**Roll No…………….. Total No. of Pages:……**

**ST-1 (SET-I)**

**4th SEMESTER 2023-24**

**22CS007- Database Management System**

**Time allowed: 90 Minutes Max. Marks: 40**

**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 of the following is the correct definition of a functional dependency in a relational database?
   1. A relationship between two tables
   2. **A constraint that specifies the relationship between attributes in a table**
   3. A set of operations to manipulate data in a table
   4. A technique to optimize query performance in a database
2. In an ER model, a weak entity is one that:
   1. Has a weak relationship with another entity
   2. Lacks attributes
   3. **Depends on another entity for its existence**
   4. Cannot participate in relationships
3. Which type of join returns only the matching rows between two tables?
   1. **Inner join**
   2. Left join
   3. Right join
   4. Full outer join
4. Which SQL clause is used to limit the number of rows returned in a SELECT statement?
   1. **LIMIT**
   2. TOP
   3. ROWNUM
   4. FETCH
5. Which technique is used to prevent unauthorized users from viewing data by altering the database schema?
   1. Access controls
   2. Database encryption
   3. **Data masking**
   4. Database normalization
6. What is the purpose of a cursor's WHERE CURRENT OF clause?
   1. It specifies the condition for closing the cursor.
   2. It specifies the condition for deleting a row from the cursor.
   3. **It specifies the condition for updating the current row of the cursor.**
   4. It specifies the condition for declaring a new cursor.
7. Which control structure is used to transfer the control from a function back to the caller?
   1. BREAK
   2. CONTINUE
   3. EXIT
   4. **RETURN**
8. Which of the following is a common vulnerability in database security?
   1. Buffer overflow
   2. Firewall misconfiguration
   3. Weak encryption algorithm
   4. **All of the above**
9. In the context of functional dependencies, which of the following is the highest normal form?
   1. 1NF (First Normal Form)
   2. 2NF (Second Normal Form)
   3. 3NF (Third Normal Form)
   4. **BCNF (Boyce-Codd Normal Form)**
10. Which normal form eliminates partial dependencies in a relational model?
    1. First normal form (1NF)
    2. Second normal form (2NF)
    3. **Third normal form (3NF)**
    4. Boyce-Codd normal form (BCNF)

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

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

1. Snapshot of the data in the database at a given instant of time is called
   1. Database Schema
   2. **Database Instance**
   3. Database Snapshot
   4. All of the above
2. Which of the following is not a DDL command?
3. TRUNCATE
4. ALTER
5. CREATE
6. **UPDATE**
7. Which of the following is true about the HAVING clause?
   1. Similar to the WHERE clause but is used for columns rather than groups.
   2. Similar to WHERE clause but is used for rows rather than columns.
   3. **Similar to WHERE clause but is used for groups rather than rows.**
   4. Acts exactly like a WHERE clause.
8. Which of the following is the correct order of a SQL statement?
   1. SELECT, GROUP BY, WHERE, HAVING
   2. **SELECT, WHERE, GROUP BY, HAVING**
   3. SELECT, HAVING, WHERE, GROUP BY
   4. SELECT, WHERE, HAVING, GROUP BY
9. Which statement is used to get all data from the student table whose name starts with p?
10. SELECT \* FROM student WHERE name LIKE '%p%';
11. **SELECT \* FROM student WHERE name LIKE 'p%';**
12. SELECT \* FROM student WHERE name LIKE '\_p%';
13. SELECT \* FROM student WHERE name LIKE '%p';

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

1. Create table customers with columns like 'customer\_id', 'first\_name', 'last\_name', 'eamil' and ' phone'. Insert one record in customers table and display all the records in it.

Update a customer's email.

Solution:

**CREATE TABLE customers (**

**customer\_id INT PRIMARY KEY,**

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

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

**email VARCHAR(100),**

**phone VARCHAR(20)**

**);**

**-- insert record**

**INSERT INTO customers (customer\_id, first\_name, last\_name, email, phone)**

**VALUES (1, 'John', 'Doe', 'john@example.com', '555-123-4567');**

**-- display records**

**select \* from customers;**

**-- update customer's email**

**UPDATE customers**

**SET email = 'john.doe@example.com'**

**WHERE customer\_id = 1;**

**-- display records**

**select \* from customers;**

1. Create table customers with columns like 'customer\_id', 'first\_name', 'last\_name', 'eamil' and ' phone'.

Insert one record in customers table and display all the records in it.

Delete customer's record whose id is 1

Count the number of records in table

Solution:

**CREATE TABLE customers (**

**customer\_id INT PRIMARY KEY,**

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

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

**email VARCHAR(100),**

**phone VARCHAR(20)**

**);**

**-- insert record**

**INSERT INTO customers (customer\_id, first\_name, last\_name, email, phone)**

**VALUES (1, 'John', 'Doe', 'john@example.com', '555-123-4567');**

**-- display records**

**select \* from customers;**

**-- delete record**

**DELETE FROM customers**

**WHERE customer\_id = 1;**

**-- display records**

**select \* from customers;**

**-- count the number of records in table**

**SELECT COUNT(\*) AS count from customers;**

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

Create table Authors with attributes author\_id, author\_name.

We assume that each book has only one author. However, in real-world scenarios, a book may have multiple authors.

So, your task is to decompose the Books table to remove this redundancy.

Solution:

**-- Create Books table**

**CREATE TABLE Books (**

**book\_id INT PRIMARY KEY,**

**title VARCHAR(100),**

**category\_id INT,**

**author\_id INT,**

**price DECIMAL(10, 2),**

**published\_date DATE**

**);**

**drop table if exists Authors;**

**-- Create Authors table**

**CREATE TABLE Authors (**

**author\_id INT PRIMARY KEY,**

**author\_name VARCHAR(100)**

**);**

**/\***

**We assume that each book has only one author. However, in real-world scenarios, a book may have multiple authors.**

**So, we'll decompose the Books table to remove this redundancy.**

**\*/**

**-- Create a new table for BookAuthors**

**CREATE TABLE BookAuthors (**

**book\_id INT PRIMARY KEY,**

**author\_id INT**

**);**

**-- Insert records into BookAuthors table**

**INSERT INTO BookAuthors (book\_id, author\_id)**

**VALUES (1001, 1),**

**(1002, 2),**

**(1003, 3),**

**(1004, 1),**

**(1005, 2);**

**-- Remove author\_id from the Books table**

**ALTER TABLE Books**

**DROP COLUMN author\_id;**