**Solution SET-A**

 **Subject: -ADBMS**

 **IInd Year MCA. III Semester I st Mid Term Examination, Oct. – 2018**

Ans 1 a) An object database is a database management system in which information is represented in the form of objects as used in object-oriented programming.

b) An OODBMS is aimed at applications where an object-centric viewpoint is appropriate

An ORDBMS is optimized for applications in which large data collections are the focus, even though objects may have a rich structure and be fairly large,

c) Atomicity: A database follows the all or nothing rule, i.e., the database considers all transaction operations as one whole unit or atom. Thus, when a database processes a transaction, it is either fully completed or not executed at all.
Consistency: Ensures that only valid data following all rules and constraints is written in the database. When a transaction results in invalid data, the database reverts to its previous state, which abides by all customary rules and constraints.

d) An **object-oriented database** management system (**OODBMS**) is a database management system that supports the creation and modeling of data as objects.**OODBMS** also includes support for classes of objects and the **inheritance** of class properties, and incorporates methods, subclasses and their objects.

e) A **nested table** is one **table** placed inside of another, where the larger **table**functions as a container for the smaller one. **Nested tables** are a way for you to organize objects, such as images or text, in evenly spaced rows and columns.

Ans 2 a) Advantages of Object Oriented DBMSs (OODBMS)

1)Enhanced modeling capabilities  2)Extensibility
3)Removal of impedance mismatch  4)Expressive power
 5)Support for schema evolution 6)Support for long-duration transactions
 7)Applicability to advanced database applications  8)Improved performance  8)Reusability

b) **Homogeneous Database:-** In a homogeneous distributed database, all the sites use identical DBMS and operating systems. Its properties are −

• The sites use very similar software.

• The sites use identical DBMS or DBMS from the same vendor.

• Each site is aware of all other sites and cooperates with other sites to process user requests.

• The database is accessed through a single interface as if it is a single database.

Ans 3 a)” A logically interrelated collection of shared data (and a description of this data) physically distributed over a [computer](http://ecomputernotes.com/fundamental/introduction-to-computer/what-is-computer) network."

A DDBMS has the following characteristics:

1. A collection of logically related shared data;
2. The data is split into a number of fragments;
3. Fragments may be replicated;
4. Fragments/replicas are allocated to sites;
5. The sites are linked by a communications network;
6. The data at each site is under the control of a DBMS;
7. The DBMS at each site can handle local applications, autonomously;
8. Each DBMS participates in at least one global application

b) **DATA FRAGMENTATION**

Data fragmentation allows you to break a single object into two ormore segments or fragments. The object might be a user’s database, a system database, or a table. Each fragment can be stored at any site over a computer network. Information about data fragmentation is stored in the distributed data catalog (DDC), from which it is accessed by the TP to process user requests.

There are three types of data fragmentation strategies:

• **Horizontal fragmentation** refers to the division of a relation into subsets (fragments) of tuples (rows). Each fragment is stored at a different node, and each fragment has unique rows. However, the unique rows all have the same attributes (columns). In short, each

fragment represents the equivalent of a SELECT statement, with the WHERE clause on a single attribute.

• **Vertical fragmentation** refers to the division of a relation into attribute(column)subsets. Each subset (fragment) is stored at a different node, and each fragment has unique columns—with the exception of the key column, which is common to all fragments.

• **Mixed fragmentation** refers to a combination of horizontal and vertical strategies. In other words, a table may be divided into several horizontal subsets (rows), each one having a subset of the attributes (columns).

Ans 4- Object oriented database layers architecture :-the six layer architecture model for object oriented data model.

 1. Interaction layer The interaction layer is first layer of Six Layers Architecture Model for Object Oriented Databases. In this layer, user can interact with the databases. The user can send the data to databases as well as data can be retrieved from database to user.

2.A pplication- is the second layer in this model. making that assumption, if by chance an early adopter choose an ODB who's architecture was ill suited for their applications needs, the reasoning lead immediately to the conclusion that no ODB was suited to solve their needs.

3 Administration Layer -This layer is responsible for management of administrative information. This layer can change responsibility as per requirement.

4 Security Layer -The security layer play important role in this model. The security layer is responsible to provide the full security to data and also provide to the security of application used to manage the data also.

5 Virtual Layer - the main advantage in their approach is that the memory requirement of each slice processor is very small and is independent of input size. In this model the virtual layer manage the data virtually. This time the large volume of data are managed. The concept of virtual is to put the data outside the memory.

6 Paging -The paging layer is responsible to divide the data in the form of pages. The pages are managed easily. The data are divided into pages as the same size of page frame; the page frame is that dividing memory in equal number of partitions. In this way large volume of data can be managed efficiently.

 OR

**Distributed concurrency control**

Concurrency control in DBMS ensures the parallel execution of transaction without interleaving the transaction. In DDBMS, we implement concurrency control assuming few points. The assumptions are stated as follows:

* Each site in DBMS participates in a transaction and commits the transaction from the respective sites.
* All the replicas present in DDBMS are updated

**Single-lock Manager Approach**

Single – lock manager approach states that there should be one dedicated site in DDBMS that will manage all the locks, which are requested by the other sites in DBMS.

distributes the transaction in all the other site distributes the transaction in all the other remaining sites in the network. Then, when one partial transaction in one site needs to lock a data item Q. It sends a lock request to sites LS(0). The data item Q can be locked depending on the priority of the request. If the priority is low. Then the lock request can be delayed until the request can be granted

Distributed lock manager Approach :-the lock managers reside at every site and they control the locking operation of their own site. The distribution of lock manager helps to avoid the point of failure, thus making the system more robust to failure. However, deadlock management in this approach is quite difficult to handle.

Two protocols to implement distributed lock manger approach:

1) Majority protocol

2) Biased protocol

**Majority Protocol** :- The name majority comes from the fact majority protocol handles majority of replica of a item Q, which is required to be locked by some transaction. This protocol works on the replica that is replicated and also the data that is not replicated. We will describe this protocol for the aforementioned two conditions as follows:-

* Since this protocol follows distributed-lock manager approach, all the sites in the network consisting DDBMS have their own lock manager who is responsible for locking the data and also unlocking the data stored at their own site.
* In case of data that has no replicas present in the network and a transaction wants to lock that data item. Say Q, stored at site S(i), the requesting site sends a message to S(i) starting the operation. The request cannot be granted until the state of Q is incompatible whenever the lock request is granted by the lock manager at site S(i), a message has been sent to the requesting node starting the success of the operation.
* In case of data Q that has replicas stored at n different sits in the network, whenever lock request is sent to any site that stores Q, that request needs to be sent to more than n sites holding the replicas of Q. This is the most important task in this protocol because until the majority of the sites holding Q are not being informed about the locking operation, the transaction does not operate on Q. Another point that needs to follow this case is whenever the WRITE operation takes place in any replicas of Q, all the replica should also write the updated value of Q.

 **2. Biased Protocol** :-

As we know that in case of the READ operation, we generally use shared lock instead of exclusive lock, but for WRITE operation, only exclusive lock is used. Unlike majority protocols, biased protocol uses shared lock to READ a data item on a single site. Similar to majority protocol, in biased protocol. All the sites have their own lock manager who can lock and unlock the data items stored in his/her own site.

When a data item Q needs to be locked by some transaction by executing shared lock, that initiator site should send a request to the site holding the data item. No need to send the lock request message to all the other sites holding the replicas of Q. When a data item Q needs locked by some transaction by executing exclusive lock, the initiator site must send the request message to all the other sites holding the data item Q.

 SET-B

 Ans 1. a) Need a object database when you have a business need for high performance on complex data. It is a good when you have all three factors: business need, high performance, and complex data.

b) **Dynamic binding** also called **dynamic** dispatch is the process of linking procedure call to a specific sequence of code (method) at run-time. ... **Dynamic binding** is also known as **late binding** or run-time **binding**. **Dynamic binding** is an object oriented programming concept and it is related with polymorphism and inheritance.

Encapsulation:- is Binding the data with the code that manipulates it. It keeps the data and the code safe from external interference

c) The basic **difference between** two languages is that **SQL** executes the single query at a time where as, **Pl**/**SQL** executes the block of code at once. **SQL** is a Structured Query Language where as, **PL**/**SQL** is a Procedural Language/ Structured Query Language.

d) The basic user-**defined** data **type** in C is the **structur** , or **struct**. (C **structures** are analogous to the records found in some other languages.) ... Since a **complex** number consists of a real and imaginary part, we need a way of holding both these quantities in one data **type**, and a **structure** will do just the trick.

e) A **distributed transaction** is a database **transaction** in which two or more network hosts are involved. Usually, hosts provide **transactional** resources, while the **transaction** manager is responsible for creating and managing a global **transaction** that encompasses all operations against such resources.

## Ans 2 a) Advantages of ORDBMSs

There are following advantages of ORDBMSs:

**Reuse and Sharing:** The main advantages of extending the Relational data model come from reuse and sharing. Reuse comes from the ability to extend the DBMS server to perform standard functionality centrally, rather than have it coded in each application.

**Increased Productivity:** ORDBMS provides increased productivity both for the developer and for the, end user

**Use of experience in developing RDBMS:** Another obvious advantage is that .the extended relational approach preserves the significant body of knowledge and experience that has gone into developing relational applications.

b) **Heterogeneous Database**

In a heterogeneous distributed database, different sites have different operating systems, DBMS products and data models.

Its properties are −

• Different sites use dissimilar schemas and software.

• The system may be composed of a variety of DBMSs like relational, network, hierarchical or object oriented.

• Query processing is complex due to dissimilar schemas.

• Transaction processing is complex due to dissimilar software.

Ans 3 a) **The basic types of distributed DBMS are as follows:**

1. Client-server architecture of Distributed system.

* A client server architecture has a number of clients and a few servers connected in a network.
* A client sends a query to one of the servers. The earliest available server solves it and replies.
* A Client-server architecture is simple to implement and execute due to centralized server system.



2. Collaborating server architecture.

* Collaborating server architecture is designed to run a single query on multiple servers.
* Servers break single query into multiple small queries and the result is sent to the client.
* Collaborating server architecture has a collection of database servers. Each server is capable for executing the current transactions across the databases.



3. Middleware architecture.

* Middleware architectures are designed in such a way that single query is executed on multiple servers.
* This system needs only one server which is capable of managing queries and transactions from multiple servers.
* Middleware architecture uses local servers to handle local queries and transactions.
* The softwares are used for execution of queries and transactions across one or more independent database servers, this type of software is called as middleware.

b) Replicated data are subject to the mutual consistency rule. The **mutual consistency**

**rule** requires that all copies of data fragments be identical. Therefore, to maintain data

consistency among the replicas, the DDBMS must ensure that a database update is

performed at all sites where replicas exist.

Three replication scenarios exist :

• A **fully replicated** database stores multiple copies of each database fragment at

multiple sites. In this case , all database fragments are replicated. A fully replicated

database can be impractical due to the overhead it imposes on the system.

• A **partially replicated** database stores multiple copies of some database fragments

at multiple sites. Most DDBMSs are able to handle the partially replicated database

well.

• An **unreplicated** database stores each database fragment at a single site. Therefore,

there are no database fragments.

## Ans 4. Features of OODBMS :-In OODBMS, every entity is considered as object and represented in a table. Similar objects are classified to classes and subclasses and relationship between two object is maintained using concept of inverse reference.Some of the features of OODBMS are as follows:1. ComplexityOODBMS has the ability to represent the complex internal structure (of object) with multilevel complexity.2. InheritanceCreating a new object from an existing object in such a way that new object inherits all characteristics of an existing object.3. EncapsulationIt is an data hiding concept in OOPL which binds the data and functions together which can manipulate data and not visible to outside world.4. PersistencyOODBMS allows to create persistent object (Object remains in memory even after execution). This feature can automatically solve the problem of recovery and concurrency.

 OR

The different distributed commit protocols are −

* One-phase commit
* Two-phase commit
* Three-phase commit

Distributed One-phase Commit

Distributed one-phase commit is the simplest commit protocol. Let us consider that there is a controlling site and a number of slave sites where the transaction is being executed. The steps in distributed commit are −

* After each slave has locally completed its transaction, it sends a “DONE” message to the controlling site.
* The slaves wait for “Commit” or “Abort” message from the controlling site. This waiting time is called **window of vulnerability**.
* When the controlling site receives “DONE” message from each slave, it makes a decision to commit or abort. This is called the commit point. Then, it sends this message to all the slaves.
* On receiving this message, a slave either commits or aborts and then sends an acknowledgement message to the controlling site.

Distributed Two-phase Commit

Distributed two-phase commit reduces the vulnerability of one-phase commit protocols. The steps performed in the two phases are as follows −

**Phase 1: Prepare Phase**

* After each slave has locally completed its transaction, it sends a “DONE” message to the controlling site. When the controlling site has received “DONE” message from all slaves, it sends a “Prepare” message to the slaves.
* The slaves vote on whether they still want to commit or not. If a slave wants to commit, it sends a “Ready” message.
* A slave that does not want to commit sends a “Not Ready” message. This may happen when the slave has conflicting concurrent transactions or there is a timeout.

**Phase 2: Commit/Abort Phase**

* After the controlling site has received “Ready” message from all the slaves −
	+ The controlling site sends a “Global Commit” message to the slaves.
	+ The slaves apply the transaction and send a “Commit ACK” message to the controlling site.
	+ When the controlling site receives “Commit ACK” message from all the slaves, it considers the transaction as committed.
* After the controlling site has received the first “Not Ready” message from any slave −
	+ The controlling site sends a “Global Abort” message to the slaves.
	+ The slaves abort the transaction and send a “Abort ACK” message to the controlling site.
	+ When the controlling site receives “Abort ACK” message from all the slaves, it considers the transaction as aborted.

Distributed Three-phase Commit

The steps in distributed three-phase commit are as follows −

**Phase 1: Prepare Phase**

The steps are same as in distributed two-phase commit.

**Phase 2: Prepare to Commit Phase**

* The controlling site issues an “Enter Prepared State” broadcast message.
* The slave sites vote “OK” in response.

**Phase 3: Commit / Abort Phase**

The steps are same as two-phase commit except that “Commit ACK”/”Abort ACK” message is not required.