SQL

* What is SQL :-

- SQL stands for structured Query Language.
- SQL lets you access and manipulate databases.
- SQL became a stands of the (ANSI) American National Standards Institute in 1986, and of the International Organization For Standardization (ISO) in 1987.

* What can SQL do :-

- SQL can execute queries against a databases.
- SQL can reterive data from a databases.
- SQL can insert records in a databases.
- SQL can update records in a databases.
- SQL can delete records in a databases.
- SQL can create new databases.
- SQL can create new tables in a databases.
- SQL can create stored products in a databases.
- SQL can create views in a database.
- SQL can set permissions on tables, products and views.

* Using SQL in your web site:

- An ~~RDMS~~ RDBMS databasese program (ie. MS Access., SQL server, MySQL).
- To use a server side scripting language, like PHP / ASP
- To use SQL to get the data you went
- TO use HTML / CSS to style the page.

Page No.
Date :

* RDBMS :-

- RDBMS stands for Relational Database Management system.

- RDBMS is the basis for SQL, and for all modern databases system such as MS SQL server, IBM DB2, Oracle, MySQL, Microsoft Acess.

- The data in RDBMS is stored in database object called tables. A tables is a collection of related data entevies. and it consists of columns and rows.

Page No.

Date :

* Set up :-

Page No.
Date :

* SQL Commands :-

open MySQL shell => CLI => Command Line Interface.

- \SQL [switching to sql mode.]
- \connect root@localhost : 3309.
  [your pc username and port code.]
- admin@123. [password]
- y \n [your pc work (y) and other pc work (n)]
-

- Pc in all databases list
  show databases ;

- Create new database
  create database databasename ;

- use your select databases.
  use databasenames ;

- Pc in show all table list
  show tables ;

- Create new table and colums
  create table tablename (columnname datatype,
  -- ,
  -- ,
  -- ,
  --);

Page No.
Date :

- Insert data in table.
  insert into tablename values (.,.,.),
  (.,.,.),
  (.,.,.);

- show table with data.
  select * from tablename;

- show structur of table
  describe tablename;

- exit sql.
  \ quit.

Page No.
Date :

* SQL Data types :-

- Numeric
  bit , tinyint , smallint , int , bigint ,
  decimal , numeric , Float , real

- character / string
  char , varchar , Text

- Date / Time
  Date , Time , Datetime , Timestamp , year .

- Miscellaneous
  Json , XML

Page No.
Date :

* Query file using vscode and make a sql file.

=> open vscode and make a new folder.
=> Make a sql file (.sql extention) in vscode.
=> sql file in write a sql commande.
=> copy path this sql file and
=> open Mysql - command line client.
=> Enter password => use database => select table
=> and fire this file.

→ source G:\sql\demo.sql ;
file path

* Constraints :-

-o Not null :
Ensure that a column does not have a null value

-o Default :
Provides a default value for a column when none is specified.

-o Unique :
Ensures that all the value in a column are different.

-o primary :
Identifies each row / record in a database's table uniquely.

-o check
Ensures that all values in a column satisfy certain conditions.

-o Index
Creates and retrieves data from the database very quickly.

Page No.
Date :

* Retrive / select Query :-

Syntax :-

```
Select * From table_name;
```

- show select column of table

```
Select column1 , column2 , -- , columnN
From table_name;
```

- one time show full table and add show table column extra

```
Select * , colname From tablename ;
```

- Alias name :-

```
Select colname " alias_name" From table name;
Select colname as " alias_name" From tablename;
```

* Where clause :-

- The where clause is used to Filter records.
- It is used to extract only those records. that fulfill a specified condition.

Syntax

```
Select column1 , column2 , - - -
From tablename
where condition ;
```

[condition =, <, >, !=, <>, <=, >=]

Example :-

=> select name from product_master
where name = "keyboard" ;

=> select na costprise from product_master
where costprise < 1000 ;

=> select costprise from product_master
where costprise > 1000 ;

=> select costprise from product_master
where costprise < 500 and/or costprise > 1000

=> select costprise from product_master
where name != | <> "keyboard" ;

* Update statement :-

=> The update satement is used to modify t
existing crecords. in a tulele.

Syntax :-

Update <tulele_name>
set column1 = Value1, column2 = valu
where condition ;

Note :- Be careful when updating orecords
a tulue! notice the where culuse in
update satement. The where culuss
specifies which orecords. that shouu

to be updated. If you omit the where clause all record in the table will be updated!

Example :-

```
update client_master
set state = "Gujarat"
where clientno = "c00005";
```

* Delete Statement :-

=> The delete statement is used to delete existing records in a table.

Syntax :-

-o Delete from <table_name> ; // all record delete.

-o Delete from <table_name>
where condition ; // selected row deleted.

Example :-

```
delete from client_master
where clientno = "c00006";
```

Note :- This statement is delete only record (row). and as it is column (table-structure).

* Drop Statement :-

=> The drop table statement is used to delete or drop an existing table in a database.

Syntax :-

Drop table <table_name> ;
// all record and data structure delete

Example :-

Drop table client_master ;

Note :- Be careful before dropping a table. deleteing a table will result in loss of complete information stored in the table !

Page No.
Date :

* Primary key :-

A primary key is used to ensure data in the specific column is unique. It is a column cannot have null values. It is either an existing table column or a column that is specifically generated by the database according to a defined sequence.

- A primary key is used to ensure data in the specific column is unique.
- It uniquely identifies a record in the relational database tables.
- only one primary key is allowed in a tables.
- It is a combination of unique and not null constraints.
- It does not allow not null values.
- Its values cannot be deleted from the parent.
- It constraint can be implicitly defined on the temporary tables.
- a table can allow composite primary key.

Page No.
Date :

Syntax :-

1). MySQL, SQL server, oracle, MS Access :-

```
Create table <table_name> (
    colname    datatype primary key,
    ---,  ---,
    );
```

2). MySQL

```
Create table <table_name> (
    colname  datatype, --, --, --,
    primary key (colname)
    );
```

3) MySQL, SQL server, oracle, MS Access :-

```
Create table <table_name> (
    colname datatype, --, --, --,
    constraint  PK_tablename
    primary key (colname).
    );
```

Ex :-

```
Creat table client_master (
    client no  varchar (50), name varchar (50),
    city varchar (50), pincod numeric (6),
    state varchar (50), baldue numeric (8),
    constraint  PK_client_master (.
    primary key (client no)
    );
```

Page No.
Date :

* composite primary key :-

=> A composite key in SQL can be defined as a combination of multiple columns, and these columns are used to identify all the rows that are involved uniquely. Even though a single column can't identify any row uniquely, a combination of over one column can uniquely identify any record.

Syntax :-

```
create table mytable_name> (
    colname datatype, --, --, --, --,
constraint PK-tablename.
primary key ( col1, col2, col3 )
);
```

Example :

```
create table mybag (
    colour varchar(30),
    compartment int,
    capacity varchar(30),
    constraint PK_mybag
    primary key (colour, compartment, capacity)
);
```

// Bag colour = 3 = Red, Blue, Cheeem = RGB.
// compartment = 3 = 4, 5, 6.
// capacity = 3 = 25L, 30L, 35L.

insert into mybag value ("R", 4, "30L"); // R430L
insert into mybag value ("G", 4, "30L"); // G430L

* Primary key on Alter table :-

To create a primary key constraint on the column when the table is already created.

Syntax :

- Alter table persons
  Add primary key (colname) ;

- Alter table <table_name>.
  Add constraint PK_tablename
  primary key (colname, colname).

* Drop a primary key constraint

To drop a primary key constraint.

Syntax :

Alter table <table_name>
drop primary key ;

Alter table <table_name>.
drop constraint PK_tablename.

Page No.
Date :

* Foreign key.

- A foreign key is a column or group of column in a relational database table that provides a link between data in two tables.
- It refers to the field in a table which is the primary key of another tables.
- Where as more than one foreign key are allowed in a tables.
- It can contain duplicate values and a table in a relational database.
- It can also contain null values.
- It's value can be deleted from the child table.

Syntax :-

1)

```
create table <table_name> (
    colname datatype primary key,
    --, --, --, --, --,
    <Foreign colname> datatype references <ptabna
);
```

2)

```
create table <table_name> (
    colname datatype primary key,
    --, --, --, --, --,
    Foreign key (colname)
    References ptabname (primary for parent tabe
);
```

3)

```
create table <table_name> (
    colname datatype primary key,
    --, --, --, --, --, --, --,
    constraint FK_table_name
    foreign key (colname)
    References ptabname (primary for parant t
);
```

// p tab = parent table

Page No.
Date :

* SQL Foreign key on alter table

To create a foreign key constraint on the column when the table is already created.

Syntax :-

-o Alter table <table_name>
Add foreign key (colname)
Reference ptabname (pcolname);

-o Alter table <table_name>
Add constraint FK_tablename
Foreign key (colname),
reference ptablename (pcolname);

* Drop a foreign key constraint.

To drop a foreign key constraint ....

Syntax :-

-o Alter table <tablename>
Drop foreign key (colname);

-o Alter table <table_name>
Drop foreign key <constraint_name>;

-o Alter table <table_name>
Drop constraint <constraint_name>;

Page No.
Date :

* Order by keyword :-

=0 The order by keyword is used to sort the result-set in ascending or descending order.

-0 The order by keyword sorts the records in ascending order by default. To sort the records in descending order, use key the DESC keyword.

Syntax :-

-0 select * from <table_name>
order by columnname ; // acending order

-0 select * from <table_name>
order by colname desc ; // descending order

-0 select * from <table_name>
order by colname , colname ;

* In operator & Not In operator :-

-0 The in operator always you to specify multiple values in a where clause.

-0 The in operator is a shorthand for multiple or conditions.

Page No.
Date :

Syntax :-

-o select * from <table_name>.
where colname In (value1, value2 --);

-o select * from <table_name>.
where colname In (select colname
from colname);

Not In :-

select * from <table_name>
where colname Not In (value1, value2-);

* Between operator :-

The Between operator selects values within a given range. The values can be numbers, text or dates.

Syntax :-

-o select * from <table_name>.
where colname Between value1 and value2;

-o select * from <table_name>.
where colname Not between
value1 and value2 ;

Page No.
Date :

* like operator :-

- The like operator is used in a where clause to search for a specified pattern in a column.
  - The percent sign (%) represents zero, one or multiple characters.
  - The underscore sign (_) represents one single character.

Note :-

Ms Acess uses an asterisk (*) instead of the percent sign (%), and a question mark (?) instead of the underscore. (_).

Syntax :-

```
Select * from <tablename>
where colname like pattern;
```

Tip :-

You can also combine any number of condition using And or OR operator.

Page No.
Date :

*. wildcard Patterns :-

Pattern	Description
"a%"	start with "a"
"%a"	end with "a"
"_a%"	second character "a"
"%a_"	second last character "a"
"%a%"	word in character " any position.
"_a__"	length specific word
"a%o"	start with a end with o.

Page No.
Date :

* Distinct Statement :-

-o The select Distinct statement is used to return only distinct (different) value.

-o Inside a table, a column often contains many duplicate values; and sometimes you only want to list the different (distinct) values.

Syntax :-

```
select Distinct colname, --, --
Fowm <table_name>;
```

* Limit, top, fetch.

=> Syntax :-

Limit

```
select * fowm <tab-name> limit number;
```

Top

```
select top number * <Fowm tab_name>;
```

Fetch

```
select * Fowm <tab_name>
Fetch First number Row only;
```

Page No.
Date :

* String function :-

- Ascii() - Return the ascii value of first character
  select ascii ("Name"). from table_name.

- char_length(). - Return the length of string.
  select char_length ("Name") ;

- character_length() - Return length of the string.
  select character_length ("Name") ;

- concat() - add several string together
  select concat ("--","--","--","--") ;

- Left() - starting from left extract number
  select left ("colname", Number) ;

- Length() - Return the length of the string
  select length ("colname") ;

- Locate() - search and return position
  Select locate ("U", "[illegible]") ;

- Lower() - Convert text lower-case :
  selecte lower ("colname") ;

- Lpad() - left pad the string "--", to a total length
  select lpad ("Name", 10, "very") ;

- ltrim() - Remove leading space from a string
  select ltrim ("---name") ;

Page No.
Date :

-o Mid () - Extract a substring from a string.
select mid ("---", 5, 3);
// start 5, extract 3 character)

-o Repeat () - Repeat a string number time.
select repeat ("name", number);

-o Replace () - Replace name.
select Replace ("Name", "Name", "new");

-o Reverse () - Reverse string
select Reverse ("Name");

-o Right () - Extract number character right side
select Right ("Name", Number);

-o Rpad () - Right-pad the string with "--", total number
select Rpad ("Name", Number, "add character");

-o Rtrim () - Remove trailing spaces from a string
select Rtrim ("Name---");

-o Strcmp () - compare two string
select Strcmp ("string1", "string2");

-o Substr () - start at position extract character
select substr ("Name", Number, extract number);

-o substring () - start at position extract character
select substring ("string", start pos, extract number);

Page No.
Date :

- Trim() - Remove leading and trailing space
  select trim(" -- Name -- ");

- Ucase() - convert the text upper case :-
  select ucase(" string ");

- Upper() - convert the text upper case :-
  select upper(" string ");

Page No.
Date :

* Numeric Function :-

-o ABS() - Return the absolute value of a number.
select ABS(-243.5); // 243.5

-o Asin() - Return the arc sine of a number :
select Asin(0.25);

-o Acosin() - Return the arc cosine of a number.
select Acosin(value);

-o Atan() - Return the arc tan of a number.
select Atan(value);

-o Avg() - Return the average value of column.
select avg(colname) from table_name.

-o ceil() - Return the smallest int value that is geter than or equal to value.
select ceil(25.60); // 26.

-o Floor - Return the bigest int value that is less than or equale value.
select Floor(25.60); // 25

-o count - return number of row in table.
select count(colname) from table_name;

-o Degrees - convert to radian value into degrees.
select Degrees(1.5); // 85.9436692

Page No.
Date :

-o Div() - Integer division
select 10 div 5; // 2

-o Log(), Log10, Log2 :- Return logarithm.
select log(2);
select log10(2);
select log2(2);

-o Max() - Find the max value in column of table.
select max(colname) from table_name;

-o Min() - Find the min value of column in table.
select min(colname) from table_name;

-o Mod() - Return the remainder (int).
select Mod(18, 4) // 2

-o PI() - Return the value of PI (π).
select PI();

-o Pow() - Return first power, second power ----
select pow(base number, exponent number).
// same power().

-o Radians() - convert a degree value into radians:
select radians(180); // 3.1415.

-o Rand() - Return random decimal number.
select rand();

-o Round() - Round the number
select Round(value, position AFter point);

-o sign() - Return the sign of number
if number > 0 => 1
if number < 0 => -1
if number = 0 => 0.
select sign (value);

-o SQrt()- Return the square root of a number.
select Sqrt (value);

-o sum() - return sum of column in the table.
select sum (colname) From table_name

-o Truncate() - return a number truncated to 2 decimal places:
select truncate (135.375, 2); // 135.37

Page No.
Date :

check constraint :-

-o The check constraint is used to limit the value range that can be placed in a column.

-o If you difine a check constraint on a column it will allow only certain values for this column.

-o If you define a check constraint on a table it can limit the value in certain column based on values in other column in the row.

Syntax :-

1) create table <table-name> (
column datatype, --, --,
check (column condition)
);

2) create table <table-name> (
column datatype check (column condition),

3) create table <table-name> (
column datatype, --, --,
constraint chk-table-name
check (column condition)
);

* Alter table statements :-

-o The Alter table statement is used to add, delet or modify columns, in an existing table.

-o The Alter table statemant is also used to add and drop various constraints on an existing table.

-o Syntax :-

-o Alter table table_name
Add column column_name datatype ;
// add column

-o Alter table table_name.
Drop column column_name ;
// drop column.

Notice : that a some database system don't allow deleting a column.

-o Alter table table_name
Modify column column_name datatype ;
// Modify datatype.

Page No.
Date :

* One yeur in huw many .... :-

=0 1 yeur 365.3 , 365.25 , 365 day.
-0 1 yeur 12 month.
-0 1 yeur 52 week.
-0 1 yeur 2 samasteur.
-0 1 yeur 4 quaurteur.
(1. Jun - March
2. April - June
3. July - sep.
4. octombeur - Dec).
-0 1 yeur 8766 Hours.
-0 1 yeur 525960 Minute.
-0. 1 yeur. 31,557,600 second.

* one day in huw many ..... :-
-0 1 day 24 hours.
-0 1 day 1440 Minute.
-0 1 day 86,400 second.

* Time.
-0 1 huur 60 Minute.
1 Minute 60 second.
1 second 1000 Millisecond.

* Leap yeur Fourmulu.

Any yeur that is enveniy divisible by 4.
is a leap yeur : 1988, 1992, 1996, ...

Page No.
Date :

* ① Date functions :-

-o ADDDate() - add unit value, a date and return da

adddate (date, Interval value add unit)
adddate (date, days).

-o Addtime() - adds a time interval to a time / date
and then return the time / datetime.

addtime (date time, addtime)
Required. The time interval to add to datime.
Both positive and negative values are allowed

-o Curdate() - return the current date.
Select curdate();
Select curdate() + 1; // next day date.
(string) ("yyyy-MM-DD") or (numeric) (yyyyMMDD).

-o Current_date() - return the current date.
Select current_date();
Select current_date() + 1; // add one day

-o Current_time() - return current time();
Select current_time();
Select current_time() + 1; // add 1 second.

(string - "HH-MM-ss") (numeric - HHMMSS)

Page No.
Date :

-o. Current_timestamp() - return the current date and time.

```
select current_timestamp();
select current_timestamp() + 1; // add 1 second
```

-o curtime() - return current time()

```
select curtime();
select curtime() + 1;
```

-o Date() - Extract the date part.

```
select date("date");
```

-o Datediff() - return the number of days between two date values.

```
select datediff(date1, date2);
```

-o Date_add() - same works add date function).

```
select date_add(date, Interval value add unit)
```

-o date_formate() - Formate a date:

```
select date_format(date, "Format");
```

%Y	year	4 digit value.
%y	year	2 digit value.
%M	Month	Month full name.
%m	Month	numeric month
%D	Day	1st, 2nd, 3rd ---.
%d	Day	01 to 31.
%H	Hour	(00 to 23).
%h	Hour	(00 to 12).
%i	Minute	(00 to 59).
%S	second	(00 to second).
%p	AM or PM	-
%r	AM or PM	- hh:mm:ss (AM/PM).

- Date_sub() - subtract days from a date and return

```
select date_sub(date, Interval value unit);
```

- Day() - return the day of the month for date

```
select day(date);
```

- Dayname() - return the weekday name for a day

```
select dayname(date);
```

- Dayofmonth() - return the day of the month for a

```
select dayofmonth(date);
```

- Dayofweek() - return the weekday index number

```
select dayofweek(date);
```

- Dayofyear() - return the day of the year

```
select dayofyear(date);
```

- Extract() - extract the unit from date

```
select extract(unit from "date");
```

- From_days() - numeric day to convert date.

```
select from_days(number);
```

- Localtime() - return current date and time.

```
select localtime();
```

- Localtimestamp() - return current date and time.

```
select localtimestamp();
select localtimestamp() + 1;
```

Page No.
Date :

-o Makedate() - create and return a date based on a year and a number of days value
select makedate(year, day);

-o Maketime() - create and return a time value based on an hour, minute and second value.
select maketime(hour, minute, second);

-o Monthname() - return the name of the month
select monthname("date");
select monthname(curdate());

-o Now() - return current date and time:
select now();
select now()+1;

-o Period_add() - add specified number of months
select period_add(period, number);
(period format - yymm or YYMM)

-o period_diff() - return the diff. between two pe-
select period_diff(period1, period2);

-o Quarter() - return the quarter of the year.
select quarter(date);
select quarter(curdate());

-o Hour, minute, second - return hour, minute, second
select hour/minute/second(datetime);

- sec_to_time() - second convert to time.
  select sec_to_time(seconds);

- str_to_date() - string return a date.
  select str_to_date(string, format);

- subdate() - substract unit from a date.
  select subdate(date, Interval value unit);

- subtime() - substract time.
  select subtime(date time, interval time);

- sysdate() - return system date.
  select sysdate();
  select sysdate()+1;

- Time_format() - formate a time.
  select time_formate(time, formate)
  %H - Hours, %i - Minute, %s - second

- Time_to_sec() - convert a time value into second
  select time_to_sec(time);

- Timediff() - return the diff. between two time
  select timediff(time1, time2);

- Timestamp() - return a datetime value based on the arguments
  select timestamp(expression, time)

Page No.
Date :

-o To_days() - returns the number of days between the date and year 0.
select to_days(date);
This function opposite of the from_days().

-o week() - returns the week number of a date
select week(date);

-o weekday() - returns the weekday number for a dat
select weekday(date);

-o weekofyear() - returns the week number for
select weekofyear("date");

-o year() - returns the year part of a date.
select year(date);

-o yearweek() - returns the year and week numbe
select yearweek("date");
select yearweek(curdate());

* Join :-

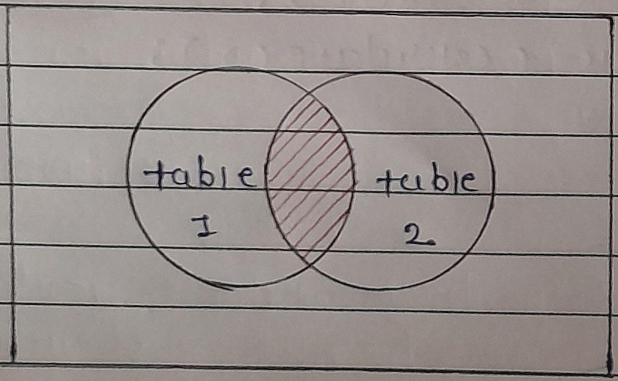
My SQL Joins are used with select it is used to retrive data from multiple table it is performmed whenever you need to fetch records from two or more tables.

There are three types of mysql Joins

1) Inner Join for sometimes called simple Join
2) left Join for sometime called left Join
3) Right Join for sometime called right Join

1) Inner Join :-

The mysql inner Join is used to return all rows multiple to table where the Join condition is at is the must common type of Join.



Inner Join

Syntax :

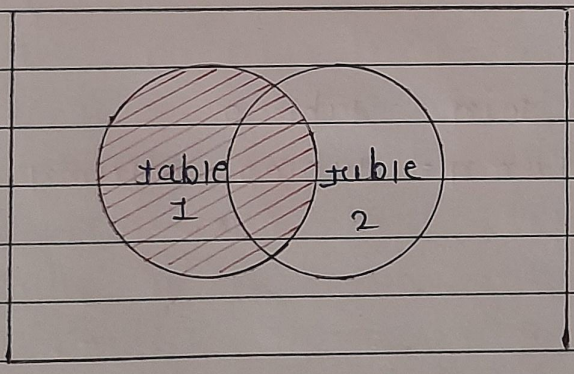
```
Select columns
From table 1
Inner Join table 2
On table 1. column = table 2. column;
//two tables
```

Page No.
Date :

```
Select a.colname, b.colname.
Forom Table1 a, Table2 b
where a.colname = b.colname ;
// condition use and.
```

* Left (outer) Join :-

The left outer Join returns all rows from the left hand table specified in the on column and only those rows from the table where the join condition is fullfilled.



Left Join

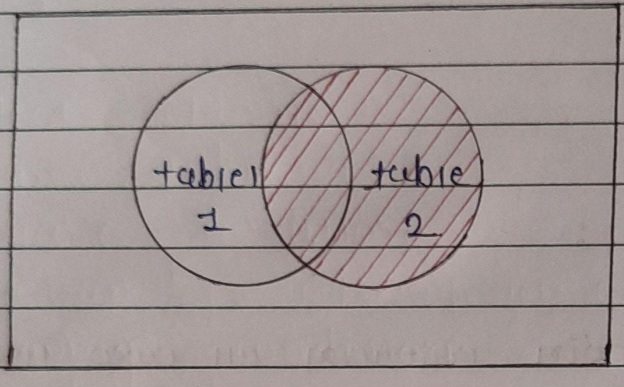
Syntax :

```
Select columns
From table 1
left Join table 2
on table1. column = table2. column ;
```

Page No.
Date :

* Right (outer) Join :-

The right (outer) join returns all rows from right hand table specified in the on condition and only those rows from the another table where the join condition is fullfilled...



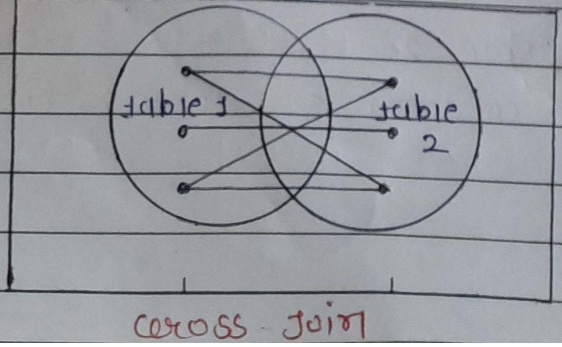
Right Join.

Syntax :-

```
Select columns
from table 1
right [outer] join table 2
on table 1.column = table2.column ;
```

* Cross Join :-

when might need that types of join envision that you have to find all conditions of a products and a color in that case a cross join would be highly advantages.



Cross Join

Syntax :

```
Select column
From table 1
cross Join table 2 ;
```

* Group by :-

The group by statement groups rows that have the same value into summary rows, like "find the number of customers in each country."

The group by statement is often used with aggregate function (count(), Max(), Min(), SUM(), Avg()) to group the result-set by one or more columns.

Syntax :-

-o
```
select columns
From tablename
where condition
group by column name
order by column name;
```

-o
```
select columns
From table 1
(Join type) table 2
on table1. colname = table2. colname
group by colname ;
    // group by with Join
```

* Having clause :-

The having calause was added to SQL because the where keyword cannot be used with aggregate functions.

Syntax :-

```
select columns
fowm tubie_name
where condition
Gwoup by column_name
Having condition
order by column_name(s);
```

* Views :-

A views contains wows and column, just like a weal tables. The Fields in a view are Fields fwom one ow mowe weal tables in the databases. you can add SQL statements and function to a view and pwesent the data as if the data were coming fwom one single tubie.

Syntaoc :-

```
Cweate view view_name As
select columns
fwom tubie_name
where condition ;        // cweate view.

select * fwom view_name ;    // Fiwe Qwewy.
```

-0 updating a view.

A view can be updated with the create or Replace view statement.

syntax :-

Create or Replace view view-name as
select colums
From tablename
where condition;