1. Over Clouse

Used when we work with

1. **Aggregated FN**
2. **Windows FN (agaaa parhain gaa)**

Has 2 Components

**🡪 1. Order By (asc ,desc)**

**🡪 2. Partition By (divides values in Parts) \_- Table ko Similar Values me Break krtaa ha**

when we used simple aggrigate function its return only group by values. to get All Rows Values sum we use sum(sum()) gives error . Over() clouse give forece to get this

1. **Simple Aggregate Vs ( Aggregate into Aggrigate ) Over Clouse**

CREATE TABLE sales\_table (

id INT PRIMARY KEY,

region VARCHAR(50),

sales DECIMAL(10, 2)

);

INSERT INTO sales\_table (id, region, sales)

VALUES

(1, 'North', 5000.50), (2, 'North', 7500.75),(3, 'South', 10000.00),

(4, 'South', 12500.25),(5, 'East', 8000.00),(6, 'East', 9500.50),

(7, 'West', 6000.00), (8, 'West', 7000.00);

---\_\_\_\_\_\_\_\_ 1. Simple Aggregate Group by---- \_\_\_\_\_\_\_\_\_\_\_\_

--- Region Bas Values

select

region,

sum(sales) totalsales

from sales\_table

group by region

---\_\_\_\_\_\_\_\_ 2. Aggregate into Aggregate ---- \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

-- All Regions Values Sum

select

region,

sum(sales) totalsalesPerRegion,

sum(sum(sales)) over() AllRegionSales

from sales\_table

group by region

---\_\_\_\_\_\_\_\_ 3. Calculate OverAll Averge of Per Region ---- \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

-- Average ka liyaa hum Total Region ke Value Required hoti ha

select

region,

sum(sales) totalsalesPerRegion,

sum(sum(sales)) over() AllRegionSales,

(sum(sales)\*100)/sum(sum(sales)) over() PerAvergeOfRegion

from sales\_table

group by region

--\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Without OVER() clouse \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

select

region,

sum(sales) overallRegionSale,

avg(sales) AS AverPerRegion,

s.total\_sales\_all\_regions,

s.Total\_Sales\_AVERG

from sales\_table

cross join (

SELECT

SUM(sales) AS total\_sales\_all\_regions,

Avg(sales) as Total\_Sales\_AVERG

FROM sales\_table

) s

group by region , s.total\_sales\_all\_regions,s.Total\_Sales\_AVERG

1. Over Clouse + Partition

create table persons(

id int primary key identity,

name varchar(25),

gender varchar(25),

age int,

salary int

)

insert into persons values ('Zain','Male',26,1400),('Anum','Female',24,17000),

('Amir','Male',25,19000),('Naima','Female',27,22000),('Saima','Female',24,15000)

,('Osama','Male',28,18000),('Afzal','Male',24,19000),('Sidra','Female',22,16000)

,('Farhan','Male',24,19000),('Muqaddas','Female',24,17000),('Irfan','Male',24,12000)

--\_\_\_\_\_\_\_\_ 1. Simple Group \_\_\_\_\_\_\_\_

select

gender,

count(gender)

from persons

group by gender

-- error -- Mujhaa harr Record kaa Samnaaaa Total Male , Female show krnaa han

select

name,

gender,

count(gender)

from persons

group by name,gender

--\_\_\_\_\_\_\_\_ 2. 1st Way (sub\_Query) \_\_\_\_\_\_\_\_ we used in Carts this type of Values

select name,persons.gender,totalGender from persons

inner join

(

select

gender,

count(gender) totalGender

from persons

group by gender

) genders

on genders.gender = persons.gender

--\_\_\_\_\_\_\_\_ 3. \_\_\_\_\_\_\_\_

select

name,

persons.gender,

genders.totalGender,

g.totalGender

from persons

inner join

(

select

gender,

count(gender) totalGender

from persons

group by gender

) genders

on genders.gender = persons.gender

cross join (

select

count(gender) totalGender

from persons

) g

--Max,Min,Avg salary of groupby Gender (2 male,female)

select name,persons.gender,totalGender,maxSalary,minSalary,avargeSalary from persons

inner join

(

select

gender,

count(gender) totalGender,

Max(salary) maxSalary,

Min(salary) minSalary,

Avg(salary) avargeSalary

from persons

group by gender

) genders

on genders.gender = persons.gender

--\_\_\_\_\_\_\_\_ 4. 2nd Way (Over Clouse) \_\_\_\_\_\_\_\_ we used in Carts this type of Values

select

name,

gender,

salary,

count(gender) over(partition by gender) genderTotal,

count(count(gender)) over() genderTotal

from persons

--same work

select

name,

gender,

salary,

count(gender) over(partition by gender) genderTotal,

Max(salary) over(partition by gender) MaxSalary,

Min(salary) over(partition by gender) MaxSalary

from persons

1. Over Clouse + Order by

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ order by \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CREATE TABLE orders (

order\_id INT PRIMARY KEY,

customer\_id INT,

order\_date DATE,

order\_amount DECIMAL(10,2)

);

INSERT INTO orders (order\_id, customer\_id, order\_date, order\_amount)

VALUES (1, 1, '2022-01-01', 100.00),

(2, 1, '2022-02-01', 200.00),

(3, 2, '2022-01-15', 150.00),

(4, 3, '2022-02-15', 75.00),

(5, 2, '2022-02-28', 225.00);

SELECT customer\_id, order\_date, order\_amount,

ROW\_NUMBER() OVER(ORDER BY order\_date asc) AS row\_num

FROM orders;

SELECT customer\_id, order\_date, order\_amount,

ROW\_NUMBER() OVER(ORDER BY order\_date desc) AS row\_num

FROM orders;

**//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Partition + order by \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

SELECT customer\_id, order\_date, order\_amount,

SUM(order\_amount) OVER(PARTITION BY customer\_id ORDER BY order\_date ASC) AS running\_total

FROM orders;

1. Different Group By vs Over Clause

**Group By :** give only aggregated values for the column that are not Include in Group by.

**Over Clouse :** you can retrieve aggregated and Non-Aggregated Values