Internal Assessment Test 1 – December 2023 (Set -3)

Sub:	Database Management System			Sub Code:	21CS53	Branch:	ISE	
Date:	19-12-2023	Duration:	90 min's	Max Marks:	50	Sem / Sec:	V/ A,B,C	OBE

	Answer any FIVE FULL QUESTIONS	MARKS	CO	RBT
1	Define DBMS. Explain with necessary diagram the Database Component Modules.	[10]	CO1	L2
2	Describe the three-schema architecture in DBMS. Also explain the different categories of data model.	[10]	CO1	L2
3	Draw and explain different notations used in ER-Diagram.	[10]	CO1	L2
4	List and explain the advantages of using the DBMS approach.	[10]	CCO1	L2
5	Explain CREATE, DROP, RENAME and different ALTER commands with a suitable example.	[10]	CO2	L2
6	Create the table named 'emp' with the attributes: empno, empname, dept, designation, salary, doj, place. Write SQL statements for the following: i. Display all the fields of employee table. ii. Display details of employee number and their salary. iii. Delete the employee whose empno is 101. iv. Display total salary of employee which is greater than 120000. v. Add a column named phone_no in an employee table using alter command vi. Change the column name from place to state using alter command. vii. Rename the emp table name as "Employee"	[10]	CO2	L3

Scheme:

Question No	Content split	Marks spiltup	Total Marks	CO
1.	.DBMS .Diagram .Explanation	2M 3M 5M	10M	CO1
2.	.3-Schema architecture .Categories of data model	5M 5M	10M	CO1
3.	.Notations .Explanation	5M 5M	10M	CO1
4.	.Advantages	10M	10M	CCO1
5.	.Syntaxes for commands .Examples	5M 5M	10M	CO2
6.	.Table creation .Queries	5M 5M	10M	CO2

1.Define DBMS. Explain with necessary diagram the Database ComponentModules.

A **database management system (DBMS)** is a collection of programs that enables users to create and maintain a database. The DBMS is hence a *general-purpose software system* that facilitates the processes of *defining, constructing,* and *manipulating* databases for various applications. **Defining** a database involves specifying the data types, structures, and constraints for the data to be stored in the database. **Constructing** the database is the process of storing the data itself on some storage medium that is controlled by the DBMS.

Database Component Modules:

The database and the DBMS catalog are usually stored on disk. Access to the disk is controlled primarily by the **operating system (OS),** which schedules disk input/output. A higher-level **stored data manager** module of the DBMS controls access to DBMS information that is stored on disk, whether it is part of the database or the catalog.The stored data manager may use basic OS services for carrying out low- level data transfer between the disk and computer main storage, but it controls other aspects of data transfer, such as handling buffers in main memory.

Many DBMSs have their own **buffer management** module to schedule disk read/write.

Stored data manager controls access to DBMS information that is stored on disk, whether it is part of the database or the catalog.

Top half figure:

- ✓ it shows interfaces for the DBA staff, casual users who work with interactive interfaces to formulate queries
- ✓ application programmers who create programs using some host programming languages
- ✓ parametric users who do data entry work by supplying parameters to predefined transactions.
- ✓ the DBA staff works on defining the database and tuning it by making changes to its definition using the DDL and other privileged commands

DBA staff:

- ✓ The DDL compiler processes schema definitions, specified in the DDL, and stores descriptions of the schemas (meta-data) in the DBMS catalog
- ✓ The catalog includes information such as the names and sizes of files, names and data types of data items, storage details of each file, mapping information among schemas, and constraints

Casual users:

- ✓ interact using some form of interface, which we call the interactive query interface
- ✓ queries are parsed and validated for correctness of the query syntax, the names of files anddata elements, and so on by a query compiler that compiles them into an internal form
- ✓ This internal query is subjected to query optimization
- ✓ **query optimizer** is concerned with the rearrangement and possible reordering of operations, elimination of redundancies, and use of correct algorithms and indexes during execution.
- ✓ It consults the system catalog for statistical and other physical information about the stored data and generates executable code that performs the necessary operations for the query and makes calls on the runtime processor

Application programmers

- ✓ write programs in host languages such as Java, C, or C++ that are submitted to a precompiler
- ✓ precompiler extracts DML commands from an application program
- ✓ commands are sent to the DML compiler for compilation
- ✓ rest of the program is sent to the host language compiler
- ✓ The object codes for the DML commands and the rest of the program are linked, forming a canned transaction
- ✓ An example is a bank withdrawal transaction where the account number and the amount may be supplied as parameters

In the lower part of Figure,

- ✓ the runtime database processor executes

(1) the privileged commands
(2) the executable query plans, and

(3) the canned transactions with runtime parameters.

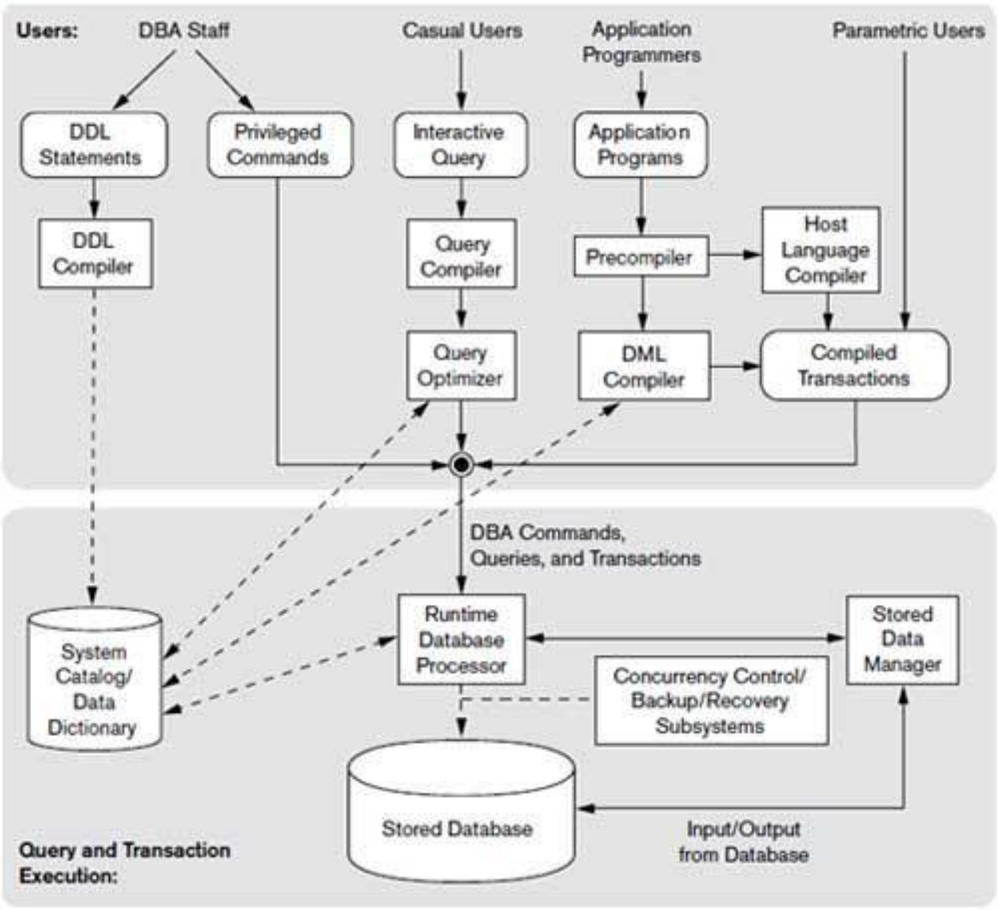
✓ It works with the system catalog and may update it with statistics

✓ It also works with the stored data manager, which in turn uses basic operating system services for carrying out low-level input/output (read/write) operations between the disk and main memory

✓ The runtime database processor handles other aspects of data transfer, such as management of buffers

concurrency control and backup and recovery systems, integrated into the working of the runtime database processor for purposes of transaction management.

Database System Environment:



2.Describe the three-schema architecture in DBMS. Also explain the differentcategories of data model.

The Three-Schema Architecture :

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

In this 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.

Categories of Data Models

1. High-level or conceptual data models

-provide concepts that are close to the way many users perceive data
-use concepts such as **entities, attributes, and relationships.**

✓ An **entity** represents a real-world object or concept, such as an employee or a project from the miniworld that is described in the database.

✓ An **attribute** represents some property of interest that further describes an entity, such as the employee’s name or salary.

✓ A **relationship** among two or more entities represents an association among the entities., for example, a works-on relationship between an employee and a project.

2.Low-level or physical data models

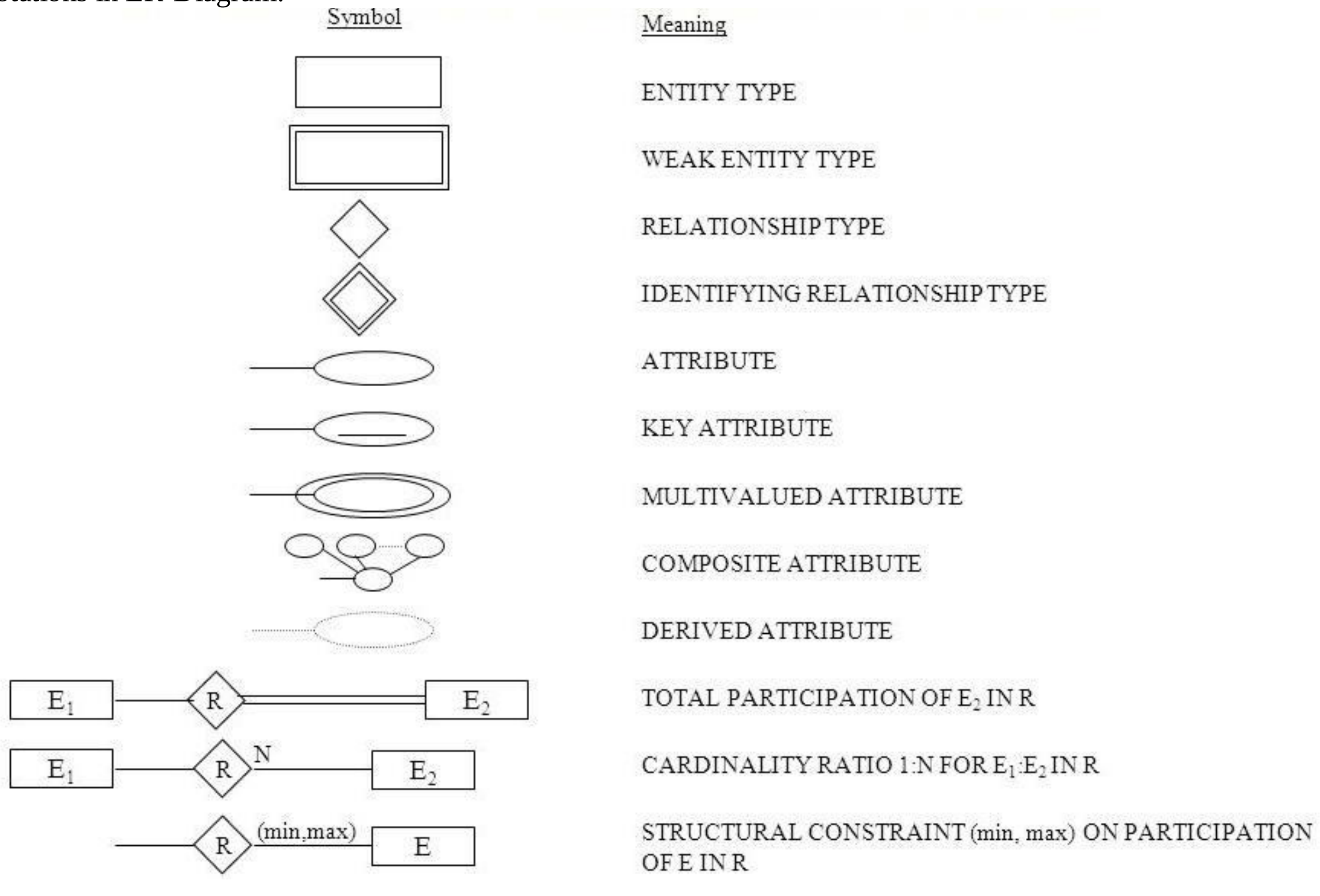
- provide concepts that describe the details of how data is stored on the computer storage media, typically magnetic disks.
- Concepts provided by low-level data models are generally meant for computer specialists, not for end users

3. Representational (or implementation) data models

- provide concepts that may be easily understood by end users but that are not too far removed from the way data is organized in computer storage
-in between high level and low level
- hide many details of data storage on disk but can be implemented on a computer system directly.

3.Draw and explain different notations used in ER-Diagram.

Notations in ER-Diagram:



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

Advantages of Using the DBMS Approach

1. Controlling Redundancy:

Data redundancy (such as tends to occur in the "file processing" approach) leads to wasted storage space, duplication of effort (when multiple copies of a datum need to be updated), and a higher likelihood of the introduction of inconsistency. On the other hand, redundancy can be used to improve performance of queries. Indexes, for example, are entirely redundant, but help the DBMS in processing queries more quickly.
A DBMS should provide the capability to automatically enforce the rule that no inconsistencies are introduced when data is updated.

2. Restricting Unauthorized Access:

A DBMS should provide a security and authorization subsystem, which the DBA uses to create accounts and to specify account restrictions. Then, the DBMS should enforce these restrictions automatically.

3. Providing Persistent Storage for Program Objects:

Object-oriented database systems make it easier for complex runtime objects (e.g., lists, trees) to be saved in secondary storage so as to survive beyond program termination and to be retrievable at a later time.

4. Providing Storage Structures and Search Techniques for Efficient Query Processing:

Database systems must provide capabilities for efficiently executing queries and updates. 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.

5. Providing Backup and Recovery:

A DBMS must provide facilities for recovering from hardware or software failures. The backup and recovery subsystem of the DBMS is responsible for recovery.
The recovery subsystem could ensure that the transaction is resumed from the point at which it was interrupted so that its full effect is recorded in the database.

6. Providing Multiple User Interfaces:

Many types of users with varying levels of technical knowledge use a database, a DBMS should provide a variety of user interfaces.For example, query languages for casual users, programming language interfaces for application programmers, forms and/or command codes for parametric users, menu-driven interfaces for stand-alone users.

7. 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 efficiently.

8. Enforcing Integrity Constraints:

Most database applications have certain integrity constraints that must hold for the data.
The simplest type of integrity constraint involves specifying a data type for each data item.
A more complex type of constraint that frequently occurs involves specifying that a record in one file must be related to records in other files.This is known as a referential integrity constraint.
Another type of constraint specifies uniqueness on data item values, this is known as a key or uniqueness constraint.

9. Permitting Inferencing and Actions Via Rules:

In a **deductive** database system, one may specify *declarative* rules that allow the database to infer new data! E.g., Figure out which students are on academic probation. Such capabilities would take the place of application programs that would be used to ascertain such information otherwise.
Active database systems go one step further by allowing "active rules" that can be used to initiate actions automatically.

5.Explain CREATE, DROP, RENAME and different ALTER commands with asuitable example.

1.Create:

It is used for creating the new table structure into a db.It is data definition language command.

Syntax:

```
  Create table < tablename>(columnname1 datatype(size),columnname2 datatype(size));
```

Example:

```
   Create table student(sid integer(20),sname varchar(20),address varchar(20));
```

2.Drop:

It is used to delete the entire tables in the database.

Syntax:

```
   drop table <tablename>;
```

Example:

```
   Drop table student;
```

3.Rename:

It is used to remove the table.

Syntax:

```
  Rename <oldname> to <newname>;
```

Example:

```
   Rename emp to employee;
```

4.Alter:

This command is used to alter or modify the structure of table.Alter having 4 types of commands.

i.alter with add:

This command is used to add extra column in to the table.

Syntax:

```
  Alter table <tablename> add columnname datatype(size);
```

Example:

```
  Alter table emp add primarykey(empno);
```

ii.alter with modify:

this command is used to modify the column of a table.

Syntax:

```
 Alter table <tablename> modify columnname datatype(size);
```

Example:

```
 Alter table emp modify salary float(10,2);
```

iii.alter with drop:

It is used to drop the unneccessary column of a table.

Syntax:

```
  Alter table <tablename> drop column <columnname>;
```

Syntax:

```
 Alter table emp drop column doj;
```

iv.alter with rename:

It is used to rename a column of a table.

Syntax:

```
 Alter table <tablename> rename column <oldname> to <newname>;
```

Example:

```
 Alter table emp rename column sal to salary;
```

6. Create the table named ‘emp’ with the attributes: empno, empname, dept,designation, salary, doj, place. Write SQL statements for the following:

i.Display all the fields of employee table.

iiDisplay details of employee number and their salary.

iii.Delete the employee whose empno is 101.

iv.Display total salary of employee which is greater than 120000.

v.Add a column named phone_no in an employee table using altercommand

vi.Change the column name from place to state using alter command.

vii.Rename the emp table name as “Employee”

```
Create table emp(empno int(20),empname varchar(20),dept varchar(20),designation varchar(20),salary int(20),doj varchar(20),place varchar(20));
i.select * from emp;
ii.select empno,salary from emp;
iii.delete emp where empno=101;
iv.select  salary from emp where salary>12000;
v.alter table emp add(phoneno);
vi.alter table emp rename column place to state;
vii.rename emp to Employee;
```