Joins

Joins -- Joins are used to combine rows from two or more columns, tables based on related column between them.

A relational database consists of multiple related tables linking together using common columns, which are known as *foreign key* columns.

We use join whenever we want to fetch data from two or more tables.

JOINS are used with SELECT statement. It is used to retrieve data from multiple tables. It is performed whenever you need to fetch records from two or more tables.

Joins help retrieving data from two or more database tables.

The tables are mutually related using primary and foreign keys.

Join statement is used to combine data or rows from two or more tables based on a common field between them. Different types of Joins are as follows:

- INNER JOIN(Simple Join)
- LEFT JOIN
- RIGHT JOIN

These Joins are explained below :--

(1.) Inner Join --> Inner join returns the record or selects the record which have matching values in both tables based on foreign key.

To get complete employeedata we need to query data from both employeedata and city tables data.

That’s why joins come into the play.

A join is a method of linking data between one (*self-join*) or more tables based on values of the common column between the tables.

Query for Inner join :-

```
select * from table1 inner join table2 on table1.id = table2.id;
```

The inner join clause compares each row from the first table with every row from the second table.

If values from both rows satisfy the join condition, the inner join clause creates a new row whose column contains all columns of the two rows from both tables and includes this new row in the result set. In other words, the inner join clause includes only matching rows from both tables.

If the join condition uses the equality operator (=) and the column names in both tables used for matching are the same, and you can use the `USING` clause instead :-

```
SELECT column_listFROM table_1INNER JOIN table_2 USING (column_name);
```

Example for inner join :--

```
create table employeedata(empId int primary key auto_increment,empName varchar(20),empSalary int);


desc employeedata;


insert into employeedata(empName,empSalary) values("Amit",33000);
insert into employeedata(empName,empSalary) values("Raj",5000);
insert into employeedata(empName,empSalary) values("Dinesh",25000);
insert into employeedata(empName,empSalary) values("Shiva",17000);
insert into employeedata(empName,empSalary) values("Sudhir",13000);
insert into employeedata(empName,empSalary) values("Kiran",30000);
insert into employeedata(empName,empSalary) values("Rakesh",28000);


select * from employeedata;


create table city(cityId int primary key auto_increment, cityName varchar(20));
desc city;
```

```
insert into city(cityName) value("Mumbai");
insert into city(cityName) value("Delhi");
insert into city(cityName) value("Pune");
insert into city(cityName) value("Banglore");
insert into city(cityName) value("Chennai");

select * from city;

select * from employeedata;
desc employeedata;

alter table employeedata add column cityId int;

alter table employeedata add foreign key (cityId) references city(cityId);

update employeedata set cityId=1 where empId=1;
update employeedata set cityId=5 where empId=2;
update employeedata set cityId=4 where empId=3;
update employeedata set cityId=4 where empId=4;
update employeedata set cityId=2 where empId=5;
update employeedata set cityId=3 where empId=6;
update employeedata set cityId=1 where empId=7;

select * from employeedata inner join city on employeedata.cityId = city.cityId order by empId;

select employeedata.empId,employeedata.empName,employeedata.empSalary,city.cityName from employeedata inner join city on employeedata.cit

select employeedata.empId,employeedata.empName,employeedata.empSalary,city.cityName,city.cityId from employeedata inner join city on empl
```