

ASSIGNMENT 12

Aim: Design at least 10 SQL queries for suitable database application using SQL DML statement: all types of Join,Sub Query

1.Create table Customers with schema(ID,name,age,address,salary)

```
mysql> create table Customers
-> ( id int primary key,
-> name varchar(233),
-> age int,
-> address varchar(233),
-> salary int);
Query OK, 0 rows affected (0.16 sec)

mysql> desc Customers;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int  | NO   | PRI | NULL    |       |
| name  | varchar(233) | YES |     | NULL    |       |
| age   | int  | YES  |     | NULL    |       |
| address | varchar(233) | YES |     | NULL    |       |
| salary | int  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

Customer table is created with attribute(id,name,age,address,salary)

2)Create table Orders with schema(O_id,o_date,customer_id,amount)

```
mysql> create table orders
-> ( O_ID int primary key,
-> o_date date,
-> customers_id int,
-> amount int,
-> foreign key(customers_id) references Customers(id));
Query OK, 0 rows affected (0.36 sec)

mysql> desc orders;
+-----+-----+-----+-----+-----+-----+
| Field      | Type  | Null  | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| O_ID       | int   | NO    | PRI  | NULL    |       |
| o_date     | date  | YES   |      | NULL    |       |
| customers_id | int   | YES   | MUL  | NULL    |       |
| amount     | int   | YES   |      | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

mysql>
```

Orders Table is created with attribute(O_ID,o_id,customers_id(foreign key),amount)

3)Insert 5 record to each table keeping few customers ids common to both the tables

```
mysql> insert into orders values(2,'2022-07-13',4,14000);
Query OK, 1 row affected (0.21 sec)

mysql> insert into orders values(3,'2022-07-15',6,19000);
Query OK, 1 row affected (0.03 sec)

mysql>
```

Data inserted in Orders Table

```
mysql> insert into customers(id,name,age,address,salary)values(1,'Anuj',22,'UK',30000),(2,'Shailesh',24,'Pune',40000),(4,'Akash',30,'Mumbai',12000),(6,'Rejui',67,'UP',12000);
Query OK, 4 rows affected (0.04 sec)
Records: 4 Duplicates: 0 Warnings: 0
```

Data inserted in Customers Table

4) Performs the inner joins on customers and order table to enlist the id,name,amount and o_date

```
mysql> select customers.id,customers.name,orders.amount,orders.o_date from orders INNER JOIN customers ON orders.customers_id=customers.id;
+-----+-----+-----+-----+
| id | name | amount | o_date |
+-----+-----+-----+-----+
| 1 | Anuj | 12000 | 2022-07-12 |
| 4 | Akash | 14000 | 2022-07-13 |
| 6 | Rejul | 19000 | 2022-07-15 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

inner join keyword selects records that have matching values in both tables

5) Perform the left outer join on customers and orders table to enlist the id,name,amount and o_date

```
mysql> select customers.id,customers.name,orders.amount,orders.o_date from orders Left JOIN customers ON orders.customers_id=customers.id;
+-----+-----+-----+-----+
| id | name | amount | o_date |
+-----+-----+-----+-----+
| 1 | Anuj | 12000 | 2022-07-12 |
| 4 | Akash | 14000 | 2022-07-13 |
| 6 | Rejul | 19000 | 2022-07-15 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Left join keywords returns all records from left table(customers), and the matching records from right table(Orders).The result is 0 records from right side if there is no match

6) Perform the right outer join on customers and orders table to enlist the id,name,amount and o_date

```
mysql> select customers.id,customers.name,orders.amount,orders.o_date from orders right JOIN customers ON orders.customers_id=customers.id;
```

id	name	amount	o_date
1	Anuj	12000	2022-07-12
2	Shailesh	NULL	NULL
4	Akash	14000	2022-07-13
6	Rejul	19000	2022-07-15

```
4 rows in set (0.00 sec)
```

Right Joins Keywords returns all records from right table(Orders) and the matching records from left table table(Customers).

7)Performs the full outer joins on customers and orders table to enlist the id,name,amount and o_date by using union all set Operation

```
mysql> select Customers.id,Customers.name,Orders.o_date,orders.amount from Customers left outer join Orders on Customers.id=Orders.customers_id union select Customers.id,Customers.name,Orders.o_date,Orders.amount from Customers right outer join Orders on Customers.id=Orders.customers_id;
```

id	name	o_date	amount
1	Anuj	2022-07-12	12000
2	Shailesh	NULL	NULL
4	Akash	2022-07-13	14000
6	Rejul	2022-07-15	19000

```
4 rows in set (0.00 sec)
```

Full outer join with the help of union because mysql do not support Full outer join

8)Performs the self joins on customers table to enlist the pair of customers belonging to same address

```
mysql> select A.name, B.name from customers A ,customers B where A.address=B.address;
```

name	name
Anuj	Anuj
Shailesh	Shailesh
Akash	Akash
Rejul	Rejul

```
4 rows in set (0.00 sec)
```

A self join is a regular join, but the table is joined with itself.

9)Perform cross / cartesian join on customers and order table to enlist the id,name,amount, and o_date

```
mysql> SELECT ID, NAME, AMOUNT, O_DATE
-> FROM CUSTOMERS, ORDERS;
```

ID	NAME	AMOUNT	O_DATE
1	Anuj	19000	2022-07-15
1	Anuj	14000	2022-07-13
1	Anuj	12000	2022-07-12
2	Shailesh	19000	2022-07-15
2	Shailesh	14000	2022-07-13
2	Shailesh	12000	2022-07-12
4	Akash	19000	2022-07-15
4	Akash	14000	2022-07-13
4	Akash	12000	2022-07-12
6	Rejul	19000	2022-07-15
6	Rejul	14000	2022-07-13
6	Rejul	12000	2022-07-12

```
12 rows in set (0.00 sec)
```

The CARTESIAN JOIN or CROSS JOIN returns the Cartesian product of the sets of records from two or more joined tables. Thus, it equates to an inner join where the join-condition always evaluates to either True or where the join-condition is absent from the statement

10)Design the sub query with select statement for displaying all the details of the customers having salary greater than 15000

```
mysql> select *from Customers where id in (select id from customers where salary>20000);
```

id	name	age	address	salary
1	Anuj	22	UK	30000
2	Shailesh	24	Pune	40000
6	Rejul	67	UP	120000

```
3 rows in set (0.00 sec)
```

their is all the record whose salary is greater than 20000

11)Create a backup table 'cust_bkp' of the table customers by using insert statement with subquery

```
mysql> create table cust_bkp as select *from Customers;
Query OK, 4 rows affected (0.22 sec)
Records: 4 Duplicates: 0 Warnings: 0

mysql> select *from cust_bkp;
+----+-----+-----+-----+-----+
| id | name   | age  | address | salary |
+----+-----+-----+-----+-----+
| 1  | Anuj   | 22   | UK      | 30000  |
| 2  | Shailesh | 24   | Pune    | 40000  |
| 4  | Akash  | 30   | Mumbai  | 12000  |
| 6  | Rejul  | 67   | UP      | 120000 |
+----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> delete from cust_bkp;
Query OK, 4 rows affected (0.21 sec)

mysql> insert into cust_bkp as select *from customers;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version 5.7.26 at line 1
mysql>
mysql> insert into cust_bkp(id,name,age,address,salary) select *from customers;
Query OK, 4 rows affected (0.03 sec)
Records: 4 Duplicates: 0 Warnings: 0

mysql> select *from cust_bkp;
+----+-----+-----+-----+-----+
| id | name   | age  | address | salary |
+----+-----+-----+-----+-----+
| 1  | Anuj   | 22   | UK      | 30000  |
| 2  | Shailesh | 24   | Pune    | 40000  |
| 4  | Akash  | 30   | Mumbai  | 12000  |
| 6  | Rejul  | 67   | UP      | 120000 |
+----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

12 Update a salaries by 10% of all the customers(in Customer table) having age greater than or equals 24 by using sub queries with update clause(by using backup table cust_bkp)

```
mysql> update Customers set salary=(salary/10)+salary where id in(select id from cust_bkp where age >=24);
Query OK, 3 rows affected (0.04 sec)
Rows matched: 3 Changed: 3 Warnings: 0

mysql> select *from customers;
+----+-----+-----+-----+-----+
| id | name   | age  | address | salary |
+----+-----+-----+-----+-----+
| 1  | Anuj   | 22   | UK      | 30000  |
| 2  | Shailesh | 24   | Pune    | 48400  |
| 4  | Akash  | 30   | Mumbai  | 14520  |
| 6  | Rejul  | 67   | UP      | 145200 |
+----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Update those customer salary whose age is greater than or equals 24

13)Delete all customers having age greater than 26 by using delete clause with sub query

```
mysql> delete from cust_bkp where id in(select id from customers where age>26);  
Query OK, 2 rows affected (0.04 sec)
```

```
mysql> select *from cust_bkp;
```

```
+-----+-----+-----+-----+-----+  
| id | name   | age | address | salary |  
+-----+-----+-----+-----+-----+  
| 1  | Anuj   | 22  | UK      | 30000  |  
| 2  | Shailesh | 24  | Pune    | 40000  |  
+-----+-----+-----+-----+-----+  
2 rows in set (0.00 sec)
```

delete record whose age is greater than 26