**General Instructions:**

* **Follow the instructions given in each section.**
* **Make sure that you attempt the questions in order.**

**SECTION-A (10\*1 mark=10 marks)**

***(All questions are compulsory)***

1. In normalization, the process of decomposing a table into multiple tables to eliminate redundancy is known as:
   1. Normalization
   2. Denormalization
   3. **Decomposition**
   4. Redundancy elimination
2. When granting permissions in a DBMS, the GRANT statement is used to:
   1. **Assign privileges to a user or role**
   2. Revoke privileges from a user or role
   3. Create a new user or role
   4. Delete a user or role
3. Which of the following is an example of an entity in an ER model?
   1. Relationship
   2. Attribute
   3. Table
   4. **Customer**
4. A foreign key in a relational model:
   1. **Links two tables together**
   2. Enforces entity integrity
   3. Determines the order of columns in a table
   4. Defines a primary key in a table
5. Which architecture allows multiple users to access the database simultaneously without interference?
   1. **Client-Server architecture**
   2. Two-tier architecture
   3. Three-tier architecture
   4. Centralized architecture
6. Which SQL keyword is used to sort data in ascending order in a SELECT statement?
   1. SORT
   2. **ORDER BY**
   3. GROUP BY
   4. ASC
7. Which control structure allows the execution to jump to a specific case based on the value of an expression?
   1. IF-ELSE
   2. FOR
   3. WHILE
   4. **SWITCH**
8. Which security mechanism ensures that only authorized users can access the database?
   1. **Role-based access control**
   2. Encryption
   3. Intrusion detection systems
   4. Secure Socket Layer (SSL)
9. Which of the following statements about cursors is true?
   1. Cursors can only be used for read operations.
   2. Cursors are used to execute stored procedures.
   3. **Cursors allow for sequential access to query results.**
   4. Cursors can only be used with SQL Server databases.
10. Rectangles in ER diagram represents?
    1. Table
    2. Attribute
    3. Tuple
    4. **Entity Set**

**SECTION-B (5\*2 mark=10 marks)**

***(All questions are compulsory)***

1. \_\_\_\_\_\_\_\_\_\_\_\_\_ is a classical approach to database design?
   1. Left – Right approach
   2. Right – Left approach
   3. **Top – Down approach**
   4. Bottom – Up approach
2. In which of the following cases a DML statement is not executed?
   * 1. When existing rows are modified.
     2. **When a table is deleted.**
     3. When some rows are deleted.
     4. All of the above
3. Shared locks are applied while performing
4. **Read operations**
5. Write operations
6. A & B both
7. None of the above
8. Find the cities name with the condition and temperature from table 'whether' where condition = sunny or cloudy but temperature >= 60.
9. SELECT city, temperature, condition FROM weather WHERE condition = 'cloudy' AND condition = 'sunny' OR temperature >= 60
10. SELECT city, temperature, condition FROM weather WHERE condition = 'cloudy' OR condition = 'sunny' OR temperature >= 60
11. **SELECT city, temperature, condition FROM weather WHERE condition = 'sunny' OR condition = 'cloudy' AND temperature >= 60**
12. SELECT city, temperature, condition FROM weather WHERE condition = 'sunny' AND condition = 'cloudy' AND temperature >= 60
13. What SQL command can be used to add columns to a table?
14. **ALTER TABLE TableName ADD ColumnName**
15. ALTER TABLE TableName ADD COLUMN ColumnName
16. MODIFY TABLE TableName ADD ColumnName
17. MODIFY TABLE TableName ADD COLUMN ColumnName

**SECTION-C(Coding Question) (4x5 marks=20 marks)**

1. Create table 'book' with attributes: book\_id, title, author, publication\_year, genre.

Add multiple records in it. Select books with titles containing a specific keyword (e.g., 'novel')

Select the earliest publication year among all books. Remove all records from book, but keeps the table structure.

Solution:

**CREATE TABLE book (**

**book\_id INT PRIMARY KEY,**

**title VARCHAR(255),**

**author VARCHAR(100),**

**publication\_year INT,**

**genre VARCHAR(50)**

**);**

**INSERT INTO book (book\_id, title, author, publication\_year, genre)**

**VALUES (1, 'Example Book', 'John Doe', 2020, 'Fiction'),(2, 'Another Book', 'Jane Smith', 2018, 'Mystery'),**

**(3, 'Great Novel', 'Michael Johnson', 2015, 'Drama'),**

**(4, 'Adventure Awaits', 'Samantha Lee', 2021, 'Adventure');**

**select \* from book;**

**-- Select books with titles containing a specific keyword (e.g., 'novel'):**

**SELECT \* FROM book**

**WHERE title LIKE '%novel%';**

**-- Select the earliest publication year among all books**

**SELECT MIN(publication\_year) AS earliest\_year FROM book;**

**-- remove all records from book, but keeps the table structure**

**TRUNCATE TABLE book;**

1. Create a table called "employees": attributes- emp\_id, emp\_name, emp\_age, emp\_department

Insert multiple records into the "employees" table

Select employees aged 25 or younger from the HR department:

Select employees aged 30 or older from the IT department or Marketing department

Select employees aged 30 or older from the IT department and Finance department

Solution:

**CREATE TABLE employees (**

**emp\_id INT PRIMARY KEY,**

**emp\_name VARCHAR(50),**

**emp\_age INT,**

**emp\_department VARCHAR(50)**

**);**

**INSERT INTO employees (emp\_id, emp\_name, emp\_age, emp\_department)**

**VALUES**

**(1, 'John Smith', 30, 'IT'),**

**(2, 'Jane Doe', 25, 'HR'),**

**(3, 'Michael Johnson', 35, 'Finance'),**

**(4, 'Emily Williams', 28, 'Marketing');**

**-- Select employees aged 25 or younger from the HR department:**

**SELECT \* FROM employees WHERE emp\_age <= 25 AND emp\_department = 'HR';**

**-- Select employees aged 30 or older from the IT department or Marketing department**

**SELECT \* FROM employees WHERE (emp\_age >= 30 AND emp\_department = 'IT') OR emp\_department = 'Marketing';**

**-- Select employees aged 30 or older from the IT department and Finance department**

**SELECT \* FROM employees WHERE (emp\_age >= 30 AND emp\_department = 'IT') OR emp\_department = 'Finance';**

1. Create table Order with arrtibutes order\_id, customer\_id, customer\_name, order\_date, product\_id, product\_name, quantity, total\_price.

Decompose the "orders" table into three smaller tables: "customers," "products," and "order\_items" to reduce redundancy.

Solution:

**-- Original table**

**CREATE TABLE orders (**

**order\_id INT PRIMARY KEY,**

**customer\_id INT,**

**customer\_name VARCHAR(50),**

**order\_date DATE,**

**product\_id INT,**

**product\_name VARCHAR(50),**

**quantity INT,**

**total\_price DECIMAL(10, 2)**

**);**

**-- Decomposed tables**

**CREATE TABLE customers (**

**customer\_id INT PRIMARY KEY,**

**customer\_name VARCHAR(50)**

**);**

**CREATE TABLE products (**

**product\_id INT PRIMARY KEY,**

**product\_name VARCHAR(50)**

**);**

**CREATE TABLE order\_items (**

**order\_id INT,**

**product\_id INT,**

**quantity INT,**

**total\_price DECIMAL(10, 2),**

**PRIMARY KEY (order\_id, product\_id)**

**);**

**/\***

**In this example, we decomposed the "orders" table into three smaller tables: "customers," "products," and "order\_items" to reduce redundancy.**

**\*/**

1. Write a PL/SQL program to check whether a date falls on weekend i.e. SATURDAY or SUNDAY.

Input: 05-07-2023

Output: The day of the given date is WEDNESDAY and it does not fall on the weekend

Solution:

**DECLARE**

**dt1 DATE := TO\_DATE('05-07-2023', 'DD-MM-YYYY');**

**get\_day VARCHAR2(15);**

**BEGIN**

**get\_day := RTRIM(TO\_CHAR(dt1, 'DAY'));**

**IF get\_day IN ('SATURDAY', 'SUNDAY') THEN**

**dbms\_output.new\_line;**

**DBMS\_OUTPUT.PUT\_LINE**

**('The day of the given date is '||get\_day||' and it falls on weekend');**

**ELSE**

**dbms\_output.new\_line;**

**DBMS\_OUTPUT.PUT\_LINE ('The day of the given date is '||get\_day||' and it does not fall on the weekend');**

**END IF;**

**DBMS\_OUTPUT.PUT\_LINE ('Execution done successfully.');**

**END;**

**/**