CS1301 –Data Base Management Systems

1. What is DBMS and what's the goal of it?

   DBMS is a collection of interrelated data and a set of programs to access that data. The goal is to provide the environment that is both convenient and efficient to use in retrieving and storing data base information.

2. What are the advantages of DBMS?
   - Centralized Control
   - Data Independence allows dynamic change and growth potential.
   - Data Duplication is eliminated with control Redundancy.
   - Data Quality is enhanced.

3. What are the Disadvantages of DBMS?
   - Cost of software, hardware and migration is high.
   - Complexity of Backup
   - Problem associated with centralization.

4. What are the disadvantages of File Systems?
   - Data Redundancy and Inconsistency.
   - Difficulty in accessing data.
   - Data Isolation
   - Integrity problems
   - Security Problems
   - Concurrent access anomalies

5. Define Entity, Entity Set, and extensions of entity set. Give one example for each.
   - Entity – object or thing in the real world. Egs., each person, book, etc.
   - Entity set – set of entities of the same entity that share the same properties or attributes. Eg., Customer, loan, account, etc
   - Extensions of entity set – individual entities that constitute a set. Eg., individual bank customers

6. Define and give examples to illustrate the four types of attributes in database.
   - Simple and Composite attribute
     - Simple (address)
     - Composite (street, city, district)
   - Single-valued and multi-valued attribute
     - Single-valued (roll no)
     - Multi-valued (colors = {R,B,G})
   - Null attribute
     - No values for attribute
   - Derived attribute
     - Age derived from Birth date

7. Define relationship and participation.
    - Relationship is an association among several entities.
    - Association between entity sets is referred as participation.

8. Define mapping cardinality or cardinality ratio.

    Mapping cardinality or cardinality ratio is the way to express the number of entities to which another entity can be associated through a relationship set. These are most useful in describing binary relationship sets.

9. Explain the four types of mapping cardinality with example.

    For a binary relationship set R between entity sets A and B, the mapping cardinality must be one of the following: (Draw the diagrams also)
    - 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. Eg., Roll no entity in Student info entity set and marks details entity set.
    - One-to-many

        An entity in A is associated with any number of entities in B. But an entity in B is associated with at most one entity in A. Eg., one customer with many loans.
    - Many-to-one

        An entity in A is associated with at most one entity in B. But an entity in B can be associated with any number of entities in A. Eg., street and city associated to a single person.
    - Many-to-many

        An entity in A is associated with any number of entities in B. But an entity in B can be associated with any number of entities in A. Eg., same loan by several business partners.

10. Differentiate total participation and partial participation. (i.e., write Definition and Example with illustration)
    - Total – participation of an entity set, E in relationship, R is total if every entity in E participates in at least one relationship in R. Eg., loan entity set.
    - Partial – if only some entities in E participate in R. Eg., payment weak entity set.

11. Define E-R diagram.

    Overall structure of a database can be expressed graphically by E-R diagram for simplicity and clarity.

12. Define weak Entity set. Give an example and explain why it is weak entity set.
    - Entity set with no sufficient attributes to form a primary key.
        - Payment entity set is weak since duplication exists

13. Define discriminator or partial key of a weak entity set. Give example.

    Set of attributes that allow distinction to be made among all those entities in the entity set that depend on one particular strong entity. Eg., payment_no in payment entity set. It is also called as partial key.

14. Explain Referential Integrity.

    Referential Integrity means relationship between tables. Foreign keys are used. Foreign key is the column whose values are derived from the Primary key of the same or some other table.

    Format for creating a foreign key is given below.

    Syntax: create table <table name>(columnname data type (size) constraint constraint_name **references** parent table name);

15. Define Instances and schemas.

    Instances:

    The collection of information stored in the database at a particular moment is called the instance of the database.

    Schemas:

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

16. Define and explain the two types of Data Independence.

    Two types of data independence are

    i. Physical data independence
    ii. Logical data independence

    Physical data independence:

    The ability to modify the physical schema without causing application programs to be rewritten in the higher levels. Modifications in physical level occur occasionally whenever there is a need to improve performance.

    Logical data independence:

    The ability to modify the logical schema without causing application programs to be rewritten in the higher level (external level). Modifications in physical level occur frequently more than that in physical level, whenever there is an alteration in the logical structure of the database.

17. Define transaction.

    A transaction is a collection of operations that performs a single logical function in a database application. Each transaction is a unit of both atomicity and consistency. Properties of transaction are atomicity, consistency, isolation, and durability.

18. Define the type types of DML.

    Two types:

    - Procedural DML
    - Non-procedural DML

Procedural DML:

It requires a user to specify what data are needed and how to get those data.

Non-Procedural DML:

It requires a user to specify what data are needed without specifying how to get those data.

19. List out the functions of DBA.
    - Schema definition
    - Storage structure and access-method definition
    - Schema and physical modification
    - Granting of authorization for data access
    - Integrity constraint specification

20. What is the need for DBA?

    The need of DBA is to have central control of both the data and the programs that access those data. The person who has such central control over the system is called the DataBase Administrator.

21. Explain DML pre-compiler.

    DML precompiler converts DML statements embedded in an application program to normal procedure calls in the host language. The precompiler must interact with the DML compiler to generate the appropriate code.

22. Define file manager and buffer manager.

    File manager:

    File manager manages the allocation of space on disk storage and the data structures used to represent information stored on disk.

    Buffer manager:

    Buffer manager is responsible for fetching data from the disk storage into the main memory, and deciding what data to cache in memory.

23. Define Data Dictionary.

    DDL statements are compiled into a set of tables that is stored in a special file called data dictionary. Data dictionary contains the meta-data, which in turn is data about the data.

24. Give the syntax for creating the table with composite primary key.

    Multicolumn primary key is called composite primary key.

    Syntax: create table <table name>(columnname1 data type (size), columnname2 data type (size), constraint name **primary key** (columnname1, columnname2));

25. Write a query to display loan number, branch name where loan amount is between 500 and 1000 using comparison operators.
    Query: select loan no, branch name from loan where amount>=500 and amount<=1000;

26. Find the names of all branches with customers who have an account in the bank and who live in the Harrison city using Equi-join.
    Query: select branch_name from customer,account,depositor where cust_city='harrison' and customer.cust_name=depositor.cust_name and depositor.acc_no=account.acc_no;

27. Find the names of all branches with customers who have an account in the bank and who live in the Harrison city using Sub-Queries.
    Query: select branch_name from account where acc_no in(select acc_no from depositor where cust_name in(select cust_name from customer where cust_city='harrison'));

28. Select the rows from borrower such that the loan numbers are lesser than any loan number where the branch name is Downtown.
    Query: select * from borrower where loan_no < any (Select loan_no from loan where branch_name='downtown');

29. Define self-join and give an example query to illustrate self-join with a sample table.
    Joining of a table to itself is called self-join. i.e., it joins one row in a table to another row.
    Query: select emp.emp_name, mngr.emp_name from employ emp, employ mngr where emp.manager_no=mngr.emp_no;

    Sample table: Employ

Emp_no	Emp_name	Manager_no
E001	Basu navindgi	E002
E002	Rukmini	E005
E003	Carol	E004
E004	Cynthia	-
E005	Ivan	-

30. What is a view and give an example query to create a view from an existing table.
    Any relation that is not a part of the logical model but which is made visible to the user as a virtual (imaginary) relation is called a view.

    Query: create **view** custall (name, city) as (Select cust_name, cust_city from customer);

31. Define Degree and Domain of a relation.

Degree:

Number of attributes 'n' of its relation schema is called a degree of a relation. eg. Account table degree is 3, since three attributes are there in that relation.

Domain:

Set of permitted values for each attribute (or) data type describing the types of values that can appear in each column is called a domain of a relation. eg. Set of all account numbers of the account table.

32. Define how a relation is defined mathematically.

A relation is defined mathematically as a subset of a Cartesian product of a list of domains.

For eg., in account table,

$D_1 \rightarrow$ set of all acc_nos

$D_2 \rightarrow$ set of all branch_names

$D_3 \rightarrow$ set of all balances

And the relation account is a subset of $D_1 \times D_2 \times D_3$

33. Define super key and give example to illustrate the super key.

Set of one or more attributes taken collectively, allowing to identify uniquely an entity in the entity set.

Eg1. {SSN} and {SSN, Cust_name} of customer table are super keys.

Eg2. {Branch_name} and {Branch_name, Branch_city} of Branch table are super keys.

34. Define candidate key and give example to illustrate the candidate key.

Super keys with no proper subset are called the candidate keys. Otherwise it is called minimal super key. Candidate key is nothing but the primary key used in SQL.

Eg1. {SSN} is the candidate key for the super keys {SSN} and {SSN, Cust_name} of customer table.

Eg2. {Branch_name} is the candidate key for the super keys {Branch_name} and {Branch_name, Branch_city} of Branch table.

35. List out the six fundamental operators and 4 additional operators in relational algebra.

Six Fundamental operators:

- Selection (σ)
- Projection (π)
- Union ($\cup$)
- Set Difference (-)
- Cartesian Product (X)
- Rename (ρ)

Four Additional operators:

- Set Intersection ($\cap$)
- Natural Join (*)
- Division ($\div$)
- Assignment ($\leftarrow$)

36. Which operators are called as unary operators and explain why they are called so.

Unary operators:

- Selection (σ)
- Projection (π)
- Rename (ρ)

These operators are called as unary operators because they operate on only one relation.

37. Which operators are called as binary operators and explain why they are called so.

Binary operators:

- Union ($\cup$)
- Set Difference (-)
- Cartesian Product (X)

These operators are called as binary operators because they operate on pairs of relations.

38. Write a relational algebra expression to find those tuples pertaining to loans of more than 1200 made by the Perryridge branch.

Relational algebra expression:

$\sigma_{branch\_name = \text{“perryridge”} \wedge amount > 1200}$ (loan)

39. Explain the use of set difference operator and give an example to illustrate the same.

Use of set difference operator:

Allows finding tuples that are in one relation but are not in another relation.

Example: Find all customers of the bank who have an account but not a loan.

Relational Algebra Expression:

π_{Cust_name} (depositor) - π_{cust_name} (borrower)

40. Explain the two conditions needed for the set difference operation (union operation) to be valid.

Two conditions are

- Relations, r and s must be of the same arity ie., they must have same number of attributes.
- Domains of the i^{th} attribute of r and the i^{th} attribute of s must be same for all i.

41. Explain with one example why the additional operators are separated from the fundamental operators?

Additional operators are used instead of fundamental operators to reduce the complexity of long relational algebra expressions.

Eg. $r \cap s = r - (r - s)$

Intersection can be used for repeated set difference operations.

42. Explain theta join.

Theta join is an extension to the natural join operation that allows combining a Selection operation with condition, θ and a Cartesian product into a single operation.

$r \bowtie_\theta s = \sigma_\theta (r \times s)$

$$r \div s = \pi_{R-S}(r) - \pi_{R-S}(\pi_{R-S}(r) \times s) - \pi_{R-S,S}(r))$$

43. Define how an expression can be expressed in tuple relational calculus and list out the notations used.

A query or expression can be expressed in tuple relational calculus as

$\{ t \mid P(t)\}$

which means the set of all tuples 't' such that predicate P is true for 't'.

Notations used:

- $t[A] \rightarrow$ the value of tuple 't' on attribute, A
- $t \in r \rightarrow$ tuple 't' is in relation 'r'
- $\exists \rightarrow$ there exists

  Definition for 'there exists' ($\exists$):

  $\exists t \in r(Q(t))$

  which means there exists a tuple 't' in relation 'r' such that predicate Q(t) is true.
- $\forall \rightarrow$ for all

  Definition for 'for all' ($\forall$):

  $\forall t \in r(Q(t))$

  which means Q(t) is true for all tuples 't' in relation 'r'.
- $\Rightarrow \rightarrow$ Implication

  Definition for Implication ($\Rightarrow$):

  $P \Rightarrow Q$ means if P is true then Q must be true.

44. Write an expression to find the names of all customers who have a loan from the Perryridge branch in tuple relational calculus.

Expression in tuple relational calculus:

$\{t \mid \exists\ s \in$ borrower $(t[cust\_name] = s[cust\_name] \wedge \exists\ u \in$ loan $(u[loan\_no] = s[loan\_no] \wedge u[branch\_name] =$"perryridge"$))\}$

45. Write an expression to find all customers who have an account at all branches located in Brooklyn in tuple relational calculus.

    $\{t \mid \forall u \in branch (u[branch\_city] = \text{“Brooklyn”} \Rightarrow \exists s \in depositor (t[cust\_name] = s[cust\_name] \wedge \exists w \in account (w[acc\_no] = s[acc\_no] \wedge w[branch\_name] = u[branch\_name]))) \}$

46. Define safety of expression in tuple relational calculus.

    Query or expression $\{ t \mid P(t)\}$ is safe if all values that appear in the result are values from dom(p), where dom(p) is domain of P. Else it is unsafe. For example, the tuples selected in the tuple relational calculus are unsafe.

47. Define domain relational calculus.

    Domain relational calculus uses domain variables that take on values from an attribute's domain rather than values for an entire tuple.

    An expression in the domain relational calculus is of the form

    $\{<x_1,x_n, \ldots x_n> \mid (P(x_1,x_n, \ldots x_n))\}$

    where $x_1,x_n, \ldots x_n$ represents domain variables.

    P represents a formula composed of atoms or domain variables.

48. Write an expression to find all customers who have an account, a loan or both at the Perryridge branch in domain relational calculus.

    $\{ <c> \mid \exists l (<c,l> \in borrower \wedge \exists b,a (<l,b,a> \in loan \wedge b = \text{“Perryridge”})) \vee \exists a (<c,a> \in depositor \wedge \exists b,bal (<a,b,bal> \in account \wedge b = \text{“Perryridge”}))\}$

49. Define and give the general format used for generalized projection. Give one example expression to illustrate the same.

    Generalized projection extends the projection operation by allowing arithmetic functions to be used in the projection list.

    General format used for Generalized projection is

    $\pi_{F1\ F2\ \ldots\ Fn}(E)$

    where

    E is the relational algebra expression.

    F1 F2 … Fn are the arithmetic expressions involving constants and attributes in the schema of E. Special case these can be simply an attribute or a constant.

    Example expression for illustration:

    $\pi_{acc\_no,\ branch\_name,\ balance\ +\ 100}\ (account)$

50. What is the use of outer join and list out the three types of outer join with the notations used in relational algebra?

    Natural join combines only the common columns. So some information will be lost, if it has no common column. So outer join is used. It avoids this loss of information.

    Three types of outer join:

    - Left outer join ($\supset\!\!\bowtie$)

- Right outer join (⟖)
- Full outer join (⟗)

51. Write a relational algebraic expression to delete all accounts at branches located in Brooklyn.

    $r1 \leftarrow \sigma_{branch\_city = \text{“Brooklyn”}} (account \bowtie branch)$
    $r2 \leftarrow \pi_{branch\_name,\ acc\_no,\ balance} (r1)$
    $account \leftarrow account - r2$
    where r1 and r2 are temporary relations.

52. Write a relational algebraic expression to insert the information about Smith with his new account number ‘A-157’ taken at the Perryridge branch with Rs.1200.
    Relational algebraic expression for insertion:

    $account \leftarrow account \cup \{(\text{“A-157”}, \text{“Perryridge”}, 1200)\}$
    $depositor \leftarrow depositor \cup \{(\text{“Smith”}, \text{“A-157”})\}$

53. Write a relational algebraic expression to insert a gift amount of Rs. 200/- for all loan customers of the Perryridge branch.
    Relational algebraic expression for insertion:

    $r_1 \leftarrow \sigma_{branch\_name = \text{“Perryridge”}} (borrower \bowtie loan)$
    $r_2 \leftarrow \pi_{branch\_name,\ loan\_no} (r_1)$
    $loan \leftarrow loan \cup (r_2 \times \{(200)\})$
    $borrower \leftarrow borrower \cup \pi_{cust\_name,\ loan\_no} (r_1)$

54. Write a relational algebraic expression to update by adding the balances more than 10000 with Rs.60/- interest and otherwise Rs. 50/- interest.
    Relational algebraic expression for updating:

    $account \leftarrow \pi_{branch\_name,\ acc\_no,\ balance \leftarrow balance + 60} (\sigma_{balance > 10000} (account)) \cup \pi_{branch\_name,\ acc\_no,\ balance \leftarrow balance + 50} (\sigma_{balance \leq 10000} (account))$

55. Define materialized views and explain the use of such views.
    Definitions:

    Certain database systems allow view relations to be stored, but they make sure that if the actual relations used in the view definition change then the view is kept up to date. Such views are called <u>materialized views</u>.

    The process of keeping views up to date is called <u>view maintenance</u>.

    Use of materialized views:

    If the views are used frequently then materialized views are used. But benefits of materialization must be weighed against the storage cost and the added overhead of updates.

56. Differentiate assertions and triggers. (i.e., write definitions)

    An assertion is a predicate expressing a condition that we wish the database to satisfy always.

A Trigger is a statement that is executed automatically by the system as a side effect of a modification to the database.

57. List out the two requirements of triggers in database.
    - Specify the conditions under which the trigger is to be executed.
    - Specify the actions to be taken when the trigger executes.

58. Define Functional Dependencies.

    Let $X \subseteq R$ and $Y \subseteq R$, then the Functional dependency (FD) $X \rightarrow Y$ holds on R if, in any relation r(R), for all pairs of tuples t_1 and t_2 in r such that $t_1[X] = t_2[X]$, it is also the case that $t_1[Y] = t_2[Y]$.

59. List out the Functional Dependencies in the following relation and explain why.

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_2	c_1	d_2
a_2	b_2	c_2	d_2
a_2	b_2	c_2	d_3
a_3	b_3	c_2	d_4

    - $A \rightarrow C$ holds, since for all tuples with same value for A holds in B too.
    - $AB \rightarrow D$ holds, same case.
    - $D \rightarrow B$ holds, same case

60. Define Closure of Functional Dependency.

    Let F be a set of FDs, then the closure of F is the set of all FDs logically implied by F. It is denoted by F^+

61. Define the three Armstrong's Axioms or rules of inference.
    - Reflexivity rule:
        - If α is a set of attributes and $Y \subseteq X$, then $X \rightarrow Y$ holds.
    - Augmentation rule:
        - If $X \rightarrow Y$ holds and Z is a set of attributes, then $ZX \rightarrow ZY$ holds.
    - Transitivity rule:
        - If $X \rightarrow Y$ holds and $Y \rightarrow Z$ holds, then $X \rightarrow Z$ holds.

62. Define union, decomposition, and Pseudo-transitivity rules.
    - Union rule:
        - If $X \rightarrow Y$ holds and $X \rightarrow Z$ holds, then $X \rightarrow YZ$ holds.
    - Decomposition rule:
        - If $X \rightarrow YZ$ holds, then $X \rightarrow Y$ holds and $X \rightarrow Z$ holds.
    - Pseudo-transitivity rule:
        - If $X \rightarrow Y$ holds and $ZY \rightarrow W$ holds, then $XZ \rightarrow W$ holds.

63. List all the F^+ using all the rules for the schema, R = (A, B, C, G, H, I) with the given Functional Dependencies 'F'

$A \rightarrow B$
$A \rightarrow C$
$CG \rightarrow H$
$CG \rightarrow I$
$B \rightarrow H$

- F^+ for the given F are
  - $A \rightarrow H$, transitivity rule
  - $A \rightarrow BC$, union rule
  - $CG \rightarrow HI$, union rile
  - $AG \rightarrow H$, pseudo-transitivity rule
  - $AG \rightarrow I$, pseudo-transitivity rule

64. Define normalization of data and denormalization.
    - Normalization:
      Process of decomposing an unsatisfactory relation schema into satisfactory relation schemas by breaking their attributes, so as to satisfy the desirable properties.
    - Denormalization:
      Process of storing the join of higher normal form relations as base relation, which is in the lower normal form.

65. Explain shortly the four properties or objectives of normalization.
    - To minimize redundancy
    - To minimize insertion, deletion, and updating anomalies.
    - Lossless-join or non-additive join decomposition
    - Dependency preservation

66. Define Partial Functional Dependency.
    A FD $x \rightarrow y$ is a partial dependency if some attribute $A \in x$ can be removed from x and the dependency still holds; ie., for some $A \in x$, $(x - \{ A \}) \rightarrow y$.

67. Define transaction-processing systems.
    The concept of transaction provides a mechanism for describing logical units or database processing. Transaction-processing systems are systems with large databases and many concurrent users work with those systems.

68. Define read only transaction.
    The database operations in a transaction do not update the database but only retrieve data and that transaction is called a read-only transaction.

69. When does the transaction go into an active state and partially committed state?

A transaction goes into an active state immediately after it starts execution, where it can issue read and write operations. When the transaction ends, it moves into the partially committed state.

70. What is called as committed state?

Once a transaction is committed, it has concluded its execution successfully and all its changes must be recorded permanently in the database.

71. Define ACID property.

A–Atomicity, maintained by transaction manager component.
C–Consistency, maintained by application programmer by coding with integrity constraints.
I – Isolation, maintained by concurrency control component.
D–Durability, maintained by recovery management component.

72. What is isolation of ACID properties?

The execution of a transaction should not be interfered with any other transactions executing concurrently.

73. Define cascading rollback.

An uncommitted transaction will be rolled back because of the failure of the first transaction, from which other transactions reads the data item. This phenomenon of wasting the desirable amount of work is called cascading rollback.

74. What is blind write?

If a transaction writes a data item without reading the data is called blind write. This sometimes causes inconsistency.

75. Define serial schedule.

A schedule, S is serial if for every transaction T participating in the schedule and all the operations of T are executed consecutively in the schedule; otherwise the schedule is called Non-serial schedule.

76. What is the use of locking?

It is used to prevent concurrent transactions from interfering with one another and enforcing an additional condition that guarantees serializability.

77. What is called as a time stamp?

A time stamp is a unique identifier for each transaction generated by the system. Concurrency control protocols use this time stamp to ensure serializability.

78. What is shared lock and Exclusive lock?

Shared lock allows other transactions to read the data item and write is not allowed. Exclusive lock allows both read and write of data item and a single transaction exclusively holds the lock.

79. When does a deadlock occur?

Deadlock occurs when one transaction T in a set of two or more transactions is waiting for some item that is locked by some other transaction in the set.

80. What is meant by transaction rollback?

If a transaction fails for reasons like power failure, hardware failure or logical error in the transaction after updating the database, it is rolled back to restore the previous value.

81. Write the reasons for using concurrent execution.

- ✓ High throughput

  Number of transactions that are executed in a given amount of time is high.
- ✓ Low delay

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

82. Define recoverable schedule.

Recoverable schedule is the one where for each pair of transactions T_i and T_j such that T_j reads a data item previously written by T_i, the commit operation of T_i appears before the commit operation of T_j.

83. Define Query optimization.

The DBMS must devise an execution strategy for retrieving the result of the query from the database files. Process of choosing a suitable execution strategy for processing a query is known as Query optimization.

84. Define Distributed database systems.

A distributed system consists of a collection of sites, connected together via some kind of communication network, in which

- Each site is a full database system site in its own right.
- But the sites have agreed to work together so that a user at any site can access data anywhere in the network exactly as if the data were all stored at the user's own site.

85. What are the advantages of distributed system?

- Reliability: the probability that the system is up and running at any given moment.
- Availability: the probability that the system is up and running continuously throughout a specific period.

86. What is query processor?

The objective of minimizing network utilization implies the query optimization process itself needs to be distributed as well as the query execution process.

87. Explain Client/Server.

Client/Server refers primarily to architecture, or logical division of responsibility, the client is the application called the front end. The server is the DBMS, which is the back end.

90. What are the rules that have to be followed during fragmentation?

The three correctness rules are:

- Completeness: ensure no loss of data
- Reconstruction: ensure Functional dependency preservation
- Disjoint ness: ensure minimal data redundancy.

91. What are the objectives of concurrency control?

- To be resistant to site and communication failure.
- To permit parallelism to satisfy performance requirements.
- To place few constraints on the structure of atomic actions.

92. What is multiple-copy consistency problem?

It is a problem that occurs when there is more than one copy of data item in different locations and when the changes are made only in some copies not in all copies.

93. What are the types of Locking protocols?

- Centralized 2 phase locking
- Primary 2 phase locking
- Distributed 2 phase locking
- Majority locking

94. What are the failures in distributed DBMS?

- The loss of message
- The failure of a communication link
- The failure of a site
- Network partitioning

95. What is replication?

The process of generating and reproducing multiple copies of data at one or more sites is called replication.

96. What are the types of replication?

- Read-only snapshots
- Updateable snapshots
- Multimaster replication
- procedural replication

97. What are the applications of Data warehouse?

- OLAP (Online Analytical Processing)
- DSS (Decision Support System)
- OLTP (On Line Transaction Processing)

98. What is parallel database?

In some of the architectures multiple CPUs are working in parallel and are physically located in a close environment in the same building and communicating at a very high speed. The databases operating in such environment is known as parallel databases.

99. Define Data warehousing.

Data warehouse is a "subject oriented, integrated, non-volatile, time variant collection of data in support of managements decisions".

100. Define data mining.

The data mining refers to the mining or discovery of new information in terms of patterns or rules from vast amount of data.

16 Marks

1. Explain Distributed Relational Database design in detail.
    - Objectives of data allocation and fragmentation
    - Data Allocation:
        - Strategic Objectives: Centralized, Fragmented, Complete Replication, Selective Replication.
    - Fragmentation:
        - Need for fragmentation
        - Correctness of fragmentation
        - Types of fragmentation

2. Explain about Transparencies in a DDBMS.
    - Distribution transparency
        - Fragmentation transparency
        - Location transparency
        - Replication transparency
        - Local mapping transparency
        - Naming transparency
    - Transaction transparency
        - Concurrency transparency
        - Failure transparency
    - Performance transparency
    - DBMS transparency.

3. Explain the Functions and Architecture of a DDBMS.
    - Functions of DDBMS
    - Reference Architecture for DDBMS
    - Reference Architecture for Federated MDBS
    - Component Architecture for DDBMS

4. Explain in detail the recovery techniques in distributed databases.
    - Failures in a distributed environment
    - How failures affect recovery
    - Distributed recover protocols
    - Two-phase commit (2PC)
    - Termination protocols
    - Recovery protocols
    - Election protocols
    - Communication topologies for 2PC
    - Three-phase commit (3PC)
    - Network partitioning

5. Discuss about the X/Open Distributed Transaction Processing model.

- X/Open established the Distributed Transaction Processing (DTP) working group with the objective of fostering appropriate programming interface for Transaction Processing.
- X/Open interfaces with diagram
- Transaction Manager
- Resource Manager
- Procedures of TX interface
- X/Open interfaces in a distributed environment

6. Explain about distributed query optimization.

- Query optimization.
- Techniques in query optimization
- Distributed query transformations
- Reconstruction Algorithms
- Generic relational algebra tree
- Reduction techniques for various types of fragmentation
- Distributed joins.

7. Explain Distributed Deadlock Management.

- What is Deadlock?
- What is Deadlock Management?
- Deadlock Detection.
- Diagram for Distributed Deadlock.
- Types of Deadlock Detection.

8. Explain Object oriented Concepts in database and storing objects in Relational Database.

Concepts:

- Abstraction, Encapsulation, Information Hiding.
- Objects and Attributes
- Object Identity
- Methods and Messages
- Classes
- Subclasses, Superclasses, Inheritance
- Overriding and Overloading
- Polymorphism and Dynamic Binding
- Complex Objects

Storing objects in Relational Database:

- Mapping Classes to Relations
- Accessing Objects in the Relational Database
- Explain with e.g.

9. Explain about issues in OODBMS.

3 problematic areas for RDBMS

a) Long duration transaction

b) Versions

c) Schema Evolution.

Architecture:

1.Client-Server:

2.Storing and executing methods:

10. Explain briefly on Object Oriented DBMSs-Standards and Systems.

- Object Management Group
  The common object request broker architecture
- Object Data standards ODMG 3.0,1999
  Object Data management group
  The Object model
  The Object Definition Language
  The Object Query Langage

11. Explain the advantages and disadvantages of OODBMS.

Advantages

Disadvantages

12. Discuss briefly about THE WEB?

i).Definition of WWW.

ii).Hyper Text Transfer Protocol

iii).Hyper Text Mark Up Language

iv).Uniform Resource Locator

v).Static and Dynamic WEB Page

13. Web-DBMS Architecture

i)Traditional two-tier client-server architecture

ii)Three-tier architecture

iii) Advantages

iv) Disadvantages

14. Explain about scripting languages,CGI and HTTP cookies

- Javascript and Jscript
- CGI
- Advantage
- Disadvantage
- HTTP Cookies

15. Define Oracle internet Platform and explain briefly the various services provide by it?

- Oracle Internet Platform
- N_tier architecture based on standards such as
- Oracle Internet Application Server(iAS)

- Communication Services
- HTTP server modules
- Business logic services
- Presentation Services
- Caching Services
- Oracle Database cache
- Oracle web cache

16. MICROSOFT'S WEB SOLUTION PLATFORM
    - Microsoft's web solution platform
    - OLE
    - COM
    - DCOM
    - UNIVERSAL DATA ACCESS
    - Active server pages & ActiveX Data objects
    - Web Page generation using MS Access

17. Explain in detail about active database concepts and triggers:
    -Generalised model for active databases and oracle triggers.
    -Design and implementation issues for active databases.
    -Potential Applications for active databases.
    -Triggers in SQL

18. What are temporal databases?Explain in detail .
    -Temporal database concepts
    -Time representation,calendars and time dimensions
    -Incoporating time in relational databases using tuple versioning.
    -Incorporating time in object oriented databses using attribute versioning.
    -Temporal querying and the TSQL2 language.
    -Time series data.

19. Write notes on deductive databases:
    -Definition of deductive databases.
    -Prolog /datalog notation.
    -Clausal form and horn clauses.
    -Interpretations of rules.
    -Datalog programs and their safety.
    -Relational operations.
    -Non-recursive datalog queries.

20. Explain mobile databases in detail.
    i) Mobile databases
    ii)Characteristics of mobile environment
    iii)Data management issues
    iv)Application:Intermittently synchronized databases

22. Explain the characteristics of biological data in GDB?

- haracteristic 1
- Characteristic 2
- Characteristic
- Characteristic 4
- Characteristic 5
- Characteristic 6 .
- Characteristic 7
- Characteristic 8
- Characteristic 9

23. Write short notes on – Multimedia Databases.

i)Nature of Multimedia Data & Applications
ii)Data Management Issues
iii)Open Research Problems
iv) Multimedia Database Applications

24. Explain in detail the Parallel & Spatial Data base.

i) Parallel Data Base

- Architecture of parallel data bases :
- Key elements of parallel DB processing :
- Query Parallelism :

ii) Spatial Data Base

- Spatial DB characteristics
- Spatial Data Model
- Spatial data base queries
- Techniques of Spatial DB Query

25. Define Data mining? Briefly explain the different types of knowledge discovered during data mining.

Data Mining

Types of knowledge discovered during Data mining :

Association Rule

- Classification
- Clustering
- Approaches to other Data mining problems
- Applications of Data Mining