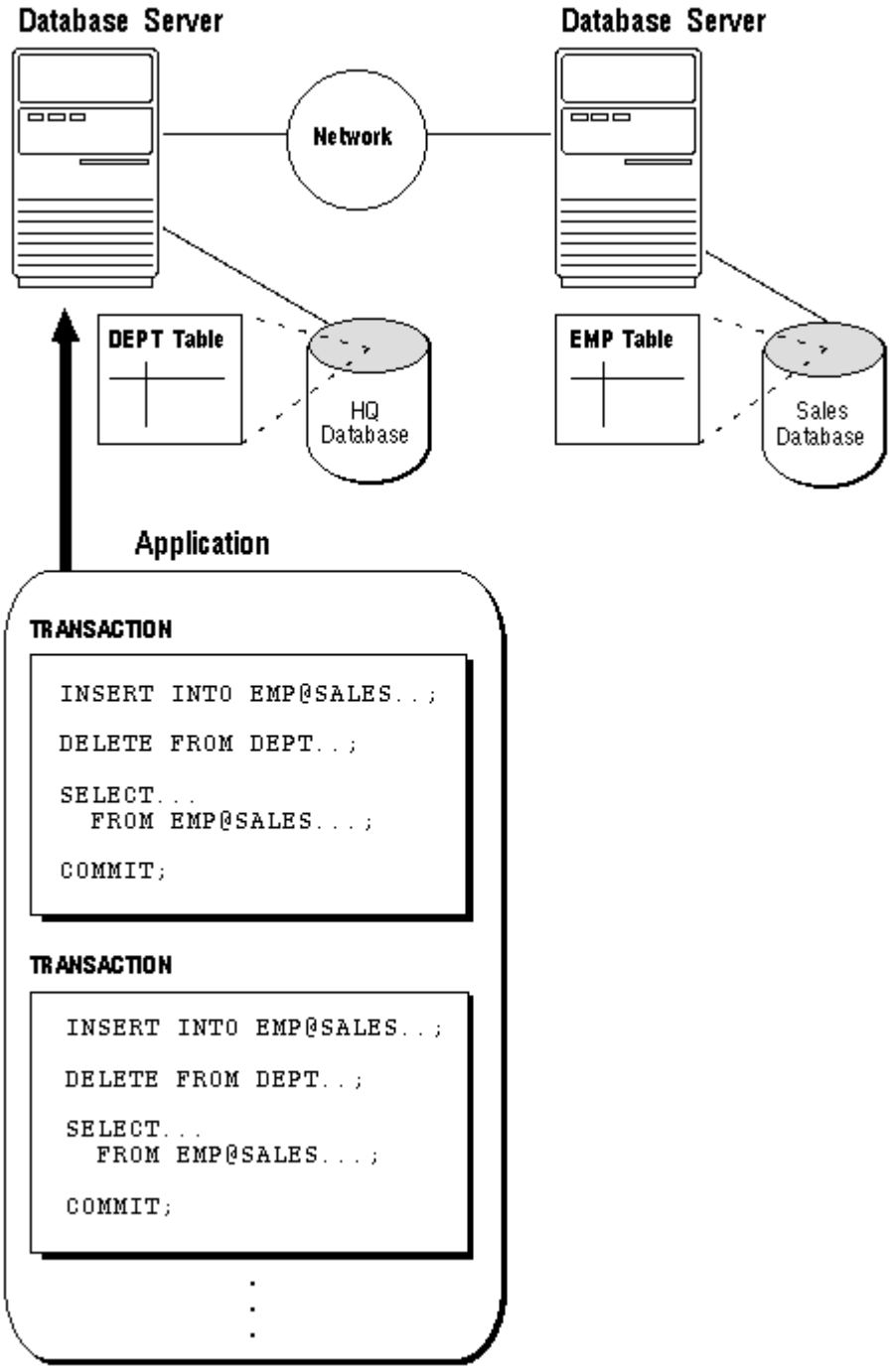



**Figure 21 - 1. An Example of a Distributed DBMS Architecture**

## Site Autonomy

*Site autonomy* means that each server participating in a distributed database is administered independently (for security and backup operations) from the other databases, as though each database was a non-distributed database. Although all the

databases can work together, they are distinct, separate repositories of data and are administered individually. Some of the benefits of site autonomy are as follows:

- Nodes of the system can mirror the logical organization of companies or cooperating organizations that need to maintain an "arms length" relationship.
- Local data is controlled by the local database administrator. Therefore, each database administrator's domain of responsibility is smaller and more manageable.
- Independent failures are less likely to disrupt other nodes of the distributed database. The global database is partially available as long as one database and the network are available; no single database failure need halt all global operations or be a performance bottleneck.
- Failure recovery is usually performed on an individual node basis.
- A data dictionary exists for each local database.
- Nodes can upgrade software independently.

**Schema Objects and Naming in a Distributed Database**

A schema object (for example, a table) is accessible from all nodes that form a distributed database. Therefore, just as a non-distributed local DBMS architecture must provide an unambiguous naming scheme to distinctly reference objects within the local database, a distributed DBMS must use a naming scheme that ensures that objects throughout the distributed database can be uniquely identified and referenced.

To resolve references to objects (a process called *name resolution*) within a single database, the DBMS usually forms object names using a hierarchical approach. For example, within a single database, a DBMS guarantees that each schema has a unique name, and that within a schema, each object has a unique name. Because uniqueness is enforced at each level of the hierarchical structure, an object's local name is guaranteed to be unique within the database and references to the object's local name can be easily resolved.

Distributed database management systems simply extend the hierarchical naming model by enforcing unique database names within a network. As a result, an object's*global object name* is guaranteed to be unique within the distributed database, and references to the object's global object name can be resolved among the nodes of the system.

For example, Figure 21 - 2 illustrates a representative hierarchical arrangement of databases throughout a network and how a global database name is formed.

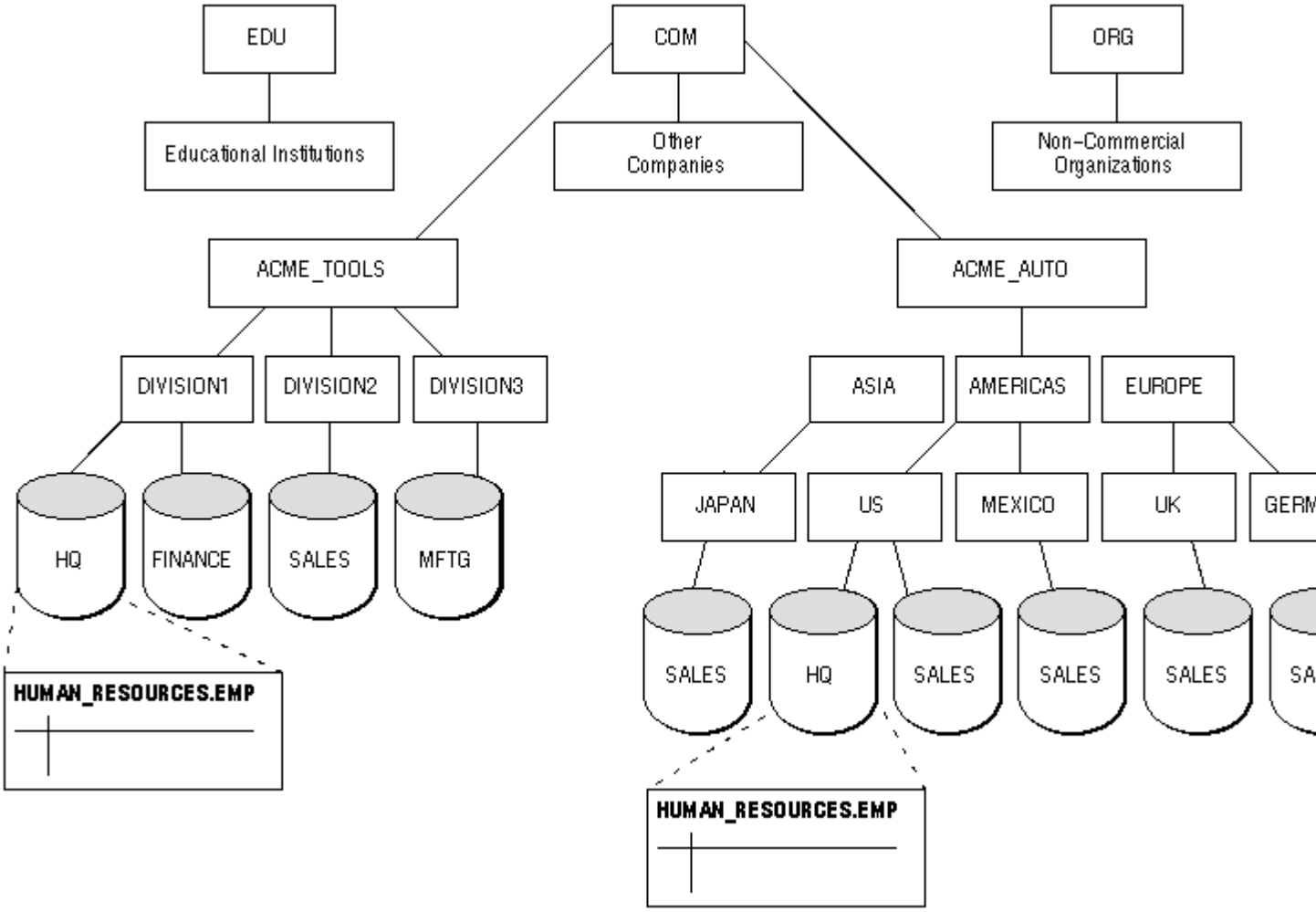



**Figure 21 - 2. Network Directories and Global Database Names**

## 2. Parallel databases

A **parallel database** system seeks to improve performance through parallelization of various operations, such as loading data, building indexes and evaluating queries.[1] Although data may be stored in a distributed fashion, the distribution is governed solely by performance considerations. Parallel databases improve processing and input/output speeds by using multiple CPUs and disks in parallel. Centralized and client–server database systems are not powerful enough to handle such applications. In parallel processing, many operations are performed simultaneously, as opposed to serial processing, in which the computational steps are performed sequentially. Parallel databases

can be roughly divided into two groups, the first group of architecture is the multiprocessor architecture, the alternatives of which are the followings:

**Shared memory architecture**

Where multiple processors share the main memory space.

**Shared disk architecture**

Where each node has its own main memory, but all nodes share mass storage, usually a storage area network. In practice, each node usually also has multiple processors.

**Shared nothing architecture**

Where each node has its own mass storage as well as main memory.

The other architecture group is called hybrid architecture, which includes:

- Non-Uniform Memory Architecture (NUMA), which involves the non-uniform memory access.
- Cluster (shared nothing + shared disk: SAN/NAS), which is formed by a group of connected computers.

## Introduction

- Parallel machines are becoming quite common and affordable
  - Prices of microprocessors, memory and disks have dropped sharply
  - Recent desktop computers feature multiple processors and this trend is projected to accelerate
- Databases are growing increasingly large
  - large volumes of transaction data are collected and stored for later analysis.
  - multimedia objects like images are increasingly stored in databases
- Large-scale parallel database systems increasingly used for:
  - storing large volumes of data
  - processing time-consuming decision-support queries
  - providing high throughput for transaction processing

## Parallelism in Databases

- Data can be partitioned across multiple disks for parallel I/O.
- Individual relational operations (e.g., sort, join, aggregation) can be executed in parallel
  - data can be partitioned and each processor can work independently on its own partition.
- Queries are expressed in high level language (SQL, translated to relational algebra)

- makes parallelization easier.

- Different queries can be run in parallel with each other. Concurrency control takes care of conflicts.
- Thus, databases naturally lend themselves to parallelism.

**Intraquery Parallelism**

- Execution of a single query in parallel on multiple processors/disks; important for speeding up long-running queries.
- Two complementary forms of intraquery parallelism :
  - **Intraoperation Parallelism** – parallelize the execution of each individual operation in the query.
  - **Interoperation Parallelism** – execute the different operations in a query expression in parallel.

  the first form scales better with increasing parallelism because the number of tuples processed by each operation is typically more than the number of operations in a query

## 3. Concurrency control in Databases

Concurrency control (CC) is a process to ensure that data is updated correctly and appropriately when multiple transactions are concurrently executed in DBMS (Connolly & Begg, 2015).

In general, concurrency control is an essential part of TM. It is a mechanism for correctness when two or more database transactions that access the same data or data set are executed concurrently with time overlap. According to Wikipedia.org, if multiple transactions are executed serially or sequentially, data is consistent in a database. However, if concurrent transactions with interleaving operations are executed, some unexpected data and inconsistent result may occur. Data interference is usually caused by a write operation among transactions on the same set of data in DBMS. For example, the lost

update problem may occur when a second transaction writes a second value of a data content on top of the first value written by a first concurrent transaction. Other problems such as the dirty read problem, the incorrect summary problem are also identified in the Connolly and Begg's textbook.

There are two main kinds of concurrency control mechanisms:

1. Pessimistic (conservative) concurrency control.

The pessimistic concurrency control delays the transactions if they conflict with other transactions at some time in the future by locking or a time-stamping technique.

1. Optimistic concurrency control.

According to Kung and Robinson (1981), the optimistic concurrency control, that assumes that the conflict is rare, allows concurrent transactions to proceed without imposing delays to ensure serializability then check conflict only at the end, when a transaction commits. Notice that there is another mechanism, semi-optimistic technique, which uses lock operations in some situations (if they may violate some rules), and does not lock in other circumstances.

1. The pros and cons of the pessimistic and optimistic concurrency control mechanisms

Both pessimistic and optimistic concurrency control mechanisms provide different performance, e.g., the different average transaction completion rates

or throughput, depending on transaction types mix, computing level of parallelism, and other events.

According to Vallejo, Sanyal, Harris, Vallejo, Beivide, Unsal, Valero (2011), there is a tradeoff between the concurrency control techniques. Their pros and cons are shown below:

For pessimistic concurrency control, the strength is:

- Guarantee that all transactions can be executed correctly.

- Data is properly consistent by either rolling back to the previous state (Abort operation) or new content (Commit operation) when the transaction conflict is cleared.

- Database is relatively stable and reliable.

Its weakness is:

- Transactions are slow due to the delay by locking or time-stamping event.

- Runtime is longer. Transaction latency increases significantly.

- Throughput or the amount of work (e.g. read/write, update, rollback operations, etc.)

is reduced.

For optimistic concurrency control, the strength is:

- Transactions are executed more efficiently.

- Data content is relatively safe.

- Throughput is much higher.

Its weakness is:

- There is a risk of data interference among concurrent transactions since it transactions conflict may occur during execution. In this case, data is no longer correct.

- Database may have some hidden errors with inconsistent data; even conflict check is performed at the end of transactions.

- Transactions may be in deadlock that causes the system to hang.

Many users often encounter the data interference issue in database management system in stock markets. The simple example addresses the inconsistent data issue as shown below.

Multiple analysts or investors from Fidelity Investments, LLC access a client's fund in a database for stock trading from either Dow, Bonds or Mutual Funds. Both analysts A, B perform some transactions to transfer some amount of money to buy stocks from different funds for their daily works. In this

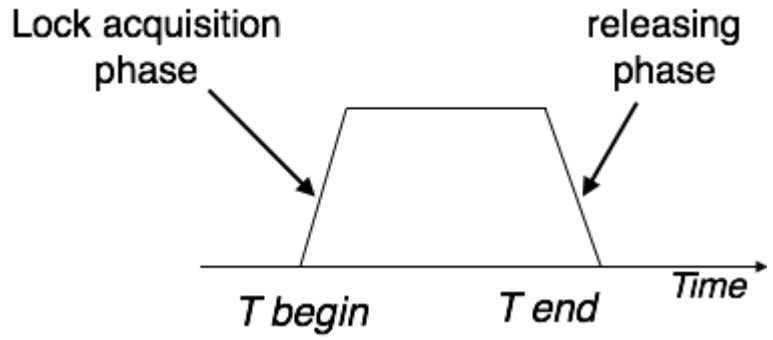
scenario, the nested transaction is used in database. At scheduled step T7, a transaction manager faces inconsistent data (i.e. $800 or $1000) in the client's balance. The Operating System can execute either thread A for a balanced result of 1000 or thread B for a balanced result of 800. Either case the balance content is incorrect. For pessimistic technique, the transaction B may be delayed for a long time or in a deadlock. For optimistic technique, it goes through with incorrect data in the database. Furthermore, the micro execution of the of tware threads is unknown, ambiguous and out of control. Database is in the unknown state.

| Steps | Operations | Analyst A's Transactions | Analyst B's Transactions | Client's balance Bal $US |
|---|---|---|---|---|
| T1 | Beginning | Start | | 1000 |
| T2 | A reads balance | Read(Bal) | | 1000 |
| T3 | Transfer 100 to Fund A | Bal = Bal – 100 | Start | 1000 |
| T4 | B reads balance | | Read(Bal) | 1000 |
| T5 | A updates balance | Write(Bal) | Bal = Bal - 200 | 900 |
| | Transfer 200 to | | | |

| | | | | |
|---|---|---|---|---|
| | Fund B | | | |
| T6 | B updates balance | | Write(Bal) | 800 |
| T7 | A aborts, B commits | Rollback | Commit | 1000? or 800? |
| T8 | End | End | End | Conflict! |

## Two-Phase Locking 2PL

This locking protocol divides the execution phase of a transaction into three parts. In the first part, when the transaction starts executing, it seeks permission for the locks it requires. The second part is where the transaction acquires all the locks. As soon as the transaction releases its first lock, the third phase starts. In this phase, the transaction cannot demand any new locks; it only releases the acquired locks.



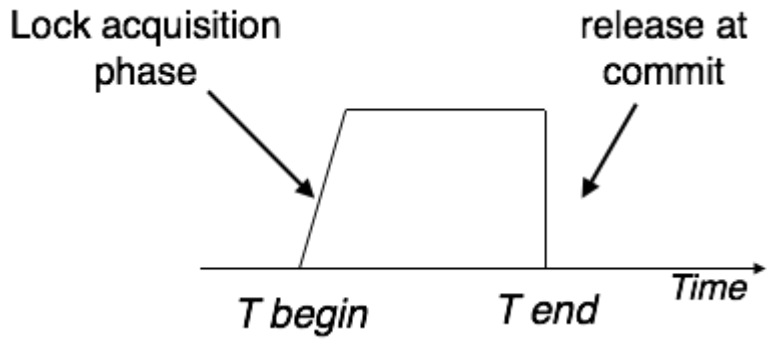

Two-phase locking has two phases, one is **growing**, where all the locks are being acquired by the transaction; and the second phase is shrinking, where the locks held by the transaction are being released.

To claim an exclusive (write) lock, a transaction must first acquire a shared (read) lock and then upgrade it to an exclusive lock.

## Strict Two-Phase Locking

The first phase of Strict-2PL is same as 2PL. After acquiring all the locks in the first phase, the transaction continues to execute normally. But in

contrast to 2PL, Strict-2PL does not release a lock after using it. Strict-2PL holds all the locks until the commit point and releases all the locks at a time.




Strict-2PL does not have cascading abort as 2PL does.

**Features of OODBMS**

The mandatory features are tabulated as below:

| Features of OODBMS from General databases | Features of OODBMS from Object Oriented databases |
|---|---|
| Orthogonal Persistence of data<br>Able to handle large databases<br>Controlled Concurrency<br>Restoring or data Recovery<br>Query facility on adhoc basls | Construction of complex Objects<br>identily of an object<br>Feature of Classes and types<br>Property of encapsulation<br>Property of Inheritance<br>Property of overriding combined with late binding<br>Property of Extensibility<br>Property of Computational completeness |

These mandatory features are described as:

**Feature of Persistence:** This feature of OODBMS includes the survival of data as well as persistence should be orthogonal and implicit. 1ne orthogonal implies each object should be persistent as such and the user should not have to explicitly move or copy data to make it persistent. In particular, a database can store, individual objects and the volatile mainmemory of an application can contain collections of objects.

**Able to handle large databases:** This feature· includes the optimal "management of very large databases using techniques like Data clustering, Data buffering, Query optimization, Access path selection and Index management.

**Controlled Concurrency:** This feature guarantees harmonious coexistence among users. Working simultaneously on the database and enjoying controlled sharing. By

allowing multiple transactions to run concurrently will improve the performance of the system in terms of increased throughout or improved response time. Ensuring consistency in spite of concurrent execution of transaction require additional effort which is performed by the concurrency controller system of DBMS.

**Restoring or Data Recovery:** This feature indicates the restoration of the system to a state that existed before the software or hardware based crash such as processor or disk failure. The recovery refers to the various strategies and procedures involved in protecting your database against data loss and reconstructing the data such that no data- is lost after failure.

**Query facility on basis:** This feature includes the facility of applying query that should be efficient using query optimization and application independent that can work on any database.

**Construction of Complex Objects:** This feature enables the OODBMS to construct complex objects like tuples sets, lists and arrays from the simple objects like integers, characters, byte strings Boolean and float using the constructors and appropriate operators.

**Identity of an object:** This feature ensures that each object is assigned an Object Identifier (OID) when it is created. Object identity assists OODBMS to uniquely identify an object, thereby automatically providing entity integrity. In fact, as object identity ensures system-wide uniqueness, it provides a stronger constraint than the relational data model's entity integrity, which requires any uniqueness within a relation.

**Feature of Classes and types:** This feature supports the notion of classes and types for defining a set of similar objects. Objects that have the safe attributes and respond to the same messages can be grouped together to form a class. The attributes and associated methods are defined once for the' class rather than separately for each object. The type of variables and expressions help to do the type checking at compile time, to check the correctness of the programs.

**Property of encapsulation:** This property of OODBMS implies that an object contains both the data structure and the set of operations that can be used to manipulate it. An object is said to encapsulate (hide) data and program. This means that the user cannot see the inside of the object but can use the object by calling the program part of the object.

**Property of Inheritance:** This property of OODBMS implies that feature of objects by which instances of a class can have access to data and programs contained in a previously defined class, without those definitions being restarted. The different types of inheritance used for refusing the code are substitution inheritance, inclusion inheritance, constraint inheritance and specialization.

**Property of overriding combined with late binding:** This property of OODBMS implies the ability to use the same message to objects different classes and have them behave differently. Thus we can define the message "+" for both the addition of numbers and the concatenation -joining) of characters, even though both these operations are completely different. This feature provides the' ability to use the same word to invoke different methods, according to similarity of meaning. Here the late binding is being done as the- system cannot bind operation names to programs at compile time and thus, operation names are resolved at run-time.

**Property of Extensibility:** This property of OODBMS implies that newdata types to be built from existing types. The ability to factor out common properties of several classes and form them into a super class that can be shared with subclasses can greatly reduce redundancy within system. The usage of both system defined types and user-defined types *is same*.

**Property of Computational Completeness:** This feature of OODBMS implies that does can employ any computable function using the reasonable connectivity to any existing programming language. This feature makes OODBMS more powerful than a database system which only stores and retrieves data and performs simple computations on· atomic values.

Similarities and differences between interface and abstract class

There are few similarities and differences between interface and abstract class in java software development language as bellow.

**Similarities**

Interface can not be instantiated. Same way, you can not instantiate abstract class.

That means you can not create object of interface or abstract class.

**Difference**

Differences between interface and abstract class in java software development language are as bellow.

| Interface | Abstract Class |
|---|---|
| We can use interface keyword to declare interface. | We can use abstract keyword to declare abstract class. |
| Interface can hold only abstract methods(without implementation). | Abstract class can hold abstract(without implementation) as well as non abstract methods. |
| Interface can be implemented using implements keyword. | Abstract class can be extended using extends keyword. |
| We can achieve multiple inheritance using interface as we can implement multiple interfaces to any class. | Abstract class doesn't support multiple inheritance as we can not extend more than one class. |
| Interface can not hold main method, static methods or constructor. | Abstract class can hold main method, static methods or constructor. |
| Also it can hold only static and final variables and mandatory to initialize them. | It can hold static, non static, final, non final variables and also it is not mandatory to initialize them. |
| We can achieve 100% abstraction using interface as all methods are abstract by default. It can not hold concrete methods. | We can achieve partial(0% to 100%) abstraction using abstract class as it can hold abstract as well concrete methods too. |
| When you add a new method in existing interface it breaks all its implementation and you need to provide an implementation in all clients which is not good. | By using abstract class you can provide default implementation in super class by creating concrete method. It is not required to provide its implementation in sub class. |

## Web Based Architecture

### Architecture of Web-based systems

#### Client Server and 2 tier Web Architectures

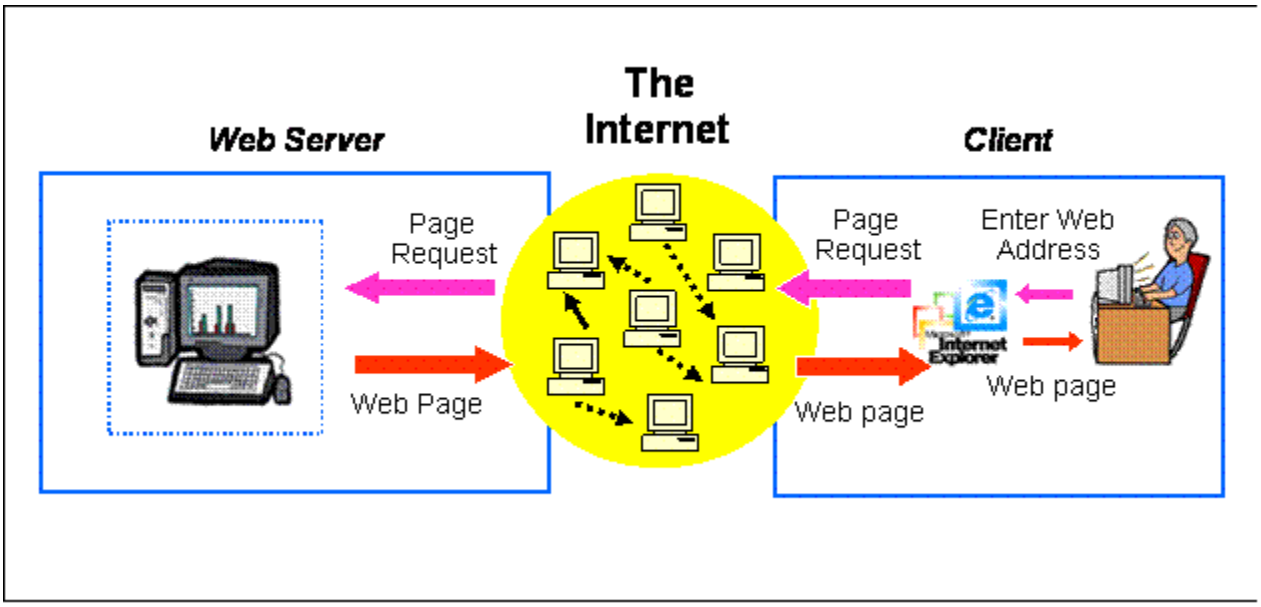
Typically, when you are browsing the Internet, you will be using *Web Browser* software such as Internet Explorer or Mozilla Firefox. The computer which is running a browser is called a *client*, whilst the machine which is providing Web pages is called a *server*.

When you dial up to an Internet Service Provider ( ISP ) e.g. Blue Yonder, Aol, your computer is forming a network connection to a *Web server*. In this situation, your computer is in effect a *client*, which is linked to an ISP *Web server*. The web server, as the name suggests, serves your browser with Web pages (e.g. HTML, ASPX, JSP pages etc).

This simple scenario, where the Web server is connected to one or more clients is known as a 2 tier architecture model.

Figure B below demonstrates how Web pages are accessed via a browser, using a 2 tier architecture.

**Figure B - A simple diagram of 2 tier Client/server architecture**




## 3-Tier architectures

Generally computing applications consist of three different and distinct types of functionalities.

Data Management

Functional Logic

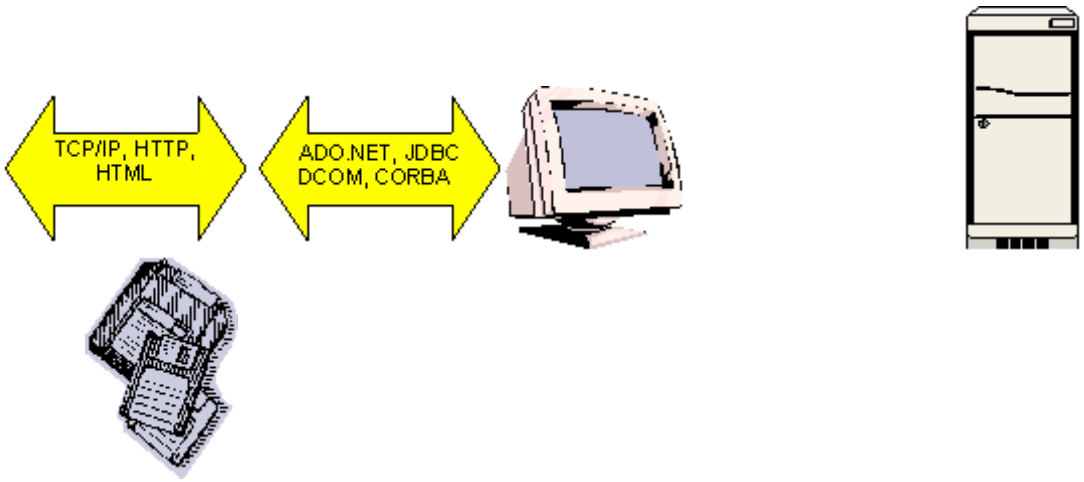
Presentation Services

- **Presentation Services**: These manifest themselves in the form of information display and user data input facilities. Generally the front-end for user

interaction. For example logging in requires interaction in the form of collecting username and password information using a HTML-form.

- **Functional logic**: Every application includes some data processing and this may also involve database interactivity. For example user authentication requires the logic unit to read username-password combinations from a database and compare until a good comparison (hopefully) is arrived at.
- **Data Management**: Data, its storage, insertion and retrieval, its management and alteration is central to computing applications. For example a database management system (DBMS) is required for the management of usernames and associated passwords, their owners, etc.

**Figure C - An illustration of a 3 tier architecture model**




1st tier- Web browser 2nd tier- Web server 3rd tier- Data storage

## n-Tier Architectures

It is also important to note that Web application architecture sometimes is referred to as multi-tier. In effect we could have more than 3-tiers in circumstances when the Web server requires to access one or more application servers for specialised services. This is known as n-Tier architecture

Each layer within an N-tier architecture could be thought of as 'logical components' interacting with the layer above or below. Layers provide a means of grouping functionality within the application structure.

Some benefits of this approach include flexibility of component location - each layer may be held on a different server, this facilitates scalable applications capable of handling heavier server loads. Additionally each layer is encapsulated making it possible to change one layer without affecting another.

## ONLINE QUESTIONS

## UNIT-I

| questions | opt1 | opt2 | opt3 | opt4 | opt5 | opt6 | answer |
|---|---|---|---|---|---|---|---|
| Let us suppose that in a distributed database, during a transaction T1, one of the sites, say S1, is failed. When recovers, the site S1 has to check its log file (log based recovery) to decide the next move on the transaction T1. If the log contains a record, what the site S1 has to do? | Need not do anything | Perform Undo | Perform Redo | Abort the transaction | | | Perform Undo |
| Global Wait-for graph is used for _____ in Distributed database. | Handling concurrency control | Handling failures | Handling deadlock | None of these | | | Handling concurrency control |
| In Distributed database, _____ are the transactions for which a log is found in the log file, but neither a log nor an log is found. | In-doubt transactions | Serialized transactions | Cascadeless transactions | Distributed transactions | | | Distributed transactions |
| While using commit protocols for handling atomicity issues, the distributed databa | Two-phase commit protocol | Three-phase commit protocol | Both of these | None of these | | | Two-phase commit protocol |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| se system may enter into a situation called Blocking problem. Which of the following commit protocols can avoid Blockin g problem? | | | | | | | |
| Which of the following concurrency control protocol is suitable for an application wh ere frequency of read operation is much greater than that of write operation? | Majority protocol | Quorum-consensus protocol | Biased protocol | Single lock protocol | | | Majority protocol |
| Which of the following parallel database a rchitectures is/are mainly used by distributed databa se system? | Shared Memory | Shared Disk | Shared Nothing | Hierarchical | | | Shared Memory |
| Which of the following is not a promise of distributed databa se? | Network Transpar ency | Replication Transapare ncy | Fragmenta tion Transpare ncy | None of these | | | Network Transparency |
| All sites in a distributed databa se must commit at exactly the same instant. TRUE/FALSE | TRUE | FALSE | | | | | TRUE |
| The real use of the Two-phase commit protocol is _____ | Deadloc k will not occur | Concurrenc y control can be avoided | Atomicity, i.e, all-or-nothing co mmits at all sites | None of these | | | Concurrency control can be avoided |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Read one, write all available protocol is used to increase _____ in a distributed database system. | Availability | Robustness | Both Availability and Robustness | Neither Availability nor Robustness | | | Both Availability and Robustness |
| Which of the following is more suitable for parallelizing a single query? | Coarse-granularity parallelism | Fine-granularity parallelism | Both of these | None of these | | | Coarse-granularity parallelism |
| A regular desktop machine (with two to four processors) can be mentioned as _____ machine. | Fine-granularity parallelism | Coarse-granularity parallelism | Massively parallel | None of these | | | Fine-granularity parallelism |
| If we parallelize the execution of large number of small transactions in a database system, then which of the following would be increased? | Response time | Rotational latency | Throughput | All of these | | | Rotational latency |
| Which of the following would improve the availability of resources especially for reading in a distributed database system? | Fragmentation of database | Replication of database | Defragmentation of database | Tuning of database | | | Replication of database |
| _____ is very much required to process a query in a distributed database. | Global System Catalog | Database Views | Triggers | None of these | | | Global System Catalog |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| A fragmentation technique wherein every tuple of a table is assigned to one or more fragments as a result of fragmentation is called _____ | Vertical Fragmentation | Horizontal Fragmentation | Hybrid Fragmentation | None of these | | | Horizontal Fragmentation |
| In a distributed database application, if we have very many number of read only queries than update queries then _____ allocation technique is advantageous. | Hybrid Fragmentation | Horizontal Fragmentation | Replication | Vertical Fragmentation | | | Hybrid Fragmentation |
| Which of the following would be the advantage of Database Fragmentation? | Most of the operations are local to any sites | Reduced Network Traffic | Parallel processing | All of these | | | Reduced Network Traffic |
| The Data Model Which describes how the data is actually stored is: | Internal model | External model | Logical model | None of these | | | External model |
| Data about data is normally termed as | Directory | Data bank | Meta Data | None of these | | | Meta Data |

## UNIT-II

| questions | opt1 | opt2 | opt3 | opt4 | opt5 | opt6 | answer |
|---|---|---|---|---|---|---|---|

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Encapsulation, types, classes, inheritance, overriding combined with late binding, and extensibility are all features of this type of database.** | multidimensional database | relational database | object-oriented | open source | | | object-oriented |
| A Database Management System (DBMS) may be used by, or combined with, this type of transaction manager. | DB2 | CICS | Oracle | Cognos | | | CICS |
| **The acronym ACID is typically used in reference to:** | transactions | CRM analytics | data integrity | stubs | | | transactions |
| **The basic unit of this SQL variant is called a block** | PL/SQL | XQL | SQLJ | TMQL | | | PL/SQL |
| **Which of the following tree structures is unique in that the tree organization varies depending on which nodes are most frequently accessed?** | B-tree | binary tree | quad tree | splay tree | | | splay tree |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **What concept would a data modeler NOT be expected to know?** | class | object | UML | FMP | | | FMP |
| **Which technology was Microsoft's first attempt at an object-oriented database interface?** | MQSeries | ADO | DAO | UDA | | | DAO |
| **At this level of normalization, each column in a table that is not a determiner of the contents of another column, must itself be a function of the other columns in the table.** | 1NF | 2NF | 3NF | DKNF | | | 2NF |
| **This a program tool allows you to reorganize and summarize selected columns and rows of data in a spreadsheet or database table to obtain a desired report.** | pivot table | PERT chart | quad tree | ODBC | | | pivot table |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **In a relational database, this is the data structure that holds a single piece of data,such as first name or phone number.** | catalog | field | entity | object | | | field |
| **This is a common standard language used to update and get information from a database.** | VB | TMQL | PHP | SQL | | | SQL |
| **This is the job title of someone who directs or performs all activities related to maintaining a successful database environment.** | DBA | MOLAP | PHP | RFC | | | DBA |
| **This is a central repository for all or significant parts of the data that an enterprise's various business systems collect.** | data warehouse | business information warehouse | knowledge warehouse | None | | | data warehouse |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **This is a family of relational database management system products from IBM that serve a number of different operating system platforms.** | congnos | DB2 | ADO | Universal Data Access | | | DB2 |
| **This is a type of program that lets one or more computer users create, manage, and access data in a database.** | B2B | DBMS | multidimensional database | None | | | DBMS |
| **In a relational database, this is a data structure that organizes the information about a single topic into rows and columns.** | block | record | tuple | table | | | table |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| The primary difference between the Relational database (RDB) and Object Oriented database (OODB) models is: | OODB incorporates methods in with the definition of the data structure, while RDB does not | OODB supports multiple objects in the same database while RDB only supports a single table per database | RDB allows the definition of the relationships between the different tables, while OODB does not allow the relationships to be defined between objects | RDB supports indexes, while OODB does not support indexes | | | OODB incorporates methods in with the definition of the data structure, while RDB does not |
| The predominant way of storing data today is using which type of database models? | Hierarchical | Network | Object Oriented | Relational | | | Relational |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Two different terms are used to describe the characteristics of interest for an entity. They are attributes and: | classes | entities | properties | traits | | | properties |
| A reflexive association is one where one class is: | broken down into special cases | combined with multiple other classes | combined with one other class | linked back to itself | | | linked back to itself |

## UNIT-III

| questions | opt1 | opt2 | opt3 | opt4 | opt5 | opt6 | answer |
|---|---|---|---|---|---|---|---|
| Common gateway interface is used to | generate executable files from web content by web server | generate web pages | stream videos | none of the mentioned | | | generate executable files from web content by web server |
| An alternative of javascript on | VBScript | ASP.NET | JSP | none of the | | | VBScript |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| windows platform is | | | | mentioned | | | |
| The main difference between PHP and Perl is that | Perl runs slower than PHP | Perl is a server-side technology but PHP is client-side | PHP embeds the script into the HTML code whereas Perl scripts print HTML | Only PHP can be used to display databases on the web | PHP is a web server extension whereas Perl uses CGI | | Only PHP can be used to display databases on the web |
| Which of the following is NOT related to Search Engine technologies? | metawords | Natural Language Processing | spider | robot.txt | inverted index | | metawords |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| The different ways to send data to the CGI program | Arguments of the CGI program | Environment variables | Standard input | All of these | | | All of these |
| The web server then responds back to the _____ ____ accordingly: | Web Browser | Web Server | Server | All of these | | | Web Browser |
| The CGI output must consist of any of these _____ header files: | 1 | 5 | 3 | 4 | | | 3 |
| XML uses the features of | HTML | XHTML | VML | SGML | | | SGML |
| The tags in XML are | Case insensitive | Case sensitive | Browser dependent | None of these | | | Case sensitive |
| _____ programs are automatically loaded and operates as a part of browser. | Utilities | Widgets | Plug-ins | Add-ons | | | Plug-ins |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Which of the following statements is false regarding “Cookies”? | Cookies are programs which run in the background of the web-client | Cookies have the potential of being used to violate the privacy of users | Cookies are very helpful in keeping track of users in developing online shopping cart applications, personalized portals and in advertising on web sites | Cookies cannot contain more than 4Kb of data | Cookies usually contain data in the form of name = value pairs | | Cookies are programs which run in the background of the web-client |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Which of the following statements is true regarding HTTP? | Web browsers use only HTTP as a communication protocol with servers | It does not maintain any connection information on previous transactions | It is designed to route information based on content | It refers to resources using their Universal Resource Identifier (URI) | It does not carry browser information to the server. | | It does not maintain any connection information on previous transactions |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Well formed XML document means | It contains a root element | It contain an element | It contains one or more elements | Must contain one or more elements and root element must contain all other elements | | | Must contain one or more elements and root element must contain all other elements |
| Microsoft XML schema data types for hexadecimal digits representing octates | UID | UXID | UUID | XXID | | | UUID |
| Which of the following statements is not true about XML Schemas: | They are used to define the content and structure of data. | They define a set of symbols and the relationships of those symbols. | They are themselves XML documents. | They have their own syntax. | | | They have their own syntax. |
| ADO.NET provides the ability to create and process in-memory databases called: | views | relations. | tables. | datasets. | | | datasets. |
| A report generated by a reporting system is delivered to the | Push | Pull | RFM | OLAP | | | Pull |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| appropriate users via a user-accessed Web site. This system uses which of the following report modes? | | | | | | | |
| Before use of DBMS information was stored using _____. | File Management System | Cloud Storage | Data System | None of These | | | File Management System |
| 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 |
| 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 |
| A recursive relationship is a relationship between an entity and _____ . | itself | a subtype entity | an archetype entity | an instance entity | | | itself |
| A characteristic of a file server is which of the following? | Manages file operations and is shared on a network. | Manages file operations and is limited to one PC. | Acts as a fat client and is shared on a network. | Acts as a fat client and is limited to one PC. | | | Manages file operations and is shared on a network. |
| Client/server security includes which of the following? | Physical security | Change control security | Measures taken to secure all systems | All of these | | | All of these |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| An application program interface (API) is which of the following? | The same thing as ODBC. | Middleware that does not provide access to a database. | Middleware that provides access to a database. | The same thing as JDBC. | | | Middleware that provides access to a database. |
| A server cannot serve on which of the following levels? | Client Layer | Business Layer | Database Layer | All of the above. | | | Client Layer |
| A database server is responsible for which of the following? | Database storage | Data processing logic | Data presentation logic | All of the above. | | | Database storage |
| A transitive dependency is which of the following? | A functional dependency between two or more key attributes. | A functional dependency between two or more nonkey attributes. | A relation that is in first normal form. | A relation that is in second normal form. | | | A functional dependency between two or more nonkey attributes. |
| If no multivalued attributes exist and no partial dependencies exist in a relation, then the relation is in what normal form? | First normal form | Second normal form | Third normal form | Fourth normal form | | | Second normal form |
| A transaction for which all committed changes are permanent is called: | atomic. | consistent. | isolated. | durable. | | | durable. |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| What is the overall responsibility of the DBA? | Facilitate the development and use of the database | Create and populate tables | Development, operation, and maintenance of the database and its applications | Both the first and third answers above are correct. | | | Both the first and third answers above are correct. |

## UNIT-IV

| questions | opt1 | opt2 | opt3 | opt4 | opt5 | opt 6 | answer |
|---|---|---|---|---|---|---|---|
| What is Hypertext Transfer Protocol (HTTP)? | The protocol to copy files between computers | The transfer protocol to transfer Web pages to a browser | The database access protocol for SQL statements | | The hardware/software protocol that limits access to company data | | The transfer protocol to transfer Web pages to a browser |
| Common Gateway Interface (CGI) is which of the following? | An interface that accepts and returns data that may be written in SQL. | An interface that accepts and returns data that may be written in any language that produces an executable file. | A small program that executes within another application and is stored on the server. | A small program that executes within another application and is stored on the client. | | | An interface that accepts and returns data that may be written in any language that produces an executable file. |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| An intrusion detection system does not perform which of the following? | Tries to identify attempts to hack into a computer system. | May monitor packets passing over the network. | May transmit message packets to the correct destination. | Set up deception systems that attempt to trap hackers. | | | May transmit message packets to the correct destination. |
| Features of XML include which of the following? | The tags are used to describe the appearance of the content. | Addresses the structuring and manipulation of the data involved. | Governs the display of information in a Web browser. | All of the above. | | | Addresses the structuring and manipulation of the data involved. |
| An intranet provides which of the following? | Connectivity to all customers | Connectivity to selected customers | Connectivity to no customers | Connectivity to all of the above | | | Connectivity to no customers |
| Client-side extensions: | add functionality to the server. | add functionality to the browser. | add functionality to the firewall. | add functionality to the network. | | | add functionality to the browser. |
| Web servers should be configured in which of the following ways? | unauthorized access is restricted. | unauthorized access is unrestricted. | unauthorized access is impossible. | unauthorized access is possible. | | | unauthorized access is restricted. |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| A proxy server is used for which of the following? | To provide security against unauthorized users | To process client requests for Web pages | To process client requests for database access | To provide TCP/IP | | | To provide security against unauthorized users |
| Which of the following is true concerning the standard network protocol for connecting a client to a Web or database server via the Internet? | Both TCP and IP are needed. | TCP is needed but IP is not needed. | TCP is not needed but IP is needed. | Neither TCP and IP are needed. | | | Both TCP and IP are needed. |
| Which of the following is true about Microsoft Active Server Pages (ASP)? | It is stored in .cfm files. | It is executed on the client. | It does not use tags. | It allows for coding of custom tags in an HTML file. | | | It allows for coding of custom tags in an HTML file. |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Websites without a database attached to them have which of the following characteristics? | The ability to generate data via SQL. | The inability to use a browser to display Web pages. | Static information using HTML or JavaScript. | The need to use TCP/IP as the network protocol. | | | Static information using HTML or JavaScript. |
| The @active data warehouse architecture includes which of the following? | At least one data mart | Data that can extracted from numerous internal and external sources | Near real-time updates | All of the above. | | | All of the above. |
| To run a compiled Java program, the machine must have what loaded and running? | Java virtual machine | Java compiler | Java bytecode | A Web browser | | | Java virtual machine |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| _____ is an open source DBMS product that runs on UNIX, Linux and Windows. | MySQL | JSP/SQL | JDBC/SQL | Sun ACCESS | | | MySQL |
| What is sent to the user via HTTP, invoked using the HTTP protocol on the user's computer, and run on the user's computer as an application? | A Java application | A Java applet | A Java servlet | None of the above is correct. | | | A Java applet |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| What programming language(s) or scripting language(s) does Java Server Pages (JSP) support? | VBScript only | Jscript only | Java only | All of the above are supported | | | Java only |
| What is bytecode? | Machine-specific code | Java code | Machine-independent code | None of the above is correct. | | | Machine-independent code |
| JDBC stands for: | Java Database Connectivity | Java Database Components | Java Database Control | None of the above is correct. | | | Java Database Connectivity |
| How does Tomcat execute a JSP? | As a CGI script | As an independent process | By one of Tomcat's threads | None of the above is correct. | | | By one of Tomcat's threads |
| A database management system (DBMS) is a: | hardware system used to create, maintain, and provide controlled access to a database. | hardware system used to create, maintain, and provide uncontrolled access to a database. | software system used to create, maintain, and provide controlled access to a database. | software system used to create, maintain, and provide uncontrolled access to a database. | | | software system used to create, maintain, and provide controlled access to a database. |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| An enterprise data model is: | a graphical model that shows the high-level entities for an organization. | a graphical model that shows all entities for an organization. | a non-graphical model that shows the high-level entities for an organization. | a non-graphical model that shows the all entities for an organization. | | | a graphical model that shows the high-level entities for an organization. |
| Which of the following types of databases are the most common? | Personal | Workgroup | Department | Enterprise | | | Department |

## UNIT-V

| questions | opt1 | opt2 | opt3 | opt4 | opt5 | opt6 | answer |
|---|---|---|---|---|---|---|---|
| If the XML data instance conforms to the DTD, the document is said to be: | type-invali | type-vali | not-type-vali | an HTML document. | | | type-vali |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| XML is: | a subset of SGML only. | a hybrid of document processing and database processing only. | a standardized yet customizable way to describe the content of documents only. | XML is all of the above. | | | XML is all of the above. |
| The document that is used by XSLT to indicate how to transform the elements of the XML document to another format is a(n): | HTML page. | DOCTYPE procedure. | stylesheet. | stored procedure. | | | stylesheet. |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| If an XML document does not have a DTD, then by definition it is: | not-type-vali | type-vali | an HTML document. | None of the above is correct. | | | not-type-vali |
| A distributed database has which of the following advantages over a centralized database? | Software cost | Software complexity | Slow Response | Modular growth | | | Modular growth |
| A transaction manager is which of the following? | Maintains a log of transactions | Maintains before and after database images | Maintains appropriate concurrency control | All of the above. | | | All of the above. |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Location transparency allows for which of the following? | Users to treat the data as if it is at one location | Programmers to treat the data as if it is at one location | Managers to treat the data as if it is at one location | All of the above. | | | All of the above. |
| Some of the columns of a relation are at different sites is which of the following? | Data Replication | Horizontal Partitioning | Vertical Partitioning | Horizontal and Vertical Partitioning | | | Vertical Partitioning |
| Storing a separate copy of the database at multiple locations is which of the following? | Data Replication | Horizontal Partitioning | Vertical Partitioning | Horizontal and Vertical Partitioning | | | Data Replication |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Which of the following is a disadvantage of replication? | Reduced network traffic | If the database fails at one site, a copy can be located at another site. | Each site must have the same storage capacity. | Each transaction may proceed without coordination across the network. | | | Each site must have the same storage capacity. |
| A distributed database can use which of the following strategies? | Totally centralized at one location and accessed by many sites | Partially or totally replicated across sites | Partitioned into segments at different sites | All of the above | | | All of the above |
| Which of the following is not one of the stages in the evolution of distributed DBMS? | Unit of work | Remote unit of work | Distributed unit of Work | Distributed request | | | Unit of work |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| The @active data warehouse architecture includes which of the following? | At least one data mart | Data that can extracted from numerous internal and external sources | Near real-time updates | All of the above. | | | All of the above. |
| A goal of data mining includes which of the following? | To explain some observed event or condition | To confirm that data exists | To analyze data for expected relationships | To create a new data warehouse | | | To explain some observed event or condition |
| A data warehouse is which of the following? | Can be updated by end users. | Contains numerous naming conventions and formats. | Organized around important subject areas. | Contains only current dat | | | Organized around important subject areas. |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| A snowflake schema is which of the following types of tables? | Fact | Dimension | Helper | All of the above | | | All of the above |
| The generic two-level data warehouse architecture includes which of the following? | At least one data mart | Data that can extracted from numerous internal and external sources | Near real-time updates | All of the above. | | | Data that can extracted from numerous internal and external sources |
| Fact tables are which of the following? | Completely denoralized | Partially denoralized | Completely normalized | Partially normalized | | | Completely normalized |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Data transformation includes which of the following? | A process to change data from a detailed level to a summary level | A process to change data from a summary level to a detailed level | Joining data from one source into various sources of data | Separating data from one source into various sources of data | | | A process to change data from a detailed level to a summary level |
| The extract process is which of the following? | Capturing all of the data contained in various operational systems | Capturing a subset of the data contained in various operational systems | Capturing all of the data contained in various decision support systems | Capturing a subset of the data contained in various decision support systems | | | Capturing a subset of the data contained in various operational systems |
| A star schema has what type of relationship between a dimension and fact table? | Many-to-many | One-to-one | One-to-many | All of the above. | | | One-to-many |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| A multifield transformation does which of the following? | Converts data from one field into multiple fields | Converts data from multiple fields into one field | Converts data from multiple fields into multiple fields | All of the above | | | All of the above |
| ODBC minimum SQL grammar contains which of the following? | INSERT, UPDATE, DELETE only | Stored Procedures only | Literals for date, time and timestamp only | CREATE VIEW, DROP VIEW only | | | INSERT, UPDATE, DELETE only |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| When using ODBC, which of the following processes ODBC requests and submits specific SQL statements to a given type of data source? | Data source | Driver | Driver manager | DBMS | | | Driver |
| The ODBC core API consists of which of the following functions? | Commit or rollback transactions only | Connect to data sources with driver-specific information only | Connect to data sources only | Both 1 and 3 above are in the OBDC core API. | | | Both 1 and 3 above are in the OBDC core API. |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| _____ ___ is a simple object model that is easier to understand and use than OLE DB, and is frequently used for database applications. | ASP | XML | ODBC | ADO | | | ADO |
| ODBC core SQL grammar contains which of the following?: | INSERT, UPDATE, DELETE only | Stored procedures only | Full SELECT (includes subqueries) only | Both 1 and 3 above are contained in the ODBC core SQL. | | | Both 1 and 3 above are contained in the ODBC core SQL. |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| _____ ___ is considered to be one of the foundations of data access in the Microsoft world. | ODBC | OLE DB | JPCD | ADO | | | OLE DB |
| An ODBC data structure that identifies a database and the DBMS that processes it is called a(n): | data source. | driver manager. | driver. | API. | | | data source. |
| In OLE DB, actions that an object can perform are called: | properties. | collections. | methods. | abstractions | | | methods. |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| In ODBC, a file that can be shared among database users is called a: | system data source. | file data source. | user data source. | SQL text file. | | | file data source. |
| _____ is an object-oriented interface that encapsulates data-server functionality. | ODBC | OLE DB | JPCD | ADO | | | OLE DB |
| Data mining is best described as the process of | identifying patterns in data. | deducing relationships in data. | representing data | simulating trends in data. | | | identifying patterns in data. |
| Data used to build a data mining model. | validation data | validation data | test data | hidden data | | | validation data |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Database query is used to uncover this type of knowledge. | deep | hidden | shallow | multidimensional | | | shallow |
| Which of the following is not a characteristic of a data warehouse? | contains historical data SC | designed for decision support | stores data in normalized tables | promotes data redundancy | | | stores data in normalized tables SC |
| Which are the programs that locate and gather information on the web? | Web spider | Search engine | Autobots | Web crawlers | | | Web crawlers |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Which program is used to access information that is not accessible to web crawlers? | Advance web crawlers | Spider traps | Autobots | Deep web crawlers | | | Deep web crawlers |
| _____ __ and _____ __ are two measure of the effectiveness of an information retrieval system. | Precision, sysonyms | Homonyms, ontologies | Web crawler, spider | Precision, recall | | | Precision, recall |
| _____ __ deal with the prediction of value rather than a | Regression | Multiway splits | Recall | Precision | | | Regression |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| class. | | | | | | | |
| Which is a type of classifier that has been found to give very accurate classification across a range of application? | Binary split | Multiway split | Overfitting | Support vector machine | | | Support vector machine |
| Data in the real world is | incomplete | noisy | dirty | all the above | | | all the above |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| What is the most appropriate special association that indicates that multiple textbooks make up a course required reading list? | aggregation association | generalization association | n-ary association | reflexive association | | | n-ary association |
| What is the special association that indicates that one object can be broken down into multiple special cases? | composition association | generalization association | n-ary association | reflexive association | | | generalization association |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| A reflexive association is one where one class is | broken down into special cases | combined with multiple other classes | combined with one other class | linked back to itself | | | linked back to itself |
| . In relational database model, after conceptually designing your database, the information contained in a single class would be stored in a | database | field | property | table | | | table |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| An aggregation association is drawn using which symbol | a line which loops back onto the same table | small closed diamond at the end of a line connecting two tables | small open diamond at the end of a line connecting two tables | small triangle at the end of a line connecting the aggregated item and multiple component items | | | small open diamond at the end of a line connecting two tables |
| Why is it a good idea to use special associations when drawing class diagrams | It is necessary. It is impossible to draw some diagrams without using these special structures | Specialized tools are available to create these special associations | They convey more information about the underlying structure of the database, making them easier to understand | They hide the detail concerning the database structure, making them easier to understand | | | They convey more information about the underlying structure of the database, making them easier to understand |
| What is the special association that combines different items from multiple classes to | composition association | generalization association | n-ary association | reflexive association | | | composition association |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| build a new object ? | | | | | | | |
| A data wareh ouse is ___ | updated by end u sers | contains numerous na ming conventions and format | organized a round impor tant subject area s. | contains on ly current da ta | | | organized a round impor tant subject area s. |
| An operat ional syste m is _____ | used to run the busin ess in real time and is based on curren t data | used to support decision making and is based on current data | used to support deci sion making and is based on historical data | used to support deci sion making and is based on historical data | | | used to support decision making and is based on current data |
| Data cleani ng is | Large collectio n of data mostly stored in a computer system | The removal of noise errors and incorrect input from a database | The systematic description of the syntactic structure of a specific database | None of these | | | The removal of noise errors and incorrect input from a database |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Data dictionary is | Large collection of data mostly stored in a computer system | The removal of noise errors and incorrect input from a database | The systematic description of the syntactic structure of a specific database. | None of these | | | The systematic description of the syntactic structure of a specific database. |
| Decision support systems are essential for | Day–to-day operation of an organization. | Providing statutory information. | Top level strategic decision making. | Ensuring that organizations are profitable. | | | Top level strategic decision making. |
| Decision support systems are used by | Line managers. | Top-level managers. | Middle level managers. | System users | | | Top-level managers. |
| Decision support systems are used for | Management decision making | Providing tactical information to management | Providing strategic information to management | Better operation of an organization | | | Providing strategic information to management |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Which is the most unlikely function of a marketing division of an organization. | advertising | sales analysis | order processing | customer preference analysis | | | order processing |