Software Engineering Assignment

MODULE: 5 (DATABASE)

Q1. What do you understand By Database.

Ans.

- database is a structured collection of data that is organized and stored in a way that allows for efficient retrieval, management, and manipulation of information.
- Databases are used to store and manage vast amounts of data, making it easy to access and update information as needed.
- They are a fundamental component of modern information systems and are used in a wide range of applications, including websites, business applications, scientific research, and more.

Q2. What is Normalization?

Ans.

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate undesirable characteristics like Insertion, Update, and Deletion Anomalies.
- Normalization divides the larger table into smaller and links them using relationships.

- The normal form is used to reduce redundancy from the database table.

Q3. What is Difference between DBMS and RDBMS?

Ans.

DBMS:

- Data stored is in the file format.
- Individual access of data elements.
- No connection between data.
- No support for distributed database.
- DBMS supports a single user.
- The software and hardware requirements are higher.

RDBMS:

- Data stored is in table format.
- Multiple data elements are accessible together.
- Data in the form of a table are linked together.
- Support distributed database.
- DBMS supports a multiple user.
- The software and hardware requirements are higher.

Q4. What is MF Cod Rule of RDBMS Systems?

Ans.

- The MF Cod Rule of RDBMS Systems states that for a system to qualify as an RDBMS, it must be able to manage database entirely through the relational capabilities.

- Rule 0 of the MF Cod Rules states that the system must qualify as relational, as a database, and as a management system. For a system to qualify as an RDBMS, that system must use its relational facilities exclusively to manage the database.

Q5. What do you understand By Data Redundancy?

Ans.

- In DBMS, when the same data is stored in different tables, it causes data redundancy.
- Sometimes, it is done on purpose for recovery or backup of data, faster access of data, or updating data easily. Redundant data costs extra money, demands higher storage capacity, and requires extra effort to keep all the files up to date.
- Data redundancy means the occurrence of duplicate copies of similar data. It is done intentionally to keep the same piece of data at different places, or it occurs accidentally.

- In DBMS, when the same data is stored in different tables, it causes data redundancy.
- Sometimes, it is done on purpose for recovery or backup of data, faster access of data, or updating data easily.
- Redundant data costs extra money, demands higher storage capacity, and requires extra effort to keep all the files up to date.

Software Engineering Assignment

TASK

Q1. Create Table Name : Student and Exam.

- create table student(Rollno int auto_increment not null primary key,Name varchar(15) not null,Branch varchar(20) not null);
- insert into student (name,branch) values("jay","Computer science"),("suhani","Electronics and Com."),("Kirti","Electronics and Com");
- create table exam(Rollno int not null ,foreign key(Rollno) references student(Rollno),scode varchar(20),marks int(100),pcode varchar(20));

- insert into exam(Rollno,scode,marks,pcode) 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");

Q2. Create table given below

- create table emp(FirstName varchar(20) not null,LastName varchar(20) not null,Address varchar(25) not null,City varchar(15) not null,Age int(5) not null);

- insert into emp(FirstName,LastName,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",56),("Bugs","Bunny","567 Carrot Street","Rascal",58),("Wiley","Coyote","999 Acme Way","Canyon",61),("Cat","Woman","234 Purrfect Street","Hairball",32),("Tweety","Bird","543","Itotlow" ,28);

Q3.Create table Employee and Incentive.

Table Name: Employee

- create table Employee(Employee_id int auto_increment not null primary key,First_name varchar(15) not null,Last_name varchar(15) not null, Salary int(10) not null,Joining_date datetime not null,Department varchar(20) not null);

- insert into Employee(First_name,Last_name,Salary,Joining_date,Department)values('John','Abraham',10000000,'2013-01-01 12:00:00','Banking'),('Michael','Clarke',800000,'2013-01-01 12:00:00','Insurance'),('Roy','Thomes',700000,'2013-02-01 12:00:00','Banking'),('Tom','Jose',600000,'2013-02-01 12:00:00','Insurance'),('Jerry','Pinto',650000,'2013-

02-01 12:00:00','Insurance'),('Philip','Mathew',750000,'2013-01-01 12:00:00','Service'),('TestName1','123',650000,'2013-01-01 12:00:00','Service'),('TestName2','Lname%',600000, '13-02-01 12:00:00','Insurance');

- create table Incentive(Employee_ref_id int not null,Incentive_date date not null,Incentive_amount int(10) not null);

- insert into Incentive (Employee_ref_id,Incentive_date,Incentive_amount)values(1,'2013-02-01',5000),(2,'2013-02-01',3000),(3,'2013-02-01',4000),(1,'2013-01-01',4500),(2,'2013-01-01',3500);

A. Get First_Name from employee table using Tom name “Employee Name”.
   - SELECT First_name FROM Employee WHERE First_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 max_salary FROM employee GROUP BY Department ORDER BY max_salary 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 Employee.First_name,incentive.Incentive_amount FROM Employee JOIN incentive ON Employee.Employee_id=incentive.Employee_ref_id WHERE incentive.Incentive_amount > 3000;

G. Create After Insert trigger on Employee table which insert records in view table.

- ➢ create table viewtable(employee_id int auto_increment primary key ,first_name varchar(15) not null,last_name varchar(15) not null,salary int(10) not null,joining_date datetime not null,department varchar(20));

- create trigger emp_3_insert_trigger after insert on viewtable for each row insert into emp_3(first_name,last_name,salary,joining_date, department)values ('John','Abraham',10000000,'2013-1-01 12:00:00 ','Banking');

Q4.Create table given below: Salesperson and Customer.

- create table SALESPERSON(SNO int auto_increment not null primary key,SNAME varchar(15) not null,CITY varchar(20) not null,COMM float(5) not null);

- insert into SALESPERSON(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);

- create table CUSTOMER(CNM int(5) primary key not null ,CNAME varchar(15) not null,CITY varchar(15) not null,RATING int(5) not null,SNO int(10),foreign key(SNO) references SALESPERSON(SNO));

- insert into CUSTOMER(CNM,CNAME,CITY,RATING,SNO)values(201,'Hoffman','London',100,1001),(202,'Giovanne','Roe',200,1003),(203,'Liu','San Jose',300,1002),(204,'Grass','Barcelona',100,1002), (206,'Clements','London',300,1007),(207,'Pereria','Roe',100,1004);

❖ **Retrieve the below data from above table**

A)All orders for more than $1000.

➢ Select * from CUSTOMER where order >= 1000;

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

➢ select SNAME,CITY,COMM from SALESPERSON where (CITY='London' and COMM>'.12');

C)All salespeople either in Barcelona or in London

➢ SELECT * FROM salesperson WHERE (CITY = 'Barcelona' OR CITY = 'London');

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

➢ SELECT * FROM salesperson WHERE COMM > 0.10 AND COMM < 0.12;

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

➢ SELECT *FROM CUSTOMER WHERE NOT (RATING <= 100 AND CITY = 'Rome');