Internal Assessment Test I – September 2025

Sub:	Introduction to DBMS					Sub Code:	BCS755A	Branch:	ECE		
Date:	30-09-2025	Duration:	90 min's	Max Marks:	50	Sem/Sec:	VII – OPEN ELECTIVE			OBE	

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1	Define database. Elaborate component modules of DBMS and their interaction with diagram.	2+8	**CO1**	**L1**
2	List and explain the advantages of using the DBMS approach. Draw and Explain different notations used in ER-Diagram	6 4	**CO2**	**L1**
3	Describe the three-schema architecture. Why do we need mappings between schemas levels? Explain with necessary diagrams.	10	**CO1**	**L2**
4	Explain an algorithm for ER-to-Relational mapping with example.	5+5	**CO2**	**L2**
5	Explain different DDL,DML,TCL,DCL,and DQL statements used in SQL with examples	10	**CO3**	**L2**
6	Consider the Sailors-Boats-Reserves DB described s (sid, sname, rating, age) b (bid, bname, color) r (sid, bid, date) Write each of the following queries in SQL. 1. Find the colors of boats reserved by Alber. 2. Find all sailor ids of sailors who have a rating of at least 8 or reserved boat 103. 3. Find the names of sailors who have not reserved a boat whose name contains the string "storm". Order the names in ascending order. 4. Find the sailor ids of sailors with age over 20 who have not reserved a boat whose name includes the string "thunder".	10	**CO3**	**L3**

SOLUTION

1. Define database.

A database is an organized collection of data that is stored and managed in such a way that it can be easily accessed, managed, and updated. It allows users to store, retrieve, and manipulate data efficiently.

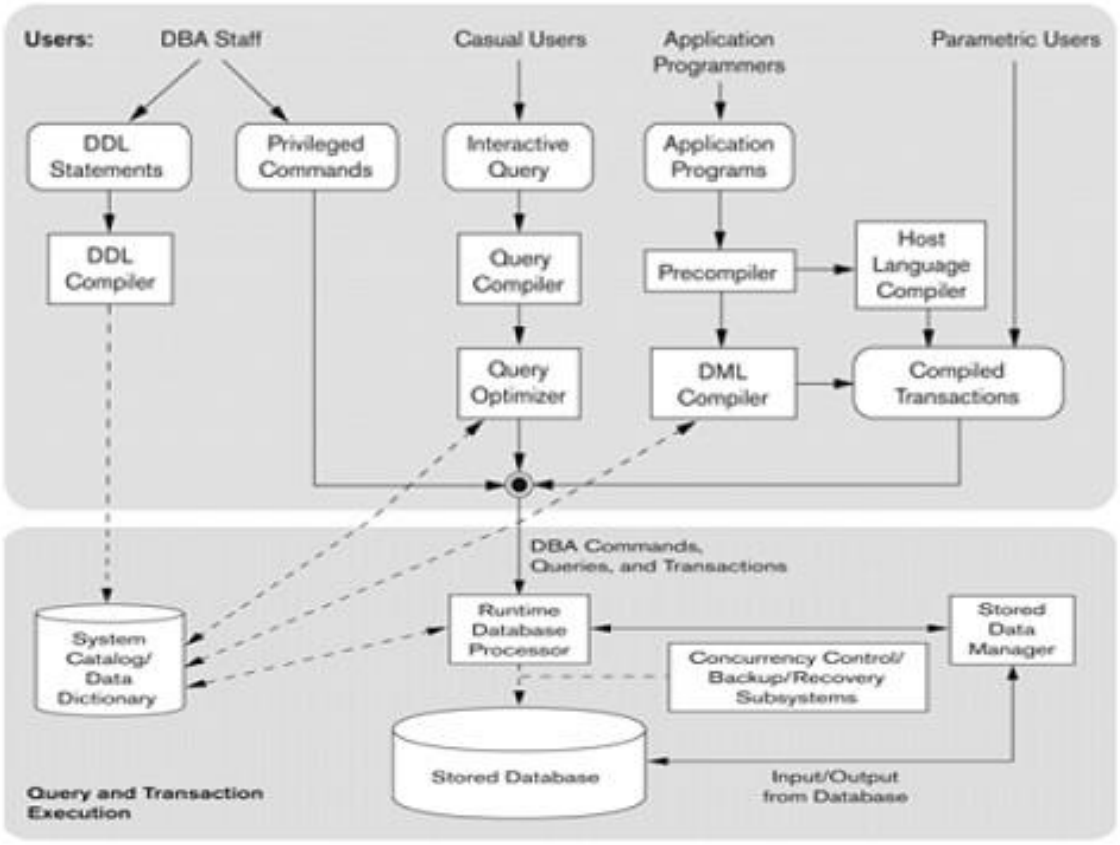
Elaborate component modules of DBMS and their interaction with diagram

A DBMS is a complex software system • discuss the types of software components that constitute a DBMS and the types of computer system software with which the DBMS interacts • two parts: o The top part: various users and their interfaces o The lower part: storage of data and processing of transactions • Access to the disk is controlled primarily by the operating system (OS) • buffer management module to schedule disk read/write, because this has a considerable effect on performance • A higher-level stored data manager module: controls access to DBMS information that is stored on disk, whether it is part of the database or the catalog • DDL Statement: used by the DBA for defining the database and tuning it • DDL Compiler: processes schema definitions and stores descriptions of the schemas in the DBMS catalog.

Privileged commands: The commands which are exclusively used by the DBA • Interactive query: The interface by which the casual users and persons with occasional need for information from the database interact • Query compiler: Queries are parsed and validated for correctness of the query syntax • Query optimizer: Responsible for optimising the query and its execution.

Pre-compiler: Extracts DML commands from an application program written in a host programming language • DML compiler: Compilation of pre-compiled DMLC commands into object code for database access

Host language compiler: The rest of the program is sent to the host language compiler. • Compiled transactions: Canned transactions are executed repeatedly by parametric users, who simply supply the parameters to the transactions.



2. List and explain the advantages of using the DBMS approach.

i. Controlling Redundancy • redundancy in storing the same data multiple times leads to several problems o there is the need to perform a single logical update storage space is wasted when the same data is stored repeatedly o Files that represent the same data may become inconsistent

ii. Restricting Unauthorized Access • A DBMS should provide a security and authorization subsystem • most users will not be authorized to access all information in the database

iii. Providing Persistent Storage for Program Objects • Databases can be used to provide persistent storage for program objects and data structures

iv. Providing Storage Structures and Search Techniques for Efficient Query Processing • the DBMS must provide specialized data structures and search techniques to speed up disk search for the desired records – index • The DBMS often has a buffering or caching module that maintains parts of the database in main memory buffers • The query processing and optimization module of the DBMS is responsible for choosing an efficient query execution plan for each query based on the existing storage structures.

v. Providing Backup and Recovery
responsible for recovery • the database is restored to the state it was in before the transaction started executing

vi. Providing Multiple User Interfaces • a DBMS should provide a variety of user interfaces o query languages for casual users, o programming language interfaces for application programmers, o forms and command codes for parametric users, and o menu-driven interfaces and natural language interfaces for standalone users

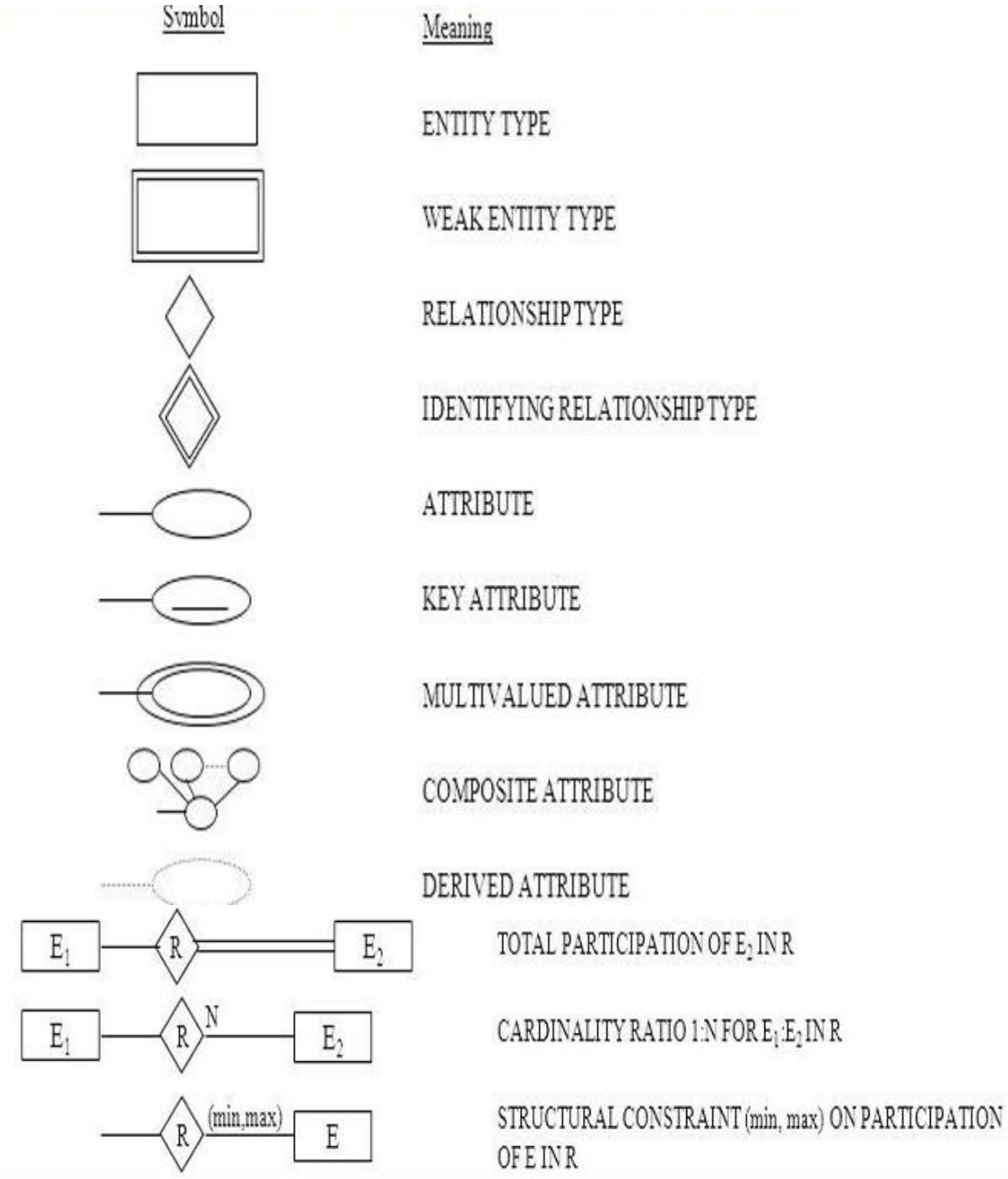
vii. Representing Complex Relationships among Data • A DBMS must have the capability to represent a variety of complex relationships among the data, to define new relationships as they arise, and to retrieve and update related data easily and efficientl

viii. Enforcing Integrity Constraints • The simplest type of integrity constraint involves specifying a data type for each data item • referential integrity constraint: specifying that a record in one file must be related to records in other files • Primary key constraint: uniqueness on data item values.

ix. Permitting inferencing and Actions Using Rules: A trigger is a form of a rule activated by updates to the table, which results in performing some additional operations to some other tables, sending messages, and so on.

x. Additional Implications of Using the Database Approach
Potential for Enforcing Standards: The DBA can enforce standards in a centralized database environment o Reduced Application Development Time: Development time using a DBMS is estimated to be one-sixth to one-fourth of that for a traditional file system o Flexibility: Modern DBMSs allow certain types of evolutionary changes to the structure of the database without affecting the stored data and the existing application programs o Availability of Up-to-Date Information: As soon as one user's update is applied to the database, all other users can immediately see this update o Economies of Scale: reduce the wasteful overlap activities thereby reducing the overall costs of operation and management.

Draw and Explain different notations used in ER-Diagram



3. Describe the three-schema architecture. Why do we need mappings between schemas levels? Explain with necessary diagrams.

In three architecture, schemas can be defined at the following three levels:

1. The internal level has an internal schema, -describes the physical storage structure of the database. -uses a physical data model and describes the complete details of data storage and access paths for the database.

2. The conceptual level has a conceptual schema, -describes the structure of the whole database for a community of users -hides the details of physical storage structures and concentrates on describing entities, data types, relationships, user

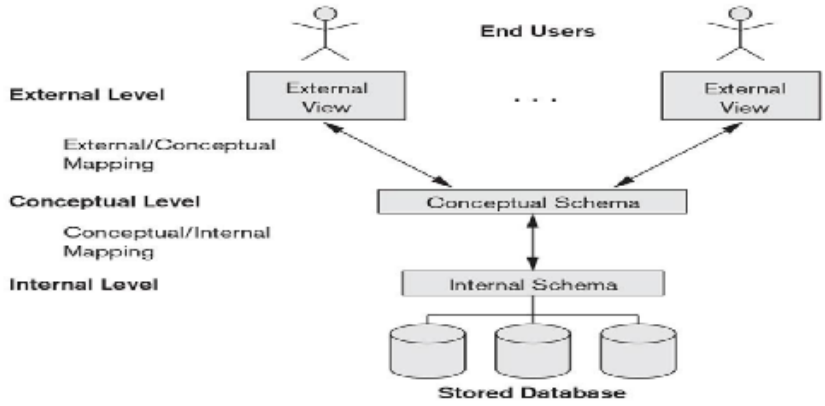
operations, and constraints -representational data model is used to describe the conceptual schema when a database system is implemented -This implementation conceptual schema is often based on a conceptual schema design in a high-level data model.

3. The external or view level includes a number of external schemas or user views, -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. -As in the previous level, each external schema is typically implemented using a representational data model, possibly based on an external schema design in a high-level data model.

The DBMS must transform a request specified on an external schema into a request against the conceptual schema, and then into a request on the internal schema for processing over the stored database. If the request is a database retrieval, the data extracted from the stored database must be reformatted to match the user's external view. The processes of transforming requests and results between levels are called mappings. These mappings may be time-consuming, so some DBMSs—especially those that are meant to support small databases—do not support external views. Even in such systems, however, a certain amount of mapping is necessary to transform requests between the conceptual and internal levels.

The Three-Schema Architecture

The goal of the three-schema architecture is to separate the user applications from the physical database.



4. **Explain an algorithm for ER-to-Relational mapping with example.**

Map Strong Entities

Create a relation for each strong entity.
Include all simple attributes.
Primary key of entity → Primary key of relation.

Map Weak Entities

Create a relation for each weak entity.
Include all attributes.
Include primary key of owner entity as foreign key.
Composite primary key = (owner PK + weak entity partial key).

Map Binary Relationships

1:1 relationship: Add foreign key to either entity (usually total participation side).
1:N relationship: Add foreign key to the "many" side.

M:N relationship: Create a new relation with primary keys of both entities as composite PK.

Map Multi-Valued Attributes

Create a separate relation.
Include owner entity PK as foreign key.
Composite primary key = (owner PK + multi-valued attribute).

Map N-ary Relationships (more than 2 entities)

Create a new relation.
Include primary keys of all participating entities as foreign keys.
Composite primary key = combination of all participating PKs.
Include attributes of the relationship, if any.

5. **Explain different DDL,DML,TCL,DCL,and DQL statements used in SQL with examples**

Common DDL Statements:

Statement	Description	Example
`CREATE`	Create a new table or database	`CREATE TABLE Student(sid INT PRIMARY KEY, sname VARCHAR(50), age INT);`
`ALTER`	Modify structure of an existing table	`ALTER TABLE Student ADD COLUMN dept VARCHAR(20);`
`DROP`	Delete a table or database	`DROP TABLE Student;`
`TRUNCATE`	Remove all records from a table, structure remains	`TRUNCATE TABLE Student;`

Common DML Statements:

Statement	Description	Example
`INSERT`	Insert new data into a table	`INSERT INTO Student(sid, sname, age) VALUES(1, 'Alice', 20);`
`UPDATE`	Modify existing data	`UPDATE Student SET age=21 WHERE sid=1;`
`DELETE`	Remove specific data	`DELETE FROM Student WHERE sid=1;`

3. DQL – Data Query Language

Purpose: Used to **query or retrieve data** from tables.

Statement:

- `SELECT`

Example:

```sql
SELECT sname, age FROM Student WHERE age > 18;
```

4. DCL – Data Control Language

Purpose: Used to **control access/permissions** on database objects.

Common DCL Statements:

Statement	Description	Example
`GRANT`	Give privileges to users	`GRANT SELECT, INSERT ON Student TO User1;`
`REVOKE`	Remove privileges from users	`REVOKE INSERT ON Student FROM User1;`

5. TCL – Transaction Control Language

Purpose: Used to **manage transactions** in a database to ensure data integrity.

Common TCL Statements:

Statement	Description	Example
`COMMIT`	Save all changes made in the transaction	`COMMIT;`
`ROLLBACK`	Undo changes made in the current transaction	`ROLLBACK;`
`SAVEPOINT`	Set a point to rollback to	`SAVEPOINT sp1;`

6. Consider the Sailors-Boats-Reserves DB described

s (sid, sname, rating, age)
b (bid, bname, color)
r (sid, bid, date)
Write each of the following queries in SQL.

1. Find the colors of boats reserved by Alber.
   SELECT DISTINCT b.color
   FROM b
   JOIN r ON b.bid = r.bid
   JOIN s ON s.sid = r.sid

```
WHERE s.sname = 'Alber';
```

2. Find all sailor ids of sailors who have a rating of at least 8 or reserved boat 103.

```
SELECT DISTINCT s.sid
FROM s
LEFT JOIN r ON s.sid = r.sid
WHERE s.rating >= 8 OR r.bid = 103;
```

3. Find the names of sailors who have not reserved a boat whose name contains the string “storm”. Order the names in ascending order.

```
SELECT s.sname
FROM s
WHERE s.sid NOT IN (
   SELECT r.sid
   FROM r
   JOIN b ON r.bid = b.bid
   WHERE b.bname LIKE '%storm%'
)
ORDER BY s.sname ASC;
```

4. Find the sailor ids of sailors with age over 20 who have not reserved a boat whose name includes the string “thunder”.

```
SELECT s.sid
FROM s
WHERE s.age > 20
AND s.sid NOT IN (
   SELECT r.sid
   FROM r
   JOIN b ON r.bid = b.bid
   WHERE b.bname LIKE '%thunder%'
);
```