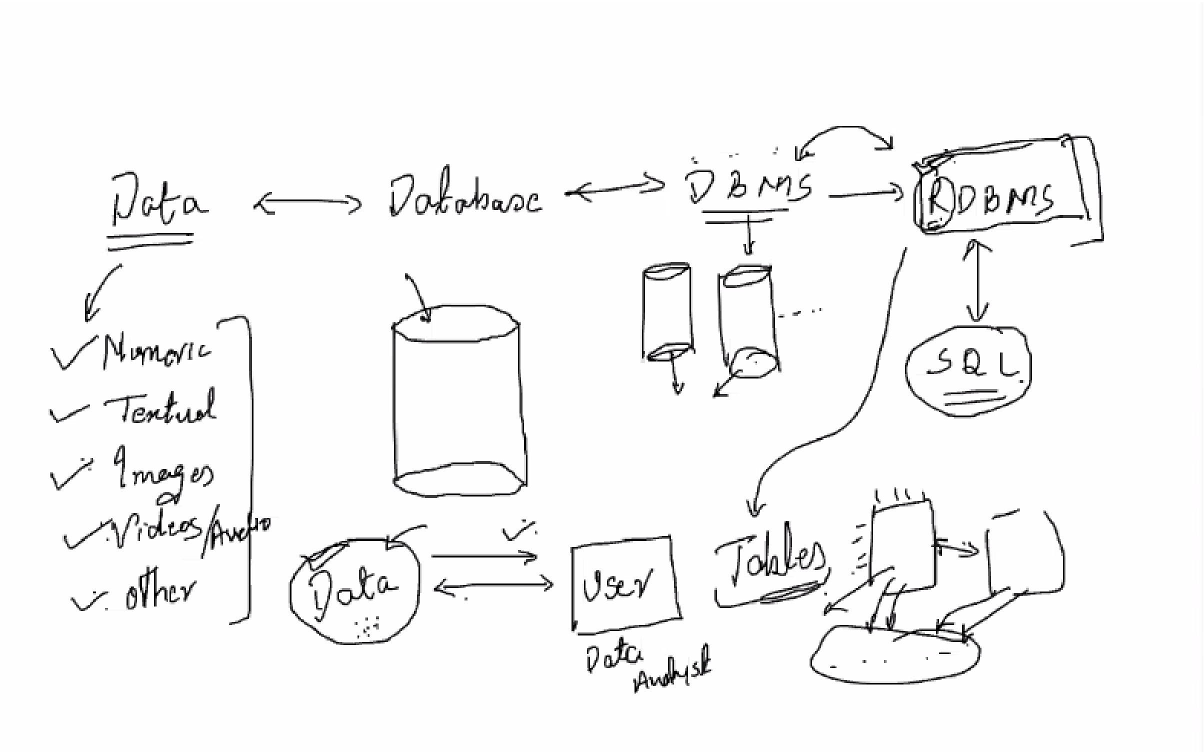
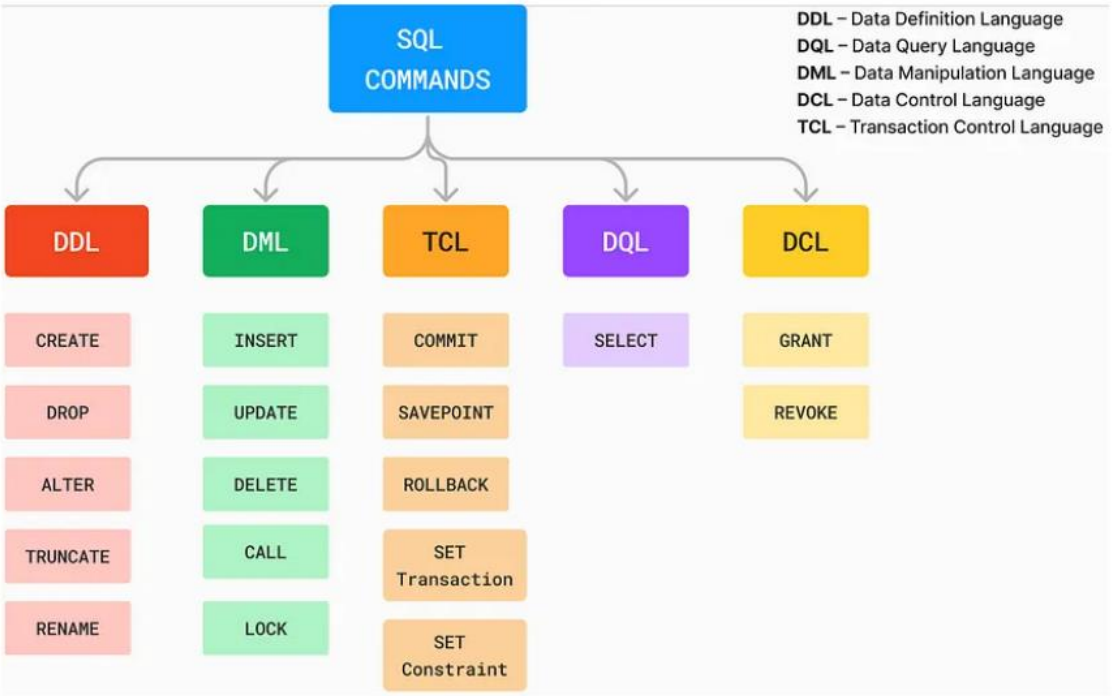
# SQL COMMANDS



## Create Database

```
➔ create database <database-name>;
```



## Drop Database

- ➔ drop database <database-name>;

## Create Table

- ➔ create table <table-name> (column-name dtype);
- ➔ E.g=

-use <database-name>;

-create table courses (Course_id char(5), Course_Title varchar(30), Time_duration int, Student_intake int);

## Drop Table

- ➔ drop table <table-name>;

# Select all data from table

- ➔ select * from <table-name>

# Alter the table

**I want to add a column**

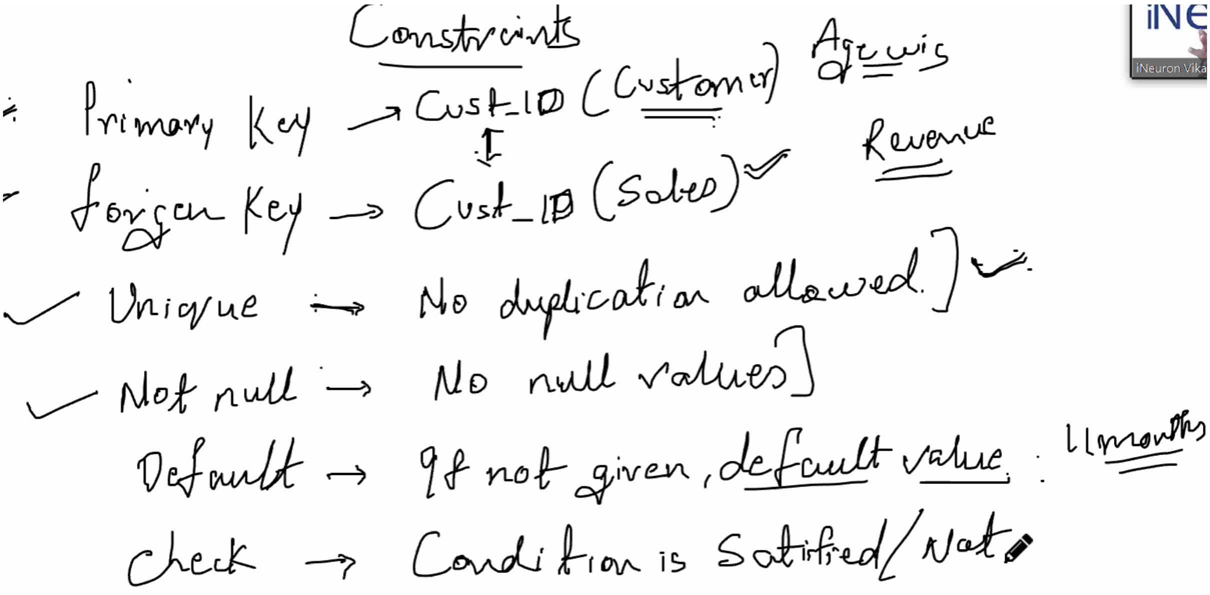
- ➔ Alter table <table-name> add column <column-name> dtype;
- ➔ E.g= Alter table Courses
  add column Mode_of_delivery Varchar(40);

**I want to change a column**

- ➔ Alter table <table-name> Change column <old-column-name> <new-name> varchar(20);
- ➔ E.g= = Alter table Courses
  Change column Mode_of_delivery  Delievery_mode Varchar(20);

# CONSTRAINTS

- ➔ Constrainsts means restrictions
- ➔ Contrainsts are
  - Primary key
  - Foreign key
  - Unique
  - Not Null
  - Default



Constraints

Primary Key → Cust_ID (Customer) Agewis

Forigen Key → Cust_ID (Sales) Revenue

Unique → No duplication allowed.

Not null → No null values

Default → If not given, default value : 11months

Check → Condition is Satisfied/Not

-Constraints will apply on column

-If we use primary constraint on the column then automatically unique and not null constraint is applied.

## DIFFERENCE BETWEEN TRUNCATE AND DROP

-If we use truncate then all data which is in table will deleted.

-If we use drop then entire table is got deleted.

-Delete is use to delete specific rows

# DML

## INSERT

-Insert into <table-name>

values(first,second,third,….);

## INSERT MULTIPLE VALUE

-Insert into <table-name>

Values(first,second,third column values,…),(first,second,third,……);

## INSERT VALUE IN PARTICULAR COLUMN

-Insert into <table-name> (first column,second column)

Values(first value,second value)

# CONSTRAINTS

## Unique constraint

- ➔ Alter table <table-name> change column <old-name> <new-name> <dtype> Unique;

## Not Null Constraint

- ➔ Alter table <table-name> change column <old-name> <new-name> <dtype> Not Null;

You make old-name and new-name of column same or you can change it also.

**Note=**

1)Update is for row level

2)Alter is for column level

## DEFAULT CONSTRAINT

- ➔ Alter table <table-name> change column <old-column-name> <new-column-name> <dtype> default="<default-value>"

NOTE=Now if we not inserting the value for that column for which we set the default value then it will take that default value if we not inserting value for that column.

## CHECK CONSTRAINT

- ➔ Alter table <table-name> change column <old-column-name> <new-column-name> <dtype> check <condition>

E.g=>

- ➔ Alter table <table-name> change column student_intake student_intake int check (student_intake>100)

## PRIMARY KEY CONTRAINT

- ➔ Alter table <table-name> change column <old-column-name> <new-column-name> <dtype> Primary key

## If we want to do some changes in database like update the data delete the data at that time we have to do this

- ➔ set sql_safe_updates=0;

## And if you want that no one can do the changes in database then:-

- ➔ set set sql_safe_updates=1;

# UPDATE THE DATA

```
Update <table-name>
Set <column-name> = <value which you want to set>
Where <column-name> = <value>;
```

E.g=

```
Update Courses
Set Students_intake = 160
Where Course_id =”ML101”;
```

# DESCRIBE

```
Describe <table-name>;
```

# OFFSET AND LIMIT

```
Select * from city
Limit 10
Offset 5;
```

NOTE=It means from 6th row give the 10 rows data because we do offset 5 so means don’t give till 5th row give after 5th row means from 6th row.

# ORDER BY

```
Select * from city
Order by <column-name> Asc/desc;
```

### MULTIPLE COLUMN ORDER

Select * from city

Order by <column-one> <column-two> desc;

QUESTION=Select 3$^{rd}$ MAX POPULATION DATA FROM COUNTRY TABLE

Select country,continent,population from country

Order by population desc limit 1 offset 2;

-Note=By default it is Ascending

## IF WE WANT TO SAVE THE TABLE INTO .CSV FORMAT THEN FOLLOW THIS STEPS:

1)Right click on the table which you want to export and save it in the csv format and click on Table data export wizard.

2)Then select the path where you want to select and process next.

## IF WE WANT TO IMPORT THE CSV FILE IN THE SCHEMA

E.G WORLD SCHEMA/DATABASE IS THERE AND YOU HAVE 3 CSV FILES city.csv, country.csv, country_language.csv now you want to import this 3 csv file and create the table like city, country,country_language this 3 tables you want in the world schema and this 3 tables data you have in the form of csv then to import that csv follow this steps:

1)Right click on the Tables and then click on table data import wizard and then do the other process like select csv and continue the further process and do same for all csv file.

# CREATE TABLE

- create table Courses (Course_id char(5),Course_Title Varchar(30),Time_duration int,Students_intake int);

# COMMANDS

- select * from courses;
- Alter table Courses add column Mode_of_Delivery Varchar(40);
- Alter table Courses add column Faculty Varchar(25);
- Alter table Courses Change column Mode_of_delivery Delivery_Mode Varchar(20);

## ## DML

### #INSERTION

- Insert into Courses

  values ("DA101","Data Analytics",6,100,"Live Sessions","Saurabh");

### # Multiple rows insertion

- Insert into Courses

  values ("DA102","Data Analytics 2.0",6,100,"Live Sessions","Saurabh"),("DS101","Data Science",11,150,"Live Sessions","Mayank");

- Insert into Courses (Course_id,Course_Title,Faculty)

  values ("ML101","Machine Learing Fundamentals","Suraj");

- Insert into Courses (Course_id,Course_Title,Time_duration,Faculty)

  values ("ML101","Machine Learing 2.0","Bappi Sir");

- Alter table Courses Change Column Course_ID Course_ID Varchar(10) Unique;
- select * from courses;

- Alter table Courses Change Column Course_ID Course_ID Varchar(10) Unique;

- Insert into Courses

  values ("ML101","Machine Learing 2.0",12,150,"Live Classes","Bappi Sir");

- Alter table courses
  change column course_title Course_Title Varchar(30) Not Null;
- Insert into Courses (Course_ID,Time_duration,Students_intake,Delivery_mode,Faculty)

values ("ML102",12,150,"Live Classes","Vivek Sir");

# # Check

- use ineuron_dup;
- select * from courses;
- update Courses

  set Time_duration = 9

  where Course_Id = "ML101";

- update Courses

  set Students_intake = 160

  where Course_id = "ML101";

# # Default

- Alter table courses
  change column Delivery_Mode Delivery_Mode Varchar(20) default "Live Session";
- Insert into Courses(Course_id,Course_Title,Time_Duration,Students_intake,Faculty)

  values ("ML102","Machine Learing 2.0",12,150,"Bappi Sir");

select * from courses;

# # Check

- Alter table courses change column Students_intake Students_intake int check (Students_intake>=100);
- Insert into Courses

  values ("GA101","Generative AI",12,90,"Hybrid Mode","Sunny Sir");

- Describe Courses;
- select * from courses;

# # Primary Key

- ➔ Alter table courses change column Course_id Course_id varchar(30) Primary key;
- ➔ Insert into Courses

  values ("GA101","Generative AI",12,190,"Hybrid Mode","Sunny Sir");

# ## Creating database and importing data

- ➔ Create Database world2;
- ➔ use world2;

# # Managing Data Output

# # Limit

```
select * from city

limit 10;
```

# # Limit + Offset

```
select * from city

limit 10

offset 5;
```

# # Ordering the data output--------

## # One Column

```
select * from city

order by Population desc;
```

## ## Multiple Columns

```
select * from country

order by continent, country desc;
```

```
    ➔ use world;
select Country,Continent,Population from country
order by Population desc limit 1 offset 2;

select * from country order by continent, Region desc, IndepYear desc;
```

## DATA FILTERING -----------------------------------

# Where Operator

```
    ➔ select * from country
    where Continent = "Asia";

    ➔ select * from country
    where indepyear = 1991;
```

# Relational Operators (>,<,=,>=,<=,<>)

# Numeric Values

```
    ➔ select country,continent,GNP from country
    where Indepyear <> 1947
    order by GNP desc;
```

# textual Values

```
    ➔ select * from country
    where country < "Japan";
```

- ➔ select * from country

  where indepyear is null;

# # Logical Operators

## # And

- ➔ select * from country

  where population > 3401200 and lifeExpectancy < 60 and GNP > 5411;

## # Or

- ➔ select * from country

  where population > 3401200 or lifeExpectancy < 60 or GNP < 8400;

## # Not

- ➔ select * from country

  where not population > 3401200;

# # Like, IN and Between Operators

## # In

- ➔ Select * from country where Continent = 'Asia' or Continent = 'Africa' or Continent = 'Europe'

  order by Continent;

- ➔ select * from country

  where continent in ('Asia','Europe','Africa');

- ➔ alter table country change column country Country text;

- ➔ select Country,Continent,Region from country

```
where lifeexpectancy in (45.9,74.1,75.1);
```

## ## Write a query to fetch the data of all countries from Asia,Africa, Europe.

### # Between

```
➔ select Country,Continent,Region from country
where lifeexpectancy between 45 and 56;
```

### # NOT BETWEEN

```
➔ select * from country
where lifeexpectancy not between 44 and 56.5
order by LifeExpectancy;
```

### # Like

```
➔ select * from address
where state like 'Delhi';
```

### # All countries where the name starts with 'A'.

```
use world;
```

### ## Starting from 'A'

```
➔ select * from country
where Country like 'A%';
```

### ## Having 'A'

```
➔ select * from country
where Country like '%A%';
```

**## Ending with 'A'**

```
➔ select * from country
where Country like '%A';
```

**## Fixed Characters starting from A**

```
➔ select * from country
where country like 'A____';
```

```
➔ select * from country
where country like '__A__%';
```

```
➔ use orders;
select * from product
where product_desc like '%Nokia%';
```

```
➔ select * from product
where product_desc like '% Tab %' or product_desc like '% Tab';
```

NOTE=

## % ---> any number of characters

## _ ---> fixed number of charecters

## Space ----> Individualise the phrase.

## A would mean ending with a.

## __ - Exactly 2 charecters

## % - any number of charecters

## "%___A" ---> Means text ending with atleast 4 chatrecters ending with A

## ## "__A%" ---> Means text starting with atleast 3 charecters and 3rd letter being A

## ## "__A__%" ---> Means text starting with at least 5 charecters and 3rd letter to be A.

## # Aliases

### # Aliases for columns

```
use world;

select Country as C,continent as Con from country;


# Aliases for Expression


select Country, Continent, Population/1000000 as Pop_in_mn from country

order by pop_in_mn desc;


Select * from country

Order by continent, region desc,IndepYear desc;

Select * from country

where country like 'A____';




# Build in Functions in SQL

        ## Aggregate Functions

## Sum /Average/ Max/ Min/ Count/ Count distinct


# Total No. of Customers/No. of Customers who placed orders/No. of Orders

use orders;
```

```
select count(customer_id)  as Total_Customers from online_customer;

select count(distinct customer_id)  as Total_Customers from order_header;


select

count(distinct customer_id)  as Active_Customers,

count(order_id) as Total_Orders from order_header;


select count(distinct Order_id) as Total_orders, count(product_id) as Products,
sum(product_quantity) as Total_quantity from order_items;

select * from order_items;


use world;


Alter table Country change column country Country_name text;

select * from country;

Alter table Country drop column Code2;


select Sum(Population) as Avg_LE from country;


## Country with highest life expectancy---

select min(lifeExpectancy) from country;

select * from country;


        # Group by Operator

select * from country;

select Continent,round((sum(SurfaceArea)/1000000),2)
Total_Surface_Area_mn,round(sum(Population)/1000000,2) Total_Population_mn

,round(avg(LifeExpectancy),2) Avg_LE,sum(GNP) Total_GNP from country

group by Continent

Order by Total_Population_mn desc;


select Continent, count(Country_name) as No_of_Countries, sum(Population) from country
```

```sql
group by Continent;

   # Having

select Continent,
count(Country_name) Total_countries,
round((sum(SurfaceArea)/1000000),2) Total_Surface_Area_mn,
round(sum(Population)/1000000,2) Total_Population_mn,
round(avg(LifeExpectancy),2) Avg_LE,
sum(GNP) Total_GNP from country
group by Continent
having continent in ('Asia','Europe','North America')
order by Total_Population_mn desc;



####    set global sql_mode = (select replace(@sql_mode,'only_full_group_by',"'"));

## Write a query to find the name of the country with the highest population.

select Country_name from country
order by population desc
limit 1;

use orders;
select * from address;
## String Functions
   # Concat -  Joins two or more strings

select * from address;
select concat(Address_line1,', ',Address_line2,', ',city,', ',state,', ',pincode,', ',country) as Address from address;


   # Upper/Lower

select lower(city) as City from address;


           # Length

select address_line1,length(address_line1) as length from address;

   # Substring

select * from online_customer;

select Customer_fname,concat(substring(Customer_fname,1,3),'_',substring(Customer_lname,1,3)) as Password from online_customer;

select Customer_fname,substring(Customer_fname,-3,3),length(substring(Customer_fname,-3,3)) from online_customer;

select Customer_fname,substring(Customer_fname,-1,3) from online_customer;


   # Replace -

select address_line1,replace(address_line1,'H.NO.','House Number ') from address;

   # Trim - Removes whitespaces from left and right of the text.


#### set sql_safe_updates = 0;


   # Left/Right

select customer_fname,right(Customer_fname,4) from online_customer;
```

#Practise question prompt

I have a data set consisting of the following columns
code text
country text
continent text
region text
surfaceArea double
Indepyear int

population int
LifeExpectancy double
GNP double
GNPOld double
LocalName text
GovernmentForm text
HeadOfState text
please provide 20 tough SQL questions for practice

## write a sql query select middle characters from the string if the length of the string is even then it should take 2 middle characters and if the length of the string is odd then it should take one middle character.

```
select Customer_fname,

case

   when (length(Customer_fname)/2) %2 = 0 Then
substring(Customer_fname,(length(Customer_fname)/2),2)

   else substring(Customer_fname,(length(Customer_fname)/2),1)

end as middle_characters

from online_customer;
```

# 16/6/2024

```
   ## Date and Time Functions

   # Current Date

use orders;

select current_date();

   # Current time

select current_time();
```

```
   # Now/Current_timestamp
select current_timestamp();
select current_user();
   # Date

select date('2024-06-15 10:17:13') as Date;
   # Time
select time('2024-06-15 10:17:13') as Time;
   # Year
select year(customer_creation_date) Inception_Year,
count(customer_creation_date) Customers_Aqu from online_customer
group by Inception_Year
order by Customers_Aqu desc;

  # Month
select monthname(customer_creation_date) Month,
count(customer_creation_date) Customers_Aqu from online_customer
group by Month
order by Customers_Aqu desc;

   # Day
select day(customer_creation_date) as Day from online_customer;
   # Dayname
select
dayname(customer_creation_date) as Weekday,
count(Customer_creation_date) as Customer_Aqu
from online_customer
```

```
group by Weekday
Order by Customer_Aqu desc;

   # Dateadd
select
Customer_creation_date,
date_add(customer_creation_date, interval 10 Day) as final_date
from Online_customer;

   # Datesub
Select
Customer_creation_date,
date_sub(customer_creation_date, interval 10 Day) as final_date
from Online_customer;

         # Datediff
Select
Customer_creation_date,
date_sub(customer_creation_date, interval 10 Day) as final_date,
datediff(customer_creation_date,date_sub(customer_creation_date, interval 10 Day)) as Inter
from Online_customer;

Select
Customer_creation_date,
date_sub(customer_creation_date, interval 10 Day) as final_date,
-round(datediff(customer_creation_date,current_date()) / 365, 2) as Inter
```

```
from Online_customer;

select year(current_date());

## Numeric Functions
   # Abs - Absolute value of a number
select abs(-44.38) as Modu;

   # Round
select round(-44.38,1) round_up_number;
   # Ceiling/Floor
select ceiling(44.38);
select floor(44.38);
   # Sqrt
select sqrt(100);
   # Power
select power(2,3);
   # Rand
select round(rand()*10000,0) as Random_num;
   # Mod

   # Sign
   # Truncate
Select truncate(48.936547, 2);
   # Log
   # exp
select exp(2);

## Conditional Functions
use world;
   # If
select
country_name,
if(Population > 25434098,"Above_Ave","Below_Ave") as Pop_Category
from country;

select
if(Population > 25434098,"Above_Ave","Below_Ave") as Pop_Category,
count(country_name) as No_of_Countries
from country
group by Pop_Category;

select
if(Population/SurfaceArea > 650,"Denesly Pop",
if(Population/SurfaceArea > 450 and Population/SurfaceArea < 650,"Av_Pop","Low_Pop_Density")) as Pop_Den_Cat,
count(Country_name) as Num
from country
group by Pop_Den_Cat
order by Pop_Den_Cat desc;

   # Case When Operator
```

```
select Country_name, Case
when Population > 25434098 then "Above Average"
when Population < 25434098 then "Below Average"
else "Equal to Average"
end as Pop_Cat
from country;

select Country_name, Case
when Population/SurfaceArea > 650 then "Densely Pop"
when Population/SurfaceArea < 650 and Population/SurfaceArea > 450 then "Average"
else "Below Average"
end as Pop_Cat
from country;

select * from country;

   # Coalesce
use orders;
select Order_Id, Payment_mode, Coalesce(Payment_mode,"Not Available") new_payment_mode from order_header
where Payment_mode is null;

use orders;

select * from Country;
select order_id,Payment_mode,Coalesce(Payment_mode,"Not Available") new_payment_mode

from order_header where Payment_mode is null;
```

```
select order_id, payment_mode, coalesce(payment_mode, 'Not avaliable ') new_payment_mode from order_header where payment_mode is null;
```

## Nullif

```
Select Payment_mode,nullif(payment_mode,"Credit Card") from order_header;
```

## IfNull

```
select Order_date, ifnull(order_date,"NA") from order_header;

select payment_mode, ifnull(payment_mode,"NA") as NPM from order_header;

select Order_date, ifnull(Order_date,"NA") as NOD from order_header;
```

```
select order_id,Payment_mode,Coalesce(Payment_mode,"Not Available") new_payment_mode from order_header where Payment_mode is null;
```

## Window Functions

- ➔ Note= we do not use alias with where or having.
- ➔ If we use partition in rank and dense rank then it will do ranking in partition e.g if we write partition by Continent then it will ranking in the partition here it will do ranking in the partition of continent if continent is Asia,Africa so it will give ranking in Asia first then give Africa and in each continent it will start ranking from 1.

TODAY=

```
## Window Functions

#Rank

select rank() over(order by GNP desc) as Ranking,

Country_name, continent, GNP

from country;


select rank() over(order by GNP/population desc) as Ranking,

Country_name, continent, GNP, round(GNP*1000/Population,2) as GNP_Per_Cap

from country

limit 5;


#Denserank

select dense_rank() over(partition by Continent,Region order by GNP/population desc,GNP desc) as Dense_Ranking,

Country_name, Region, continent, GNP, round(GNP*1000/Population,2) as GNP_Per_Cap

from country;


#Ntile

select country_name,continent,region,LifeExpectancy,

ntile(3) over(order by LifeExpectancy desc) as Clusters

from country;


#Sum Over

use world;
```

```
select country_name, continent, GNP, sum(GNP) over(order by GNP desc) as Cum_GNP

from country;


#Avg Over


use world;

select country_name, continent, GNP, round(avg(GNP) over(partition by continent order by GNP desc),2) as Avg_GNP

from country;


#Lag and Lead Functions

select country_name, continent, GNP, lag(GNP,2) over(order by GNP desc) as Pre_GNP,

(lag(GNP,2) over(order by GNP desc) - GNP) as GNP_Change

from country;


select country_name, continent, GNP, lead(GNP,2) over(order by GNP desc) as Post_GNP,

(lead(GNP,2) over(order by GNP desc) - GNP) as GNP_Change

from country;


# Joins in SQL


use ineuron;

create Table Demographic (Id Varchar(20), age int, gender char(1), salary int, city varchar(20));
create Table Professional (Id Varchar(20), Name Varchar(20), Dept
varchar(10), Manager varchar(20));


Insert into Demographic

values
(201,25,"M",20000,"Beng"),(202,32,"F",25000,"Mum"),(203,40,"F",20000,"
Mum"),(204,23,"M",22000,"Che");


Insert into Professional

values (202,"Shree","Mar","Ram"),
(204,"Ram","Fin","Atul"),(211,"Priya","HR","Raj"),(212,"Ritu","Ops","Amar")
;


select * from Professional;

    ## Inner Join


select demographic.id,Age,Gender,Dept,Manager

from demographic

inner join Professional

on demographic.id = Professional.id;


select d.id,Age,Gender,Dept,Manager

from demographic d

inner join Professional p

on d.id = P.id;
```

22/6/2024

```
    ## Inner Join

select demographic.id,Age,Gender,Dept,Manager
from demographic
inner join Professional
on demographic.id = Professional.id;

select d.id,Age,Gender,Dept,Manager
from demographic d
inner join Professional p
on d.id = P.id;

   ## Outer Join
select * from demographic;
select * from professional;
use ineuron;

## Left Outer

select d.id,Age,Gender,Dept,Manager
from demographic d
Left join professional p
on d.id = P.id;

select p.id,Age,Gender,Dept,Manager
from professional p
```

```
Left join demographic d
on d.id = P.id;


## Right Outer

select p.id,Age,Dept,Manager,Salary
from demographic d
right join professional p
on d.id = P.id;

## Joining 3 tables

use world;
select * from city;
select * from country;
select * from countrylanguage;

use world;
select id as City_ID,Name as City_Name,
District,ci.Population,Country_name,Continent,Region,count(Language)
No_of_languages
from country c
right join city ci
on ci.countrycode = c.code
left join countrylanguage cl
on ci.Countrycode = cl.countrycode
```

```
group by City_id;
```

# Full Outer Join

```
select d.id,Age,Gender,Dept,Salary from demographic d
left join
professional p
on d.id = p.id
union
select p.id,age,Gender,Dept,Salary
from demographic d
right join professional p
on d.id = p.id;
```

# Subqueries

## Use Cases of Subqueries

```
use hr_emp;
select * from employees;
```

## Fetch all data of the department where mangaer id is 186

```
select * from employees
where department_id = (select department_id from employees
where manager_id = 186); ## Single row Subquery

select department_id from employees
```

```
where manager_id = 186;

## Fetch the data of all employees where manager_id are either 50,70,186

select * from employees
where department_id in (select distinct department_id from employees
where manager_id in (60,80,186)); ## Multi Row Subquery


## Provide the details of all employees earning equal to Purv.
select * from employees
where salary = (select salary from employees where first_name = 'Purv');

## Provide the details of all employees earning equal to Ajay.



## fetch data of all employees where salary > Overall average salary;

select * from employees
where salary > (select avg(salary) from employees);

## Fetch the Data of all employees who are earning more than vaerage in
their respective departments.

select * from employees e where salary >
```

```
(Select avg(t.salary) from employees t where t.department_id = e.department_id);


use hr_emp;

select * from employees a where salary > (select avg(salary) from employees b

group by b.department_id having b.department_id=a.department_id);


select * from employees e

join ( select department_id, avg(salary) as avg_salary from employees

   group by department_id) as dept_avg on e.department_id = dept_avg.department_id

where e.salary > dept_avg.avg_salary;



select salary from employees

where first_name = 'Ajay' and last_name = 'Mishra';


## Database Objects

# Views

select * from country;

select * from city;

select * from countrylanguage;
```

```
create view City_Summary as
select id as City_ID,Name as City_Name,
District,ci.Population,Country_name,Continent,Region,count(Language)
No_of_languages
from country c
right join city ci
on ci.countrycode = c.code
left join countrylanguage cl
on ci.Countrycode = cl.countrycode
group by City_id;

select * from city_summary;

use hr_emp;

create view emp_more_than_avg_salary as
select * from employees
where salary > (select avg(salary) from employees);

select first_name,last_name,department_id,manager_id from
emp_more_than_avg_salary;



drop view emp_more_than_avg_salary;
```

ALL NOTES

```
use ineuron;

use ineuron;

create database ineuron_dup;

use ineuron_dup;

create table Courses (Course_id char(5),Course_Title
Varchar(30),Time_duration int,Students_intake int);


select * from courses;

Alter table Courses add column Mode_of_Delivery Varchar(40);

Alter table Courses add column Faculty Varchar(25);

Alter table Courses Change column Mode_of_delivery Delivery_Mode
Varchar(20);

select * from courses;


## DML

Insert into Courses

values ("DA101","Data Analytics",6,100,"Live Sessions","Saurabh");

select * from courses;

# Multiple rows insertion

Insert into Courses

values ("DA102","Data Analytics 2.0",6,100,"Live
Sessions","Saurabh"),("DS101","Data Science",11,150,"Live
Sessions","Mayank");


Insert into Courses (Course_id,Course_Title,Faculty)
```

```
values ("ML101","Machine Learing Fundamentals","Suraj");


Insert into Courses (Course_id,Course_Title,Time_duration,Faculty)

values ("ML101","Machine Learing 2.0","Bappi Sir");


Alter table Courses Change Column Course_ID Course_ID Varchar(10) Unique;

select * from courses;


Alter table Courses Change Column Course_ID Course_ID Varchar(10) Unique;

Insert into Courses

values ("ML101","Machine Learing 2.0",12,150,"Live Classes","Bappi Sir");


Alter table courses change column course_title Course_Title Varchar(30) Not Null;

Insert into Courses (Course_ID,Time_duration,Students_intake,Delivery_mode,Faculty)

values ("ML102",12,150,"Live Classes","Vivek Sir");


set sql_safe_updates = 0;

# Check

use ineuron_dup;

select * from courses;

update Courses

set Time_duration = 9 where Course_Id = "ML101";
```

```
update Courses

set Students_intake = 160 where Course_id = "ML101";


# Default

Alter table courses change column Delivery_Mode Delivery_Mode Varchar(20) default "Live Session";

Insert into Courses(Course_id,Course_Title,Time_Duration,Students_intake,Faculty)

values ("ML102","Machine Learing 2.0",12,150,"Bappi Sir");


select * from courses;

# check


Alter table courses change column Students_intake Students_intake int check (Students_intake>=100);

Insert into Courses

values ("GA101","Generative AI",12,90,"Hybrid Mode","Sunny Sir");


Describe Courses;

select * from courses;

# Primary Key

Alter table courses change column Course_id Course_id varchar(30) Primary key;

Insert into Courses

values ("GA101","Generative AI",12,190,"Hybrid Mode","Sunny Sir");


## Creating database and importing data
```

```
Create Database world2;
use world2;

use world;

# Managing Data Output

select * from city;
select * from country;
describe country;
Alter table country change column Name Country char(52);
select * from countrylanguage;

# Limit

select * from city
limit 10;
# Limit + Offset

select * from city
limit 10
offset 5;


# Ordering the data output--------
# One Column
```

```
describe city;
Alter table city change column Name City char(35);
select * from city
order by City desc;

select * from city
order by Population desc;



## Multiple Columns
select * from country
order by continent, country desc;

use world;
select Country,Continent,Population from country
order by Population desc limit 1 offset 2;

select * from country order by continent, Region desc, IndepYear desc;

## DATA FILTERING --------------------------------
# Where Operator

select * from country
where Continent = "Asia";

select * from country
```

```
where indepyear = 1991;



# Relational Operators (>,<,=,>=,<=,<>)
     # Numeric Values
select country,continent,GNP from country
where Indepyear <> 1947
order by GNP desc;

   ## textual Values

select * from country
where country < "Japan";

select * from country
where indepyear is null;
create database world;
use world;

Describe country;
select * from country;
# Logical Operators
     # And
select * from country
where population > 3401200 and lifeExpectancy < 60 and GNP > 5411;
  # Or
```

```sql
select * from country

where population > 3401200 or lifeExpectancy < 60 or GNP < 8400;

   # Not


select * from country

where not population > 3401200;


select * from country

where population > 3401200 or lifeexpectancy < 60 or GNP < 8400;


# Like, IN and Between Operators

     # In


Select * from country where Continent = 'Asia' or Continent = 'Africa' or Continent = 'Europe'

order by Continent;


select * from country

where continent in ('Asia','Europe','Africa');


alter table country change column country Country text;

select * from Country;


select Country,Continent,Region from country

where lifeexpectancy in (45.9,74.1,75.1);
```

```
## Write a query to fetch the data of all countries from Asia,Africa, Europe.
    # Between

select Country,Continent,Region from country
where lifeexpectancy between 45 and 56;

select * from country
where lifeexpectancy not between 44 and 56.5
order by LifeExpectancy;

create database orders2;
use orders2;

# Like
use orders;
select * from address;
select * from address
where state like 'Delhi';

# All countries where the name starts with 'A'.
use world;

## Starting from 'A'
select * from country
where Country like 'A%';
```

```
## Having 'A'
select * from country
where Country like '%A%';

## Ending with 'A'
select * from country
where Country like '%A';

## Fixed Characters starting from A
select * from country
where country like 'A____';

select * from country
where country like '__A__%';

use orders;
select * from product
where product_desc like '%Nokia%';

select * from product
where product_desc like '% Tab %' or product_desc like '% Tab';

## % ---> any number of characters
## _ ---> fixed number of charecters
## Space ----> Individualise the phrase.
## A would mean ending with a.

## __ - Exactly 2 charecters

## % - any number of charecters


## "%___A" ---> Means text ending with atleast 4 chatrecters ending with A

## "__A%" ---> Means text starting with atleast 3 charecters and 3rd letter being A

## "__A__%" ---> Means text starting with at least 5 charecters and 3rd letter to be A.



# Aliases


# Aliases for columns

use world;

select Country as C,continent as Con from country;


# Aliases for Expression


select Country, Continent, Population/1000000 as Pop_in_mn from country

order by pop_in_mn desc;


Select * from country

Order by continent, region desc,IndepYear desc;

Select * from country

where country like 'A____';
```

```sql
# Build in Functions in SQL
    ## Aggregate Functions
## Sum /Average/ Max/ Min/ Count/ Count distinct

# Total No. of Customers/No. of Customers who placed orders/No. of Orders
use orders;
select count(customer_id)  as Total_Customers from online_customer;
select count(distinct customer_id)  as Total_Customers from order_header;

select
count(distinct customer_id)  as Active_Customers,
count(order_id) as Total_Orders from order_header;

select count(distinct Order_id) as Total_orders, count(product_id) as
Products, sum(product_quantity) as Total_quantity from order_items;
select * from order_items;

use world;

Alter table Country change column country Country_name text;
select * from country;
Alter table Country drop column Code2;

select Sum(Population) as Avg_LE from country;
```

```
## Country with highest life expectancy---
select min(lifeExpectancy) from country;
select * from country;

    # Group by Operator
select * from country;
select Continent,round((sum(SurfaceArea)/1000000),2)
Total_Surface_Area_mn,round(sum(Population)/1000000,2)
Total_Population_mn
,round(avg(LifeExpectancy),2) Avg_LE,sum(GNP) Total_GNP from country
group by Continent
Order by Total_Population_mn desc;

select Continent, count(Country_name) as No_of_Countries,
sum(Population) from country
group by Continent;

   # Having

select Continent,
count(Country_name) Total_countries,
round((sum(SurfaceArea)/1000000),2) Total_Surface_Area_mn,
round(sum(Population)/1000000,2) Total_Population_mn,
round(avg(LifeExpectancy),2) Avg_LE,
sum(GNP) Total_GNP from country
group by Continent
having continent in ('Asia','Europe','North America')
```

```
order by Total_Population_mn desc;




####    set global sql_mode = (select
replace(@sql_mode,'only_full_group_by',"'"));


## Write a query to find the name of the country with the highest
population.


select Country_name from country
order by population desc
limit 1;


use orders;
select * from address;
## String Functions
   # Concat -  Joins two or more strings


select * from address;
select concat(Address_line1,', ',Address_line2,', ',city,', ',state,', ',pincode,',
',country) as Address from address;


   # Upper/Lower
select lower(city) as City from address;


    # Length
```

```
select address_line1,length(address_line1) as length from address;
   # Substring
select * from online_customer;
select Customer_fname,concat(substring(Customer_fname,1,3),'_',substring(Customer_lname,1,3)) as Password from online_customer;
select Customer_fname,substring(Customer_fname,-3,3),length(substring(Customer_fname,-3,3)) from online_customer;
select Customer_fname,substring(Customer_fname,-1,3) from online_customer;


   # Replace -
select address_line1,replace(address_line1,'H.NO.','House Number ') from address;
   # Trim - Removes whitespaces from left and right of the text.


#### set sql_safe_updates = 0;


   # Left/Right
select customer_fname,right(Customer_fname,4) from online_customer;



   ## Date and Time Functions
   # Current Date
use orders;
select current_date();
   # Current time
select current_time();
```

```
   # Now/Current_timestamp
select current_timestamp();
select current_user();
   # Date

select date('2024-06-15 10:17:13') as Date;
   # Time
select time('2024-06-15 10:17:13') as Time;
   # Year
select year(customer_creation_date) Inception_Year,
count(customer_creation_date) Customers_Aqu from online_customer
group by Inception_Year
order by Customers_Aqu desc;

select * from Order_header;

  # Month
select monthname(customer_creation_date) Month,
count(customer_creation_date) Customers_Aqu from online_customer
group by Month
order by Customers_Aqu desc;

   # Day
select day(customer_creation_date) as Day from online_customer;
   # Dayname
select
```

```
dayname(customer_creation_date) as Weekday,
count(Customer_creation_date) as Customer_Aqu
from online_customer
group by Weekday
Order by Customer_Aqu desc;

   # Dateadd
select
Customer_creation_date,
date_add(customer_creation_date, interval 10 Day) as final_date
from Online_customer;

   # Datesub
Select
Customer_creation_date,
date_sub(customer_creation_date, interval 10 Day) as final_date
from Online_customer;

    # Datediff
Select
Customer_creation_date,
date_sub(customer_creation_date, interval 10 Day) as final_date,
datediff(customer_creation_date,date_sub(customer_creation_date,
interval 10 Day)) as Inter
from Online_customer;

Select
```

```
Customer_creation_date,
date_sub(customer_creation_date, interval 10 Day) as final_date,
-round(datediff(customer_creation_date,current_date()) / 365, 2) as Inter
from Online_customer;

select year(current_date());

## Numeric Functions
   # Abs - Absolute value of a number
select abs(-44.38) as Modu;

   # Round
select round(-44.38,1) round_up_number;
   # Ceiling/Floor
select ceiling(44.38);
select floor(44.38);
   # Sqrt
select sqrt(100);
   # Power
select power(2,3);
   # Rand
select round(rand()*10000,0) as Random_num;
   # Mod

   # Sign
   # Truncate
```

```
Select truncate(48.936547, 2);
   # Log


   # exp
select exp(2);

## Conditional Functions
use world;
   # If
select
country_name,
if(Population > 25434098,"Above_Ave","Below_Ave") as Pop_Category
from country;

select
if(Population > 25434098,"Above_Ave","Below_Ave") as Pop_Category,
count(country_name) as No_of_Countries
from country
group by Pop_Category;

select
if(Population/SurfaceArea > 650,"Denesly Pop",
if(Population/SurfaceArea > 450 and Population/SurfaceArea < 650,"Av_Pop","Low_Pop_Density")) as Pop_Den_Cat,
count(Country_name) as Num
from country
group by Pop_Den_Cat
```

```
order by Pop_Den_Cat desc;

   # Case When Operator
select Country_name, Case
when Population > 25434098 then "Above Average"
when Population < 25434098 then "Below Average"
else "Equal to Average"
end as Pop_Cat
from country;

select Country_name, Case
when Population/SurfaceArea > 650 then "Densely Pop"
when Population/SurfaceArea < 650 and Population/SurfaceArea > 450 then "Average"
else "Below Average"
end as Pop_Cat
from country;

select * from country;

   # Coalesce
use orders;
select Order_Id, Payment_mode, Coalesce(Payment_mode,"Not Available") new_payment_mode from order_header
where Payment_mode is null;

use orders;
```

```
select * from Country;

select order_id,Payment_mode,Coalesce(Payment_mode,"Not Available")
new_payment_mode

from order_header where Payment_mode is null;


select order_id, payment_mode, coalesce(payment_mode, 'Not avaliable ')
new_payment_mode from order_header where payment_mode is null;


    ## Nullif

Select Payment_mode,nullif(payment_mode,"Credit Card") from
order_header;

    ## IfNull

select Order_date, ifnull(order_date,"NA") from order_header;

select payment_mode, ifnull(payment_mode,"NA") as NPM from
order_header;

select Order_date, ifnull(Order_date,"NA") as NOD from order_header;


select order_id,Payment_mode,Coalesce(Payment_mode,"Not Available")
new_payment_mode from order_header where Payment_mode is null;


## Finding the middle charecter in a string

use orders;

select customer_fname as Customer_First_Name, case

when mod(length(customer_fname),2) = 0 then
substring(customer_fname,length(customer_fname) div 2 , 2)

else  substring(customer_fname,(length(customer_fname) div 2)+1 , 1)

end as Middle_Character from online_customer;
```

```
select length(customer_fname) div 2 from online_customer;

select Customer_fname,
case
   when (length(Customer_fname)/2) % 2 = 0 Then
substring(Customer_fname,(length(Customer_fname)/2),2)
   else substring(Customer_fname,(length(Customer_fname)/2),1)
end as middle_characters
from online_customer;

use hr_emp;
select * from employees;
Create Database hr_emp;
use world;
select * from country;

## Window Functions
#Rank
select rank() over(order by GNP desc) as Ranking,
Country_name, continent, GNP
from country;

select rank() over(order by GNP/population desc) as Ranking,
Country_name, continent, GNP, round(GNP*1000/Population,2) as
GNP_Per_Cap
from country
```

```
limit 5;


#Denserank

select dense_rank() over(partition by Continent,Region order by GNP/population desc,GNP desc) as Dense_Ranking,

Country_name, Region, continent, GNP, round(GNP*1000/Population,2) as GNP_Per_Cap

from country;


#Ntile

select country_name,continent,region,LifeExpectancy,

ntile(3) over(order by LifeExpectancy desc) as Clusters

from country;


#Sum Over

use world;

select country_name, continent, GNP, sum(GNP) over(order by GNP desc) as Cum_GNP

from country;


#Avg Over


use world;

select country_name, continent, GNP, round(avg(GNP) over(partition by continent order by GNP desc),2) as Avg_GNP

from country;
```

```
#Lag and Lead Functions

use world;

select country_name, continent, GNP, lag(GNP,2) over(order by GNP desc) as Pre_GNP,

(lag(GNP,2) over(order by GNP desc) - GNP) as GNP_Change

from country;


select country_name, continent, GNP, lead(GNP,2) over(order by GNP desc) as Post_GNP,

(lead(GNP,2) over(order by GNP desc) - GNP) as GNP_Change

from country;


# Joins in SQL


use ineuron;

create Table Demographic (Id Varchar(20), age int, gender char(1), salary int, city varchar(20));

create Table Professional (Id Varchar(20), Name Varchar(20), Dept varchar(10), Manager varchar(20));


Insert into Demographic

values
(201,25,"M",20000,"Beng"),(202,32,"F",25000,"Mum"),(203,40,"F",20000,"
Mum"),(204,23,"M",22000,"Che");


Insert into Professional

values (202,"Shree","Mar","Ram"),
(204,"Ram","Fin","Atul"),(211,"Priya","HR","Raj"),(212,"Ritu","Ops","Amar")
;
```

```
select * from Professional;
    ## Inner Join

select demographic.id,Age,Gender,Dept,Manager
from demographic
inner join Professional
on demographic.id = Professional.id;

select d.id,Age,Gender,Dept,Manager
from demographic d
inner join Professional p
on d.id = P.id;

   ## Outer Join
select * from demographic;
select * from professional;
use ineuron;

## Left Outer

select d.id,Age,Gender,Dept,Manager
from demographic d
Left join professional p
on d.id = P.id;
```

```
select p.id,Age,Gender,Dept,Manager
from professional p
Left join demographic d
on d.id = P.id;


## Right Outer

select p.id,Age,Dept,Manager,Salary
from demographic d
right join professional p
on d.id = P.id;

## Joining 3 tables

use world;
select * from city;
select * from country;
select * from countrylanguage;

use world;
select id as City_ID,Name as City_Name,
District,ci.Population,Country_name,Continent,Region,count(Language)
No_of_languages
from country c
right join city ci
on ci.countrycode = c.code
```

```
left join countrylanguage cl
on ci.Countrycode = cl.countrycode
group by City_id;

# Full Outer Join

select d.id,Age,Gender,Dept,Salary from demographic d
left join
professional p
on d.id = p.id
union
select p.id,age,Gender,Dept,Salary
from demographic d
right join professional p
on d.id = p.id;

# Subqueries
    ## Use Cases of Subqueries
use hr_emp;
select * from employees;

## Fetch all data of the department where mangaer id is 186

select * from employees
where department_id = (select department_id from employees
where manager_id = 186); ## Single row Subquery
```

```
select department_id from employees
where manager_id = 186;

## Fetch the data of all employees where manager_id are either 50,70,186

select * from employees
where department_id in (select distinct department_id from employees
where manager_id in (60,80,186)); ## Multi Row Subquery


## Provide the details of all employees earning equal to Purv.
select * from employees
where salary = (select salary from employees where first_name = 'Purv');

## Provide the details of all employees earning equal to Ajay.


## fetch data of all employees where salary > Overall average salary;

select * from employees
where salary > (select avg(salary) from employees);

## Fetch the Data of all employees who are earning more than vaerage in their respective departments.
```

```
select * from employees e where salary >

(Select avg(t.salary) from employees t where t.department_id = e.department_id);
```

```
use hr_emp;

select * from employees a where salary > (select avg(salary) from employees b

group by b.department_id having b.department_id=a.department_id);
```

```
select * from employees e

join ( select department_id, avg(salary) as avg_salary from employees

   group by department_id) as dept_avg on e.department_id = dept_avg.department_id

where e.salary > dept_avg.avg_salary;
```

```
select salary from employees

where first_name = 'Ajay' and last_name = 'Mishra';
```

## Database Objects

# Views

- View is nothing but the virtual table which stores the table.
- If we made any changes in the database then changes being reflected in view also automatically.
- It will take less memory as compare to table.

- If one view is already exist then if we want to replace that view with another view but it has same name so if you use create syntax then it will give error so at that time use
    - create or replace <view-name> as <query>

```
select * from country;

select * from city;

select * from countrylanguage;



create view City_Summary as

select id as City_ID,Name as City_Name, District,ci.Population,Country_name,Continent,Region,count(Language) No_of_languages

from country c

right join city ci

on ci.countrycode = c.code

left join countrylanguage cl

on ci.Countrycode = cl.countrycode

group by City_id;


select * from city_summary;


use hr_emp;


create view emp_more_than_avg_salary as

select * from employees

where salary > (select avg(salary) from employees);
```

```
select first_name,last_name,department_id,manager_id from
emp_more_than_avg_salary;
```

```
drop view emp_more_than_avg_salary;
```

```
use world;

create view Gistofcountries as

select country_name, Continent, Region, SurfaceArea, IndepYear,
Population, Lifeexpectancy, GNP

from country;


select * from Gistofcountries;


update Gistofcountries

set Lifeexpectancy = 76.3

where Name = "Angola";


SET SQL_SAFE_UPDATES= 0;
```

```
use ineuron;



create view Common_Data as
select d.id, Name, Age, Gender, Salary, city, Dept, Manager
from demographic d
join professional p
on d.id = p.id;

select * from Common_data;

create or replace view Common_Data as
select d.id, Name, Age, Gender, Salary, city, Dept, Manager
from demographic d, professional p
where d.id = p.id;

select * from Common_data;
use world;
```

# Stored Procedure

- Store procedure is used to store the query and use it directly by calling the stored procedure .
- Here we use Delimiter bcz in procedure we store the query so it is one query and procedure is another query so to differentiate we use delimiter in procedures.
- And if we will not use delimiter and we write like this
    - create procedure AllCityData()
      begin
      select * from city;

end ;
then it is giving issue or error.

- View takes more space and time than store procedure.
- To see the query which is in stored procedure.
  - ❖ Right click on the stored procedure
  - ❖ Click on copy to clipboard
  - ❖ Click on select statement
  - ❖ Then control + V

```
Delimiter //
create procedure AllCityData()
begin
select * from city;
end //
Delimiter ;

call AllCityData;

Delimiter //
create procedure allthecitydata()
begin
select * from city;
end //
Delimiter ;

Delimiter //
create procedure Continent_LE()
Begin
```

```
select Continent, round(Avg(lifeExpectancy),2) as Avg_LE
from country
group by continent
order by Avg_LE desc;
end //
Delimiter ;

Delimiter //

create procedure City_Population_Data(In First_Country varchar(20), Second_Country Varchar(20))
Begin
select Name,Country_name, ci.Population
from city ci
left join country co
on ci.countrycode = co.code
where co.country_name in (First_Country,Second_Country);
end //
delimiter ;

call City_Population_Data('Italy','India');
```

### CTE

- In this we create the virtual table which is not store and we do some operations with that table.

```
use hr_emp;
```

```
➔ with Departmentwiseavgsalary as(
select department_id,count(manager_id) as No_of_Managers,round(avg(salary),0) as Avg_Salary
from employees
group by department_id
order by Avg_salary desc)
select * from Departmentwiseavgsalary;
```

```
➔ use hr_emp;
with Deptavgsalary as(
select department_id,avg(salary) as Avg_Salary
from employees
group by department_id)
select first_name,last_name,e.department_id,salary
from employees e
left join
Deptavgsalary das
on e.department_id = das.department_id
where e.salary > das.Avg_salary;

use ineuron;
```

```
➔ use world;
Delimiter //
create procedure LifeExp()
```

```
Begin
    select Continent, avg(lifeexpectancy) as AvglifeExp
    from country
    group by continent;
End //
Delimiter ;

call lifeexp();

➔ Delimiter //
create procedure LifeExp_cont(IN Continent_name varchar(20))
Begin
    select Continent, avg(lifeexpectancy) as AvglifeExp
    from country
    group by continent
   having continent = Continent_name;
End //
Delimiter ;

call lifeExp_cont("Africa");

➔ select Continent, avg(lifeexpectancy) as AvglifeExp
    from country
    group by continent
   having continent = "Asia";
```

```
➔ use hr_emp;
select * from employees;
```

```
➔ Delimiter //
create procedure No_of_Emp(IN Startdate date,In Enddate date)
Begin
    select count(employee_id) from employees
   where hire_date between startdate and enddate;
end //
Delimiter ;
```

```
➔ select count(employee_id) from employees
   where hire_date between '16-06-1977' and '23-01-1983';
```

```
call No_of_Emp('16-06-1977','23-01-1983')
```

```
➔ Delimiter //
create procedure Total_Emp(IN Department Varchar(20))
Begin
    select department_id, count(employee_id) as No_of_Emp, Avg(Salary) as Avg_Salary from employees
   where department_id = Department;
end //

Delimiter ;

Call Total_Emp(50);
```

```
## Grant and Revoke Permissions

use hr_emp;
Grant select,insert,create,update,delete,drop
on employees
to 'newuser';

use hr_emp;
Revoke insert,create,update,delete,drop
on employees
from 'newuser';



Create table jshdbjdsh(Name varchar(30));
commit;
Insert Into jshdbjdsh values ("ABC");
Commit;
Insert Into jshdbjdsh values ("PQR");
Alter table jshdbjdsh change column Name  Full_name char(20);
rollback;

use hr_emp;
```

DRIVE LINK FOR TABLES

https://drive.google.com/drive/u/1/folders/1S4tXUL58l-AjnJI5gI7Khro8GOWvU6bY

PROMPT TO CREATE THE ASSIGNMENT OR PROJECTS

➔ So I have a database consisting of the following tables

city, country and country language.

The city table consists of the following columns

ID int

Name text

CountryCode text

District text

Population int

The country table consists of the following columns

Code text

Country_name text

Continent text

Region text

SurfaceArea double

IndepYear int

Population int

LifeExpectancy double

GNP double

GNPOld double

LocalName text

GovernmentForm text

HeadOfState text

Capital int

The country language table consist of the following columns

CountryCode text

Language text

IsOfficial text

Percentage double

Please create a project on sequel questions including the following sub topics

1. Intermediate level queries
2. Common table expressions
3. Joins in sql
4. Sub queries in SQL
5. Stored procedures and views in SQL
6. Commonly asked interview questions in sequel

STEPS=

1)Go to the speech notes it will convert speech into text then go to the chatgpt and paste it that prompt