* [INNER JOIN:](http://www.tutorialspoint.com/sql/sql-inner-joins.htm" \t "_blank) returns rows when there is a match in both tables.
* [LEFT JOIN:](http://www.tutorialspoint.com/sql/sql-left-joins.htm) returns all rows from the left table, even if there are no matches in the right table.
* [RIGHT JOIN:](http://www.tutorialspoint.com/sql/sql-right-joins.htm) returns all rows from the right table, even if there are no matches in the left table.

**group by:**

The SQL **GROUP BY**clause is used in collaboration with the SELECT statement to arrange identical data into groups.

The GROUP BY clause follows the WHERE clause in a SELECT statement and precedes the ORDER BY clause.

Syntax:

The basic syntax of GROUP BY clause is given below. The GROUP BY clause must follow the conditions in the WHERE clause and must precede the ORDER BY clause if one is used.

SELECT column1, column2

FROM table\_name

WHERE [ conditions ]

GROUP BY column1, column2

ORDER BY column1, column2

Primarykey-------@@@@-----

A primary key is a field in a table which uniquely identifies each row/record in a database table. Primary keys must contain unique values. A primary key column cannot have NULL values.

A table can have only one primary key, which may consist of single or multiple fields. When multiple fields are used as a primary key, they are called a**composite key**.

If a table has a primary key defined on any field(s), then you can not have two records having the same value of that field(s).

**Note:** You would use these concepts while creating database tables.

Create Primary Key:

Here is the syntax to define ID attribute as a primary key in a CUSTOMERS table.

CREATE TABLE CUSTOMERS(

ID INT NOT NULL,

NAME VARCHAR (20) NOT NULL,

AGE INT NOT NULL,

ADDRESS CHAR (25) ,

SALARY DECIMAL (18, 2),

PRIMARY KEY (ID)

);

To create a PRIMARY KEY constraint on the "ID" column when CUSTOMERS table already exists, use the following SQL syntax:

ALTER TABLE CUSTOMER ADD PRIMARY KEY (ID);

**NOTE:** If you use the ALTER TABLE statement to add a primary key, the primary key column(s) must already have been declared to not contain NULL values (when the table was first created).

For defining a PRIMARY KEY constraint on multiple columns, use the following SQL syntax:

CREATE TABLE CUSTOMERS(

ID INT NOT NULL,

NAME VARCHAR (20) NOT NULL,

AGE INT NOT NULL,

ADDRESS CHAR (25) ,

SALARY DECIMAL (18, 2),

PRIMARY KEY (ID, NAME)

);

To create a PRIMARY KEY constraint on the "ID" and "NAMES" columns when CUSTOMERS table already exists, use the following SQL syntax:

ALTER TABLE CUSTOMERS

ADD CONSTRAINT PK\_CUSTID PRIMARY KEY (ID, NAME);

Delete Primary Key:

You can clear the primary key constraints from the table, Use Syntax:

Foreign key----@@@@@\_\_\_\_-

A foreign key is a key used to link two tables together. This is sometimes called a referencing key.

Foreign Key is a column or a combination of columns whose values match a Primary Key in a different table.

**The relationship between 2 tables matches the Primary Key in one of the tables with a Foreign Key in the second table.**

If a table has a primary key defined on any field(s), then you can not have two records having the same value of that field(s).

Example:

Consider the structure of the two tables as follows:

CUSTOMERS table:

CREATE TABLE CUSTOMERS(

ID INT NOT NULL,

NAME VARCHAR (20) NOT NULL,

AGE INT NOT NULL,

ADDRESS CHAR (25) ,

SALARY DECIMAL (18, 2),

PRIMARY KEY (ID)

);

ORDERS table:

CREATE TABLE ORDERS (

ID INT NOT NULL,

DATE DATETIME,

CUSTOMER\_ID INT references CUSTOMERS(ID),

AMOUNT double,

PRIMARY KEY (ID)

);

If ORDERS table has already been created, and the foreign key has not yet been set, use the syntax for specifying a foreign key by altering a table.

ALTER TABLE ORDERS

ADD FOREIGN KEY (Customer\_ID) REFERENCES CUSTOMERS (ID);

DROP a FOREIGN KEY Constraint:

To drop a FOREIGN KEY constraint, use the following SQL

https://ssl.gstatic.com/ui/v1/icons/mail/images/cleardot.gif