Unit-4

SQL:-

Nested queries:-

1. It involves a query that is placed within another query.
2. Output of the inner query is used by the outer query
3. A Nested query has 2 select statement one for the inner query and another for the outer query.
4. Inner query is executed first and returns a set of values that are then used by the outer query.
5. Outer query consists of operators like <, >, =, <>

Ex:-

```
Create table emp (id int, name varchar(20), sal int);
insert all into emp (id, name, sal)
values (1, 'ravi', 1000)
into emp (id, name, sal)
values (2, 'sethu', 4000)
into emp (id, name, sal)
values (3, 'ravi', 3000)
into emp(id, name, sal)
values (4, 'sony', 4000).
Select * from dual;
Select * from emp;
```

Select id, sal from emp where sal < (select max(sal) from emp);

Joins :-

Join is a SQL operation that combines data from two tables based on related column. It allows you to merge rows from different tables into a single result set.

Types :-

There are 4 types of joins

1. Inner Joins
2. Left join
3. Right Join.
4. Full Join.

1. Inner Join :-

It returns only the rows that have matching values in both tables.

2. Left Join :-

It returns all rows from the left table and matching rows from the right table. It gives null values when the values are not matched.

3. Right Join :-

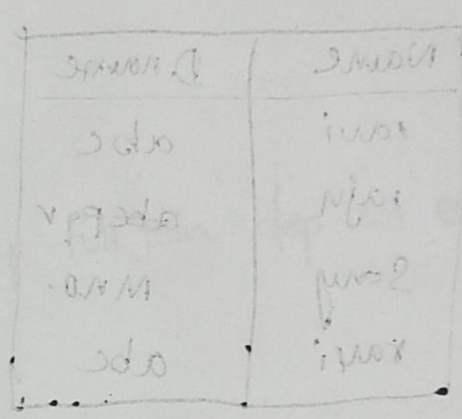
It returns rows from right table and matching rows from the left table. It gives null values when the values are not matched.

4. Full Join :-

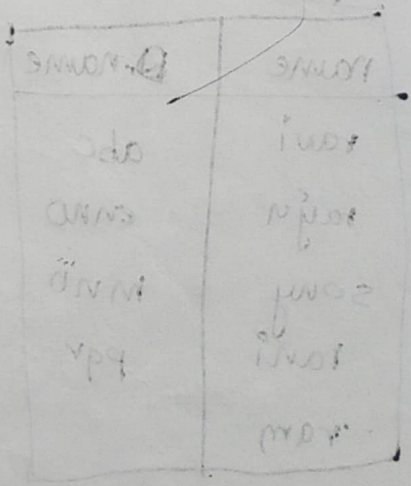
It returns rows from both tables including null values where there are no matches.

Ex:-

```
Create table emp (name varchar(5), deptid int);
insert all into emp (name, deptid)
 values ('ravi', 12)
 into emp (name, deptid)
 values ('sony', 13)
 into emp (name, deptid)
 values ('raju', 14)
 into emp (name, deptid)
 values ('rani', 12)
  Select * from dual;
```



```
Create table dep (dname varchar(5), deptid int);
 insert all into dep (dname, deptid)
 Values ('Abc', 12)
  into dep (dname, deptid)
   values ('mno', 13)
   into dep (dname, deptid)
    values ('pqr', 14)
  into dep (dname, deptid)
   value ('def', 15).
  Select * from dual;
```



Select * from emp;

Select * from dep;

Select emp.name, dep.dname from emp inner join dep on emp.deptid = dep.deptid;

output :-

Name	D.name
ravi	abc
raju	~~abc~~pqr
Sony	mno
ravi	abc

for example

name	D.name
ravi	abc
rani	abc
Sony	mno
raju	pqr

Insert into emp values ('ram', 16);

Select * from emp;

Select * from dep;

select emp.name, dep.dname from emp left join dep on emp.deptid = dep.deptid;

output :-

name	D.name
ravi	abc
raju	mno
sony	mno
rani	pqr
ram	

select emp.name, dep.dname from emp right join dep on emp.deptid = dep.deptid;

name	D.name
ravi	abc
sony	mno
raju	mno
rani	pqr
	def

select emp.name, depdname from emp full join dep on emp.deptid = dep.deptid;

name	Dname
ravi	abc
sony	mno
raju	mno
rani	pqr
ram	
	def

Sql functions :-

There are 4 types of functions.

1. Date functions
2. numeric functions
3. string functions
4. Conversion functions.

1. Date function :-

These function returns current date. the date function are sysdate() or current date()

2. Numeric function :-

* These are used to perform mathematical calculations on numeric data.

* some of them are

1. abs() → absolute value.

   Eg:- abs(-10) = 10

2. Sin() → Trignometric function

   Eg:- sin(30)

3. cos() → Trignometric function

   Eg:- cos(60)

4. mod() → modular value.

   Eg:- mod(10,3)

5. power() → exponential value

   Eg:- power(2,3)

3. String function :-
These functions are performed on characters. Some of them are -

1. concat() → combines two strings
   Eg:- concat ('Hi',' ','hello')
2. length() → Gives total no. of characters
   Eg:- length ('hello')
3. Lowercase() → converts upper to lowercase
   Eg:- Lower ('HELLO')
4. Uppercase() → converts lower to uppercase
   Eg:- upper ('hello')
5. reverse() → reverse the string.
   Eg:- reverse ('hi')

4. Conversion functions :-
These functions converts one datatype into another datatype. some of them are -

1. to-char()
2. to-numeric().

Creating tables with relationship :-

There are 3 types of relationships. They are

1. one to one relationship
2. one to many relationship
3. many to many relationship

These relationships are implemented in tables using keys (primary key and foreign key)

1. One to one relationship :-

Each row of table is exactly related to only one row

Ex :- create table ~~is exactly~~ student (id int primary key, name varchar (5));

insert into student values (1, 'ravi');
insert into student values (2, 'rani');

output :-

student

Id	Name
1	ravi
2	Rani

In the above example id is primary key. one student contains only one id number.

2. One to many relationship :-

Each row of the first table can be related to multiple rows in second table. The first table is

known as parent table and the second table is known as child table.. TO implement this relationship we have to use both the keys primary and foreign keys.

Ex:-

```
Create table customer(cus-id int primary key, name varchar(5));
insert into customer values(1,'sony');
insert into customer values(2,'ravi');
insert into customer values(3,'raju');
Select * from customer;
Create table orders(order-id int primary key, product_name
        varchar(6), cus-id int, foreign key(cus-id)
        references customer(cus-id));
insert into orders values(12,'oil',1);
insert into orders values(13,'sugar',2);
insert into orders values(14,'salt',1);
Select * from orders;
```

Output :-

customer

cus-id	name
1	sony
2	Ravi
3	Raju

orders

Order_id	Product name	cus-id
12	oil	1
13	sugar	2
14	Salt	1

In the above ~~table~~ example customer and orders are the two tables. In customer table cus-id is the primary key, in orders table order_id is the primary key and cus-id is the foreign key.

one customer can order many products, i.e one to many relationship.

3. Many to Many Relationship:-

In this Relationship many rows ~~of~~ the first table can be related to multiple rows in second table. while establishing this relationship we have to create three tables they are parent table, child table and junction table. parent table, child table contains only primary keys and junction table contains the foreign keys ~~of~~ parent and child tables.

Ex:-

```
Create table customer (cus-id int primary key, name
                       varchar(5));
insert into customer values (1, 'sony');
insert into customer values (2, 'ravi');
```

insert into customer values (3, 'rayu');
Selelt * from table;

creat table orders (order_id int primary key, product_name varchar(6));

insert into ~~values~~ orders values (12, 'oil');
insert into orders values (13, 'sugar');
insert into orders values (14, 'salt');
Select * from orders;

Create table customer orders (cus_id int, foreign key (cus-id) references customer (cus-id), order-id int, foreign key (order_id) references orders (order_id));

insert into customer ~~values~~ orders values (1,12);
insert into customer orders values (2,12);
insert into customer orders values (3,12);

Select * from customer orders;

Output :-

customer

cus-id	name
1	sony
2	Ravi
3	Rayu

orders

order_id	product_name
12	oil
13	Sugar
14	Salt.

Customer orders

Cus-id	order-id
1	12
2	12
3	12

In the above example customer is the parent table and orders is the child table, customer_order is the Junction table. many customers can order many products i.e. many to many relationship.

Implementation of key & integrity Constraints :-

There are 3 types of constraints in sql they are

1. Not null
2. unique
3. checks

1. Not NULL :-

The values in column can't be null.

2. unique : The values in column can't be repeat.

3. Checks :- The each value of a column must be unique based on ~~where~~ condition.

Ex :- create table student (id int not null, name varchar(5), email varchar(10) unique, age int check(age>0));

Views :-

View is a virtual table constructed from existing table with ~~requirement~~ required attributes → It is based on select statement. ~~We can perform~~

→ we can perform insert, update, delete and drop the view.

→ If we perform any operations on views then the original table data also changed

Ex :-

```
create table student (id int, name varchar(5), age int);
insert into student values (1, 'raju', 18);
insert into student values (2, 'sony', 19);
insert into student values (3, 'ravi' 20);
select * from student;
create view bca as select name, age from student;
select * from bca;
```

Relational set operations :-

These operations are performed on tables to manipulate the data.

There are 4 types of set operations they are.

1. Union
2. Union all
3. Intersect
4. minus

When we perform these operations we need two tables. These operations uses two select statements.

The number of cols must be same in both the tables while performing set operations.

Union :-

This operation returns the rows from both the tables. duplicate rows occurs only one time.

union all :-

It ~~result~~ returns all the rows from both the tables including duplicate rows.

Intersect :-

It returns the rows that are common in both the tables.

minus :-

it returns the rows that are present in 1st table but not in 2nd table.

Ex :-

```
Create table bca (id int, name varchar(5));
Insert into bca values (1,'raju');
Insert into bca values (2,'ravi');
Select * from bca;
Create table bsc (id int, name varchar(5));
insert into bsc values (12,'sony');
insert into bsc values (1,'raju');
Select * from bsc;
select * from bca union select * from bsc;
select * from bca union all select * from bsc;
select * from bca intersect select * from bsc;
```

select * from bca minus select * from bsc;

DCL Commands:-

DCL means Data control language. These commands manage Database security, access and user permesians. the permisions are DML Operations are performed. There are two types of DCL commands. They are.

1. Grant
2. Revoke.

1. Grant:-
   It assigns permissions to the users
2. Revoke:-
   It is used to take back permission from the user.

Example:-

① Login : Scott
P.W : Tiger
Connect

user name : sys as sysdba
[press Enter

create user raju identified by bca;
grant connect to raju;

minimize the current window and perform step ②

create table emp (id int,

→ user name : Raju
password : BCA

minimize current window

name varchar(5));

insert into emp value(1,'ravi');

grant select on emp to raju;

grant insert on emp to raju;

minimize current window and perform step 2

Revoke insert on emp from Raju;

drop user raju;

Select * from sys.Emp;

insert into sys Emp values (2,'sony');

minimize current window and go to 1st window

TCL commands :-

TCL means transaction control language. they are three types of TCL commands they are:

1. Commit
2. save point
3. Roll back.

1. Commit :-

It is used to permanently from the data in database.

Ex:- commit;

2. Save point :-

It is used to temporarly store the data at particular point.

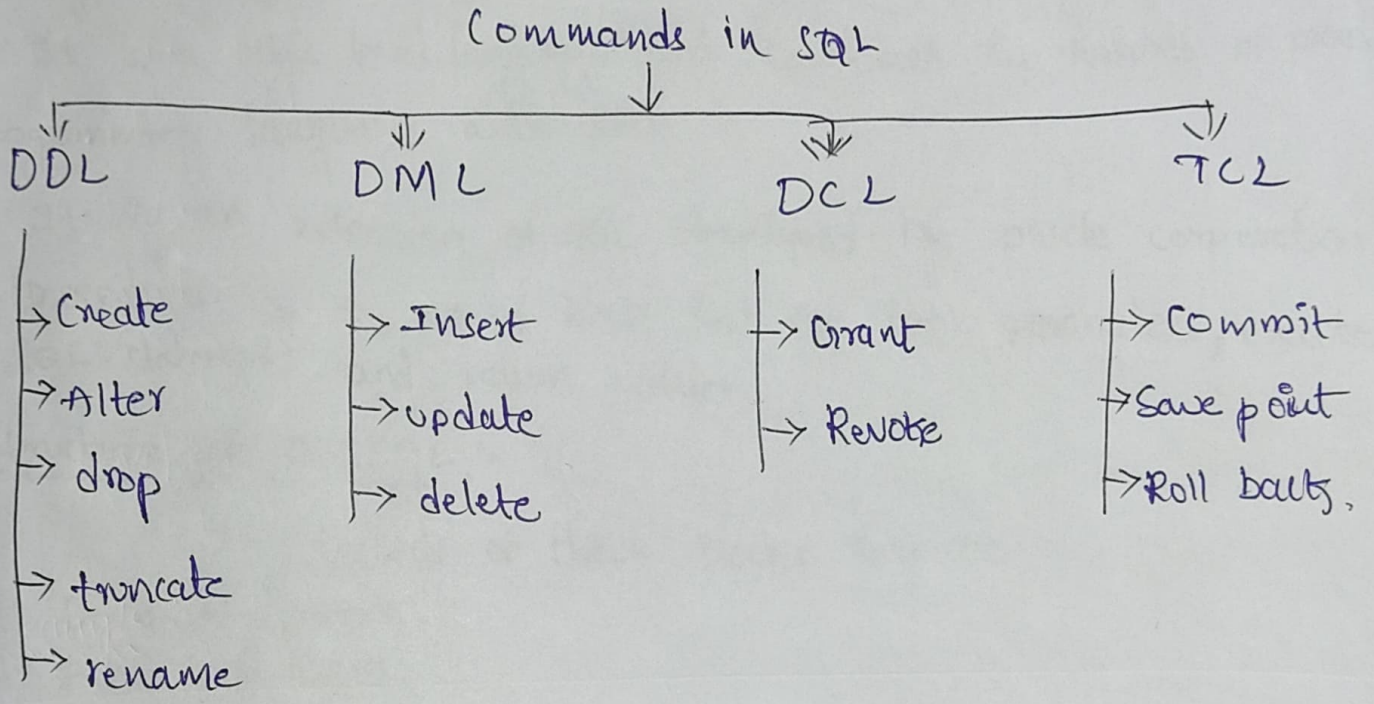
Ex:- Save point A;

Here 'A' is the name of save point.

3. Roll Back :-

It is used an undo. It reverses to the DML operations

Ex:- Roll Back;



Important Questions :-

1. Define joins? Explain the types of joins in SQL.
2. Explain the different types of relational set operations with an example?
3.