

Module : 5 Database

1. What do you understand By Database.

Ans >> Database is a collection of raw data which stores the individual data in meaningful fact. It is basically a shared collection of interrelated data tables which are designed from varied information from organisation.

2. What is Normalization?

Ans >> Normalization is an important process in database design that helps improve the database's efficiency, consistency, and accuracy. It makes it easier to manage and maintain the data and ensures that the database is adaptable to changing business needs.

3. What is Difference between DBMS and RDBMS?

Ans >> DBMS stands for Database Management System, and RDBMS is the acronym for the Relational Database Management system. In DBMS, the data is stored as a file, whereas in RDBMS, data is stored in the form of tables.

4. What is MF Cod Rule of RDBMS Systems?

Ans >> Codd's rules are proposed by a computer scientist named Dr. Edgar F. Codd and he also invent the relational model for database management. These rules are made to ensure data integrity, consistency, and usability. This set of rules basically signifies the characteristics and requirements of a relational database management system (RDBMS). In this article, we will learn about various Codd's rules.

5. What do you understand By Data Redundancy ?

Ans >> Data redundancy is when multiple copies of the same information are stored in more than one place at a time. This challenge plagues organizations of all sizes in all industries and leads to elevated storage costs, errors, and compromised analytics.

6. What is DDL Interpreter?

Ans >> DDL stands for Data Definition Language. The DDL interpreter changes the DDL statements into a specific format to make sense to the storage manager. The DDL also ensures the consistency and validity of the database. DML Compiler - DML stands for Data manipulation language .

7. What is DML Compiler in SQL?

Ans >> DML Compiler DML expands to Data Manipulation Language in DBMS. DML Compiler again as the name suggests compiles(or translates) the DML statements such as select, update, and delete statements into low-level instructions which is nothing but the machine-readable object code to make it executable.

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

Ans >> SQL key constraints are rules that define how data is managed in a database. They help ensure data integrity and consistency. The most common types of SQL key constraints include:

- **Primary Key:** Uniquely identifies each record in a table.

Ex >> CREATE TABLE Colleges (college_id INT PRIMARY KEY,college_code VARCHAR(20) NOT NULL,college_name VARCHAR(50));

- **Foreign Key:** Ensures referential integrity by linking a column or a set of columns in one table to the primary key in another table.

Ex >> CREATE TABLE Orders (order_id INT PRIMARY KEY,customer_id int REFERENCES Customers(id));

- **Unique Key:** Ensures all values in a column or a set of columns are unique.

Ex >> CREATE TABLE Colleges (college_id INT NOT NULL UNIQUE, college_code VARCHAR(20) UNIQUE, college_name VARCHAR(50));

- **Not Null Constraint:** Ensures that a column cannot have a NULL value.

Ex>> CREATE TABLE Colleges (college_id INT NOT NULL,college_code VARCHAR(20) NOT NULL, college_name VARCHAR(50));

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

ANS>> The SAVEPOINT command is used to temporarily save a transaction so that you can roll back to the point whenever required. The different states of our table can be saved at any time using different names and the rollback to that state can be done using the ROLLBACK command.

EX>>

SAVEPOINT savepoint_name;

UPDATE Student SET Name = 'Mini 'WHERE Admno = 105;

SAVEPOINT A;

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

ANS>> A trigger is a stored procedure in a database that automatically invokes whenever a special event in the database occurs. For example, a trigger can be invoked when a row is inserted into a specified table or when specific table columns are updated. In simple words, a trigger is a collection of SQL statements with particular names that are stored in system memory. It belongs to a specific class of stored procedures that are automatically invoked in response to database server events. Every trigger has a table attached to it.

Because a trigger cannot be called directly, unlike a stored procedure, it is referred to as a special procedure. A trigger is automatically called whenever a data modification event against a table takes place, which is the main distinction between a trigger and a procedure. On the other hand, a stored procedure must be called directly.

```
Trigger>> delimiter //  
create procedure getall()  
begin  
select * from stud;  
end //  
delimiter;
```

```
call getall();
```

- **Tasks**

1. Create Table Name : Student and Exam

ANS >>

```
use taskdb;
```

```
create table Student(Rollno integer primary key auto_increment, Name varchar(10), Branch  
varchar(30));
```

```
insert into Student (Name,Branch)values
```

```
('Jay','Computer Science'),
```

```
('Suhani','Electronic and Com '),
```

```
('Kriti','Electronic and Com ');
```

```
create table Exam(Rollno integer ,foreign key (Rollno) references Student (Rollno),S_code  
varchar(10),Marks integer,P_code varchar(10));
```

```
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');
```

```
select * from Student;
```

```
select * from exam;
```

Output :

Student Table :

	Rollno	Name	Branch
▶	1	Jay	Computer Science
	2	Suhani	Electronic and Com
	3	Kriti	Electronic and Com
*	NULL	NULL	NULL

Exam Table :

	Rollno	S_code	Marks	P_code
▶	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

ANS >>

use taskdb;

```
create table persondata(id integer primary key auto_increment, FirstName varchar(10),LastName  
varchar(10),Address varchar(30),City varchar(10),age integer);
```

```
insert into persondata(FirstName,LastName,Address,City,Age)values
```

```
('Mickey','Mouse','123 Fantasy Way ','Anaheim',73),
```

```
('Bat','Man','321 Cavern Ave','Gotham',54),
```

```
('Wonder','Women','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),
```

```
('Twenty','Bird','543','Itotitaw',28);
```

```
select * from persondata;
```

Output :

Persondata Table :

	id	FirstName	LastName	Address	City	age
▶	1	Mickey	Mouse	123 Fantasy Way	Anaheim	73
	2	Bat	Man	321 Cavern Ave	Gotham	54
	3	Wonder	Women	987 Truth Way	Paradise	39
	4	Donald	Duck	555 Quack Street	Mallard	65
	5	Bugs	Bunny	567 Carrot Street	Rascal	58
	6	Wiley	Coyote	999 Acme Way	Canyon	61
	7	Cat	Woman	234 Purrfect Street	Hairball	32
	8	Twenty	Bird	543	Itotitaw	28
*	NULL	NULL	NULL	NULL	NULL	NULL

3.Create table given below: Employee and Incentive

ANS >>

use taskdb;

create table Employee(

Employee_id integer primary key auto_increment,

First_name varchar(10),

Last_name varchar(10),

Salary int,

joining_date datetime,

Department varchar(20));

create table Incentive(

Employee_ref_id int , foreign key (Employee_ref_id) references Employee (Employee_id),

Incentive_date date ,

Incentive_amount int

);

```
insert into Employee(First_name,Last_name,Salary,joining_date,Department)values
('John','Abraham',1000000,'2013-01-01 12:00:00','Banking'),
('Micheal','Clarke',800000,'2013-01-01 12:00:00','Insurance'),
('Roy','Thomas',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','Services'),
('TestName1','123',650000,'2013-01-01 12:00:00','Services'),
('TestName2','Lname%',600000,'2013-02-01 12:00:00','Insurance');
```

```
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);
```

```
select * from Employee;
```

```
select * from Incentive;
```

Output >>

Employee Table :

	Employee_id	First_name	Last_name	Salary	joining_date	Department
▶	1	John	Abraham	1000000	2013-01-01 12:00:00	Banking
	2	Micheal	Clarke	800000	2013-01-01 12:00:00	Insurance
	3	Roy	Thomas	700000	2013-02-01 12:00:00	Banking
	4	Tom	Jose	600000	2013-02-01 12:00:00	Insurance
	5	Jerry	Pinto	650000	2013-02-01 12:00:00	Insurance
	6	Philip	Mathew	750000	2013-01-01 12:00:00	Services
	7	TestName1	123	650000	2013-01-01 12:00:00	Services
	8	TestName2	Lname%	600000	2013-02-01 12:00:00	Insurance
	9	Rahul	Kanara	1000000	2013-01-01 12:00:00	Banking
*	NULL	NULL	NULL	NULL	NULL	NULL

Incentive Table :

	Employee_ref_id	Incentive_date	Incentive_amount
▶	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”.

ANS>> select First_name from Employee where First_name = 'Tom';

	First_name
▶	Tom

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

ANS>> select First_name,joining_date,Salary from Employee;

	First_name	joining_date	Salary
▶	John	2013-01-01 12:00:00	1000000
	Micheal	2013-01-01 12:00:00	800000
	Roy	2013-02-01 12:00:00	700000
	Tom	2013-02-01 12:00:00	600000
	Jerry	2013-02-01 12:00:00	650000
	Philip	2013-01-01 12:00:00	750000
	TestName1	2013-01-01 12:00:00	650000
	TestName2	2013-02-01 12:00:00	600000

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

ANS>> select * from Employee order by First_name , Salary desc;

	Employee_id	First_name	Last_name	Salary	joining_date	Department
▶	5	Jerry	Pinto	650000	2013-02-01 12:00:00	Insurance
	1	John	Abraham	1000000	2013-01-01 12:00:00	Banking
	2	Micheal	Clarke	800000	2013-01-01 12:00:00	Insurance
	6	Philip	Mathew	750000	2013-01-01 12:00:00	Services
	9	Rahul	Kanara	1000000	2013-01-01 12:00:00	Banking
	3	Roy	Thomas	700000	2013-02-01 12:00:00	Banking
	7	TestName1	123	650000	2013-01-01 12:00:00	Services
	8	TestName2	Lname%	600000	2013-02-01 12:00:00	Insurance
	4	Tom	Jose	600000	2013-02-01 12:00:00	Insurance

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

ANS>> select * from Employee where First_name like '%J%';

	Employee_id	First_name	Last_name	Salary	joining_date	Department
▶	1	John	Abraham	1000000	2013-01-01 12:00:00	Banking
	5	Jerry	Pinto	650000	2013-02-01 12:00:00	Insurance
*	NULL	NULL	NULL	NULL	NULL	NULL

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

ANS>> select max(Salary),Department from Employee group by Department order by max(Salary);

	max(Salary)	Department
▶	750000	Services
	800000	Insurance
	1000000	Banking

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

ANS>> select Employee.First_name ,Incentive.Incentive_amount from Employee join Incentive on Employee.Employee_id = Incentive.Employee_ref_id where Incentive_amount>3000;

	First_name	Incentive_amount
▶	John	5000
	Roy	4000
	John	4500
	Micheal	3500

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

ANS>> create table Employee_backup(id int primary key auto_increment,

```
First_name varchar(10),  
Last_name varchar(10),  
Salary int,  
joining_date datetime,  
Department varchar(20));
```

```
delimiter //
```

```
create trigger Emplog
```

```
after insert
```

```
on Employee
```

```
for each row
```

```
begin
```

```
insert into Employee_backup(First_name,Last_name,Salary,joining_date,Department)values  
(NEW.First_name,NEW.Last_name,NEW.Salary,NEW.joining_date,NEW.Department);
```

```
end //
```

```
delimiter ;
```

```
insert into Employee(First_name,Last_name,Salary,joining_date,Department)values  
('Rahul','Kanara',1000000,'2013-01-01 12:00:00','Banking');
```

```
select * from Employee;
```

```
select * from Employee_backup;
```

	Employee_id	First_name	Last_name	Salary	joining_date	Department
▶	1	John	Abraham	1000000	2013-01-01 12:00:00	Banking
	2	Micheal	Clarke	800000	2013-01-01 12:00:00	Insurance
	3	Roy	Thomas	700000	2013-02-01 12:00:00	Banking
	4	Tom	Jose	600000	2013-02-01 12:00:00	Insurance
	5	Jerry	Pinto	650000	2013-02-01 12:00:00	Insurance
	6	Philip	Mathew	750000	2013-01-01 12:00:00	Services
	7	TestName1	123	650000	2013-01-01 12:00:00	Services
	8	TestName2	Lname%	600000	2013-02-01 12:00:00	Insurance
	9	Rahul	Kanara	1000000	2013-01-01 12:00:00	Banking
★	NULL	NULL	NULL	NULL	NULL	NULL

	id	First_name	Last_name	Salary	joining_date	Department
▶	1	Rahul	Kanara	1000000	2013-01-01 12:00:00	Banking
★	NULL	NULL	NULL	NULL	NULL	NULL

4.Create table given below: Salesperson and Customer

ANS>>

use taskdb;

create table SALSEPERSON (

SNo integer primary key ,

SNAME varchar(10),

CITY varchar(10),

COMM real);

insert into SALSEPERSON(SNo,SNAME,CITY,COMM)values

(1001,'Peel','London',0.12),

(1002,'Serres','San Jose',0.13),

(1004,'Motika','London',0.11),

(1007,'Rafkin','Barcelona',0.15),

(1003,'Axelrod','New York',0.01);

```

create table CUSTOMER(
CNM int primary key auto_increment,
CNAME varchar (10),
CITY varchar (10),
RATING int ,
SNo int , foreign key (SNo) references SALSEPERSON(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),
(205,'Clemens','London',300,1007),
(206,'Pereira','Roe',100,1004);

```

```

select * from SALSEPERSON;

select * from CUSTOMER;

```

OUTPUT:

SALSEPERSON Table :

	SNo	SNAME	CITY	COMM
▶	1001	Peel	London	0.12
	1002	Serres	San Jose	0.13
	1003	Axelrod	New York	0.01
	1004	Motika	London	0.11
	1007	Rafkin	Barcelona	0.15
*	NULL	NULL	NULL	NULL

CUSTOMER Table :

	CNM	CNAME	CITY	RATING	SNo
▶	201	Hoffman	London	100	1001
	202	Giovanne	Roe	200	1003
	203	Liu	San Jose	300	1002
	204	Grass	Barcelona	100	1002
	205	Clemens	London	300	1007
	206	Pereira	Roe	100	1004
*	NULL	NULL	NULL	NULL	NULL

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

ANS>> select SNAME,CITY from SALESPERSON where CITY='London' and COMM>0.12;

b) All salespeople either in Barcelona or in London

ANS>> select * from SALESPERSON where CITY='Barcelona' or CITY='London';

	SNo	SNAME	CITY	COMM
▶	1001	Peel	London	0.12
	1004	Motika	London	0.11
	1007	Rafkin	Barcelona	0.15
*	NULL	NULL	NULL	NULL

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

ANS>> select * from SALESPERSON where COMM between 0.10 and 0.12;

	SNo	SNAME	CITY	COMM
▶	1001	Peel	London	0.12
	1004	Motika	London	0.11
*	NULL	NULL	NULL	NULL

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

ANS>> select * from CUSTOMER where RATING>100 and CITY!='Rome';

	CNM	CNAME	CITY	RATING	SNo
▶	202	Giovanne	Roe	200	1003
	203	Liu	San Jose	300	1002
	205	Clemens	London	300	1007
*	NULL	NULL	NULL	NULL	NULL