**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. Which normal form eliminates transitive dependencies?
   1. 1NF (First Normal Form)
   2. 2NF (Second Normal Form)
   3. 3NF (Third Normal Form)
   4. **BCNF (Boyce-Codd Normal Form)**
2. Which of the following represents a one-to-many relationship in an ER diagram?
   1. Crow's foot notation
   2. Dashed line
   3. **Solid line**
   4. Dotted line
3. Which normal form ensures that every non-key attribute is fully functionally dependent on the primary key?
   1. 1NF (First Normal Form)
   2. **2NF (Second Normal Form)**
   3. 3NF (Third Normal Form)
   4. BCNF (Boyce-Codd Normal Form)
4. Which language is used to define the structure and integrity constraints of a database schema?
   1. Data Manipulation Language (DML)
   2. Data Control Language (DCL)
   3. **Data Definition Language (DDL)**
   4. Query Language (QL)
5. Which SQL keyword is used to create a new table in a database?
   1. ALTER TABLE
   2. DROP TABLE
   3. **CREATE TABLE**
   4. INSERT INTO
6. Which SQL function is used to find the average value of a column in a SELECT statement?
   1. **AVG()**
   2. COUNT()
   3. SUM()
   4. MAX()
7. Which of the following best describes the concept of data masking in database security?
   1. **Hiding sensitive data from unauthorized users**
   2. Encrypting data at rest and in transit
   3. Backing up data regularly
   4. Securing the physical location of the database server
8. Which of the following is not a valid cursor type?
   1. STATIC
   2. DYNAMIC
   3. FORWARD\_ONLY
   4. **RANDOM**
9. Which of the following is an advantage of using cursors in DBMS?
   1. Improved query performance
   2. Reduced memory consumption
   3. **Ability to process data row by row**
   4. Simpler and more concise code
10. Which component of DBMS is responsible for enforcing data integrity constraints?
    1. Query Compiler
    2. Schema Manager
    3. Data Dictionary
    4. **Constraint Manager**

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

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

1. Every time attribute A appears, it is matched with the same value of attribute B, but not the same value of attribute C. Therefore, it is true that:
   1. **A -> B**
   2. A -> C
   3. A -> (B,C)
   4. (B,C) -> A
2. Which of the following is not a valid aggregate function?
   1. COUNT
   2. **COMPUTE**
   3. SUM
   4. MAX
3. How can you change "Thomas" into "Michel" in the "LastName" column in the Users table?
   1. UPDATE User SET LastName = 'Thomas' INTO LastName = 'Michel'
   2. MODIFY Users SET LastName = 'Michel' WHERE LastName = 'Thomas'
   3. MODIFY Users SET LastName = 'Thomas' INTO LastName = 'Michel'
   4. **UPDATE Users SET LastName = 'Michel' WHERE LastName = 'Thomas'**
4. Which of the following statement is correct regarding the difference between TRUNCATE, DELETE and DROP command?
   * 1. DELETE operation can be rolled back but TRUNCATE and DROP operations cannot be rolled back.
     2. TRUNCATE and DROP operations can be rolled back but DELETE operations cannot be rolled back.
     3. DELETE is an example of DML, but TRUNCATE and DROP are examples of DDL.
     4. All are an example of DDL.
5. **I and III**
6. II and III
7. II and IV
8. II and IV
9. What is the need for our query to execute successfully on an existing view?
   1. The specified table must contain data.
   2. **We must have a SELECT privilege on the view.**
   3. We should have a SELECT privilege only on the specified table.
   4. The specified table must be in the same database or schema.

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

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

Insert multiple records into the "employees" table. Select employees that are in department of 'HR', 'Marketing'.

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 \* FROM employees**

**WHERE emp\_department IN ('HR', 'Marketing');**

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

Insert multiple records into the "employees" table. Select employees whose name start from J or E.

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 whose name start from J or E**

**SELECT \* FROM employees**

**WHERE emp\_name LIKE 'J%' OR emp\_name LIKE 'E%';**

1. Create table Books with attributes book\_id, title, category\_id, published\_date

Create table Categories with attributes category\_id, category\_name.

'Books' table is in 2NF already.

You have to remove the transitive dependency by creating a new table for categories.

Solution:

**-- Create Books table**

**CREATE TABLE Books (**

**book\_id INT PRIMARY KEY,**

**title VARCHAR(100),**

**category\_id INT,**

**published\_date DATE**

**);**

**-- Create Categories table**

**CREATE TABLE Categories (**

**category\_id INT PRIMARY KEY,**

**category\_name VARCHAR(50)**

**);**

**-- 3 NF**

**-- Remove category\_name from the Books table**

**ALTER TABLE Books**

**DROP COLUMN category\_id;**

**-- Create a new table for BookCategories**

**CREATE TABLE BookCategories (**

**book\_id INT PRIMARY KEY,**

**category\_id INT**

**);**

**-- Insert records into BookCategories table**

**INSERT INTO BookCategories (book\_id, category\_id)**

**VALUES (1001, 101),**

**(1002, 101),**

**(1003, 102),**

**(1004, 103),**

**(1005, 103);**

**-- Remove category\_name from the Categories table**

**ALTER TABLE Categories**

**DROP COLUMN category\_name;**

1. Write a PL/SQL block to differentiate between CHAR and VARCHAR2 datatype.

Solution:

**DECLARE**

**f\_name CHAR(15 CHAR);**

**l\_name VARCHAR2(15 CHAR);**

**BEGIN**

**f\_name := 'Allen ';**

**l\_name := 'Munra ';**

**DBMS\_OUTPUT.PUT\_LINE('\*' || f\_name || '\*');**

**DBMS\_OUTPUT.PUT\_LINE('\*' || l\_name || '\*');**

**END;**

**/**