**My sql**

**Database:-** it is an application which stores and collects the data

>each database has one or more distinct api for creation,copy,manage,searching and relocating the database.  
**Fill:-** unchanged or cannot update,cant organize the data in files.

**Api:-application programming interface**

**Excel:-**less secure or for security reasons we are not going to use excel

**DBMS:-database management system**

**>**stores the data in the form of tables

**Challenges of dbms**

Relations is not possible for the accessing the data

**RDBMS:-relation database management system**

>It is in the form of tables and we can also map them to the one location to another location

>it will retrieve the data very fastly

>operations will be very effective

Databases are 2 types

1. RDBMS
2. NON-RDBMS :-store the data in the form of key values (j-son format)

MY SQL DATABASE:-my sql along with sql

**SQL:-STRUCURED QUIERY LANGUAGE**

It refers to server platform in sql language

---> PREREQUISITES NEED TO CREATE A PROJECT

WE NEED

1. front-end:-view the data
2. Back-end:-interaction between data and program
3. Database:-storage--provides space to the applications

Database componenets

1. Client
2. Server

My sql uses 2 types of commands

1. DDL COMMANDS:-DATA DEFINITION LANGUAGE
2. DML COMMANDS :-DATA MANUPULATION LANGUAGE
3. DDL COMMANDS:-

Whatever we written

1. create :- to create
2. Alter :-in order to add rows and columns
3. Drop:-delete the record from db
4. Truncate:-remove the records from the table
5. Rename:-to change the table or records in the database
6. DML COMMANDS:-
7. insert:-insert the data into the table
8. Update:-update existing data into the table
9. Delete:-delete the records from the db
10. Call:-python or java programmer
11. Explain call

Data types:-

CHAR-->a fixed length of string characters are allowed

VARCHAR-->a variable length

BINARY(SIZE)-->equal to char but stores the binary strings,default it will be 1

TEXT (SIZE)---> holds a string with a maximum length 65,535 bytes

TINY TEXT --> holds the string with maximum 255

drop table employee

create table employee(emp\_name varchar (30),emp\_id int (10),emp\_role varchar (30),emp\_number int (11),emp\_age int (10));

select \* from employee

insert into employee values ('jeevan',437,'trainee',5966,26);

insert into employee values ('mani',439,'trainee',5967,25),('gopi',449,'trainee',9000,25),('hari',444,'trainee',9666,25),('ganesh',450,'trainee',9500,23),('yesh',420,'trainee',9300,24),('ashok',421,'trainee',1234,24);

insert into employee values ('mouli',444,'trainee',3333,21),('lokesh',234,'trainee',2222,19),('vardhan',222,'trainee',1222,19);

select \* from employee

use jeevan

create table product (prd\_name varchar(30),prd\_brand varchar(20),prd\_model varchar(20),prd\_storage varchar(20),prd\_price int);

insert into product values ('mobile','apple','iphone16plus','128gb',94000);

select \* from product

create table item (item\_id int,item\_name varchar(30),item\_category varchar(30),stock int,price int);

insert into item values (201,'shoe','foot wear',15,2000);

insert into item values (202,'chappal','foot wear',12,500),(203,'jeans','cloth',55,1500),(204,'t shirt','cloth',33,750),(206,'shirt','cloth',32,1000),(207,'huddie','cloth',12,1500),(208,'cap','style',50,100),(209,'spects','style',100,150),(210,'mask','style',500,10);

create table stock (ord\_no varchar(30),ord\_quantity int,ord\_category varchar(30),ord\_price int,ord\_discount varchar(30));

select \* from stock

insert into stock values (02,300,'cosoms','75000','10%'),(03,290,'shoe','88000','20%'),(04,300,'cloth','95000','10%'),(05,1000,'mask','100000','50%'),(06,220,'spects','95000','10%'),(07,800,'watch','86000','10%'),(08,100,'rings','6000','10%'),(09,220,'cloth','95000','10%'),(01,100,'chappal','10000');

insert into stock values

(2, 300, 'cosoms', 75000, '10%'),

(3, 290, 'shoe', 88000, '20%'),

(4, 300, 'cloth', 95000, '10%'),

(5, 1000, 'mask', 100000, '50%'),

(6, 220, 'spects', 95000, '10%'),

(7, 800, 'watch', 86000, '10%'),

(8, 100, 'rings', 6000, '10%'),

(9, 220, 'cloth', 95000, '10%'),

(1, 100, 'chappal', 10000, '10%');

create table surf\_stock (surf\_name varchar(30),surf\_brand varchar(20),surf\_quantity varchar(20),surf\_price int,prd\_price int);

select \* from surf\_stock

insert into surf\_stock values

('rin','art','100kg',199,19000),

('ariel','itc','100kg',400,400000),

('aroma','ikc','100kg',100,10000),

('nirma','ikt','100kg',199,19000),

('ramraj','itc','100kg',189,12000),

('kalahari','idc','100kg',179,15000),

('burma','imc','100kg',159,15000),

('jaci','kkc','100kg',149,14000);

show tables

**CLAUSES AND OPERATORS:-**

1. **Where clause:-** mainly used for filtering purpose

Syntax:-select clo\_name from table name where condition

Ex:-**select employee\_name from employee where emp\_id=’101’;**

1. **And or not**

**And :-** satisfy the condition,and used to filter

Condition condition2 result

True true true

True false false

False true false

False false false

If we wanted to display a record if all the conditions are satisfied by and are true

**Or :-**

# If any one condition satisfied then the result is true

**NOT:- DISPLAY THE RECORDS WHEN CONDITION FAILS**

**AND:-**

**SELECT COLUMN NAME FROM TABLE NAME WHERE COND1 AND COND2 AND COND3**

**OR:-**

**SELECT COLUMN NAME FROM TABLE NAME WHERE COND1 OR COND2 OR COND3**

**NOT:-**

**SELECT COLU1,COLU2 FROM TABLE NAME WHERE NOT CONDITION**

**ORDER BY:-SORT BY:-**sorting the records in a table ---> ascending or descending default value of orders in ascending

**SYNTAX:**

**SELECT COLUMN1,COLUMN2 FROM TABLE NAME ORDERBY COLUMN1,COLUMN2------------ASC (OR)DEC;**

**INSERT INTO:-** Whicj is used to insert into new records already existing table

**SYNATX:-**

**INSERT INTO TABLE NAME (COLUMN1,COLUMN2,COLUMN3,-----------------) VALUES (N1,N2,N3 -----);**

**SELECT CLAUSE:-**To display or obtain the data from particular table

**Select \* from tablename;==> display the whole table**

**Select column1 from tablename ==> display the record of column1 only**

**UPDATE:-** Modify or changing the existing values

**SYN:- UPDATE TABLENAME SET “COL1=VALUE2”,”COL2=VALUE2,--------- WHERE CONDITION;**

**DELETE:- DELETS THE EXISTING RECORDS**

**SYNTAX:-**

**Delete from table name where condition**

**LIMIT:-**Used to specify the number of records to return

**SYNTAX:-**

**Select column name from table name where condition limit number;**

**MIN AND MAX FUNCTIONS :-**

**Min():-** it return the minimum value of the selected column of a table

**Mix():**it will return the maximum value of selected column of a table

**SYNTAX:-select min/max (column name) from tablename where condition;**

**LIKE:-** Operator :- mainly used in where clause

Used to search for specific pattern

**Syntax:- select col1,col2 from tablename where col1 like pattern**

**%a--- finds names ending with ‘a’;**

**a%--- finds names starts with ‘a’;**

**\_a%---- find names whose second letter is ‘a’;**

**a\_%--- find the values atleast 2 letters**

**IN:-** allows to specify multiple values where clause

**Syntax:- select column name from tablename column name in (val1,val2,-----);**

**Select \* from where state in ‘ap’,’telengana’,’bihar’,’delhi’;**

**BETWEEN:-**Selects the middel value from a range of values

**Syntax:- select columnname from table name where column between val1 and val2;**

**AVG:-** it returs the average values of a particular column/int;

**Syntax:- select avg (column) from table name where condition;**

**SUM();** Give the total numbers in the column

**Syntax:- select sum(column) from tablename where condition;**

**COUNT:-**returns the number of records which satisfy our condition.

**Syntax:- select count of col1 from table where condition;**

**GROUPBY:-** groups the data present in the tables with same values

**Syntax:- select column name from table name where condition groupby column name orderby colname;**

**JOINS:-** there are 3 types of joins

1. Inner join(simple join)
2. Outer join (left outer join , right outer join )
3. Right join (similar way of right outer join)
4. **inner join :-** in order to return all the rows from multiple tables where the join condition is satisfied.

==> it is the most commanly used join

**Syntax:- select column from table1 innerjoin table2 on table1\_column,table2\_column2;**

1. **outer join:-** it will returns all rows from left hand-side table and all the rows in the right hand side table by satisfying the join condition

Left hand side table combined with right hand side

**Syntax:- select column from table1 left outer join table2 on table1 column=table2 column**

1. **Right outer join:-** it will return all rows from the right hand table rows on the table by satisfying the join condition.

**Syntax:- select column from table1 right outer join table2 on table1 column=table2 column**

1. **Self join:-** data in the table are combined /joined with the samedata or rows in a same table

**Syntax:- select columnname from table1,table2 where condition ;**

1. **cross join :-** it will return all the records from both the tables (table1,table2)

**Syntax:- select column name from table 1 cros join table2;**