## #To see how many databases exist

SHOW DATABASES;

## #Creation of Database

CREATE DATABASE COFFEE\_STORE;

## #Using Database for further operations

USE COFFEE\_STORE;

## #Creation of Table

CREATE TABLE PRODUCTS(

id INT auto\_increment primary key,

name varchar(30),

price decimal(3,2)

);

## #Select Operations

SELECT \* FROM products;

create table customers(

id int auto\_increment primary key,

first\_name varchar(30),

last\_name varchar(30),

gender enum("M","F","T"),

phone\_number Varchar(11)

);

show tables;

## #Creation of Foreign and Primary Key

create table orders(

id int auto\_increment ,

table\_no int(2) ,

product\_id int ,

customer\_id int,

order\_time datetime,

primary key (id ,table\_no),

foreign key (product\_id) REFERENCES products(id) ,

foreign key (customer\_id) references customers(id)

);

## #Drop Primary Key

alter table orders

drop primary key ;

## #Modify column type, Length

alter table orders

modify id int auto\_increment;

desc orders;

## #Remove fields from the table

alter table orders

drop column table\_no1 ;

alter table orders;

## #Add Primary Key

alter table orders

modify id int auto\_increment primary key;

## #To add column in database

alter table products

add column coffee\_origin VARCHAR(30);

## #Show description of table

DESC products;

## #Insert values into table

insert into products

values(5,"frapaccino",4.5,"usa");

## #Insert multiple values into table at onego

**insert into products (name,price,coffee\_origin)**

**values("Filter\_Coffee",3.5,"INDIA"),**

**("Filter\_Coffee",3.5,"INDIA"),**

**("Filter\_Coffee",3.5,"INDIA");**

## #Drop Table

truncate table products;

## #Add Foreign Key

alter table orders

add constraint f\_key

Foreign Key (Address\_id) Refrerences table name(fieldname)

EG: ALTER TABLE PETS

ADD CONSTRAINT F\_KEY FOREIGN KEY (OWNER\_ID) REFERENCES PEOPLE (ID) ;

## #MODIFY column name

ALTER TABLE PRODUCTS CHANGE COFFEE\_ORIGIN TEA\_ORIGIN varchar(15);

## #MODIFY column datatype

ALTER TABLE PRODUCTS MODIFY COFFEE\_ORIGIN INT(15);

SELECT \* FROM PRODUCTS;

## #Add unique constraints

ALTER TABLE PEOPLE ADD CONSTRAINT EMAIL\_u UNIQUE(email);

## #First name and pone number of all Female customers with last name of Bluth

select first\_name, phone\_number from customers

where gender = "F" and last\_name = 'Bluth';

## #Count of Male customers with no phone number entered.

select count(\*) from customers

where gender = "M" and phone\_number is null;

## #Customers having last\_name of ‘Taylor’, ‘Bluth’ and ‘Armstrong’

SELECT \* FROM CUSTOMERS

WHERE LAST\_NAME IN (‘Taylor’,’Bluth’,’Armstrong’);

## #Customers having last\_name between ‘A’ and ‘T’. Do note last name starting with T will not be included in the list

SELECT \* FROM CUSTOMERS

WHERE LAST\_NAME between 'A' and 'T';

## #Customers having first\_name with ‘O’ in middle and starting and ending with character on each side.

SELECT \* FROM CUSTOMERS

WHERE first\_name like '\_o\_';

## #To control the number of records fetched. It will only fetch 2 records

SELECT \* FROM ORDERS

WHERE ORDER\_TIME LIKE '2017-02%'

AND CUSTOMER\_ID IN (2,4,6,8)

limit 2;

## #To control the number of records fetched along with starting position. It will fetch 5 records skipping first 2 records which satisfy the below conditions.

SELECT \* FROM ORDERS

WHERE ORDER\_TIME LIKE '2017-02%'

AND CUSTOMER\_ID IN (2,4,6,8)

LIMIT 5 OFFSET 2;

USE COFFEE\_STORE;

select \* from products;

select name, price from products

where COFFEE\_ORIGIN IN ('CANADA','INDONESIA')

ORDER BY NAME;

SELECT \* FROM ORDERS

WHERE ORDER\_TIME LIKE '2017-02%'

AND CUSTOMER\_ID IN (2,4,6,8)

LIMIT 5 OFFSET 2;

SELECT FIRST\_NAME, PHONE\_NUMBER

FROM CUSTOMERS

WHERE LAST\_NAME LIKE '%ar%'

limit 1;

select distinct(last\_name) from customers

order by last\_name;

SELECT \* FROM ORDERS

WHERE ORDER\_TIME LIKE '2017-02%'

AND CUSTOMER\_ID IN (1)

order by order\_time

LIMIT 3;

select name, price as retail\_price, coffee\_origin from products;

show databases;

use studentb;

select \* from studentb.student\_table1;

select \* from studentb.employee;

select count(\*) from studentb.nano\_tubes;

delete from studentb.nano\_tubes;

insert into studentb.nano\_tubes values(2;1;0,679005;0,701318;0,017033;0,721039;0,730232;0,017014);

create database Illikasoft;

use illikasoft;

Create table Employee ( Empid int not null primary Key , firstname varchar(30), lastname varchar(30), title varchar(2), age int, salary float );

desc employee;

insert into Employee values(1,'Ravi','Dubey', 'Mr',39,92360);

insert into Employee values(2,'Astha','', 'Ms',35,62360);

alter table Employee modify title varchar(5);

select \* from employee;

update employee set title = 'Mrs' where empid = 2;

insert into Employee values(3,'Illisha','Dubey', 'Miss',6,22360);

insert into Employee values(3,'Lika','Dubey', 'Miss',3,32360);

alter table Employee modify lastname varchar(25) not null;

insert into Employee values(7,'Astha',null, 'Ms',35,62360);

delete from Employee where empid > 4;

update employee set lastname = null where empid = 2;

alter table employee drop Title;

select \* from employee;

create table employee1(Empname varchar(30), designation varchar(30), age int(3),salary float(5));

insert into employee1 values('Ravi Dubey', 'Senior Technical Architect', 39, 92500);

insert into employee1 values('Astha', 'Senior Manager', 35, 62500);

insert into employee1 values('Illisha Dubey', 'HR Manager', 7, 42500);

insert into employee1 values('Lika Dubey', 'Accountant', 3, 22500);

select \* from employee1;

select \* from employee1 where salary >= 30000;

select Empname from employee1 where age < 30;

select empname from employee1 where designation like '%Manager%';

select \* from employee1 where empname like '%s%a%';

update employee1

set designation = 'Senior HR Manager'

where empname = 'Astha';

update employee

set age = age +1;

update employee1 set designation = 'Sr.' where designation like 'Senior%';

select \* from employee1;

## # Join SQL Operations

create database join\_sql;

use join\_sql;

create table Customers(CustomerID int primary key, CustomerName varchar(40), ContactName varchar(30),Address varchar(50),City varchar(20), PostalCode varchar(10), Country varchar(20));

create table Orders(OrderID int primary key not null, CustomerID int , EmployeeID int, OrderDate datetime, ShipperID int);

insert into Customers values(1,'Ravi Dubey', 'Ravi','5107 Forest Hill Drive' , 'Mississauga', 'L5M5A2', 'Canada');

insert into Customers values(2,'Astha', 'Astha','101, Friends Mahima Apartment' , 'Patna', '800001', 'India');

insert into Customers values(3,'Illisha Dubey', 'Astha','Friends Mahima Apartment' , 'Patna', '800001', 'India');

insert into Customers values(4,'Lika Dubey', 'Astha','Friends Mahima Apartment' , 'Patna', '800001', 'India');

insert into Customers values(9,'Kima', 'Astha','Friends Mahima Apartment' , 'Patna', '800001', 'India');

insert into Customers values(5,'Ian Botham', 'Ravi','5107 Forest Hill Drive' , 'Mississauga', 'L5M5A2', 'Canada');

insert into Customers values(6,'Malcom', 'Astha','101, Friends Mahima Apartment' , 'Patna', '800001', 'India');

insert into Customers values(7,'Kapil Dev', 'Astha','Friends Mahima Apartment' , 'Patna', '800001', 'India');

insert into Customers values(8,'Vishwatma', 'Astha','Friends Mahima Apartment' , 'Patna', '800001', 'India');

insert into Orders values(101,1,1,'2022/06/09',9255);

insert into Orders values(101,1,1,'2022/06/09',9255);

insert into Orders values(102,2,1,'2022/06/09',9255);

insert into Orders values(103,5,1,'2022/06/09',9255);

insert into Orders values(104,9,1,'2022/06/08',9255);

insert into Orders values(105,3,1,'2022/06/07',9255);

insert into Orders values(106,3,1,'2022/06/09',9255);

insert into Orders values(107,4,1,'2022/06/05',9255);

select \* from CUSTOMERS;

select \* from orders;

alter table orders add amount int;

update orders

set amount = 86

where orderid = 105;

#other way to get details fo person with maximum salary

select \* from orders where amount = ( select max(amount) from orders);

#other way to get details of person with min salary

select \* from orders where amount = ( select min(amount) from orders);

#To get list of employees having salary greater than average Salary

select \* from orders where amount >= ( select avg(amount) from orders);

#To get list of employees having amount less than average Salary

select \* from orders where amount < ( select avg(amount) from orders);

#To get 2nd maximum salary of employees having average Salary

select \* from orders where amount in (select max(amount) from orders where orderid not in(

select orderid from orders where amount in ( select max(amount) from orders)));

#To get 2nd maximum salary of employees having average Salary

select \* from orders

group by amount

order by amount desc limit 1,1;

## # Full outer join

-- Union Left join and right join results

select a.\*, b.\* from

orders a Left join customers b

on a.customerid = b.customerid

union

select a.\*, b.\* from

orders a right join customers b

on a.customerid = b.customerid;

select CUSTOMERS.\*

FROM orders JOIN customers

on orders.customerid = customers.customerid;

## #Regular Expression

SELECT \* FROM CUSTOMERS;

select \* from customers where CustomerName regexp '[D]';

select \* from customers where CustomerName regexp '[D,i]';

## #Date Processing

select sysdate(), datediff(sysdate(),'20121128') ;

select day(sysdate());

select adddate('20121128',15), adddate('20121128',interval 11 day);

select adddate('20121128',15), adddate('20121128',interval 11 month);

select adddate('20121128',15), adddate('20121128',interval 11 quarter);

select subdate('20121128',15);

select date\_format(sysdate(),'%y');

select date\_format(sysdate(),'%Y');

select date\_format(sysdate(),'%m');

select date\_format(sysdate(),'%M');

select date\_format(sysdate(),'%d');

select date\_format(sysdate(),'%D');

##it will output in 20220615;

SELECT date\_format(sysdate(),'%Y%m%d');

##it will output in 20220615;

SELECT date\_format(sysdate(),'%Y%m%d');

##it will output in 220615;

SELECT date\_format(sysdate(),'%y%m%d');

##it will output in 22June15;

SELECT date\_format(sysdate(),'%y%M%d');

##it will output in 22June15th;

SELECT date\_format(sysdate(),'%y%M%D');

select \* from customers;

select \* from orders;

create table customer\_orders

select a.customerid, b.orderid, a.customername,b.employeeid, b.amount

from customers a join orders b on a.customerid= b.customerid

where b.amount > 50;

select \* from customer\_orders;

## #Views.

create view customer\_orders1 as

select a.customerid, b.orderid, a.customername,b.employeeid, b.amount

from customers a join orders b on a.customerid= b.customerid

where b.amount > 50;

update customer\_orders1 set orderid = 111 where customerid = 2;

select \* from customer\_orders1;

## #Guidelines about Views.

* We can update noncomplex views
* If we update views, original tables on which views is based gets updated too
* If we update original tables, it reflects in view

## #Stored Procedures

DELIMITER //

create procedure get\_order\_details2(amt Int)

Begin

select orderid, customerid, amount from orders where amount > amt;

End //

## #syntax to call Stored Procedures

call get\_order\_details2(10);

## #Stored Procedures with input and output variable

DELIMITER //

create procedure get\_order\_details5(IN amt Int, OUT orderid\_o varchar(3))

Begin

select orderid from orders where amount= amt into orderid\_o;

End //

call get\_order\_details5(50,@order\_id\_o);

select @order\_id\_o as order\_id\_output;

## #DROP Procedures

DROP PROCEDURE IF EXISTS get\_order\_details1;

Table

Description automatically generated

use ravibank;

SHOW TABLES;

SHOW VARIABLES LIKE "secure\_file\_priv";

Go to location C:\ProgramData\MySQL\MySQL Server 5.7

Open the file my in notepad ++

Search for secure-file-priv. assign it to “ “

load data local infile 'D://Study//Data Science//Python//ineuron//bank1.csv' into table bank\_details4

fields terminated by ';' Enclosed by '"' Terminated by '\n' IGNORE 1 ROWS;

load data infile 'C:\\ProgramData\\MySQL\\MySQL Server 5.7\\Uploads\\bank.csv' into table bank\_details4

fields terminated by ';' Enclosed by '"' Terminated by '\n' IGNORE 1 ROWS;

load data infile 'D://bank2.csv' into table bank\_details4

fields terminated by ';' Enclosed by '"' IGNORE 1 ROWS;

desc bank\_Details4;

select count(\*) from bank\_Details4;

delete from bank\_Details4;

select marital, education,max(Age), sum(balance), count(housing) from bank\_details4 group by marital, education;

select education, char\_length(education), balance, length(Balance) from bank\_details4;

select \* from bank\_Details where job like 'a%m\_%\_';

update bank\_details set job= 'ammin.' where age = 29;

select count(\*) from bank\_Details4;

#Task 1: Try to find out average of Balance

select avg(balance) from bank\_Details4;

#Task 2: Try to find who has mininum Balance

select \* from bank\_Details4 where balance = (select min(balance) from bank\_Details4);

#alternate way provided we have only 1 person with minimum balance

select \* from bank\_details4

order by balance

limit 1;

#Task 3: Try to find who is has maximum balance

select \* from bank\_Details4 where balance = (select max(balance) from bank\_Details4);

#other way provided we have only 1 person with maximum balance

select \* from bank\_details4

order by balance desc

limit 1;

#Task 4: Try to prepare a lit of all the person who is having a loan

select \* from bank\_Details4 where upper(loan) = upper('yes');

#Task 5 : try to find out average balance for all the people whose job role is admin

select avg(balance) from bank\_details4 where job = 'admin.';

#Task 6: Try to find out a record without job whose age is below 45

select \* from bank\_details4 where age < 45 and upper(job) = upper('unknown');

#Task 7: Try to find out a record where education is primary and person is jobless;

select \* from bank\_details4 where upper(education) = upper('primary') and upper(job) = upper('unemployed');

#Task 8: Try to find out a record whose bank account is having a negative balance;

select \* from bank\_details where balance < 0;

#Task 9 : Try to find out a record who is not having house at all despite,

select age, balance,housing from bank\_details where upper(housing) = upper('no');