

2 MARKS

1. What is database

Ans: A **database** is a collection of related data. By **data**, we mean known facts that can be recorded and that have implicit meaning. For example, names, telephone numbers.

2. What is DBMS

Ans: A database management system (DBMS) is a collection of programs enabling users to create and maintain a database. More specifically, The DBMS is a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications.

3. List the advantages of database approach

Ans:

- a. Controlling Redundancy
- b. Restricting Unauthorized Access
- c. Providing Persistent Storage for Program Objects
- d. Providing Storage Structures and Search Techniques for Efficient Query Processing
- e. Providing Backup and Recovery
- f. Providing Multiple User Interfaces
- g. Representing Complex Relationships among Data
- h. Enforcing Integrity Constraints
- i. Permitting Inferencing and Actions Using Rules

4. Write the example query using between and not between

Ans) The **BETWEEN** operator is used to filter records within a range.

>SELECT *FROM Employee WHERE Salary BETWEEN 30000 AND 50000;

>SELECT * FROM Employee WHERE Salary NOT BETWEEN 30000 AND 50000;

5. Define IN and NOT IN operator in SQL with an example for each.

Ans) The **IN** operator checks if a value matches any value in a list ; **NOT IN** excludes them.

>SELECT *FROM Employee WHERE Dept_ID IN(10,20,30);

>SELECT * FROM Employee WHERE Dept_ID NOT IN(10,20,30);

6. Define nested queries with an example.

Ans) A **nested query (subquery)** is a query inside another SQL query.

>SELECT Name FROM Employee WHERE

Salary >(SELECT AVG(Salary) FROM Employee);

7. Define the datatype char and varchar with example for each.

Ans) **CHAR(n)**: Fixed-length character datatype.

Example: CHAR(10) always stores 10 characters (padded with spaces).

VARCHAR(n): Variable-length character datatype.

Example: VARCHAR(10) stores only the actual number of characters entered.

Example:

> CREATE TABLE Student (Name CHAR(10), City VARCHAR(20));

8. Define equijoin condition with an example.

Ans) **Equijoin** combines rows from two or more tables based on equality between related columns.

Example:

> SELECT E.Fname, D.Dname FROM Employee E, Department D WHERE

E.Dno = D.Dnumber;

9. Define like operator and its metacharacters.

Ans) The **LIKE** operator is used for pattern matching in text values.

Metacharacters:

%(percentile) → Represents zero or more characters.

_(underscore) → Represents exactly one character.

10. Define update command, give its syntax and an example

Ans) The **UPDATE** command modifies existing records in a table.

Syntax:

> UPDATE table_name SET column_name =value WHERE condition;

Example:

> UPDATE Employee SET Salary=Salary+5000 WHERE Dept_ID= 10;

5/10 MARKS

1. Consider the following tables

Employee(fname,minit,lname,ssn,bdate,address,sex,salary,super_ssn,dno)

Department(dname,dnumber,mgr_ssn,mgr_start_date)

Dept_locations(dnumber,dlocation)

Project(pname,pnumber,plocation,dnum)

Works_on(essn,pno,hours)

Dependent(essn,dependent_name,sex,bdate,relationship)

i) For every project located in Stafford, list the project number, the controlling department number, and the department manager's last name, address, and birth date.

```
> SELECT P.Pnumber, P.Dnum, E.Lname, E.Address, E.Bdate
   FROM PROJECT P, DEPARTMENT D, EMPLOYEE E
  WHERE P.Dnum = D.Dnumber
     AND D.Mgr_ssn = E.Ssn
     AND P.Plocation = 'Stafford';
      3M
```

ii) For each employee, retrieve the employee's first and last name and the first and last name of his or her supervisor.

```
> SELECT E.Fname, E.Lname, S.Fname, S.Lname
   FROM EMPLOYEE E, EMPLOYEE S
  WHERE E.Super_ssn = S.Ssn;
      2M
```

iii) Retrieve the names of employees who live in Houston, Texas.

```
> SELECT Fname, Lname
   FROM EMPLOYEE WHERE Address LIKE '%Houston, TX%';
      2M
```

iv) For each department that has more than 2 employees, retrieve the department number and the number of employees.

```
> SELECT Dno, COUNT(*) AS Number_of_Employees
   FROM EMPLOYEE
  GROUP BY Dno
  HAVING COUNT(*) > 2;
```

3M

v) Count the number of employees whose salaries exceed 40,000 in each department, but only for department where more than 5 employees work

```
> SELECT Dname, COUNT(*)
    FROM EMPLOYEE
   WHERE Salary > 40000
   GROUP BY Dno
   HAVING COUNT(*) > 5;
```

5M

- vi) Retrieve project number, project location, department number, name of handling department and average salaries of the employees.**

```
> SELECT P.pnumber, P.plocation, D.dnumber, D.dname, AVG(E.salary)
    FROM PROJECT P, DEPARTMENT D, EMPLOYEE E
   WHERE P.dnum = D.dnumber AND D.dnumber = E.dno
   GROUP BY P.pnumber, P.plocation, D.dnumber, D.dname;
```

5M

- vii) Show the resulting salaries if every employee working on 'ProductX' is given 25% raise and also list their names.**

```
> SELECT E.fname, E.lname, 1.25*E.salary AS increased_salary
    FROM EMPLOYEE E, WORKS_ON W, PROJECT P
   WHERE E.ssn = W.essn AND W.pno = P.pnumber
   AND P.pname = 'ProductX';
```

5M

- viii) Retrieve the list of employees and the projects they are working on ordered by department and, within each department ordered alphabetically by last name, first name.**

```
> SELECT D.dname, E.lname, E.fname, P.pname
    FROM DEPARTMENT D, EMPLOYEE E, WORKS_ON W, PROJECT P
   WHERE D.dnumber=E.dno AND E.ssn=W.essn AND
   W.pno=P.pnumber ORDER BY D.dname, E.lname, E.fname;
```

5M

2. Consider the following tables

PUBLISHER (Publisher_Name, Address, Phone)
BOOK (Book_id, Title, Publisher_Name, Pub_Year)
BOOK_AUTHORS (Book_id, Author_Name)
LIBRARY_BRANCH (Branch_id, Branch_Name, Address)
BOOK_COPIES (Book_id, Branch_id, No-of_Copies)
CARD (CARD_NO)
BOOK_LENDING (Book_id, Branch_id, Card_No, Date_Out, Due_Date)

Write the SQL queries for the following

i) Retrieve details of all books in the library id, title, name of publisher, authors, number of copies in each branch.

```
> SELECT B.BOOK_ID, B.TITLE, B.PUBLISHER_NAME,  
A.AUTHOR_NAME, C.NO_OF_COPIES, L.BRANCH_ID  
FROM BOOK B, BOOK_AUTHORS A,  
BOOK_COPIES C, LIBRARY_BRANCH L  
WHERE B.BOOK_ID=A.BOOK_ID AND  
B.BOOK_ID=C.BOOK_ID  
AND L.BRANCH_ID=C.BRANCH_ID;
```

5M

ii) Create a view of all books and its number of copies that are currently available in the Library.

```
> CREATE VIEW V_BOOKS AS  
SELECT B.BOOK_ID, B.TITLE, C.NO_OF_COPIES FROM BOOK B,  
BOOK_COPIES C, LIBRARY_BRANCH L WHERE  
B.BOOK_ID=C.BOOK_ID AND C.BRANCH_ID=L.BRANCH_ID;
```

5M

Q. Discuss the characteristics of database approach 10M

Ans)

1. Self-Describing Nature of Database System

A database system stores both data and information about the data, called metadata. Metadata describes tables, attributes, data types, relationships, and constraints. This information is stored in the system catalog or data dictionary.

The DBMS uses this information to understand and manage the database. It helps users and programs know the structure of the stored data.

2. Insulation Between Programs and Data

The database approach provides program-data independence, which means application programs are separated from the details of data storage and operations. Changes in data do not require major changes in application programs.

It has two types:

- **Program-data independence:** Changes in the data structure do not require changes in application programs.
- **Program-operation independence:** The details of how data operations are performed are hidden from application programs.

This makes programs easier to develop, maintain, and modify.

3. Support of Multiple Views of Data

Different users may require different portions of the database. The DBMS allows multiple views of the same database according to user requirements. This improves security, simplicity, and usability.

Each user can access only the information relevant to their work. Views also hide unnecessary details from users.

4. Data Sharing and Multiuser Transaction Processing

A database allows multiple users to access and share data at the same time. The DBMS controls simultaneous operations using concurrency control and transaction management. This maintains the consistency and correctness of the database.

It prevents conflicts when many users access the same data simultaneously. Thus, the database can efficiently support a large number of users.