

DBMS Assignment

1. What do you understand By Database?

Ans - A database is a structured collection of data that is organized and stored in a way that allows for efficient data retrieval, management, and manipulation. Databases are used to store and manage large volumes of structured information, making it easier to access, update, and analyze data.

2. What is Normalization?

Ans - Normalization is a process in database design that involves organizing the data in a relational database into separate tables to reduce data redundancy and improve data integrity. It typically involves breaking down large tables into smaller, related tables and establishing relationships between them. The goal of normalization is to eliminate data anomalies like insertion, update, and deletion anomalies.

3. What is the Difference between DBMS and RDBMS?

Ans - DBMS (Database Management System) is a software system that manages databases, while RDBMS (Relational Database Management System) is a specific type of DBMS that manages data in a structured manner using tables with rows and columns. RDBMS enforces the relational model, which includes features like data integrity through constraints and support for SQL (Structured Query Language).

4. What is the ACID Rule of RDBMS Systems?

Ans - ACID stands for Atomicity, Consistency, Isolation, and Durability, which are the key properties of a transaction in a reliable database system. These properties ensure that database transactions are processed reliably and maintain data integrity. For example, if a transaction fails, the ACID properties ensure that the database returns to a consistent state.

5. What do you understand By Data Redundancy?

Ans - Data redundancy refers to the duplication of data in a database system. When the same data is stored in multiple places, it can lead to inconsistency and inefficiency. Normalization is one technique used to reduce data redundancy by organizing data into separate tables and linking them with relationships.

6. What is DDL Interpreter?

Ans - DDL (Data Definition Language) Interpreter is a component of a database management system responsible for processing and executing DDL statements. DDL statements are used to define and manage the structure of the database, such as creating tables, altering table structures, and defining constraints.

7. What is DML Compiler in SQL?

Ans - DML (Data Manipulation Language) Compiler is a component of a database management system responsible for processing and executing DML statements. DML statements are used to manipulate data within the database, such as inserting, updating, and deleting records. SQL queries used for data retrieval are also considered DML statements.

8. What is SQL Key Constraints writing an Example of SQL Key Constraints?

Ans => SQL Key Constraints are rules applied to columns in database tables to enforce data integrity and establish relationships between tables. Common key constraints include:

- => PRIMARY KEY: Ensures uniqueness and identifies each record in a table.
- => FOREIGN KEY: Establishes a link between tables, ensuring referential integrity.
- => UNIQUE KEY: Enforces uniqueness but does not necessarily identify each record.
- => CHECK CONSTRAINT: Specifies a condition that data must meet.

Example of PRIMARY KEY constraint in SQL:

```
CREATE TABLE Students (  
    StudentID INT PRIMARY KEY,  
    FirstName VARCHAR(50),  
    LastName VARCHAR(50)  
);
```

9. What is a Save Point? How to create a Save Point write a Query?

Ans - A savepoint is a point within a transaction where you can mark a specific spot to which you can later roll back if needed. Savepoints are used to provide more granular control over transaction management.

syntax :- **SAVEPOINT my_savepoint;**

10. What is a Trigger, and how to create a Trigger in SQL?

Ans - A trigger is a database object that automatically performs an action when a specific event (such as an INSERT, UPDATE, DELETE, or other database operation) occurs in the database. Triggers are often used for enforcing business rules, logging changes, or maintaining data integrity.

Example of a trigger in SQL:

```
CREATE TRIGGER audit_log  
AFTER INSERT ON Customers  
FOR EACH ROW  
BEGIN  
    INSERT INTO CustomerAuditLog (CustomerID, Action, Timestamp)  
    VALUES (NEW.CustomerID, 'INSERT', NOW());
```

Task

1.Create two tables : Student and Exam.

```
CREATE TABLE Student (  
    Rollno INT PRIMARY KEY,  
    Name VARCHAR(20),  
    Branch VARCHAR(50)  
);  
  
CREATE TABLE Exam (  
    Rollno INT,  
    S_code VARCHAR(20),  
    Marks INT,  
    P_code VARCHAR(10),  
    FOREIGN KEY (Rollno) REFERENCES Student(Rollno)  
);
```

```
insert into student(Rollno, Name, Branch) values (1, 'Jay', 'Computer Science'),(2,  
'Suhani','Electronic and Com'),(3, 'Kriti','Electronic and Com');
```

```
insert into Exam(Rollno, S_code, Marks, P_code) values (1, 'CS11', 50, 'CS'),(1,'CS12', 60, 'CS'),(2,'EC101', 66, 'EC'),(2,'EC102', 70, 'EC'),(3,'EC101', 45, 'EC'),(3,'EC102', 50, 'EC');
```

2.Create table given below

```
CREATE TABLE MyTable (  
    First_Name VARCHAR(20),  
    Last_Name VARCHAR(20),  
    Address VARCHAR(50),  
    City VARCHAR(50),  
    Age INT  
);
```

```
insert into MyTable(First_Name, Last_Name, Address, City, Age) values  
('Mickey','Mouse','123 Fantasy Way','Anaheim',73),('Bat','Man','321 Cavern  
Ave','Gotham',54),('Wonder','Woman','987 Truth  
Way','Paradise',39),('Donald','Duck','555 Quack  
Street','Mallard',65),('Bugs','Bunny','567 Carrot  
Street','Rascal',58),('Wiley','Coyote','999 Acme  
Way','Canyon',61),('Cat','Woman','234 Purrfect  
Street','Hairball',32),('Tweety','Bird','543','Itotltaw',28);
```

3.Create two tables : Employee and Incentive

- a) Get First_Name from employee table using Tom name “Employee Name”.

```
SELECT First_Name  
FROM employee  
WHERE "Employee Name" = 'Tom';
```

- b) Get FIRST_NAME, Joining Date, and Salary from employee table.

```
SELECT FIRST_NAME, Joining_Date, Salary  
FROM employee;
```

- c) Get all employee details from the employee table order by First_Name Ascending and Salary descending?

```
SELECT *  
FROM employee
```

ORDER BY First_Name ASC, Salary DESC;

- d) Get employee details from employee table whose first name contains 'J'.

**SELECT *
FROM employee
WHERE First_Name LIKE '%J%';**

- e) Get department wise maximum salary from employee table order by salary ascending?

**SELECT Department, MAX(Salary) AS MaxSalary
FROM employee
GROUP BY Department
ORDER BY MaxSalary ASC;**

- f) Select first_name, incentive amount from employee and incentives table for those employees who have incentives and incentive amount greater than 3000.

**SELECT e.first_name, i.incentive_amount
FROM employee e
JOIN incentives i ON e.employee_id = i.employee_id
WHERE i.incentive_amount > 3000;**

- g) Create After Insert trigger on Employee table which insert records in view table.

**CREATE TRIGGER Employee_InsertTrigger
AFTER INSERT
ON Employee
FOR EACH ROW
BEGIN

INSERT INTO ViewTable (EmployeeID, FirstName, LastName)
VALUES (NEW.EmployeeID, NEW.FirstName, NEW.LastName);
END;**

4. Create table given below: Salesperson and Customer.

**CREATE TABLE SALESPERSON (
"(PK)SNo" INT (20),**

```

        SNAME VARCHAR (20),
        CITY VARCHAR (50),
        COMM VARCHAR (10)
    );
insert into SALESPERSON ("(PK)SNo", SNAME, CITY, COMM) VALUES
(1001, "peel", "London", .12),(1002, "Serres", "San Jose", .13),(1004,
"Motika", "London", .11),(1007, "Rafkin", "Barcelona", .15),(1003, "Axelrod",
"New York", .1);

```

TABLE 2 : CUSTOMER

```

CREATE TABLE CUSTOMER (
    "(PK)CNM." INT (20),
    CNAME VARCHAR (20),
    CITY VARCHAR (50),
    RATING INT (10),
    "(FK)SNo." INT (20) );

insert into CUSTOMER ("(PK)CNM.", CNAME, CITY, RATING, (FK)SNo)
VALUES (201, "Hoffman", "London", 100, 1001),(202, "Giovanne", "Roe",
200, 1003),(203, "Liu", "San Jose", 300, 1002),(204, "Grass", "Barcelona",
100, 1002),(206, "Clemens", "London", 300, 1007),(207, "Pereira", "Roe",
100, 1004);

```

a) All orders for more than \$1000.

```

SELECT * FROM SALESPERSON
WHERE OrderAmount > 1000;

```

b) Names and cities of all salespeople in London with commission above 0.12.

```

SELECT Name, City FROM SALESPERSON
WHERE City = 'London' AND Commission > 0.12;

```

c) All salespeople either in Barcelona or in London.

```

SELECT * FROM SALESPERSON
WHERE City IN ('Barcelona', 'London');

```

d) All salespeople with commission between 0.10 and 0.12. (Boundary values should be excluded).

SELECT * FROM SALESPERSON

WHERE Commission > 0.10 AND Commission < 0.12;

e) All customers excluding those with rating <= 100 unless they are located in Rome

SELECT * FROM CUSTOMER

WHERE (Rating > 100 OR City = 'Rome');

THANK YOU 😊