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

**ST-1 (SET-VI)**

**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. Normalization in database design aims to:
   1. **Minimize redundancy and eliminate data anomalies**
   2. Optimize query performance and speed
   3. Maximize data storage capacity
   4. Improve user interface design
2. Which concurrency control technique allows multiple transactions to access the database simultaneously while maintaining isolation?
   1. Two-Phase Locking (2PL)
   2. Timestamp Ordering
   3. **Multiversion Concurrency Control (MVCC)**
   4. Optimistic Concurrency Control (OCC)
3. Which SQL operator is used to combine multiple conditions in a WHERE clause?
   1. **AND**
   2. OR
   3. NOT
   4. XOR
4. Which SQL keyword is used to retrieve data from a database table?
   1. **SELECT**
   2. INSERT
   3. UPDATE
   4. DELETE
5. Which security measure ensures that data can be restored to its original state in the event of data loss or corruption?
   1. Role-based access control
   2. Data replication
   3. **Backup and recovery**
   4. Intrusion detection system
6. Which normal form eliminates all transitive dependencies in a relational database?
   1. First Normal Form (1NF)
   2. Second Normal Form (2NF)
   3. Third Normal Form (3NF)
   4. **Boyce-Codd Normal Form (BCNF)**
7. Which SQL statement is used to declare a cursor?
   1. OPEN
   2. SELECT
   3. **DECLARE**
   4. FETCH
8. Which SQL statement is used to close a cursor?
   1. **CLOSE**
   2. DEALLOCATE
   3. RELEASE
   4. FREE
9. In a DBMS, which control structure is used to selectively execute a block of statements based on a condition?
   1. FOR
   2. DO-WHILE
   3. **IF-ELSE**
   4. CASE
10. Which type of backup includes all the data that has changed since the last full backup?
    1. Incremental backup
    2. **Differential backup**
    3. Snapshot backup
    4. Mirror backup

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

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

1. A Key which is a set of one or more columns that can identify a record uniquely is called?
   1. Natural key
   2. **Candidate key**
   3. Not Null key
   4. Alternate key
2. Which of the following statement is true?
   1. **TRUNCATE free the table space while DELETE does not.**
   2. Both TRUNCATE and DELETE statements free the table's space.
   3. Both TRUNCATE and DELETE statement does not free the table's space.
   4. DELETE free the table space while TRUNCATE does not.
3. A sequence in SQL can generate a maximum number:
   1. 39 digits
   2. **38 digits**
   3. 40 digits
   4. 37 digits
4. Which of the following statement is correct to display all the cities with the condition, temperature, and humidity whose humidity is in the range of 60 to 75 from the 'whether' table?
   1. SELECT \* FROM weather WHERE humidity IN (60 to 75)
   2. **SELECT \* FROM weather WHERE humidity BETWEEN 60 AND 75**
   3. SELECT \* FROM weather WHERE humidity NOT IN (60 AND 75)
   4. SELECT \* FROM weather WHERE humidity NOT BETWEEN 60 AND 75
5. Which of the following SQL query is correct for selecting the name of staffs from 'staffinfo' table where salary is 10,000 or 25,000?
   1. SELECT name FROM staffinfo WHERE salary BETWEEN 10000 AND 25000;
   2. **SELECT name FROM staffinfo WHERE salary IN (10000, 25000);**
   3. Both A and B
   4. None of the above

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

1. Create a table for employee with attributes- emp\_id, emp\_name, emp\_salary, department\_id

Insert multiple records into the employees table

Create a table for departments with attibutes- department\_id, department\_name

Insert multiple records into the departments table

Perform an inner join between employees and departments:

Perform a left join between employees and departments

Perform a right join between employees and departments

Solution:

**CREATE TABLE employee (**

**emp\_id INT PRIMARY KEY,**

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

**emp\_salary DECIMAL(10, 2),**

**department\_id INT**

**);**

**INSERT INTO employee (emp\_id, emp\_name, emp\_salary, department\_id)**

**VALUES (1, 'John Doe', 50000.00, 101),**

**(2, 'Jane Smith', 60000.00, 102),**

**(3, 'Michael Johnson', 55000.00, 101),**

**(4, 'Emily Brown', 52000.00, 103),**

**(5, 'William Davis', 58000.00, 102);**

**select \* from employee;**

**CREATE TABLE departments (**

**department\_id INT PRIMARY KEY,**

**department\_name VARCHAR(50)**

**);**

**INSERT INTO departments (department\_id, department\_name)**

**VALUES (101, 'HR'),**

**(102, 'Finance'),**

**(103, 'Marketing');**

**-- Perform an inner join between employees and departments:**

**SELECT emp\_name, emp\_salary, department\_name**

**FROM employee**

**INNER JOIN departments ON employee.department\_id = departments.department\_id;**

**-- Perform a left join between employees and departments**

**SELECT emp\_name, emp\_salary, department\_name**

**FROM employee**

**LEFT JOIN departments ON employee.department\_id = departments.department\_id;**

**-- Perform a right join between employees and departments**

**SELECT emp\_name, emp\_salary, department\_name**

**FROM employee**

**RIGHT JOIN departments ON employee.department\_id = departments.department\_id;**

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

Insert multiple records into the "employees" table

Add a new column "salary" to the "employees" table

Increase the salary of employee in the "IT" department by 10% whose id is 1

In employees table rename the salary column as emp\_salary

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');**

**-- Add a new column "salary" to the "employees" table**

**ALTER TABLE employees**

**ADD COLUMN salary INT;**

**desc employees;**

**-- Increase the salary of employee in the "IT" department by 10% whose id is 1**

**UPDATE employees**

**SET salary = salary \* 1.10**

**WHERE emp\_department = 'IT' and emp\_id=1;**

**-- In employees table rename the salary column as emp\_salary**

**ALTER TABLE employees**

**RENAME COLUMN salary TO emp\_salary;**

1. Create table student with attributes student\_id, student\_name, age, address, phone\_number.

Decompose the "student" table into "student\_personal\_info" and "student\_contact\_info" tables based on attribute groups

Solution:

**-- Original table**

**CREATE TABLE student (**

**student\_id INT PRIMARY KEY,**

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

**age INT,**

**address VARCHAR(100),**

**phone\_number VARCHAR(15)**

**);**

**-- Decomposed tables**

**CREATE TABLE student\_personal\_info (**

**student\_id INT PRIMARY KEY,**

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

**age INT**

**);**

**CREATE TABLE student\_contact\_info (**

**student\_id INT PRIMARY KEY,**

**address VARCHAR(100),**

**phone\_number VARCHAR(15)**

**);**

**/\***

**In this example, we decomposed the "student" table into "student\_personal\_info" and "student\_contact\_info" tables based on attribute groups.**

**\*/**

1. Write a program in PL/SQL to display a cursor based detail information of employees from employees table.

Solution:

**-- Create the "employee" table**

**CREATE TABLE employee (**

**employee\_id NUMBER PRIMARY KEY,**

**first\_name VARCHAR2(50),**

**last\_name VARCHAR2(50),**

**department VARCHAR2(50),**

**salary NUMBER**

**);**

**-- Insert sample records into the "employee" table**

**INSERT INTO employee (employee\_id, first\_name, last\_name, department, salary)**

**VALUES (1, 'John', 'Doe', 'HR', 50000);**

**INSERT INTO employee (employee\_id, first\_name, last\_name, department, salary)**

**VALUES (2, 'Jane', 'Smith', 'Finance', 60000);**

**INSERT INTO employee (employee\_id, first\_name, last\_name, department, salary)**

**VALUES (3, 'Michael', 'Johnson', 'IT', 70000);**

**DECLARE**

**CURSOR z\_emp\_info IS**

**SELECT employee\_id,**

**first\_name,**

**last\_name,**

**salary**

**FROM employee;**

**r\_emp\_info z\_emp\_info%ROWTYPE;**

**BEGIN**

**OPEN z\_emp\_info;**

**LOOP**

**FETCH z\_emp\_info INTO r\_emp\_info;**

**EXIT WHEN z\_emp\_info%NOTFOUND;**

**dbms\_output.Put\_line('Employees Information:: '**

**||' ID: '**

**||r\_emp\_info.employee\_id**

**||' Name: '**

**||r\_emp\_info.first\_name**

**||' '**

**||r\_emp\_info.last\_name);**

**END LOOP;**

**dbms\_output.Put\_line('Total number of rows : '**

**||z\_emp\_info%rowcount);**

**CLOSE z\_emp\_info;**

**END;**

**/**