**1. GROUP BY clause**

The GROUP BY clause is used to group rows in a SELECT statement based on the values in one or more columns.

* Aggregate function Required in Select
* Group by whaan lgtaa ha 🡪 🡪 jhan DATA repeat ho rhaa ho

**Syntax:**

**SELECT column1, column2, ..., Aggregate\_function(column)**

**FROM table\_name**

**GROUP BY column1, column2, ...;**

***Basic Group By***

**Example 1 : Finding Overall Sallery par City**

-- location kaa hisabb saa total saalery chtaa hon

insert into customer values ('ahamd',45000,'Karachi')

insert into customer values ('akram',25000,'Karachi')

insert into customer values ('Akhtar',20000,'Karachi')

insert into customer values ('Huzafa',500000,'Karachi')

insert into customer values ('hafiiz',155000,'Karachi')

insert into customer values ('Malik',200000,'Quita')

insert into customer values ('Jasmeen',15000,'Islamabad')

insert into customer values ('Huzarnain',23000,'Islamabad')

select \* from customer

select Sum(salary) 'totalSalryParCity' , Location , count(\*) 'totalEmployee' from customer

group by Location

**Example 2 : Finding Over All Sales of Product**

create table products (

id int identity ,

product\_name varchar(25),

buy\_amount varchar(25),

sale\_amount int

)

------- 1st day ------- Sales

insert into products values ('apple',40,45),

('Orange',30,45),

('watermelon',110,120)

select product\_name, buy\_amount, sale\_amount from products

------- 2nd day ------- Sales

insert into products values ('apple',50,55),

('Orange',25,35),

('watermelon',95,100)

select product\_name,buy\_amount,sale\_amount from products

---------- Over All Sales of products ?? ------- is kaa lyaa hum Group By use krtaa han

select product\_name,sum(sale\_amount) as 'over all sale' from products group by product\_name

------------------ Compairing Both Result -------------------

select product\_name,buy\_amount,sale\_amount from products

select product\_name,sum(sale\_amount) as 'over all sale' from products group by product\_name

----- Profite -------

select

product\_name,

sum(buy\_amount) buyPrice ,

sum(sale\_amount) salePrice,

(sum(sale\_amount)-sum(buy\_amount)) profite

from products

group by product\_name

-----\_\_\_\_\_\_\_\_\_\_\_\_Benefits \_\_\_\_\_\_\_\_\_\_

--1. Hamara Product Repeat nhin hon ga

--2. Over all 2 days Sales Price mil gy geee

**Example 3 : Finding Over All Sales of Product**

create table customer\_order(

id int primary key identity,

customer\_id int not null,

order\_date date,

order\_total float,

order\_status varchar(25)

)

insert into customer\_order values (101,'2022-01-5',100,'Completed'),(102,'2022-01-5',50,'Completed'),

(101,'2022-02-5',200,'Completed'),(103,'2022-02-5',150,'Completed'),

(102,'2022-03-5',70,'Panding'),(101,'2022-04-5',50,'Completed'),

(102,'2021-04-5',200,'Cancelled'),(103,'2021-05-5',150,'Completed'),

(101,'2022-06-5',300,'Completed'),(103,'2022-06-5',400,'Completed')

**Example 4 : Finding Over All Total\_Order of Customer by moth, year wise**

create table customer\_order(

id int primary key identity,

customer\_id int not null,

order\_date date,

order\_total float,

order\_status varchar(25)

)

insert into customer\_order values (101,'2022-01-5',100,'Completed'),(102,'2022-01-5',50,'Completed'),

(101,'2022-02-5',200,'Completed'),(103,'2022-02-5',150,'Completed'),

(102,'2022-03-5',70,'Panding'),(101,'2022-04-5',50,'Completed'),

(102,'2021-04-5',200,'Cancelled'),(103,'2021-05-5',150,'Completed'),

(101,'2022-06-5',300,'Completed'),(103,'2022-06-5',400,'Completed')

select

year(order\_date) order\_year,

month(order\_date) order\_month,

sum(order\_total) total\_Order

--order\_status

from customer\_order

group by Year(order\_date),month(order\_date)

**Example 5 : Total\_Order of Customer by moth, year \_\_ (show , Order\_Status)**

select

year(order\_date) order\_year,

month(order\_date) order\_month,

sum(order\_total) total\_Order,

order\_status

--order\_status

from customer\_order

group by Year(order\_date),month(order\_date) , order\_status

order by Year(order\_date) asc

***Mid Level Group By + Where + Order by***

**Example 1 : Show those Customer \_ has Completed Orders \_ group by Year (No Month)**

SELECT

customer\_id,

YEAR(order\_date) AS order\_year,

SUM(order\_total) AS total\_order\_amount

FROM customer\_order

WHERE order\_status = 'Completed'

GROUP BY customer\_id, YEAR(order\_date)

order by customer\_id

**Example 2 : Show those Customer \_ has Completed Orders \_ group by Month (No Year)**

SELECT

customer\_id,

month(order\_date) AS order\_month,

SUM(order\_total) AS total\_order,

order\_status

FROM customer\_order

WHERE order\_status = 'Completed'

GROUP BY customer\_id, Month(order\_date) ,order\_status

order by customer\_id

**Example 3 : Show those Customer \_ has Panding Orders \_ group by Month (No Year)**

SELECT

customer\_id,

month(order\_date) AS order\_month,

SUM(order\_total) AS total\_order,

order\_status

FROM customer\_order

WHERE order\_status = 'Panding'

GROUP BY customer\_id, Month(order\_date) ,order\_status

order by customer\_id

**Example 4 : Show those Customer \_ has Cancel Orders \_ group by Year (No month)**

insert into customer\_order values (101,'2022-01-5',5,'Cancelled'),

(101,'2021-04-5',5,'Cancelled'),(103,'2021-09-5',3,'Cancelled'),

(101,'2022-07-5',1,'Cancelled'),(103,'2022-12-5',1,'Cancelled')

SELECT

customer\_id,

Year(order\_date) AS order\_Year,

SUM(order\_total) AS total\_order,

order\_status

FROM customer\_order

WHERE order\_status = 'Cancelled'

GROUP BY customer\_id, Year(order\_date) ,order\_status

order by customer\_id

**2. Having Clouse**

--> we can get (conditional) Data of (Group By) by using where clouse before Grouping.

(its give Conditonal data before Grouping)

--> not After Grouping ...

The HAVING clause is used to filter the data returned by a SELECT statement after it has been grouped by the GROUP BY clause. The basic syntax of a HAVING clause is:

**Where :** filter Records before grouping

**Having :** filter Records After grouping

**Group by ka baad jo data ay gaa --🡪 usaa fileter krnaa ka lya**

SELECT column1, column2, ..., aggregate\_function(column)

FROM table\_name

GROUP BY column1, column2, ...

HAVING condition;

**Example 1 : (Gift to Customer)**

**wo Customer Select kro -- (Completed) jin ka Grouping ka bad - Total Order > 100**

SELECT

YEAR(order\_date) AS order\_year,

MONTH(order\_date) AS order\_month,

customer\_id,

SUM(order\_total) AS total\_order

FROM customer\_order

GROUP BY YEAR(order\_date), MONTH(order\_date), customer\_id

HAVING SUM(order\_total) > 100;

**Example 2 : (Deactivate User) --- Select Those Users - Jinhon na 2 year saa Account Use nhin kraa**

create table Users(

id int,

userName varchar(25),

last\_login\_data date

)

insert into users values (1,'Saqib','2020-01-01'),

(1,'Saqib','2019-01-01'),(2,'Sara','2020-01-01'),

(2,'Sara','2019-01-01'),(2,'Sara','2020-01-01'),

(3, 'babar','2018-01-01'),(3,'babar','2017-01-01'),

(3,'babar','2019-01-01'), (4,'Ahmad','2021-01-01'),

(4,'Ahmad','2012-01-01'),(4,'Ahmad','2022-01-01'),

(5,'Noman','2022-01-01'),(5,'Noman','2022-04-08')

**---- 1. Get Last Date of Login ---**

select

userName,

Max(last\_login\_date) 'last activity' -- last date

from users

group by userName

**---- 2. Get Last Date of Login ---**

select

userName,

MIn (last\_login\_date) '1st activity'

from users

group by userName

**---- 3. Total Activities ---**

select

userName,

Count(last\_login\_date) 'Total Actitivies' --

from users

group by userName

**---- 4. Select those User .. having 2 years ago Last Activity ---**

select

userName,

Max(last\_login\_date) 'last activity 2 year Ago' -- last date

from users

group by userName

having Max(last\_login\_date) < DateAdd(Year,-2,GETDATE()) -- select user with last activity 2 years ago

select GETDATE();

select DateAdd(year,2,GETDATE())

select DateAdd(Month,2,GETDATE())

-- subtract 2 year from crant year --

select DateAdd(Year,-2,GETDATE())