# CS3492 Database Management Systems

## UNIT I

## RELATIONAL DATABASES

Purpose of Database System – Views of data – Data Models – Database System Architecture – Introduction to relational databases – Relational Model – Keys – Relational Algebra – SQL fundamentals – Advanced SQL features – Embedded SQL– Dynamic SQL

| Q. No. | Questions | CO | Bloom's Level |
|---|---|---|---|
| 1. | **Define database management system and its applications.** **<u>Nov/Dec 2008, 2014</u>**<br>Database management system (DBMS) is a collection of interrelated data and a set of programs to access those data.<br>Applications:<br>▪ Banking<br>▪ Airlines<br>▪ Universities<br>▪ Credit card transactions<br>▪ Tele communication<br>▪ Finance<br>▪ Sales<br>▪ Manufacturing<br>▪ Human resources | C212.1 | BTL1 |
| 2. | **What are the advantages of using a DBMS? <u>May / Dec 2007</u> What is the purpose of database management system ? <u>Nov/Dec 2014</u>**<br>a. The advantages of using a DBMS are<br>a) Controlling redundancy<br>b) Restricting unauthorized access<br>c) Providing multiple user interfaces<br>d) Enforcing integrity constraints.<br>e) Providing backup and recovery | C212.1 | BTL 1 |
| 3. | **Compare database systems with file systems. <u>NOV/DEC 2006. Apr/ May 2015.</u> What are the disadvantages of file processing system? May/ June 2016**<br>a. Data redundancy and inconsistency<br>b. Difficulty in accessing data<br>c. Atomicity of updates<br>d. Concurrent access by multiple users<br>e. Security problems | C212.1 | BTL 4 |

| | | | |
|---|---|---|---|
| 4. | **List the features of a database.**<br>• It is a persistent (stored) collection of related data.<br>• The data is input (stored) only once.<br>• The data is organized (in some fashion).<br>• The data is accessible and can be queried (effectively and efficiently). | C212.1 | BTL 1 |
| 5. | **Define** a database<br>Specifying the data types, structures, and constraints of the data to be stored using a *Data Definition Language* | C212.1 | BTL 1 |
| 6. | **Define a data model.**<br>A data model is a collection of concepts that can be used to describe the structure of a database. The model provides the necessary means to achieve the abstraction. | C212.1 | BTL 1 |
| 7. | **What are the categories of data models.**<br>**High level/conceptual data models** –provide concepts close to the way users perceive the data.<br>**Physical data models** –provide concepts that describe the details of how data is stored in the computer. These concepts are generally meant for the specialist, and not the end user.<br>**Representational data models** –provide concepts that may be understood by the end user but not far removed from the way data is organized. | C212.1 | BTL 5 |
| 8. | Define high level/conceptual data model.<br>**Entity** –represents a real world object or concept<br>**Attribute** - represents property of interest that describes an entity, such as name or salary.<br>**Relationships** –among two or more entities, represents an association among two or more entities. | C212.1 | BTL 1 |
| 9. | Define representational data models.<br>**Representational data models** are used most frequently in commercial DBMSs. They include relational data models, and legacy models such as network and hierarchical models. | C212.1 | BTL 1 |
| 10. | Define physical data model.<br>**Physical data models** describe how data is stored in files by representing record formats, record orderings and access paths. | C212.1 | BTL1 |
| 11. | What is object data model.<br>**Object data models** –a group of higher level implementation data models closer to conceptual data models | C212.1 | BTL 2 |
| 12. | **What is internal level schema.**<br>**Object data models** –a group of higher level implementation data models closer to conceptual data models | C212.1 | BTL 2 |
| 13. | What is the conceptual level schema.<br>**The conceptual level** –has a conceptual schema which describes the structure of the database for users. It hides the details of the physical storage structures, and | C212.1 | BTL 2 |

| | | | |
|---|---|---|---|
| | concentrates on describing entities, data types, relationships, user operations and constraints. Usually a representational data model is used to describe the conceptual schema. | | |
| 14. | What is an external or view level schema.<br>**The External or View level** –includes external schemas or user vies. Each external schema describes the part of the database that a particular user group is interested in and hides the rest of the database from that user group. Represented using the representational data model. | C212.1 | BTL 2 |
| 15. | List the components of DBMS.<br>The major components of database management system are:<br>• Software<br>• Hardware<br>• Data<br>• Procedures<br>• Database Access Language<br>• Users | C212.1 | BTL 1 |
| 16. | What is relational model.<br>The relational model represents the database as a collection of relations. A relation is nothing but a table of values. Every row in the table represents a collection of related data values. These rows in the table denote a real-world entity or relationship. | C212.1 | BTL 2 |
| 17. | **List some of the relational model concepts.**<br>1. **Attribute:** Each column in a Table. Attributes are the properties which define a relation. e.g., Student_Rollno, NAME,etc.<br>2. **Tables** – In the Relational model the, relations are saved in the table format. It is stored along with its entities. A table has two properties rows and columns. Rows represent records and columns represent attributes.<br>3. **Tuple** – It is nothing but a single row of a table, which contains a single record.<br>4. **Relation Schema:** A relation schema represents the name of the relation with its attributes.<br>5. **Degree:** The total number of attributes which in the relation is called the degree of the relation.<br>6. **Cardinality:** Total number of rows present in the Table.<br>7. **Column:** The column represents the set of values for a specific attribute.<br>8. **Relation instance** – Relation instance is a finite set of tuples in the RDBMS system. Relation instances never have duplicate tuples.<br>9. **Relation key** - Every row has one, two or multiple attributes, which is called relation key.<br>10. **Attribute domain** – Every attribute has some pre-defined value and scope which is known as attribute domain | C212.1 | BTL 1 |
| 18. | **List some relational integrity constraints.**<br>**1**. Domain Constraints<br>2. Key constraints<br>3. Referential integrity constraints | C212.1 | BTL1 |
| 19. | **Define domain constraints.**<br>Domain constraints can be violated if an attribute value is not appearing in the corresponding domain or it is not of the appropriate data type. | C212.1 | BTL2 |

Domain constraints specify that within each tuple, and the value of each attribute must be unique. This is specified as data types which include standard data types integers, real numbers, characters, Booleans, variable length strings, etc.

**Example:**

```
Create DOMAIN CustomerName
CHECK (value not NULL)
```

The example shown demonstrates creating a domain constraint such that CustomerName is not NULL

20. **Define key constraints.**

An attribute that can uniquely identify a tuple in a relation is called the key of the table. The value of the attribute for different tuples in the relation has to be unique.

**Example:**

In the given table, CustomerID is a key attribute of Customer Table. It is most likely to have a single key for one customer, CustomerID =1 is only for the CustomerName =" Google".

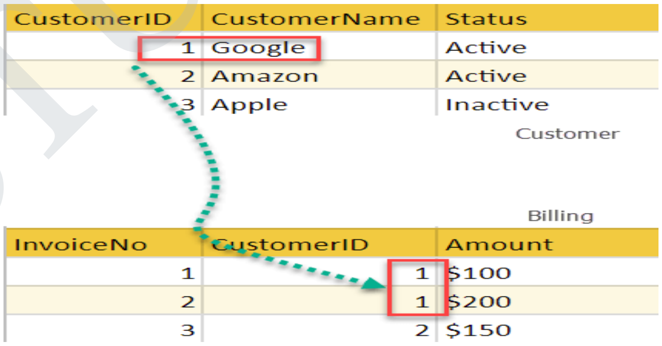
| CustomerID | CustomerName |
|---|---|
| 1 | Google |
| 2 | Amazon |
| 3 | Apple |

C212.1 BTL2

21. **Define referential integrity constraints.**

Referential integrity constraints is base on the concept of Foreign Keys. A foreign key is an important attribute of a relation which should be referred to in other relationships. Referential integrity constraint state happens where relation refers to a key attribute of a different or same relation. However, that key element must exist in the table.

**Example:**



Customer

| CustomerID | CustomerName | Status |
|---|---|---|
| 1 | Google | Active |
| 2 | Amazon | Active |
| 3 | Apple | Inactive |

Billing

| InvoiceNo | CustomerID | Amount |
|---|---|---|
| 1 | 1 | $100 |
| 2 | 1 | $200 |
| 3 | 2 | $150 |

In the above example, we have 2 relations, Customer and Billing.

Tuple for CustomerID =1 is referenced twice in the relation Billing. So we know CustomerName=Google has billing amount $300

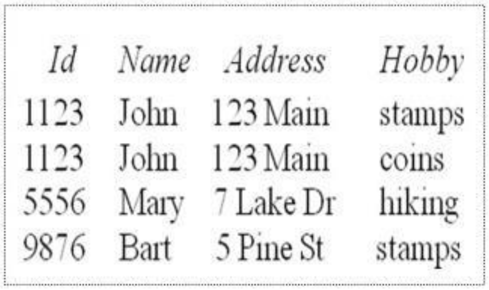
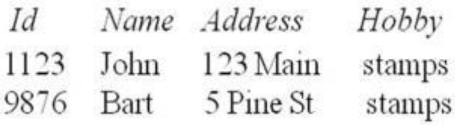
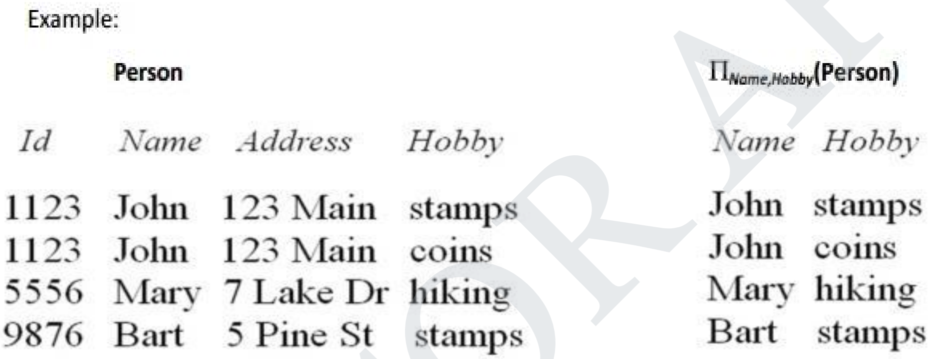
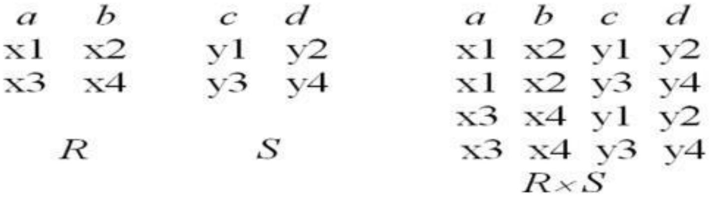
C212.1 BTL3

22. **List the operations which can be done on the relational model.**

The operations are, Insert, update, delete and select.

C212.1 BTL1

| | | | |
|---|---|---|---|
| | • Insert is used to insert data into the relation<br>• Delete is used to delete tuples from the table.<br>• Modify allows you to change the values of some attributes in existing tuples.<br>• Select allows you to choose a specific range of data. | | |
| 23. | **What are the advantages of relational model.**<br>• **Simplicity**: A relational data model is simpler than the hierarchical and network model.<br>• **Structural Independence**: The relational database is only concerned with data and not with a structure. This can improve the performance of the model.<br>• **Easy to use**: The relational model is easy as tables consisting of rows and columns is quite natural and simple to understand<br>• **Query capability**: It makes possible for a high-level query language like SQL to avoid complex database navigation.<br>• **Data independence**: The structure of a database can be changed without having to change any application.<br>• **Scalable**: Regarding a number of records, or rows, and the number of fields, a database should be enlarged to enhance its usability. | C212.1 | BTL1 |
| 24. | **What are the disadvantages of relational model.**<br>• Few relational databases have limits on field lengths which can't be exceeded.<br>• Relational databases can sometimes become complex as the amount of data grows, and the relations between pieces of data become more complicated.<br>• Complex relational database systems may lead to isolated databases where the information cannot be shared from one system to another. | C212.1 | BTL1 |
| 25. | **Define relational algebra.**<br>• Intermediate language used within DBMS.<br>• Procedural langauge | C212.1 | BTL1 |
| 26. | **List the relational algebraic operations.**<br>• *Basic operators*: select, project, union, set difference, Cartesian product<br>• *Derived operators*: set intersection, division, join | C212.1 | BTL1 |
| 27. | **Define select operation in relational algebra.**<br>Produce table containing subset of rows of argument table satisfying condition<br>$\sigma_{condition}$ *relation*<br>Example:<br>Person $\sigma_{Hobby='stamps'}$(Person) | C212.1 | BTL1 |

<table>
<tr>
<td></td>
<td>

<table>
<tr><th>Id</th><th>Name</th><th>Address</th><th>Hobby</th></tr>
<tr><td>1123</td><td>John</td><td>123 Main</td><td>stamps</td></tr>
<tr><td>1123</td><td>John</td><td>123 Main</td><td>coins</td></tr>
<tr><td>5556</td><td>Mary</td><td>7 Lake Dr</td><td>hiking</td></tr>
<tr><td>9876</td><td>Bart</td><td>5 Pine St</td><td>stamps</td></tr>
</table>

<table>
<tr><th>Id</th><th>Name</th><th>Address</th><th>Hobby</th></tr>
<tr><td>1123</td><td>John</td><td>123 Main</td><td>stamps</td></tr>
<tr><td>9876</td><td>Bart</td><td>5 Pine St</td><td>stamps</td></tr>
</table>
</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28.</td>
<td>
<b>Define project operator in relational algebra.</b><br>
Produces table containing subset of columns of argument table<br>
• Example:<br>
$\Pi_{attribute\ list}(relation)$<br>

Example:<br>
Person
<table>
<tr><th>Id</th><th>Name</th><th>Address</th><th>Hobby</th></tr>
<tr><td>1123</td><td>John</td><td>123 Main</td><td>stamps</td></tr>
<tr><td>1123</td><td>John</td><td>123 Main</td><td>coins</td></tr>
<tr><td>5556</td><td>Mary</td><td>7 Lake Dr</td><td>hiking</td></tr>
<tr><td>9876</td><td>Bart</td><td>5 Pine St</td><td>stamps</td></tr>
</table>
$\Pi_{Name,Hobby}$(Person)
<table>
<tr><th>Name</th><th>Hobby</th></tr>
<tr><td>John</td><td>stamps</td></tr>
<tr><td>John</td><td>coins</td></tr>
<tr><td>Mary</td><td>hiking</td></tr>
<tr><td>Bart</td><td>stamps</td></tr>
</table>
</td>
<td>C212.1</td>
<td>BTL1</td>
</tr>
<tr>
<td>29.</td>
<td>
<b>Define Cartesian product in relational algebra.</b><br>
• If <i>R</i> and <i>S</i> are two relations, <i>R S</i> is the set of all concatenated tuples &lt;<i>x</i>,<i>y</i>&gt;, where <i>x</i> is a tuple in <i>R</i> and <i>y</i> is a tuple in<i>S</i><br>
– (<i>R</i> and <i>S</i> need not be union compatible)<br>
• <i>R S</i> is <u>expensive to compute</u>:<br>
– Factor of two in the size of each row<br>
– Quadratic in the number of row<br>
Example:<br>

<table>
<tr><th>a</th><th>b</th></tr>
<tr><td>x1</td><td>x2</td></tr>
<tr><td>x3</td><td>x4</td></tr>
</table>
R
<table>
<tr><th>c</th><th>d</th></tr>
<tr><td>y1</td><td>y2</td></tr>
<tr><td>y3</td><td>y4</td></tr>
</table>
S
<table>
<tr><th>a</th><th>b</th><th>c</th><th>d</th></tr>
<tr><td>x1</td><td>x2</td><td>y1</td><td>y2</td></tr>
<tr><td>x1</td><td>x2</td><td>y3</td><td>y4</td></tr>
<tr><td>x3</td><td>x4</td><td>y1</td><td>y2</td></tr>
<tr><td>x3</td><td>x4</td><td>y3</td><td>y4</td></tr>
</table>
$R \times S$
</td>
<td>C212.1</td>
<td>BTL1</td>
</tr>
</table>

| | | | |
|---|---|---|---|
| 30. | **What is a table in relational database.**<br>A table is set of data elements that has a horizontal dimension (rows) and a vertical dimension (columns) in a relational database system. A table has a specified number of columns but can have any number of rows. Rows stored in a table are structurally equivalent to records from flat files. Columns are often referred as attributes or fields. | C212.1 | BTL1 |
| 31. | **Define an identifier.**<br>An identifier is an attribute that is used either as a primary key or as a foreign key. The integer datatype is used for identifiers. In cases where the number of records exceed the allowed values by the integer datatype then a big integer datatype is used. | C212.1 | BTL1 |
| 32. | **Define a primary key.**<br>A column in a table whose values uniquely identify the rows in the table. A primary key value cannot be NULLto matching columns in other tables\ | C212.1 | BTL1 |
| 33. | **What is a Foreign key?**<br>A column in a table that does not uniquely identify rows in that table, but is used as a link to matching columns in other tables. | C212.1 | BTL1 |
| 34. | **What is Relationship?**<br>A relationship is an association between two tables. For example the relationship between the table "hotel" and "customer" maps the customers to the hotels they have used. | C212.1 | BTL1 |
| 35. | **What is an index in relational database?**<br>An index is a data structure which enables a query to run at a sublinear-time. Instead of having to go through all records one by one to identify those which match its criteria the query uses the index to filter out those which don't and focus on those who do. | C212.1 | BTL1 |
| 36. | **Define view.**<br>A view is a virtual or logical table composed of the result set of a pre-compiled query. Unlike ordinary tables in a relational database, a view is not part of the physical schema: it is a dynamic, virtual table computed or collated from data in the database. Changing the data in a view alters the data stored in the database | C212.1 | BTL1 |
| 37. | **What is a Query**<br>A query is a request to retrieve data from a database with the SQL SELECT instruction or to manipulate data stored in tables. | C212.1 | BTL1 |

| | | | |
|---|---|---|---|
| 38. | **Define SQL**<br>Structured Query Language (SQL), pronounced "sequel", is a language that provides an interface to relational database systems. It was developed by IBM in the 1970s for use in System R. SQL is a de facto standard, as well as an ISO and ANSI standard. | C212.1 | BTL1 |
| 39. | **List any 5 responsibilities of the DB Manager May June 2007**<br>a) Interaction with he file manager<br>b) Translation of DML commands in to low level file system commands<br>c) Storing, retrieving and updating data in the database<br>d) Data dictionary<br>e) Indices | C212.1 | BTL1 |
| 40. | **What is Data Dictionary May2003, 2004, Nov 2006& Nov 2007.**<br>**Data dictionary** : which stores meta data about the structure of the database, in particular schema of the database. | C212.1 | BTL1 |
| 41. | **What is Data independence & what are the levels. APRIL MAY 2008, MAY /JUNE 2012, Nov 2017**<br>The ability to modify a schema definition in one level without affecting a schema definition in the next higher level called data independence.<br>**Different levels .**<br>Physical level<br>Logical level<br>View level | C212.1 | BTL1 |
| 42. | **Define instance and schemas. April/May 2012**<br>- Database change over times as information is inserted and deleted.<br>- The collection of information stored in the database at a particular moment called an **instance of the database**.<br>- The overall design of the database is called the **database schema.** | C212.1 | BTL1 |
| 43. | **What are the types of attributes. April/May 2007**<br>○ Simple<br>○ Composite<br>○ Single-valued<br>○ Multi-valued<br>○ Derived | C212.1 | BTL1 |
| 44. | **What is mapping cardinalities April/May 2009**<br>Mapping cardinalities express the number of entities to which another entity can be associated via a relationship set. | C212.1 | BTL1 |
| 45. | What are the various data base languages in SQL? **April/May 2018**<br>**Data Definition Language (DDL)**<br>Commands that define a database, including creating, altering, and dropping | C212.1 | BTL1 |

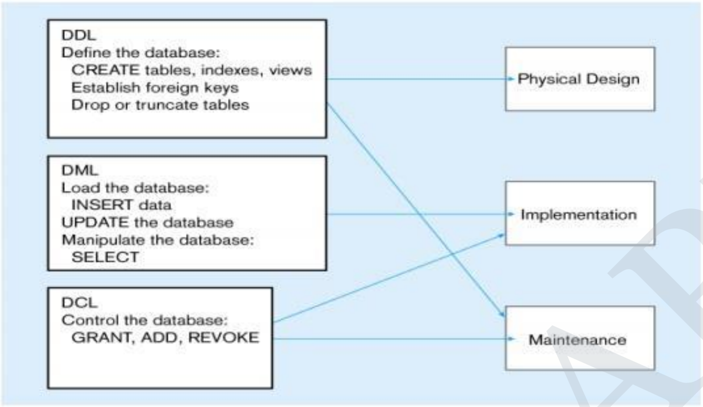
tables and establishing constraints

**Data Manipulation Language (DML)**

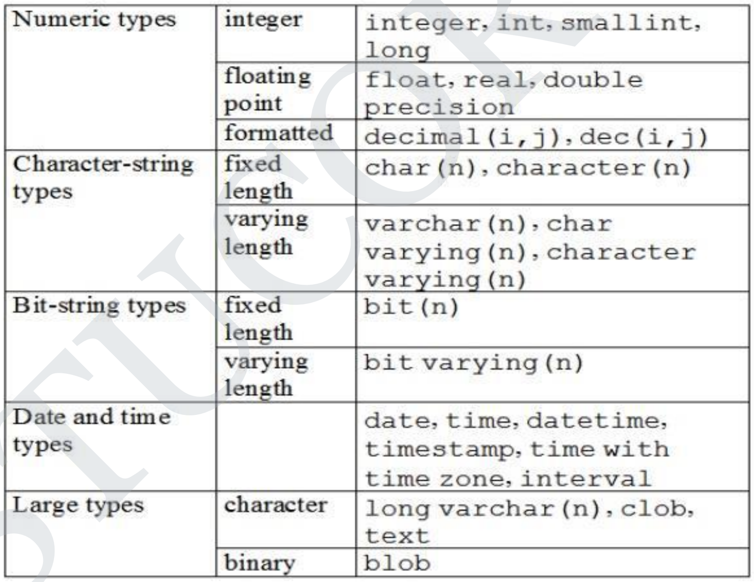
Commands that maintain and query a database

**Data Control Language (DCL)**

Commands that control a database, including administering privileges and committing data



46. List the data types in SQL.



| | | |
|---|---|---|
| Numeric types | integer | integer, int, smallint, long |
| | floating point | float, real, double precision |
| | formatted | decimal (i, j), dec (i, j) |
| Character-string types | fixed length | char (n), character (n) |
| | varying length | varchar (n), char varying (n), character varying (n) |
| Bit-string types | fixed length | bit (n) |
| | varying length | bit varying (n) |
| Date and time types | | date, time, datetime, timestamp, time with time zone, interval |
| Large types | character | long varchar (n), clob, text |
| | binary | blob |

C212.1 BTL1

47. **What are the function of DBA? April/May 2009**

- Schema definition.
- Storage structure and access –method definition.
- Schema and physical-organization modification
- Granting of authorization for data access:
- Routine maintenance

C212.1 BTL1

48. **What is Relational Model . April/May 2009**

- The relation is the only construct required to represent the associations

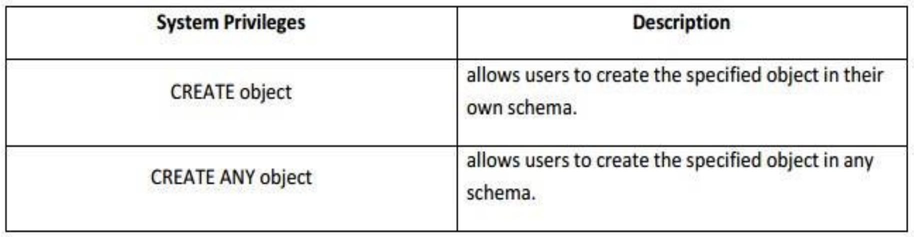
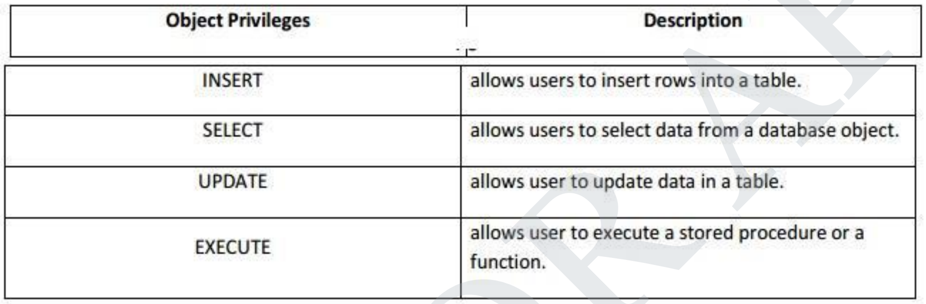
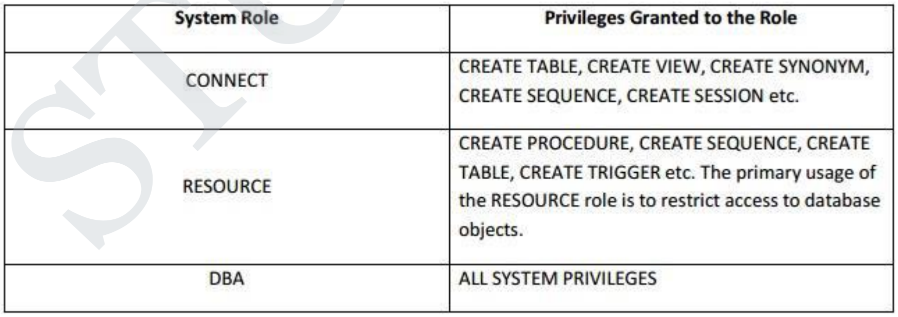
C212.1 BTL1

| | | | |
|---|---|---|---|
| | among the attributes of an entity as well as the relationship among different entities.<br>- A relation may be visualized as a named table. Each column of the table corresponds to an attribute of the relation and is named. | | |
| 49. | **Define Data Model**. **<u>April/May 2009</u>**<br>It is a collection of conceptual tools for describing data, data relationships, data semantics and consistency constraints | C212.1 | BTL1 |
| 50. | What the problems caused by the redundancy?<br>The problems caused by data redundancy are<br>• The first is that storing values multiple times will lead to waste spaces.<br>• The second problem is that when a field value changes, multiple occurrences need to be updated. For example, if customer-1 moves, it needs to change the values of Street, City, State and Zip in multiple records.<br>• The third problem occurs when forget to change the values in any of the records. The database would then have inconsistent data. | C212.1 | BTL1 |
| 51. | What is data definition language?<br>**Data Definition Language:**<br>A **data definition language** or **data description language** (**DDL**) is a syntax for defining the data structures using imperative verbs, especially database schemas. | C212.1 | BTL1 |
| 52. | What is the syntax for creating a table in SQL?<br>Syntax:<br>**CREATE TABLE *[table name]* ( *[column definitions]* ) *[table parameters]***<br>**Example**:<br>The command to create a table named employees with a few sample columns would be:<br>CREATE TABLE employees (<br>id INTEGER PRIMARY KEY,<br>first_name VARCHAR(50) NULL,<br>last_name VARCHAR(75) NOT NULL,<br>fname VARCHAR(50) NOT NULL,<br>dateofbirth DATE NULL<br>); | C212.1 | BTL2 |
| 53. | What is the syntax for removing or deleting a table in SQL?<br>**Drop** - To destroy an existing database, table, index, or view. **Syntax:**<br>DROP objecttype objectname.<br>**Example:**<br>The command to drop a table named employees would be:<br>DROP employees;<br>The Drop statement would remove the entire table (employees) from the database. | C212.1 | BTL2 |
| 54. | How to modify or alter an existing table in SQL.<br>**Alter** - To modify an existing database object.<br>An ALTER statement in SQL changes the properties of an object inside of a relational database management system (RDBMS).<br>**Syntax:** | C212.1 | BTL2 |

| | | | |
|---|---|---|---|
| | ALTER objecttype objectname parameters.<br>**Example:**<br>The command to add (then remove) a column named bubbles for an existing table named sink would be:<br>ALTER TABLE sink ADD bubbles INTEGER;<br>ALTER TABLE sink DROP COLUMN bubbles; | | |
| 55. | How to rename an existing table in SQL?<br>**Syntax:**<br>**Rename** - to rename the table.<br>**Example**<br>RENAME TABLE old_name TO new_name; | C212.1 | BTL2 |
| 56. | Define Data Manipulation Language.<br>A data manipulation language (DML) is a set of commands permitting users to manipulate data in a database. This manipulation involves inserting data into database tables, retrieving existing data, deleting data from existing tables and modifying existing data. | C212.1 | BTL1 |
| 57. | List the DML commands in SQL.<br>**SELECT:** This command is used to retrieve rows from a table.<br>**The select syntax:**<br>SELECT [column name(s)] from [table name] where [conditions].<br>**UPDATE:** This command modifies data of one or more records.<br>**The update command syntax:**<br>UPDATE table name SET column name = value where [condition].<br>**INSERT:** This command adds one or more records to a database table.<br>**The insert command syntax:**<br>INSERT INTO table name [column(s)] VALUES [value(s)].<br>**DELETE:** This command removes one or more records from a table according to specified conditions.<br>**Delete command syntax:**<br>DELETE FROM table name where [condition]. | C212.1 | BTL1 |
| 58. | List the aggregation functions in SQL.<br>• **COUNT** returns the number of rows in a specified column.<br>• **SUM** returns the sum of the values in a specified column.<br>• **AVG** returns the average of the values in a specified column.<br>• **MIN** returns the smallest value in a specified column.<br>• **MAX** returns the largest value in a specified column.<br>**Examples**:<br>Property (PropertyNo, Street, City, postcode, Type, OwnerNo, Rooms, Rent)<br>**Query:**<br>How many properties cost more than 350 per month to rent?<br><br>SELECT COUNT(*) AS count FROM property WHERE rent > 350; | C212.1 | BTL1 |
| 59. | What is subquery.<br>**Subquery**<br>• A complete SELECT statement can be embedded (subselect) within another SELECT statement.<br>• A subselect can be used in the WHERE and HAVING | C212.1 | BTL1 |

| | | | |
|---|---|---|---|
| | clauses of the outer SELECT statement (nested query).<br>• A subquery can be used immediately following a relational operator.<br>• Subquery always enclosed in parentheses.<br>**Type of subquery:**<br>▪ A scalar subquery returns a single column and a single row (singlevalue).<br>▪ A row subquery returns multiple columns, but a single row.<br>▪ A table subquery returns one or more columns and multiple rows.<br><br>**Example:**<br>List the staff who work in the branch at _163 Main St‘.<br><br>SELECT sno, fname, lname, position FROM staff WHERE bno = (SELECT bno FROM branch WHERE street = _163 Main St‘); | | |
| 60. | **How to modify a data in a table in SQL?**<br>**Modifying Data in the DB (UPDATE)**<br>**Syntax**<br>UPDATE table_name<br>SET column_name1 = data_value1 [, column_namei = data_valuei ...]<br>[WHERE search_condition]<br>**Description:**<br>▪ table_name may be either a base table or an updatable view.<br>▪ The SET clause specifies the names of one or more columns that are updated for all rows in the table.<br>▪ Only rows that satisfy the search_condition are updated.<br>▪ data_values must be compatible with the data types for the corresponding columns. | C212.1 | BTL3 |
| 61. | What is DCL?<br>**The Data Control Language (DCL)** component of the SQL language is used to create privileges to allow users access to, and manipulation of, the database.<br>There are two main commands:<br>**GRANT** to grant a privilege to a user<br>**REVOKE** to revoke (remove) a privilege from a user | C212.1 | BTL1 |

| | | | |
|---|---|---|---|
| 62. | Define GRANT command in SQL.<br>**The Syntax for the GRANT command is:**<br>GRANT privilege_name ON object_name TO {user_name \|PUBLIC \|role_name} [WITH GRANT OPTION];<br>**Description:**<br>**privilege_name** is the access right or privilege granted to the user. Some of the access rights are ALL, EXECUTE, and SELECT.<br>**object_name** is the name of an database object like TABLE, VIEW, STORED PROC and SEQUENCE.<br>**user_name** is the name of the user to whom an access right is being granted.<br>**PUBLIC** is used to grant access rights to all users.<br>**ROLES** are a set of privileges grouped together.<br>**WITH GRANT OPTION -** allows a user to grant access rights to other users.<br><br>**Example**: GRANT SELECT ON employee TO user1;<br>**Description:**<br>This command grants a SELECT permission on employee table to user1.<br>WITH GRANT option should be used carefully because if the GRANT SELECT privilege is permitted on employee table to user1 using the WITH GRANT option, then user1 can GRANT SELECT privilege on employee table to another user, such as user2 etc. Later, if you REVOKE the SELECT privilege on employee from user1, still user2 will have SELECT privilege on employee table. | C212.1 | BTL1 |
| 63. | **Write about REVOKE command in SQL.**<br>**SQL REVOKE Command:**<br>The REVOKE command removes user access rights or privileges to the database objects.<br>**The Syntax for the REVOKE command is:**<br>REVOKE privilege_name ON object_name FROM {user_name \|PUBLIC \|role_name}<br>**Example:**<br>REVOKE SELECT ON employee FROM user1;<br>**Description:**<br>This command will REVOKE a SELECT privilege on employee table from user1.When you REVOKE SELECT privilege on a table from a user, the user will not be able to SELECT data from that table anymore. However, if the user has received SELECT privileges on that table from more than one users, he/she can SELECT from that table until everyone who granted the permission revokes it. | C212.1 | BTL1 |
| 64. | **Define privileges in SQL.**<br>**Privileges:**<br>Privileges defines the access rights provided to a user on a database object. There are two types of privileges.<br>**1) System privileges -** This allows the user to CREATE, ALTER, or DROP database objects.<br>**2) Object privileges -** This allows the user to EXECUTE, SELECT, INSERT, UPDATE, or DELETE data from database objects to which the privileges apply. | C212.1 | BTL1 |
| 65. | What is system privileges?<br>**System privileges -** This allows the user to CREATE, ALTER, or DROP database objects. Few CREATE system privileges are listed below: | C212.1 | BTL1 |

<table>
<tr>
<td></td>
<td><table><tr><th>System Privileges</th><th>Description</th></tr><tr><td>CREATE object</td><td>allows users to create the specified object in their own schema.</td></tr><tr><td>CREATE ANY object</td><td>allows users to create the specified object in any schema.</td></tr></table>The above rules also apply for ALTER and DROP system privileges.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>66.</td>
<td>What is object privileges?<br><b>Object privileges</b> - This allows the user to EXECUTE, SELECT, INSERT, UPDATE, or DELETE data from database objects to which the privileges apply.<br><table><tr><th>Object Privileges</th><th>Description</th></tr><tr><td>INSERT</td><td>allows users to insert rows into a table.</td></tr><tr><td>SELECT</td><td>allows users to select data from a database object.</td></tr><tr><td>UPDATE</td><td>allows user to update data in a table.</td></tr><tr><td>EXECUTE</td><td>allows user to execute a stored procedure or a function.</td></tr></table></td>
<td>C212.1</td>
<td>BTL1</td>
</tr>
<tr>
<td>67.</td>
<td>Define roles in SQL.<br><b>Roles:</b> Roles are a collection of privileges or access rights. When there are many users in a database it becomes difficult to grant or revoke privileges to users. Therefore, if you define roles, you can grant or revoke privileges to users, thereby automatically granting or revoking privileges. You can either create Roles or use the system roles pre-defined by oracle.</td>
<td>C212.1</td>
<td>BTL1</td>
</tr>
<tr>
<td>68.</td>
<td>List the system privileges granted to role.<br><table><tr><th>System Role</th><th>Privileges Granted to the Role</th></tr><tr><td>CONNECT</td><td>CREATE TABLE, CREATE VIEW, CREATE SYNONYM, CREATE SEQUENCE, CREATE SESSION etc.</td></tr><tr><td>RESOURCE</td><td>CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER etc. The primary usage of the RESOURCE role is to restrict access to database objects.</td></tr><tr><td>DBA</td><td>ALL SYSTEM PRIVILEGES</td></tr></table></td>
<td>C212.1</td>
<td>BTL1</td>
</tr>
<tr>
<td>69.</td>
<td>Write the syntax for creating a role.<br><b>The Syntax to create a role is:</b><br>CREATE ROLE role_name [IDENTIFIED BY password];<br><b>Example:</b> To create a role called "developer" with password as "pwd",<br>CREATE ROLE developer [IDENTIFIED BY pwd];</td>
<td>C212.1</td>
<td>BTL1</td>
</tr>
</table>

| | | | |
|---|---|---|---|
| | It's easier to GRANT or REVOKE privileges to the users through a role rather than assigning a privilege directly to every user. If a role is identified by a password, then, when you GRANT or REVOKE privileges to the role, you definitely have to identify it with the password. | | |
| 70. | What is the syntax for dropping a role in SQL?<br>**The Syntax to drop a role from the database:**<br>DROP ROLE role_name;<br>**Example:** To drop a role called developer, you can write:<br>DROP ROLE testing; | C212.1 | BTL1 |
| 71. | Define TCL.<br>**TCL -** Transactional Control Language. It is used to manage different transactions occurring within a database.<br>**Examples of TCL commands include**:<br>**COMMIT** to apply the transaction by saving the database changes.<br>**ROLLBACK** to undo all changes of a transaction.<br>**SAVEPOINT** to divide the transaction into smaller sections. It defines breakpoints for a transaction to allow partial rollbacks. | C212.1 | BTL1 |
| 72. | Define COMMIT in SQL.<br>**SYNTAX:**<br>COMMIT [WORK] [ COMMENT 'text' \| FORCE 'text' [, integer] ]<br>**Description:**<br>WORK : is supported only for compliance with standard SQL. The statements COMMIT and COMMIT WORK are equivalent.<br><br>COMMENT : specifies a comment to be associated with the current transaction. The 'text' is a quoted literal of up to 50 characters that Oracle stores in the data dictionary view DBA_2PC_PENDING along with the transaction ID if the transaction becomes in-doubt.<br><br>FORCE : manually commits an in-doubt distributed transaction. The transaction is identified by the 'text' containing its local or global transaction ID. To find the IDs of such transactions, query the data dictionary view DBA_2PC_PENDING. You can also use the integer to specifically assign the transaction a system change number (SCN). If you omit the integer, the transaction is committed using the current SCN. COMMIT statements using the FORCE clause are not supported in PL/SQL. | C212.1 | BTL1 |
| 73. | Define ROLLBACK statement in SQL.<br>PURPOSE:<br>To undo work done in the current transaction. You can also use this command to manually und the work done by an in-doubt distributed transaction.<br>SYNTAX:<br>ROLLBACK [WORK][ TO [SAVEPOINT] savepoint \| FORCE 'text' ]<br>**Description:**<br>WORK : is optional and is provided for ANSI compatibility.<br><br>TO : rolls back the current transaction to the specified savepoint. If you omit this clause, the ROLLBACK statement rolls back the entire transaction.<br><br>FORCE : manually rolls back an in-doubt distributed transaction. The | C212.1 | BTL1 |

| | | | |
|---|---|---|---|
| | transaction is identified by the 'text' containing its local or global transaction ID. To find the IDs of such transactions, query the data dictionary view DBA_2PC_PENDING. ROLLBACK statements with the FORCE clause are not supported in PL/SQ | | |
| 74. | What is embedded SQL?<br>Embedded SQL statements are SQL statements written inline with the program source code of the host language. The embedded SQL statements are parsed by an embedded SQL preprocessor and replaced by host-language calls to a code library. The output from the preprocessor is then compiled by the host compiler. This allows programmers to embed SQL statements in programs written in any number of languages such as: C/C++, COBOL and Fortran. | C212.1 | BTL1 |
| 75. | What is static SQL? **Dec 2017**<br>**Static SQL**<br>The source form of a static SQL statement is embedded within an application program written in a host language such as COBOL.<br>The statement is prepared before the program is executed and the operational form of the statement persists beyond the execution of the program.<br>Static SQL statements in a source program must be processed before the program is compiled. This processing can be accomplished through the DB2 precompiler or the SQL statement coprocessor.<br>The DB2 precompiler or the coprocessor checks the syntax of the SQL statements, turns them into host language comments, and generates host language statements to invoke DB2.<br>The preparation of an SQL application program includes precompilation, the preparation of its static SQL statements, and compilation of the modified source program. | C212.1 | BTL1 |
| 76. | What is dynamic SQL?<br>**Dynamic SQL:**<br>Programs that contain embedded dynamic SQL statements must be precompiled like those that contain static SQL, but unlike static SQL, the dynamic statements are constructed and prepared at run time.<br>The source form of a dynamic statement is a character string that is passed to DB2 by the program using the static SQL statement PREPARE or EXECUTE IMMEDIATE. | C212.1 | BTL1 |

## PART-B

| Q. No. | Questions | CO | Bloom's Level |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 1. | List and describe the features and purpose of database? | C212.1 | BTL5 |
| 2. | List and explain the limitations of file processing system? | C212.1 | BTL5 |
| 3. | Compare database vs file processing system? | C212.1 | BTL2 |
| 4. | What is data model? Explain various data models with example? | C212.1 | BTL2 |
| 5. | Explain the database system architecture with neat diagram? **April/May 2017, Nov/Dec 2017** | C212.1 | BTL2 |
| 6. | List and describe the components of database management system with neat diagram. | C212.1 | BTL2 |
| 7. | Define relational algebra. Explain various relational algebraic operations with example. **Nov/Dec 2016, April/May 2017** | C212.1 | BTL2 |
| 8 | List and explain the properties of a relations with example. | C212.1 | BTL2 |
| 9. | What are the various data types in SQL? Explain them with example? | C212.1 | BTL2 |
| 10 | List and explain various database languages with example? | C212.1 | BTL2 |
| 11 | Explain DML with example? | C212.1 | BTL3 |
| 12 | Explain DDL with example? | C212.1 | BTL3 |
| 13 | Explain DCL with example? | C212.1 | BTL3 |
| 14. | Explain TCL with example? | C212.1 | BTL3 |
| 15 | Justify the need of embedded SQL. Consider the relation student (Reg No, name, mark and grade). Write embedded dynamic SQL program in C language to retrieve all the students records whose mark is more than 90. **Nov/Dec 2016, April/May 2017.** | C212.1 | BTL5 |

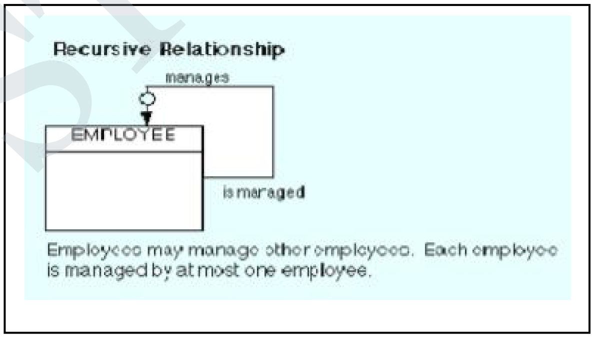
## UNIT II

Entity-Relationship model – E-R Di gr ms – Enh nced-ER Model – ER-to-Relational Mapping –

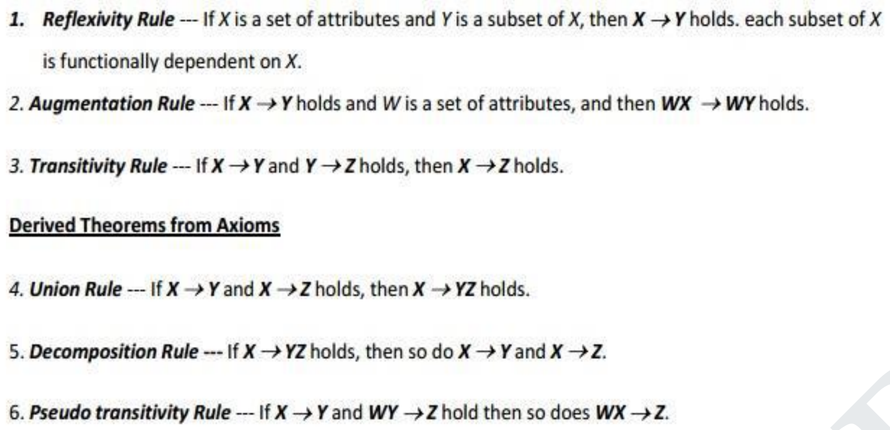
Functional Dependencies – Non-loss Decomposition – First, Second, Third Normal Forms, Dependency Preservation – Boyce/Codd Norm l Form – Multi-valued Dependencies and Fourth Normal Form – Join De endencies and Fifth Normal Form

| Q. No. | Questions | CO | Bloom’s Level |
|---|---|---|---|

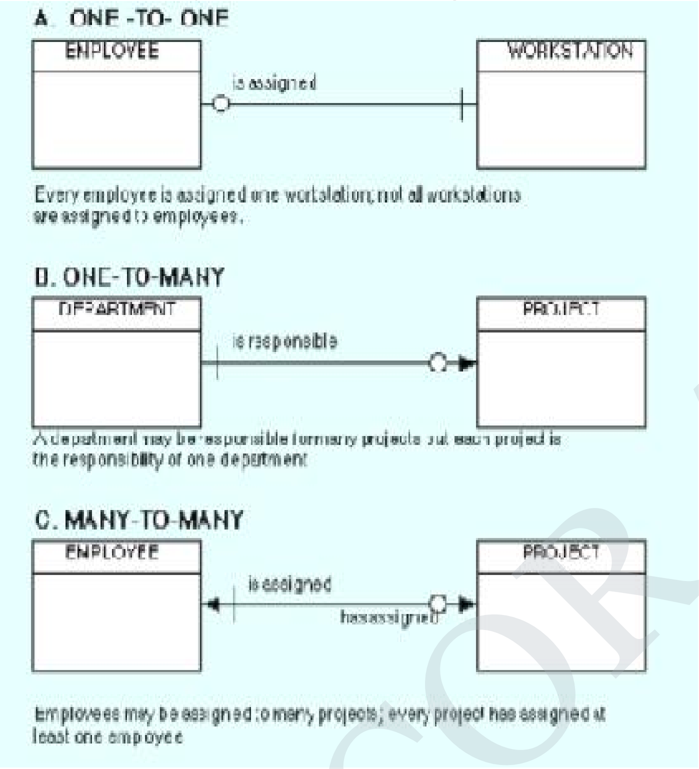
| | | | |
|---|---|---|---|
| 1. | **What is an entity relationship model? May/ June 2016**<br>The entity relationship model is a collection of basic objects called entities and relationship among those objects. An entity is a thing or object in the real world that is distinguishable from other objects. | C212.2 | BTL1 |
| 2. | **Define weak and strong entity sets? April/May 2009, April/May 2018**<br>**Weak entity set:** entity set that do not have key attribute of their own are called weak entity sets.<br>**Strong entity set:** Entity set that has a primary key is termed a strong entity set. | C212.2 | BTL1 |
| 3. | **Give the limitations of ER model? How do you overcome this? May/ June 2007**<br>The entity relationship model is a collection of basic objects called entities and relationship among those objects. An entity is a thing or object in the real world that is distinguishable from other objects. | C212.2 | BTL1 |
| 4. | **Define Specialization and Aggregation.**<br>It is the process of designating sub groupings within an entity set.It is a top down process.<br>**Specialization** which is represented by triangle. The lable ISA stands for "is a and represent, for eg that customer **is a** person.<br>**Aggregation** is a special kind of association that specifies a whole/part relationship between the aggregate (whole) and a component part. | C212.2 | BTL1 |
| 5. | **What are three characteristics of a relational database system?Nov/Dec 2008**<br>Controlling redundancy<br>Restricting unauthorized access<br>Providing multiple user interfaces | C212.2 | BTL1 |
| 6. | **Give the distinction between primary key, candidate key and super key. Nov/Dec 2006,2009**<br>Primary key – is used in a data base to avoid duplication of attributes and also makes a relation to the other database.<br>Candidate key - a key which is in the data base is called as candidate key, it might be any key attribute.<br>Super key – collection of keys of a database is called as super key | C212.2 | BTL1 |
| 7. | **Define functional dependency. Nov/Dec→2010, Apr/ May 2015**<br>A functional dependency is a constraint between two sets of attributes from the data base. A functional dependency , denoted by<br>X Y<br>Between two sets of attributes X and Y that are subsets of R specifies a constraint on the possible tuples that can form a relation instance r of R.<br>R={ A1,A2,…,An}. The constraint states that ,for any two tuples t1 and t2 in r such that t1[X]=t2[X] , we must also have t1[Y] = t2[Y].<br>**We can also say that Y is functionally dependent on X.** | C212.2 | BTL1 |

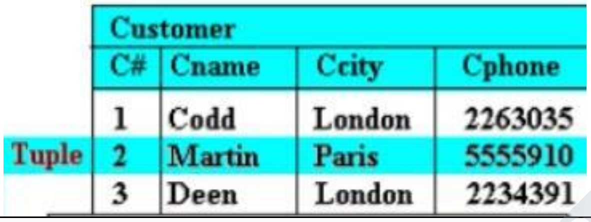
| | | | |
|---|---|---|---|
| 8. | Define an entity?<br>**<u>Entities:</u>**<br>Entity -a thing (animate or inanimate) of independent physical or conceptual existence and distinguishable.<br>**Example:**<br>In the University database context, an individual student, faculty member, a class room, a course are entities. | C212.2 | BTL1 |
| 9. | What is an entity set?<br>Entity Set or Entity Type-Collection of entities all having the same properties.<br>**Example:**<br>Student entity set –collection of all student entities.<br>Course entity set –collection of all course entities. | C212.2 | BTL2 |
| 10. | What is an attribute?<br>**<u>Attributes:</u>**<br>Attributes - Each entity is described by a set of attributes/properties.studententity<br>**Example:**<br>StudName–name of the student.<br>RollNumber–the roll number of the student.<br>Sex–the gender of the student etc.<br>All entities in an Entity set/type have the same set of attributes. | C212.2 | BTL2 |
| 11. | What is derived attributes?<br>Derived attributes are those created by a formula or by a summary operation on other attributes | C212.2 | BTL2 |
| 12. | What is a recursive relationship?<br>Recursive relationships A recursive relationship is an entity is associated with itself.<br>Example:<br>An employee may manage many employees and each employee is managed by one employee.<br><br> <br>Employees may manage other employees. Each employee is managed by at most one employee. | C212.2 | BTL2 |
| 13. | **Define cardinality?**<br>***Cardinality***<br>A business rule indicating the number of times a particular object or | C212.2 | BTL1 |

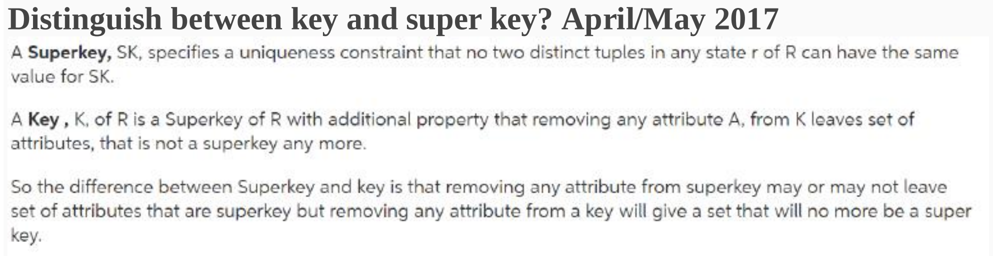
| | | | |
|---|---|---|---|
| | activity may occur. | | |
| 14. | **List the properties of a relation?**<br>All entries in a given column are of the same kind or type<br>The ordering of columns is immaterial.<br>No two tuples are exactly the same.<br>There is only one value for each attribute of a tuple.<br>The ordering of tuples is immaterial. | C212.2 | BTL1 |
| 15. | **Define a key?**<br>A key is a set of attributes that uniquely identifies an entire tuple, a functional dependency allows us to express constraints that uniquely identify the values of certain attributes.<br>First, keys will typically be used as links, ie. key values will appear in other relations to represent their associated tuples<br>Second, keys form the basis for constructing indexes to speed up retrieval of tuples from a relation. Small keys will decrease the size of indexes and the time to look up an index. | C212.2 | BTL2 |
| 16. | **Define a relational schema?**<br>A Relational Database Schema comprises<br>1. the definition of all domains<br>the definition of all relations, specifying for each<br>its intension (all attribute names), and<br>a primary key | C212.2 | BTL1 |
| 17. | **List the uses of functional dependencies?**<br>We use functional dependencies to:<br>test relations to see if they are legal under a given set of functional dependencies.<br>If a relation *r* is legal under a set *F* of functional dependencies, we say that *r* satisfies *F*.<br>specify constraints on the set of legal relations<br>We say that *F* holds on *R* if all legal relations on *R* satisfy the set of functional dependencies *F*. | C212.2 | BTL1 |
| 18. | **Define normalization**. **Nov/Dec2009.April/May2010**<br>Normalization of data is a process during which unsatisfactory relation schemas are decomposed by breaking up their attributes into smaller relation shemas that possess desirable properties. | C212.2 | BTL2 |
| 19. | **List the functional dependencies rules?** | C212.2 | BTL1 |

| | | | |
|---|---|---|---|
| | <br>1. ***Reflexivity Rule*** --- If $X$ is a set of attributes and $Y$ is a subset of $X$, then $X \rightarrow Y$ holds. each subset of $X$ is functionally dependent on $X$.<br>2. ***Augmentation Rule*** --- If $X \rightarrow Y$ holds and $W$ is a set of attributes, and then $WX \rightarrow WY$ holds.<br>3. ***Transitivity Rule*** --- If $X \rightarrow Y$ and $Y \rightarrow Z$ holds, then $X \rightarrow Z$ holds.<br>**Derived Theorems from Axioms**<br>4. ***Union Rule*** --- If $X \rightarrow Y$ and $X \rightarrow Z$ holds, then $X \rightarrow YZ$ holds.<br>5. ***Decomposition Rule*** --- If $X \rightarrow YZ$ holds, then so do $X \rightarrow Y$ and $X \rightarrow Z$.<br>6. ***Pseudo transitivity Rule*** --- If $X \rightarrow Y$ and $WY \rightarrow Z$ hold then so does $WX \rightarrow Z$. | | |
| 20. | What is normalization?<br>**Database normalization** is the process of removing redundant data from the tables in to improve storage efficiency, data integrity, and scalability.<br>In the relational model, methods exist for quantifying how efficient a database is. These classifications are called **normal forms** (or **NF**), and there are algorithms for converting a given database between them.<br>Normalization generally involves splitting existing tables into multiple ones, which must be re-joined or linked each time a query is issued. | C212.2 | BTL1 |
| 21. | **Define data Anomalies.**<br>**Data Anomalies**<br>Data anomalies are inconsistencies in the data stored in a database as a result of an operation such as update, insertion, and/or deletion. Such inconsistencies may arise when have a particular record stored in multiple locations and not all of the copies are updated. | C212.2 | BTL1 |
| 22. | **Define 1NF.**<br>Each table has a primary key: minimal set of attributes which can uniquely identify a record<br>The values in each column of a table are atomic (No multi-value attributes allowed). There are no repeating groups: two columns do not store similar information in the same table | C212.2 | BTL1 |
| 23. | **Define 2NF.**<br>where a key has more than one attribute, check that each non-key attribute depends on the whole key and not part of the key<br>for each subset of the key which determines an attribute or group of attributes create a new form. Move the *dependant* attributes to the new form.<br>Add the part key to new form, making it the primary key.<br>Mark the part key as a foreign key in the original form. | C212.2 | BTL1 |
| 24. | **Define 3NF.**<br>A relation R is in Third Normal Form (3NF) if and only if it is:<br>in Second Normal Form.<br>Every non-key attribute is non-transitively dependent on the primary key. | C212.2 | BTL1 |

| | | | |
|---|---|---|---|
| 25. | **Define BCNF**<br>**Boyce-Codd Normal Form:**<br>• A relation is in Boyce-Codd normal form (BCNF) if for every FD A B either<br>○ B is contained in A (the FD is trivial), or<br>A contains a candidate key of the relation,<br>• In other words: every determinant in a non-trivial dependency is a (super) key.<br>The same as 3NF except in 3NF we only worry about non-key Bs<br>If there is only one candidate key then 3NF and BCNF are the same | C212.2 | BTL1 |
| 26. | **List the decomposition properties. April/May 2017**<br>**Decomposition Properties**<br>• Lossless: Data should not be lost or created when splitting relations up<br>Dependency preservation: It is desirable that FDs are preserved when splitting relations up<br>Normalisation to 3NF is always lossless and dependency preserving<br>Normalisation to BCNF is lossless, but may not preserve all dependencies | C212.2 | BTL1 |
| 27. | **Define MVD.**<br>A multivalued dependency is a *full constraint* between two sets of attributes in a relation.<br>In contrast to the *functional independency*, the multivalued dependency requires that certain tuples be present in a relation. Therefore, a multivalued dependency is also referred as a *tuple-generating* dependency. The multivalued dependency also plays a role in 4NF normalization. | C212.2 | BTL1 |
| 28. | **Define 4NF.**<br>**4th Normal Form**<br>A Boyce Codd normal form relation is in fourth normal form if<br>there is no multi value dependency in the relation or<br>there are multi value dependency but the attributes, which are multi value dependent on a specific attribute, are dependent between themselves.<br>sume the following relation<br>**(a:pk1, b:pk2, c:pk3)**<br>Recall that a relation is in BCNF if all its determinant are candidate keys, in other words each determinant can be used as a primary key.<br>Because relation **R** has only one determinant **(a, b, c)**, which is the composite primary key and since the primary is a candidate key therefore R is in BCNF. | C212.2 | BTL1 |
| 29. | **Define 5NF**<br>FIFTH NORMAL FORM<br>R is in 5NF if and only if every join dependency in R is implied by the candidate keys of R<br>5NF is always achievable. | C212.2 | BTL1 |

| | | | |
|---|---|---|---|
| | a join dependency, * (A, B, …, Z), is implied by the candidate keys, K1, …, Km of R<br>ω the fact that K1, …, Km are candidate keys for R determine the fact that R has the JD * (A, B, …, Z) | | |
| 30. | Show the binary relationship in diagram?<br><br> A. ONE-TO-ONE<br>EMPLOYEE<br>WORKSTATION<br>is assigned<br>Every employee is assigned one workstation; not all workstations are assigned to employees.<br>B. ONE-TO-MANY<br>DEPARTMENT<br>PROJECT<br>is responsible<br>A department may be responsible for many projects but each project is the responsibility of one department<br>C. MANY-TO-MANY<br>EMPLOYEE<br>PROJECT<br>is assigned<br>has assigned<br>Employees may be assigned to many projects; every project has assigned at least one employee | C212.2 | BTL2 |
| 31. | List the properties of a database relation.<br>Properties of database relations are:<br>• relation name is distinct from all other relations<br>• each cell of relation contains exactly one atomic (single) value<br>• each attribute has a distinct name<br>• values of an attribute are all from the same domain<br>• order of attributes has no significance<br>• each tuple is distinct; there are no duplicate tuples<br>• order of tuples has no significance, theoretically. | C212.2 | BTL1 |
| 32. | List the various object-based logical models.<br>– Entity-relationship model<br>– Object-oriented model<br>– Semantic model<br>– Functional model | C212.2 | BTL1 |
| 33. | arious Record-based logical models<br>– Relational model (e.g., SQL/DS, DB2)<br>– Network model<br>– Hierarchical model (e.g., IMS) | C212.2 | BTL1 |

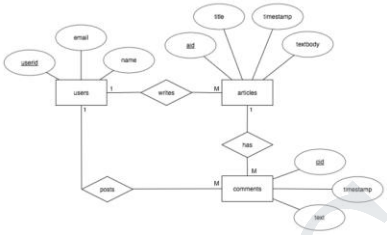
<table>
<tr><td>34.</td><td>What is a domain in a database.<br>The term domain is used to refer to a set of values of the same kind or type. For example, the column, "Cname‟ and "Ccity‟ in the following figure, both have values of type string (ie. valid values are any string). But they denote different domains, ie."Cname‟ denotes the domain of customer names while "Ccity‟ denotes the domain of city names. They are different domains even if they share common values.<br>
<table>
<tr><td></td><td colspan="4">Customer</td></tr>
<tr><td></td><td>C#</td><td>Cname</td><td>Ccity</td><td>Cphone</td></tr>
<tr><td></td><td>1</td><td>Codd</td><td>London</td><td>2263035</td></tr>
<tr><td>Tuple</td><td>2</td><td>Martin</td><td>Paris</td><td>5555910</td></tr>
<tr><td></td><td>3</td><td>Deen</td><td>London</td><td>2234391</td></tr>
</table></td><td>C212.2</td><td>BTL1</td></tr>
<tr><td>35.</td><td><b>What is candidate Keys?</b><br>Candidate keys are those keys which is candidate for primary key of a table. Such keys will full fill all the requirements of primary key which is not null and have unique records is a candidate for primary key. So thus type of key is known as candidate key. Every table must have at least one candidate key but at the same time can have several.</td><td>C212.2</td><td>BTL1</td></tr>
<tr><td>36.</td><td><b>Define the Primary Key.</b><br>Unique attribute of a table will be considered as a primary key. Primary keys are used to identify tables. There is only one primary key per table.</td><td>C212.2</td><td>BTL1</td></tr>
<tr><td>37.</td><td><b>What is Foreign Key? April/May 2018</b><br>Foreign key are those keys which is used to define relationship between two tables. When we want to implement relationship between two tables then we use concept of foreign key. It is also known as referential integrity. We can create more than one foreign key per table. foreign key is generally a primary key from one table that appears as a field in another where the first table has a relationship to the second. In other words, if we had a table A with a primary key X that linked to a table B where X was a field in B, then X would be a foreign key in B.</td><td>C212.2</td><td>BTL1</td></tr>
<tr><td>38.</td><td><b>Define alternate Key.</b><br>If any table have more than one candidate key, then after choosing primary key from those candidate key, rest of candidate keys are known as an alternate key of that table. Suppose we have a table named Employee which has two columns EmpID and EmpMail, both have not null attributes and unique value. If EmpID is considered as a primary key to that table then EmpMail is known as alternate key.</td><td>C212.2</td><td>BTL1</td></tr>
<tr><td>39.</td><td><b>Define Composite Key</b><br>When a key is created on more than one column then that key is known as composite key. Consider a table "Student" which has two columns Sid and SrefNo and these two columns are used as a primary key for record extraction, then this key is known as composite key.</td><td>C212.2</td><td>BTL1</td></tr>
</table>

| | | | |
|---|---|---|---|
| 40. | <br>**Distinguish between key and super key? April/May 2017**<br>A **Superkey,** SK, specifies a uniqueness constraint that no two distinct tuples in any state r of R can have the same value for SK.<br>A **Key ,** K, of R is a Superkey of R with additional property that removing any attribute A, from K leaves set of attributes, that is not a superkey any more.<br>So the difference between Superkey and key is that removing any attribute from superkey may or may not leave set of attributes that are superkey but removing any attribute from a key will give a set that will no more be a super key. | C212.2 | |
| 41. | List the similarities between primary and candidate keys.<br>Both Primary and Candidate keys can uniquely identify records in a table on the database.<br>Both Primary and Candidate keys have constraints UNIQUE and NOT NULL.<br>Primary key or Candidate keys can be either single column or combination of multiple columns in a table. | C212.2 | BTL1 |
| 42. | What are the characteristics of primary key?<br>Primary key is a special kind of index in that,<br>• there can be only one;<br>• it cannot be nullable<br>• it must be unique. | C212.2 | BTL1 |
| 43. | Define super key.<br>A super key is a set or one of more columns (attributes) to uniquely identify rows in a table. | C212.2 | BTL1 |
| 44. | **What is need for normalization**. **Nov/Dec 2010**<br>To ensure that the update anomalies do not occur.<br>Normal forms provide a formal frame work for analyzing relation shemas based on their keys and on the functional dependencies among their attributes.<br>A series of tests that can be carried out on individual relation schemas so that the relation database can be normalized to any degree. When a test fails , the relation violating that test must be decomposed into relations that individually meet the normalization tests | C212.2 | BTL1 |
| 45. | **Define lossless joins (or) What is non additive property. Nov/Dec 2011**<br>Lossless join property or non additive property ensures that no spurious tuples(tuples containing wrong information) are generated when a natural join operation is applied to the relations in the decomposition. | C212.2 | BTL1 |
| 46. | **Explain BCNF with example(or) How to convert a relation into BCNF. Nov/dec 2007, 2008, 2009 2010, 2011,2014**<br>Boyce-Codd Normal form: It is stricter than 3NF, meaning that every relation in BCNF is also in 3NF; however a relation in 3NF is not necessarily in BCNF. A relation is in BCNF if and only if every determinant is a candidate key (i.e) a relatioln schema R is in BCNF if whenever a functional dependency X->A holds in R, then X is a superkey of R | C212.2 | BTL1 |
| 47. | **What are pitfalls in relational database design?Nov/Dec2009 2010**<br>○ Repetition of informaition | C212.2 | BTL1 |

| | | | |
|---|---|---|---|
| | ○ Inability to represent certain information<br>○ Loss of information | | |
| 48. | **State the anomalies of 1NF. Nov/ Dec 2015.**<br>No repeating groups<br>Redundancy of data | C212.2 | BTL1 |
| 49. | **Explain the desirable properties of decomposition.**<br>Lossless-join decomposition<br>Dependency preservation<br>Repetition of information | C212.2 | BTL1 |
| 50. | **Outline the steps involved in query processing. April/May 2018.**<br>Parsing and translation<br>Optimization<br>Evaluation | C212.2 | BTL1 |
| 51. | **Explain trivial dependency? Nov/Dec 2009, 2010**<br>Functional dependency of the form α ->β. is trivial if α C β. Trivial functional dependencies are satisfied by all the relations. | C212.2 | BTL1 |
| 52. | **Why must multivalued dependencies exist in pairs? Nov/DEC 2010, 2011**<br>Multivalued dependencies: Multivalued dependencies are a result of 1NF which disallowed an attribute in a tuple to have a set of values. If we have two or more multivalued independent attributes in the same relation schema, we get into the problem of having to repeat every value of one of the attributes with every value of the other attributes with every value of the other attribute to keep the relation instances consistent. A multivalued dependency X→>Y specified on relation schema R where X and Y are subsets of R specifies the following constraint on any relation r of R. | C212.2 | BTL1 |
| 53. | **Comparison of BCNF and 3NF: Nov/Dec 2010, April/may2011**<br>3NF design is always dependency preserving and lossless.<br>dependency preserving is difficult to achieve in BCNF sometimes.<br>BCNF strictly removes transitive dependency.<br>BCNF relation is in 3NF , but reverse is not possible | C212.2 | BTL1 |
| 54. | **Why 4NF in Normal Form is more desirable than BCNF? (Nov/Dec 2014, Dec 2017)**<br>BCNF (Boyece code normal form) has all functional dependencies A to B are trivial of discriminator should be superkey. To get relation in BCNF, Splitting the relation schema not necessarily preserve all functional dependency, Loss less decomposition and dependency are main points for the normalization sometime, it is not possible to get a BCNF decomposition that is dependency, preserving. While 4NF has very similar definition as BCNF. A relational Schema is in 4NF, if all multivalued dependencies A to B are trivial and determinate A is superkey of schema. If a relational schema is in 4nf, it is already in BCNF. and 4NF decomposition preserve the all functional dependency. so 4NF is preferable than to have BCNF. | C212.2 | BTL1 |

## PART-B

| Q. NO. | QUESTIONS | CO | BLOOM'S LEVEL |
|---|---|---|---|
| 1 | **Explain The Different Types Of Normalization Nov/Dec 2009, 2010, 2011, Nov/ Dec 2014, Apr/May 2015, Nov/Dec 2016**<br>*Refer the Elmasri Navathe page no. 517* | C212.2 | BTL5 |
| 2 | **Explain In Detail About Boyce Codd Normal Form And Non Lossless Join Dependency. Nov/Dec 2009, 2010, 2011**<br>*Refer the Elmasri Navathe page no. 529* | C212.2 | BTL5 |
| 3 | **Give Detail explanation about Data Model & its types. May/ June 2009, Nov/ Dec 2014**<br>*Refer the Elmasri Navathe page no. 30* | C212.2 | BTL5 |
| 4 | **Explain the various components of ER diagram with examples Briefly.**<br>*Refer the Elmasri Navathe page no. 203* | C212.2 | BTL5 |
| 5 | **Draw ER Diagram for Difference applications ( Eg. Banking Systems . Nov/ Dec 2014, Nov/Dec 2017, Restaurant menu ordering system Apr/ May 2015, Car Rental company Nov/ Dec 2015), Car Insurance Company (Nov/Dec 2016)** | C212.2 | BTL5 |
| 6 | **Discuss the correspondence between the ER model construct and the relational model constructs. Show how each ER model construct can be mapped to the relational model. Discuss the option for mapping EER model construct. April/May 2017** | C212.2 | BTL5 |
| 7 | **Various join strategies with Example. Nov/Dec 2016**<br>**Nested loop join**<br>**Block nested loop join**<br>**Merge join**<br>**Hash join** | C213.2 | BTL6 |
| 8 | **Differentiate between foreign key constraints and referential integrity constraints with suitable example. Nov/Dec 2017** | C213.2 | BTL5 |
| 9 | **Distinguish between lossless-join decomposition and dependency preserving decomposition. Nov/Dec 2017.** | C213.2 | BTL5 |
| 10 | **Explain FNF, SNF, TNF and DCNF with an example. Nov/Dec 2016** | C213.2 | BTL1 |
| 11 | **Give detail Explanation about Query processing / Optimization overview with example. Nov/Dec 2014, May/ June 2016, Nov/Dec 2016, Nov/Dec 2017.**<br>*Elmasri Navathe page no.679* | C213.2 | BTL1 |
| 12 | | C213.2 | BTL5 |

| | | | |
|---|---|---|---|
| | A car rental company maintains a database for all vehicles in its current fleet. For all vehicles, it includes the vehicle identification number, license number, manufacturer, model, date of purchase, and color. Special data are included for certain types of vehicles.<br>• Trucks: cargo capacity.<br>• Sports cars: horsepower, renter age requirement.<br>• Vans: number of passengers.<br>• Off-road vehicles: ground clearance, drivetrain (four- or two-wheel drive).<br>Construct an ER model for the car rental company database. ***(Nov/Dec 15)*** | | |
| 13 | Construct an E-R diagram for a car insurance company whose customers cars each. Each car has associated with it zero to any number of recorde insurance policy covers one or more cars, and has one or more premium associated with it. Each payment is for a particular period of time set of date when the payment was received. ***(Nov/Dec 16)*** | C213.2 | BTL5 |
| 14 | For the ER diagram given below explain ER to Relational mapping procedures<br><br> userid<br>email<br>name<br>users<br>1<br>writes<br>M<br>articles<br>aid<br>title<br>timestamp<br>textbody<br>1<br>has<br>M<br>1<br>posts<br>M<br>comments<br>cid<br>timestamp<br>text | C213.2 | BTL1 |
| 15 | Draw E – R Diagram for the “Restaurant Menu Ordering System”, whic food items ordering and services within a restaurant. The entire restaura detailed as follows. The Customer is able to view the food items menu, c orders and obtain the final bill through the computer kept in their table. their wireless tablet PC are able to initialize a table for customers, contro to assist customers, orders, send orders to food preparation staff (chef) a customer’s bill. The food preparation staffs (Chefs), with their touch-disp system, are able to view orders sent to the kitchen by waiters. During pr able to let the waiter know the status of each item, and can send notificat completed. The system should have full accountability and logging facili support supervisor actions to account for exceptional circumstances, suc refunded or walked out on. ***(Apr/May 17)*** | C213.2 | BTL1 |

## UNIT III

## TRANSACTIONS

Transaction Concepts – ACID Properties – Schedules – Serializability – Concurrency Control – Need for Concurrency – Locking Protocols – Two Phase Locking – Deadlock – Transaction Recovery - Save Points – Isolation Levels – SQL Facilities for Concurrency and Recovery

| Q. No. | Questions | CO | Bloom's Level |
|---|---|---|---|
| 1 | **What is transaction?Nov/Dec 2010 Nov/Dec 2014**<br>Collections of operations that form a single logical unit of work are called transactions. | C212.3 | BTL1 |
| 2 | **What are the properties of transaction? Nov /Dec 2009, April/May2010 NOV/DEC 2014 , May/June 2016 (or) What are the ACID properties? APRIL/MAY-2011 , Nov/Dec 2017,**<br>The properties of transactions are:<br>Atomicity<br>Consistency<br>Isolation<br>Durability | C212.3 | BTL1 |
| 3 | **Define lock? Nov2009, Nov2010, Nov2011**<br>Lock is the most common used to implement the requirement is to allow a transaction to access a data item only if it is currently holding a lock on that item. | C212.3 | BTL1 |
| 4 | **What are the different modes of lock? Nov2009, Nov2011**<br>The modes of lock are:<br>Shared /Read<br>Exclusive /Write | C212.3 | BTL1 |
| 5 | **Define deadlock? April 2009,April 2010, April2011**<br>Neither of the transaction can ever proceed with its normal execution. This situation is called deadlock. | C212.3 | BTL1 |
| 6 | **Define the phases of two phase locking protocol April/May 2009**<br>Growing phase: a transaction may obtain locks but not release any lock.<br>Shrinking phase: a transaction may release locks but may not obtain any new locks. | C212.3 | BTL1 |
| 7 | **What is meant by log-based recovery? April 2009**<br>The most widely used structures for recording database modifications is the log.<br>The log is a sequence of log records, recording all the update activities in the database. There are several types of log records. | C212.3 | BTL1 |
| 8 | **Define shadow paging. Nov/Dec 2009** | C212.3 | BTL1 |

| | | | |
|---|---|---|---|
| | An alternative to log-based crash recovery technique is shadow paging. This technique needs fewer disk accesses than do the log-based methods. | | |
| 9 | **When is a transaction rolled back?**<br>Any changes that the aborted transaction made to the database must be undone.<br>Once the changes caused by an aborted transaction have been undone, then the transaction has been rolled back. | C212.3 | BTL1 |
| 10 | **What is a recovery scheme? April/May 2009**<br>An integral part of a database system is a recovery scheme that can restore the database to the consistent state that existed before the failure. | C212.3 | BTL1 |
| 11 | **What are two pitfalls (problem) of lock-based protocols? APRIL/MAY-2011**<br>Deadlock<br>Starvation | C212.3 | BTL2 |
| 12 | **What are the two statements regarding transaction?**<br>The two statements regarding transaction of the form: ¬ Begin transaction ¬ End transaction | C212.3 | BTL1 |
| 13 | **What is recovery management component?**<br>Ensuring durability is the responsibility of a software component of the base system called the recovery management component. | C212.3 | BTL1 |
| 14 | **When is a transaction rolled back?**<br>Any changes that the aborted transaction made to the database must be undone. Once the changes caused by an aborted transaction have been undone, then the transaction has been rolled back. | C212.3 | BTL3 |
| 15 | **What is a shadow copy scheme?**<br>It is simple, but efficient, scheme called the shadow copy schemes. It is based on making copies of the database called shadow copies that one transaction is active at a time. The scheme also assumes that the database is simply a file on disk. | C212.3 | BTL1 |
| 16 | **Give the reasons for allowing concurrency? Nov/Dec 2017**<br>The reasons for allowing concurrency is if the transactions run serially, a short transaction may have to wait for a preceding long transaction to complete, which can lead to unpredictable delays in running a transaction. So concurrent execution reduces the unpredictable delays in running transactions. | C212.3 | BTL1 |
| 17 | **What is average response time?**<br>The average response time is that the average time for a transaction to be completed after it has been submitted. | C212.3 | BTL1 |
| 18 | **What is serializability? Explain its types? NOV/DEC 2014, Nov/Dec 2016, April/May 2018**<br>A (possibly concurrent) schedule is serializable if it is equivalent to a serial schedule. Different forms of schedule equivalence give rise to the notions of:<br>• **conflict serializability**<br>• **view serializability** | C212.3 | BTL2 |

| | | | |
|---|---|---|---|
| 19 | **Define upgrade and downgrade?**<br>It provides a mechanism for conversion from shared lock to exclusive lock is known as upgrade. It provides a mechanism for conversion from exclusive lock to shared lock is known as downgrade. | C212.3 | BTL1 |
| 20 | **What is a database graph?**<br>The partial ordering implies that the set D may now be viewed as a directed acyclic graph, called a database graph. | C212.3 | BTL1 |
| 21 | **What are the two methods for dealing deadlock problem?**<br>The two methods for dealing deadlock problem is<br>deadlock detection<br>deadlock recovery. | C212.3 | BTL1 |
| 22 | **What is a recovery scheme?**<br>An integral part of a database system is a recovery scheme that can restore the database to the consistent state that existed before the failure. | C212.3 | BTL1 |
| 23 | **Define garbage collection. May/June 2016**<br>Garbage may be created also as a side effect of crashes. Periodically, it is necessary to find all the garbage pages and to add them to the list of free pages. This process is called garbage collection. | C212.3 | BTL1 |
| 24 | **Differentiate strict two phase locking protocol and rigorous two phase locking protocol. May/June 2016**<br>In strict two phase locking protocol all exclusive mode locks taken by a transaction is held until that transaction commits. Rigorous two phase locking protocol requires that all locks be held until the transaction commits. | C212.3 | BTL1 |
| 25 | **How the time stamps are implemented**<br>Use the value of the system clock as the time stamp. That is a transaction‟s time stamp is equal to the value of the clock when the transaction enters the system. • Use a logical counter that is incremented after a new timestamp has been assigned; that is the time stamp is equal to the value of the counter. | C212.3 | BTL1 |
| 26 | **What are the time stamps associated with each data item?**<br>W-timestamp (Q) denotes the largest time stamp if any transaction that executed WRITE (Q) successfully. • R-timestamp (Q) denotes the largest time stamp if any transaction that executed READ (Q) successfully. | C212.3 | BTL1 |
| 27 | **Define blocks? What are its types?**<br>The database system resides permanently on nonvolatile storage, and is partitioned into fixedlength storage units called blocks.<br>**Physical blocks**<br>The input and output operations are done in block units. The blocks residing on the disk are referred to as physical blocks.<br>**Buffer blocks**<br>The blocks residing temporarily in main memory are referred to as buffer blocks | C212.3 | BTL1 |
| 28 | **What is meant by concurrency control? Nov/Dec 2015**<br>Process of managing simultaneous operations on the database without | C212.3 | BTL1 |

| | | | |
|---|---|---|---|
| | having them interfere with one another.<br>Prevents interference when two or more users are accessing database simultaneously and at least one is updating data. | | |
| 29 | **Give an example of Two phase commit protocol. Nov/Dec 2015**<br>In distributed databases, "single phase" commit is what is used. All the necessary information for the transaction to be undone or completed are written to persistent storage (typically called a "transaction log") in an atomic step. The transaction is committed as soon as this information is permanently recorded. This is not possible in a distributed system, as there's no guarantee that the commit record is written on all participating systems. With distributed databases, Two Phase Commit solves this problem. | C212.3 | BTL2 |
| 30 | **List the four conditions for deadlock. Nov/Dec 2016**<br>1. **mutual exclusion**: at least one process must be held in a non-sharable mode.<br>2. **hold and wait**: there must be a process holding one resource and waiting for another.<br>3. **No preemption**: resources cannot be preempted.<br>4. **circular wait:** there must exist a set of processes [p1, p2, ..., pn] such that p1 is waiting for p2, p2 for p3, and so on upto pn. | C212.3 | BTL1 |
| 31 | **What is serializible schedule? April/May 2017**<br>To process transactions concurrently, the database server must execute some component statements of one transaction, then some from other transactions, before continuing to process further operations from the first. The order in which the component operations of the various transactions are interleaved is called the **schedule** | C212.3 | BTL1 |
| 32 | What type of locking needed for insert and delete operations? **April/May 2017**<br>**Share/Exclusive (for Read/Write) Locks**<br>We should allow several transactions to access the same item *A* if they all access *A′* for reading purposes only. However, if a transaction is to write an item A, it must have exclusive access to *A*. For this purpose, a different type of lock called a multiple-mode lock is used. In this scheme there are shared/exclusive or read/write locks are used. | C212.3 | BTL1 |
| 33 | **State the difference between a shared lock and an exclusive lock. April/May 2018.**<br>*Shared lock:*<br>Shared locks are placed on resources whenever a read operation (select) is performed.<br>Multiple shared locks can be simultaneously set on a resource.<br>*Exclusive lock:*<br>Exclusive locks are placed on resources whenever a write operation (INSERT, UPDATE And DELETE) are performed.<br>Only one exclusive lock can be placed on a resource at a time. | C212.3 | BTL1 |

| | | | |
|---|---|---|---|
| | i.e. the first user who acquires an exclusive lock will continue to have the sole ownership of the resource, and no other user can acquire an exclusive lock on that resource | | |
| 34 | **What is query execution plan? April/May 2017**<br>The Query Execution Plans describe the steps and the order used to access or modify data in the database. | C212.3 | BTL1 |
| 35 | **What are the costs involved in query execution? April/May 2017**<br>i. **Access cost to secondary storage.**<br>ii. Disk storage cost.<br>iii. Computation cost<br>iv. Memory usage cost and<br>v. Communication cost | C212.3 | BTL1 |
| 36 | **Define atomicity?**<br>Either all operations of the transaction are reflected properly in the database or none are. | C212.3 | BTL1 |
| 37 | **Define Durability?**<br>A transaction completes successfully then changes It has made to the database persist even if there are system failure. | C212.3 | BTL1 |
| 38 | **What is transaction-management component?**<br>Ensuring atomicity is the responsibility of the database system itself specifically, it is handled by a component called the transaction-management component. | C212.3 | BTL1 |
| 39 | **What are the two operation for accessing data in transaction?**<br>Read(x)- transfer data item x from database.<br>Write(x)- transfer data item x from the local buffer. | C212.3 | BTL1 |
| 40 | **What do you mean by read only transaction?**<br>The data base operation in a transaction do not update the database but only retrieve data, the transaction is called a read-only transaction. | C212.3 | BTL1 |
| 41 | **What are the steps followed in Executing read(x)command in transaction?**<br>1. Find the address of the disk block that contains item x.<br>2. Copy that disk block in to a buffer in main memory.<br>3. Copy item x from the buffer to the program variable named x. | C212.3 | BTL1 |
| 42 | **What are the steps followed in executing write(x) command in transaction?**<br>1. find the address of disk block that contain item x.<br>2. Copy that disk block into buffer in main memory.<br>3. Copy item x from the program variable named x into its correct location in the buffer<br>4. Store the update block from the buffer back to disk. | C212.3 | BTL1 |
| 43 | **List out the transaction states?**<br>i) active ii) Partially Committed iii) Failed iv) Abort v) committed. | C212.3 | BTL1 |
| 44 | **What are the need for concurrency?**<br>i) Improved throughput and resource utilization<br>ii) Reduced waiting time. | C212.3 | BTL1 |

| | | | |
|---|---|---|---|
| 45 | **Define Schedule**.<br>The chromosomal order in which instructions are executed in the system. | C212.3 | BTL1 |
| 46 | **When the schedule is called serial?**<br>Each serial schedule consists of a sequence of instruction from various transaction where the instruction belonging to one single transaction appear together in that schedule. | C212.3 | BTL1 |
| 47 | **When two operation in schedule are said to be conflict?**<br>i) Two operation belong to different transaction<br>ii) Two operation access the same item x<br>iii) At least one of the operation is write-item (x) | C212.3 | BTL1 |
| 48 | **Define cascading rollback?**<br>An uncommitted transaction has to be roll back because it read an item from a transaction that failure. | C212.3 | BTL1 |
| 49 | **When the schedule is said to cascade less?**<br>A schedule is said to be cascadeless or avoid cascading roll back if every transaction in the schedule reads only items that were written by committed transactions. | C212.3 | BTL1 |
| 50 | **Define lock table?**<br>System maintain record for the items that are currently locked in lock table that which could be organized as a hash file. | C212.3 | BTL1 |
| 51 | **What you mean by lock conversion?**<br>A transaction that already holds a lock on item xis allowed under certain condition to voncery the Locke from one locked state to another. | C212.3 | |

## PART-B

| Q. NO. | QUESTIONS | CO | BLOOM’S LEVEL |
|---|---|---|---|
| 1 | **Describe Log Based Recovery May/june 2008**<br>*Refer Elmasri Navathe page no.612* | C212.3 | BTL4 |
| 2 | **What is Serializability? Explain Its Types? April2009, 2011 Nov/Dec 2014 , April/May 2018** | C212.3 | BTL2 |
| 3 | **Write Short Notes On Transaction Concept & Transaction State? April 2011 Nov/Dec 2014, April/May 2018**<br>*Refer Elmasri Navathe page no.559, 562* | C212.3 | BTL3 |
| 4 | **Briefly Describe Concurrency control execution? Illustrate with a suitable example. Nov/ Dec 2015, April/May 2018**<br>*Refer Elmasri Navathe page no.583* | C212.3 | BTL1 |
| 5 | **Explain in terms of Locking mechanism and Two Phase Commit Protocol Nov/Dec 2014, April / May 2015, May/June 2016, Nov/Dec 2017** | C212.3 | BTL2 |

| | | | |
|---|---|---|---|
| | *Refer the Elmasri Navathe page no 584* | | |
| 6 | **Explain about locking protocols. May/June 2016** *Refer the Elmasri Navathe page no 584* | C212.3 | BTL2 |
| 7 | **Discuss about conflict and view serializability. April/ May 2015. Nov/ Dec 2015**<br>*Refer the Elmasri Navathe page no 570* | C212.3 | BTL5 |
| 8 | **Explain Deadlock in detail with an example. . Nov/Dec 2014 , Nov/ Dec 2015, Nov/Dec 2017**<br>*Refer the Elmasri Navathe page no 591* | C212.3 | BTL2 |
| 9 | **Briefly describe two phase locking in concurrency control techniques. (Nov/Dec 2014)**<br>*Refer the Elmasri Navathe page no 584* | C212.3 | BTL1 |
| 10 | **Explain the concepts of concurrent execution in Transaction processing system. (Nov/Dec 2014)**<br>*Refer Elmasri Navathe page no.583* | C212.3 | BTL2 |
| 11 | **Give detail explanation about ACID Properties Nov/Dec 2009,April 2011**<br>*Refer Elmasri Navathe page no.562* | C212.3 | BTL4 |
| 12 | **State and explain the lock based concurrency control with suitable examples? Nov/Dec 2017** | C212.3 | BTL4 |
| 13 | **Discuss the violations caused by each of the following: dirty read, non-repeatable read and phantoms with suitable example. April/May 2017.** | C212.3 | BTL5 |
| 14 | **Explain why timestamp-based concurrency control allows schedules that are not recoverable. Describe how it can be modified through buffering to disallow such schedules. April/May 2017.** | C212.3 | BTL1 |
| 15 | **How can you implement atomicity in transactions? Explain.** | C212.3 | BTL6 |

## UNIT IV

### *IMPLEMENTATION TECHNIQUES*

RAID – File Organization – Organization of Records in Files – Indexing and Hashing –Ordered Indices – B+ tree Index Files – B tree Index Files – Static Hashing – Dynamic Hashing – Query Processing Overview – Algorithms for SELECT and JOIN operations – Query optimization using Heuristics and Cost Estimation

## PART – A

| Q. No. | Question | CO | Bloom's Level |
|---|---|---|---|
| 1 | **What is B-Tree?**<br>□ A B-tree eliminates the redundant storage of search-key values.<br>□ It allows search key values to appear only once. | C213.4 | BTL1 |

| | | | |
|---|---|---|---|
| 2 | **What is a B+-Tree index?**<br>A B+-Tree index takes the form of a balanced tree in which every path from the root of the root of the root of the tree to a leaf of the tree is of the same length | C213.4 | BTL4 |
| 3 | **What is a hash index?**<br>A hash index organizes the search keys, with their associated pointers, into a hash file structure | C213.4 | BTL1 |
| 4 | **Define seek time.**<br>The time for repositioning the arm is called the seek time and it increases with the distance that the arm is called the seek time. | C213.4 | BTL1 |
| 5 | **Define rotational latency time.**<br>The time spent waiting for the sector to be accessed to appear under the head is called the rotational latency time. | C213.4 | BTL1 |
| 6 | **What is called mirroring?**<br>The simplest approach to introducing redundancy is to duplicate every disk. This technique is called mirroring or shadowing. | C213.4 | BTL1 |
| 7 | **What are the two main goals of parallelism?**<br>□ Load –balance multiple small accesses, so that the throughput of such accesses increases.<br>□ Parallelize large accesses so that the response time of large accesses is reduced | C213.4 | BTL1 |
| 8 | **What is an index?**<br>An index is a structure that helps to locate desired records of a relation quickly, without examining all records | C213.4 | BTL1 |
| 9 | **What are the factors to be taken into account when choosing a RAID level?**<br>□ Monetary cost of extra disk storage requirements.<br>□ Performance requirements in terms of number of I/O operations<br>□ Performance when a disk has failed and Performances during rebuild. | C213.4 | BTL1 |
| 10 | **What are the types of storage devices?**<br>Primary storage, Secondary storage, Tertiary storage, Volatile storage, Nonvolatile storage | C213.4 | BTL1 |
| | | C213.4 | BTL1 |

| | | | |
|---|---|---|---|
| 11 | **What is called remapping of bad sectors?**<br>If the controller detects that a sector is damaged when the disk is initially formatted, or when an attempt is made to write the sector, it can logically map the sector to a different physical location. | | |
| 12 | **Define software and hardware RAID systems?*(May/June 16)***<br>RAID can be implemented with no change at the hardware level, using only software modification. Such RAID implementations are called software RAID systems and the systems with special hardware support are called hardware RAID systems. | C213.4 | BTL1 |
| 13 | **Define hot swapping?**<br>Hot swapping permits the removal of faulty disks and replaces it by new ones without turning power off. Hot swapping reduces the mean time to repair. | C213.4 | BTL1 |
| 14 | **What are the ways in which the variable-length records arise in database systems?**<br>Storage of multiple record types in a file, Record types that allow variable lengths for one or more fields, Record types that allow repeating fields. | C213.4 | BTL1 |
| 15 | **What are the two types of blocks in the fixed –length representation? Define them.**<br>□<br>Anchor block: Contains the first record of a chain.<br>□ Overflow block: Contains the records other than those that are the first record of a chain. | C213.4 | BTL1 |
| 16 | **What is hashing file organization?**<br>In the hashing file organization, a hash function is computed on some attribute of each record. The result of the hash function specifies in which block of the file the record should be placed. | C213.4 | BTL2 |
| 17 | **What are called index-sequential files?**<br>The files that are ordered sequentially with a primary index on the search key are called index-sequential files. | C213.4 | BTL2 |
| 18 | **Define Primary index and Secondary Index**<br>It is in a sequentially ordered file, the index whose search key specifies the sequential order of the file. Also called clustering index. The search key of a primary index is usually but not necessarily the primary key. It is an index whose search key specifies an order different from the sequential order of the file. Also called non clustering index. | C213.4 | BTL1 |

| | | | |
|---|---|---|---|
| 19 | **Give an example of a join that is not a simple equi-join for which partitioned parallelism can be used.** ***(Nov/Dec 15)***<br>$r \text{ join } (r.A = s.B) \land (r.A < s.c)$ | C213.4 | BTL2 |
| 20 | **Differentiate static and dynamic hashing.** ***(Apr/May 15) (Nov/Dec 14,15)***<br>**Static Hashing**<br>In static hashing, when a search-key value is provided, the hash function always computes the same address. The number of buckets provided remains unchanged at all times i.e. fixed<br>Space and overhead is more<br>As file grows performance decreases<br>**Dynamic Hashing**<br>Hash function, in dynamic hashing, is made to produce a large number of values and only a few are used initially. Dynamic hashing provides a mechanism in which data buckets are added and removed dynamically and on-demand .i.e. no. of buckets not fixed.<br>Minimum space and less overhead<br>Performance do not degrade as file grows | C213.4 | BTL1 |
| 21 | **List out the mechanisms to avoid collision during hashing.*(Nov/Dec 16)***<br>□ In overflow chaining, the overflow buckets of a given bucket are chained together in a linked list.<br>□ Above scheme is called closed hashing. An alternative, called open hashing, which does not use overflow buckets, is not suitable for database applications. | C213.4 | BTL1 |
| 22 | **What are the disadvantages of B-Tree over B+ Tree?** ***(Nov/Dec 16)***<br>□ Only small fraction of all search-key values are found early<br>□ Non-leaf nodes are larger. Thus, B-Trees typically have greater depth than corresponding B+-Tree<br>□ Insertion and deletion more complicated than in B+-Trees<br>□ Implementation is harder than B+-Trees.<br>□ | C213.4 | BTL1 |
| 23 | **What is called query processing?**<br>Query processing refers to the range of activities involved in extracting data from a database. | C213.4 | BTL1 |
| 24 | **What is called a query evaluation plan?**<br>A sequence of primitive operations that can be used to evaluate be | C213.4 | BTL1 |

| | | | |
|---|---|---|---|
| | query is a query evaluation plan or a query execution plan. | | |
| 25 | **Explain “Query optimization”?*(May/June 16)***<br>Query optimization refers to the process of finding the lowest cost method of evaluating a given query. | C213.4 | BTL1 |
| 26 | **State the need for Query Optimization. *(Apr/May 15)***<br>The query optimizer attempts to determine the most efficient way to execute a given query by considering the possible query plans. | C213.4 | BTL1 |
| 27 | **What is meant by software and hardware RAID systems? May/June 2016**<br>RAID can be implemented with no change at the hardware level, using only software modification. Such RAID implementations are called software RAID systems and the systems with special hardware support are called hardware RAID systems. | C213.4 | BTL1 |
| 28 | **What is the use of RAID? April/May2009, Nov/Dec2010**<br>A variety of disk-organization techniques, collectively called redundant arrays of independent disks are used to improve the performance and reliability | C213.4 | BTL1 |
| 29 | **What is known as heap file organization? Nov/Dec 2009**<br>In the heap file organization, any record can be placed anywhere in the file where there is space for the record. There is no ordering of records. There is a single file for each relation. | C213.4 | BTL1 |
| 30 | **What is known as sequential file organization? April/May2009**<br>In the sequential file organization, the records are stored in sequential order, according to the value of a “search key” of each record. | C213.4 | BTL2 |
| 31 | **What are the types of indices?**<br>Ordered indices<br>Hash indices | C213.4 | BTL1 |
| 32 | **What are the advantages and disadvantages of indexed sequential file? APRIL/MAY- 2011**<br>The advantage of ordering records in a sequential file according to a key is that you can then search the file more quickly. If you know the key value that you want, you can use one of the relatively fast searches. The disadvantage is that when you insert, you need to rewrite at least everything after the insertion point, which makes inserts very expensive unless they are done at the end of the file. An indexed file approach keeps a (hopefully) small part of each row, and some kind of "pointer" to the row's location within the data file. This allows a search to use the index, which is ordered by the index and (again hopefully) much smaller and therefore much | C213.4 | BTL1 |

| | | | |
|---|---|---|---|
| | faster than scanning the entire data file for the indexed data. | | |
| 33 | **Compare sequential access devices versus random access devices with an example**<br>sequential access devices random access devices Must be accessed from the beginning It is possible to read data from any location Eg:- tape storage Eg:-disk storage Access to data is much slower Access to data is faster Cheaper than disk Expensive when compared with disk | C213.4 | BTL1 |
| 34 | **Explain how reliability can be improved through redundancy?**<br>The simplest approach to introducing redundancy is to duplicate every disk. This technique is called mirroring or shadowing. A logical disk then consists of two physical disks, and write is carried out on both the disk. If one of the disks fails the data can be read from the other. Data will be lost if the second disk fails before the first fail ed disk is repaired. | C213.4 | BTL1 |
| 35 | **What is database tuning? APRIL/MAY-2011**<br>Database tuning describes a group of activities used to optimize and homogenize the performance of a database. It usually overlaps with query tuning, but refers to design of the database files, selection of the database management system (DBMS), operating system and CPU the DBMS runs on. | C213.4 | BTL1 |
| 36 | **What are a block and a block number?**<br>A block is a contiguous sequence of sectors from a single track of one platter. Each request specifies the address on the disk to be referenced. That address is in the form of a block number. | C213.4 | BTL1 |
| 37 | **What are the benefits of RAID**<br>□ Data loss can be very dangerous for an organization<br>□ RAID technology prevents data loss due to disk failure<br>□ RAID technology can be implemented in hardware or software<br>□ Servers make use of RAID Technology | C213.4 | BTL1 |
| 38 | **Define Bit-Interleaved Parity**<br>□ A single parity bit is enough for error correction, not just detection, since we know which disk has failed | C213.4 | BTL1 |

| | | | |
|---|---|---|---|
| | – When writing data, corresponding parity bits must also be computed and written to a parity bit disk<br>– To recover data in a damaged disk, compute XOR of bits from other disks (including parity bit disk)<br>□ I/O operation addresses all the drives at the same time, RAID 3 cannot overlap I/O. For this reason, RAID 3 is best for single-user systems with long record applications | | |
| 39 | **What is Block-Interleaved Parity**<br>□ When writing data block, corresponding block of parity bits must also be computed and written to parity disk<br>□ To find value of a damaged block, compute XOR of bits from corresponding blocks (including parity block) from other disks | C213.4 | BTL1 |
| 40 | **What are the Types of Ordered Indices**<br>□ Dense index<br>□ Sparse index | C213.4 | BTL1 |
| 41 | **What is Multilevel Index**<br>□ If primary index does not fit in memory, access becomes expensive.<br>□ To reduce number of disk accesses to index records, treat primary index kept on disk as a sequential file and construct a sparse index on it.<br>– outer index – a sparse index of primary index<br>– inner index – the primary index file<br>□ If even outer index is too large to fit in main memory, yet another level of index can be created, and so on | C213.4 | BTL1 |
| 42 | **Define Primary and Secondary Indices**<br>□ Secondary indices have to be dense.<br>□ Indices offer substantial benefits when searching for records.<br>□ When a file is modified, every index on the file must be updated, Updating indices imposes overhead on database modification.<br>□ Sequential scan using primary index is efficient, but a sequential scan using a secondary index is expensive<br>– each record access may fetch a new block from disk | C213.4 | BTL1 |
| 43 | **Draw the structure of B+-Tree Node .**<br>□ **Typical node**<br>– $K_i$ are the search-key values<br>– $P_i$ are pointers to children (for non-leaf nodes) or pointers to records or buckets of records (for leaf nodes).<br>□ The search-keys in a node are ordered<br>$K_1 < K_2 < K_3 < . . . < K_{n-1}$ | C213.4 | BTL1 |
| 44 | **What are the Algorithms for Executing Relational Query Operations**<br>□ An RDBMS must include one or more alternative algorithms that implement each relational algebra operation (SELECT, JOIN,…) and, in many cases, that implement each combination of these operations. | C213.4 | BTL1 |

| | | | |
|---|---|---|---|
| | □ Each algorithm may apply only to particular storage structures and access paths (such index,…).<br>□ Only execution strategies that can be implemented by the RDBMS algorithms and that apply to the particular query and particular database design can be considered by the query optimization module. | | |
| 45 | **What are the Advantages and disadvantages of extendable hashing?**<br>**Advantages of extendable hashing:**<br>– Hash performance does not degrade with growth of file<br>– Minimal space overhead<br>□ **Disadvantages of extendable hashing**<br>– Extra level of indirection to find desired record<br>Bucket address table may itself become very big. | C213.4 | BTL1 |
| 46 | **What are the Cost functions for SELECT Operation**<br>**Linear Search:**<br>– [nBlocks(R)/2], if the record is found.<br>– [nBlocks(R)], if no record satisfied the condition.<br>**Binary Search :**<br>o [log2(nBlocks(R))], if equality condition is on key attribute, because SCA(R) = 1 in this case.<br>o[log2(nBlocks(R))] + [SCA(R)/bFactor(R)] – 1, otherwise. | C213.4 | BTL1 |
| 47 | **Write the Cost functions for JOIN Operation**<br>Join operation is the most time consuming operation to process.<br>□ An estimate for the size (number of tuples) of the file that results after the JOIN operation is required to develop reasonably accurate cost functions for JOIN operations.<br>□ The JOIN operations define the relation containing tuples that satisfy a specific predicate F from the Cartesian product of two relations R and S. | C213.4 | BTL1 |
| 48 | **Define Cost Estimation in Query Optimization**<br>The main aim of query optimization is to choose the most efficient way of implementing the relational algebra operations at the lowest possible cost. | C213.4 | BTL1 |
| 49 | **What is External Sorting ?**<br>It refers to sorting algorithms that are suitable for large files of records on disk that do not fit entirely in main memory, such as most database files.. | C213.4 | BTL1 |
| 50 | **What are Cost Components of Query Execution ?**<br>The cost of executing the query includes the following components:<br>– Access cost to secondary storage.<br>– Storage cost.<br>– Computation cost.<br>– Memory uses cost.<br>– Communication cost. | C213.4 | BTL1 |

## PART – B

| Q. NO. | QUESTIONS | CO | BLOOM'S LEVEL |
|---|---|---|---|
| 1 | Describe File Organization.<br>**April/May2009, 2011** | C213.4 | BTL5 |
| 2 | Define RAID and Briefly Explain RAID techniques.***(Nov/Dec 14, 15, 16) (Apr/May 15,16)*** | C213.4 | BTL5 |
| 3 | Explain Secondary storage devices. | C213.4 | BTL2 |
| 4 | Explain about static and dynamic hashing with an example | C213.4 | BTL5 |
| 5 | Explain about Multidimensional and parallel with an example | C213.4 | BTL3 |
| 6 | Explain about ordered indices with an example | C213.4 | BTL5 |
| 7 | Explain about B+ trees indexing concepts with an example ***(Nov/Dec 14)(May/June 16)*** | C213.4 | BTL5 |
| 8 | Explain about B trees indexing concepts with an example ***(Nov/Dec 14)*** | C213.4 | BTL5 |
| 9 | Illustrate indexing and hashing techniques with suitable examples. ***(Nov/Dec 15)*** | C213.4 | BTL1 |
| 10 | Explain about Query optimization with neat Diagram. ***(Nov/Dec 14,16)*** | C213.4 | BTL1 |
| 11 | Give a detailed description about Query processing and Optimization.Explain the cost estimation of Query Optimization ***(Nov/Dec 14).*** **, Nov/ Dec 2016** | C213.4 | BTL1 |
| 12 | Discuss about join order optimization and heuristic optimization algorithm. ***(Apr/May 15)*** | C213.4 | BTL1 |
| 13 | Briefly explain about Query Processing review***(May/June 16)*** | C213.4 | BTL2 |

| | | | |
|---|---|---|---|
| 14 | **Write Short Notes On Index Structure Of Files? <u>Nov/Dec 2014</u>** | C213.4 | BTL1 |
| 15 | **Give detail Explanation about Hashing & Types of Hashing. Nov/Dec 2015** | C213.4 | BTL1 |

## <u>UNIT V</u>

## ADVANCED TOPICS

Distributed Databases: Architecture, Data Storage, Transaction Processing – Object-based Databases: Object Database Concepts, Object-Relational features, ODMG Object Model, ODL, OQL – XML Databases: XML Hierarchical Model, DTD, XML Schema, XQuery – Information Retrieval: IR Concepts, Retrieval Models, Queries in IR systems.

| Q. No. | Questions | CO | Bloom's Level |
|---|---|---|---|
| 1 | **What is homogeneous distributed database and heterogeneous distributed database**<br>A homogeneous distributed database has identical software and hardware running all databases instances, and may appear through a single interface as if it were a single database. A heterogeneous distributed database may have different hardware, operating systems, database management systems, and even data models for different databases | C213.5 | BTL1 |
| 2 | **Define Distributed Database Systems. *(Nov/Dec 16)***<br>Database spread over multiple machines (also referred to as sites or nodes).Network interconnects the machines. Database shared by users on multiple machines is called Distributed Database Systems | C213.5 | BTL1 |
| 3 | **What are the types of Distributed Database** □<br>Homogeneous Distributed DB<br>Hetrogneous Distributed DB | C213.5 | BTL2 |
| 4 | **Define fragmentation in Distributed Database**<br>The system partitions the relation into several fragment and stores each fragment at different sites<br>Two approaches : Horizontal Fragmentation, Vertical Fragmentation | C213.5 | BTL4 |
| 5 | | C213.5 | BTL2 |

| | | | |
|---|---|---|---|
| | **Define Database replication.**<br>Database replication can be used on many database management systems, usually with a master/slave relationship between the original and the copies. The master logs the updates, which then ripple through to the slaves. The slave outputs a message stating that it has received the update successfully, thus allowing the sending of subsequent updates. | | |
| 6 | **What is the advantage of OODB?**<br>An integrated repository of information that is shared by multiple users, multiple products, multiple applications on multiple platforms. | C213.5 | BTL2 |
| 7 | **What is Object database System?**<br>An object database is a database management system in which information is represented in the form of objects as used in object-oriented programming. Object-relational databases are a hybrid of both approaches. | C213.5 | BTL1 |
| 8 | **What are the advantages of OODB?**<br>An integrated repository of information that is shared by multiple users, multiple products, multiple applications on multiple platforms. It also solves the following problems: 1. The semantic gap: The real world and the Conceptual model is very similar. 2. Impedance mismatch: Programming languages and database systems must be interfaced to solve application problems. But the language style, data structures, of a programming language (such as C) and the DBMS (such as Oracle) are different. The OODB supports general purpose programming in the OODB framework. 3. New application requirements: Especially in OA, CAD, CAM, CASE, object-orientation is the most natural and most convenient. | C213.5 | BTL1 |
| 9 | **How do you define types in object relational feature in oracle?**<br>Oracle allows us to define types similar to the types of SQL. The syntax is<br>CREATE TYPE t AS OBJECT ( list of attributes and methods ); | C213.5 | BTL1 |
| 10 | **Define ODMG Object model?**<br>The ODMG object model is the data model upon which the object definition langauge (ODL) and object query language (OQL) are based. | C213.5 | BTL1 |
| 11 | **Define ODL.**<br>ODL langauge is used to create object specifications:<br>□ classes and interfaces<br>- Using the specific langauge bindings to specify how ODL<br>□ constructs can be mapped to constructs in specific programming<br>□ language, such as C++, SMALLTALK, and JAVA | C213.5 | BTL1 |

| | | | |
|---|---|---|---|
| 12 | **Define Information Retrieval.**<br>It is an activity of obtaining information resources relevant to an information need from a collection of information resources | C213.5 | BTL1 |
| 13 | **Define Relevance Ranking.** ***(Nov/Dec 14)***<br>A system in which the search engine tries to determine the theme of a site that a link is coming from. | C213.5 | BTL1 |
| 14 | **Can we have more than one constructor in a class? If yes, explain the need for such a situation.** ***(Nov/Dec 15)***<br>Yes, default constructor and constructor with parameter | C213.5 | BTL1 |
| 15 | **Define XML Database.**<br>An XML database is a data persistence software system that allows data to be stored in XML format. These data can then be queried, exported and serialized into the desired format. XML databases are usually associated with document-oriented databases. | C213.5 | BTL4 |
| 16 | **Define OQL with syntax.**<br>□ Entry point to the database: needed for each query which can<br>□ be any named *persistent object*:<br>**class** Person<br>( **extent** persons<br>**key** ssn)<br>{ … … }<br>**class** Faculty extends Person ENTRY POINTS<br>( **extent** faculy )<br>{ … … }<br>**class** Department<br>( **extent** departmet **key** dname){ … … } | C213.5 | BTL1 |
| 17 | **Define Crawling and indexing the web.** ***(Nov/Dec 14)***<br>Web Crawling is the process of search engines combing through web pages in order to properly index them. These "web crawlers" systematically crawl pages and look at the keywords contained on the page, the kind of content, all the links on the page, and then returns that information to the search engine's server for indexing. Then they follow all the hyperlinks on the website to get to other websites. When a search engine user enters a query, the search engine will go to its index and return the most relevant search results based on the keywords in the search term. Web crawling is an automated process and provides quick, up to date data. | C213.5 | BTL1 |

| | | | |
|---|---|---|---|
| 18 | **How does the concept of an object in the object-oriented model differ from the concept of an entity in the entity-relationship model?*(Nov/Dec 16)***<br>An entity is simply a collection of variables or data items. An object is an encapsulation of data as well as the methods (code) to operate on the data. The data members of an object are directly visible only to its methods. The outside world can gain access to the object's data only by passing pre-defined messages to it and these messages are implemented by the methods. | C213.5 | BTL1 |
| 19 | **Is XML Hierarchical?**<br>XML documents have a hierarchical structure and can conceptually be interpreted as a tree structure, called an XML tree. XML documents must contain a root element (one that is the parent of all other elements). All elements in an XML document can contain sub elements, text and attributes. | C213.5 | BTL1 |
| 20 | **What is DTD?**<br>A document type definition (DTD) contains a set of rules that can be used to validate an XML file. After you have created a DTD, you can edit it manually, adding declarations that define elements, attributes, entities, and notations, and how they can be used for any XML files that reference the DTD file. | C213.5 | BTL1 |
| 21 | **What is the use of XML Schema?**<br>XML Schema is commonly known as XML Schema Definition (XSD). It is used to describe and validate the structure and the content of XML data. XML schema defines the elements, attributes and data types. Schema element supports Namespaces. | C213.5 | BTL4 |
| 22 | **What is Xpath and Xquery?**<br>XPath can be used to navigate through elements and attributes in an XML document. XPath is a syntax for defining parts of an XML document. XPath uses path expressions to navigate in XML documents. XPath contains a library of standard functions. XPath is a major element in XSLT and in XQuery. | C213.5 | BTL2 |
| 23 | **Define Keyword Queries.**<br>Keyword-based queries are the simplest and most commonly used forms of IR queries: the user just enters keyword combinations to retrieve documents. | C213.5 | BTL1 |
| 24 | **What are the Types of Queries in IR Systems**<br>□ Keyword Queries. Boolean Queries<br>□ Phrase Queries<br>□ Proximity Queries | C213.5 | BTL1 |

| | | | |
|---|---|---|---|
| | □ Wildcard Queries<br>□ Natural Language Queries | | |
| 25 | **State the steps to create DTD.**<br>Create a new DTD, complete the following steps:<br>1. Create a project to contain the DTD if needed.<br>2. In the workbench, click File > New > Other and select XML > DTD. Click Next.<br>3. Select the project or folder that will contain the DTD.<br>4. In the File name field, type the name of the DTD, for example MyDTD.dtd. The name of your DTD file must end with the extension .dtd<br>5. Click Next.<br>6. Optional: You can use a DTD template as the basis for your new DTD file. To do so, click the Use DTD Template check box, and select the template you want to use.<br>7. Click Finish. | C213.5 | BTL1 |
| 26 | **What is the difference between Information Retrieval and DBMS. ?**<br>S.No Information Retrieval DBMS 1 Imprecise semantics Precise semantics 2 Keyword search SQL 3 Unstructured data format Structured data 4 Reads mostly. Adds document occasionally. Expects reasonable number of updates. 5 Displays page through top k results. Generates full answer. | C213.5 | BTL1 |
| 27 | **What are Goals of Distributed Database system ?**<br>**Reliability:** In distributed database system, if one system fails down or stops working for some time another system can complete the task.<br>**Availability:** In distributed database system reliability can be achieved even if sever fails down. Another system is available to serve the client request. **Performance:** Performance can be achieved by distributing database over different locations. So the databases are available to every location which is easy to maintain. | C213.5 | BTL2 |
| 28 | ***Distributed DBMS Architectures***<br>DDBMS architectures are generally developed depending on three parameters −<br>**Distribution** − It states the physical distribution of data across the different sites.<br>**Autonomy** − It indicates the distribution of control of the database system and the degree to which each constituent DBMS can operate independently.<br>**Heterogeneity** − It refers to the uniformity or dissimilarity of the data models, system components and databases. | C213.5 | BTL1 |
| 29 | ***What are Architectural Models of DDBMS.***<br>Some of the common architectural models are –<br>□ Client - Server Architecture for DDBMS<br>□ Peer - to - Peer Architecture for DDBMS | C213.5 | BTL2 |

| | | | |
|---|---|---|---|
| | □ Multi - DBMS Architecture | | |
| 30 | **What is Client - Server Architecture for DDBMS**<br>This is a two-level architecture where the functionality is divided into servers and clients. The server functions primarily encompass data management, query processing, optimization and transaction management. Client functions include mainly user interface. However, they have some functions like consistency checking and transaction management.<br>The two different client - server architecture are –<br>□ Single Server Multiple Client<br>□ Multiple Server Multiple Client ( | C213.5 | BTL2 |
| 31 | **Write short notes on Peer- to-Peer Architecture for DDBMS**<br>In these systems, each peer acts both as a client and a server for imparting database services. The peers share their resource with other peers and co-ordinate their activities.<br>This architecture generally has four levels of schemas –<br>**Global Conceptual Schema** – Depicts the global logical view of data.<br>**Local Conceptual Schema** – Depicts logical data organization at each site.<br>**Local Internal Schema** – Depicts physical data organization at each site.<br>**External Schema** – Depicts user view of data | C213.5 | BTL2 |
| 32 | **Write short notes on Multi - DBMS Architectures**<br>This is an integrated database system formed by a collection of two or more autonomous database systems.<br>Multi-DBMS can be expressed through six levels of schemas –<br>**Multi-database View Level** – Depicts multiple user views comprising of subsets of the integrated distributed database.<br>**Multi-database Conceptual Level** – Depicts integrated multi-database that comprises of global logical multi-database structure definitions.<br>**Multi-database Internal Level** – Depicts the data distribution across different sites and multi-database to local data mapping.<br>**Local database View Level** – Depicts public view of local data.<br>**Local database Conceptual Level** – Depicts local data organization at each site.<br>**Local database Internal Level** – Depicts physical data organization at each site.<br>There are two design alternatives for multi-DBMS –<br>□ Model with multi-database conceptual level.<br>□ Model without multi-database conceptual level. | C213.5 | BTL1 |
| 33 | **Define Replication and Fragmentation.**<br>**Replication.** The system maintains several identical replicas of the relation, and stores each replica at a different site. The alternative to replication is to store only one copy of relation r.<br>**Fragmentation**. The system partitions the relation into several fragments, and stores each fragment at a different site. | C213.5 | BTL1 |
| 34 | **What are the advantages and disadvantages to replication.**<br>**Availability** If one of the sites containing relation r fails, then the relation r can be found in another site. Thus, the system can continue to process queries involving r, despite the failure of one site.<br>**Increased parallelism.** In the case where the majority of accesses to the relation r | C213.5 | BTL1 |

| | | | |
|---|---|---|---|
| | result in only the reading of the relation, then several sites can process queries involving r in parallel. The more replicas of r there are, the greater the chance that the needed data will be found in the site where the transaction is executing. Hence, data replication minimizes movement of data between sites. **Increased overhead on update.** The system must ensure that all replicas of a relation r are consistent; otherwise, erroneous computations may result. Thus, whenever is updated, the update must be propagated to all sites containing replicas. The result is increased overhead. For example, in a banking system, where account information is replicated in various sites, it is necessary to ensure that the balance in a particular account agrees in all sites. | | |
| 35 | **Define Transparency**<br>The user of a distributed database system should not be required to know where the data are physically located nor how the data can be accessed at the specific local site. This characteristic, called **data transparency**, can take several forms: | C213.5 | BTL1 |
| 36 | **What are the DISTRIBUTED TRANSACTIONS**<br>There are two types of transaction that we need to consider.<br>□ **Local transactions** are those that access and update data in only one local database;<br>□ **Global transactions** are those that access and update data in several local databases | C213.5 | BTL1 |
| 37 | **Write the System Failure Modes**<br>□ Failure of a site.<br>□ Loss of messages.<br>□ Failure of a communication link.<br>□ Network partition | C213.5 | BTL1 |
| 38 | **Define Complex Data Types .,**<br>Traditional database applications have conceptually simple datatypes. The basic data items are records that are fairly small and whose fields are atomic. | C213.5 | BTL1 |
| 39 | **What is Structured Type?**<br>Structured types allow composite attributes of E-R designs to be represented directly. For instance, we can define the following structured type to represent a composite attribute name with component attribute *firstname* and *lastname:* | C213.5 | BTL1 |
| 40 | **Write the Object-Identity and Reference Types in SQL**<br>Object-oriented languages provide the ability to refer to objects. An attribute of a type can be a reference to an object of a specified type. For example, in SQL we can define a type *Department* with a field *name* and a field *head* that is a reference to the type *Person*, and a table *departments* of type *Department*, as follows:<br>***create type*** *Department (*<br>*name* ***varchar(20),***<br>*head* ***ref****(Person)* ***scope*** *people);*<br>***create table*** *departments* ***of*** *Department;*<br>Here, the reference is restricted to tuples of the table people. | C213.5 | BTL1 |
| 41 | **What are the Object-relational Features ?**<br>Object-relational database systems are basically extensions of existing relational database systems. Changes are clearly required at many levels of the database system. However, to minimize changes to the storage-system code (relation storage, indices, etc.), the complex datatypes supported by object-relational | C213.5 | BTL1 |

| | | | |
|---|---|---|---|
| | systems can be translated to the simpler type system of relational databases.<br>Sub tables can be stored in an efficient manner | | |
| 42 | **Define Objects and Literals .**<br>Objects and literals are the basic building blocks of the object model. The main difference between the two is that an object has both an **object identifier** and a **state** (or current value), whereas a literal has a value (state) but **no object identifier**. In either case, the value can have a complex **structure. The object** state can change over time by modifying the object value. A literal is basically a constant value, possibly having a complex structure, but it does not change. | C213.5 | BTL1 |
| 43 | **What re the aspects of an object?**<br>**An object has five aspects**: identifier, name, lifetime, structure, and creation. | C213.5 | BTL1 |
| 44 | **What rae the types of literals?**<br>**There are three types of literals:** atomic, structured, and collection. | C213.5 | BTL1 |
| 45 | What are the notation of ODMG??<br>The notation of ODMG uses three concepts: ***interface, literal, and clas*** | C213.5 | BTL1 |
| 46 | **Define ODL: OBJECT DEFINITION LANGUAGE**<br>Object Definition Language (ODL) is the specification language defining the interface to object types conforming to the ODMG Object Model. Often abbreviated by the acronym ODL.This language's purpose is to define the structure of an Entity-relationship diagram. | C213.5 | BTL1 |
| 47 | **How to declare the class and element.?**<br>**Class Declarations**<br>□ interface < name > {elements = attributes, relationships, methods }<br><br>**Element Declarations**<br>□ attribute < type > < name > ;<br>□ **relationship < rangetype > < name > ;** | C213.5 | BTL1 |
| 48 | **Write the Similarities between E/R and ODL**<br>□ both support all multiplicities of relationships<br>□ both support inheritance | C213.5 | BTL1 |
| 49 | **Define XML SCHEMA**<br>XML Schema defines a number of built-in types such as string, integer, decimal date, and boolean. In addition, it allows user-defined types; these may be simple types with added restrictions, or complex types constructed using constructors such as complex Type and sequence. | C213.5 | BTL1 |
| 50 | **What are the RETRIEVAL MODELS**<br>There are the three main statistical models—Boolean, vector space, and probabilistic—and the semantic model. | C213.5 | BTL1 |

.

## PART – B

| Q. NO. | QUESTIONS | CO | BLOOM'S LEVEL |
|---|---|---|---|
| 1 | Explain about Object Oriented Databases and XM Databases. | C213.5 | BTL5 |
| 2 | Explain in detail (i) Information Retrieval (iii) Transaction processing ***(Nov/Dec 14)*** | C213.5 | BTL5 |
| 3 | Write short notes on Distributed Transactions. ***(Nov/Dec 14)*** | C213.5 | BTL6 |
| 4 | Explain in detail the Client - Server Architecture for DDBMS | C213.5 | BTL5 |
| 5 | Suppose an Object Oriented database had an object A, which references object B, which in turn references object C. Assume all objects are on disk initially? Suppose a program first dereferences A, then dereferences B by following the reference from A, and then finally dereferences C. Show the objects that are represented in memory after each dereference, along with their state. ***(Nov/Dec 15)*** | C213.5 | BTL3 |
| 6 | Suppose that you have been hired as a consultant to choose a database system for your client's application. For each of the following applications, state what type of database system (relational, persistent programming language–based OODB, object relational; do not specify a commercial product) you would recommend. Justify your recommendation.<br>(i)A computer-aided design system for a manufacturer of airplanes.<br>(ii)A system to track contributions made to candidates for public office.<br>(iii)An information system to support the making of movies.<br>***(Nov/Dec 16)*** | C213.5 | BTL3 |
| 7 | Give the DTD for an XML representation of the following nested-relational schema<br>*Emp = (ename, ChildrenSet* setof*(Children), SkillsSet* setof*(Skills))*<br>*Children = (name, Birthday)*<br>*Birthday = (day, month, year)*<br>*Skills = (type, ExamsSet* setof*(Exams)).*<br>*Exams = (year, city)* ***(Nov/Dec 16)*** | C213.5 | BTL3 |
| 8 | Explain XML Schema with an example. | C213.5 | BTL5 |
| 9 | Explain various queries in IR Systems with an example. | C213.5 | BTL5 |

| | | | |
|---|---|---|---|
| 10 | Explain ODL and OQL with an example. | C213.5 | BTL5 |
| 11 | Explain ODMG – Object Model in detail | C213.5 | BTL5 |
| 12 | Explain XML Databases. | C213.5 | BTL5 |
| 13 | Exaplain the object relational database features. | C213.5 | BTL5 |
| 14 | Explain Information Retrieval process in detail | C213.5 | BTL5 |
| 15 | Give the DTD or XML Schema for an XML representation of the nested relational schema . Nov/ Dec 2016 | C213.5 | BTL5 |