Assignment on string group by and having

Course Title: Database Management System

Course Code: CSE-2203

Name: Md. Waliul Islam Rayhan
ID: B190305034

SQL GROUP BY

In SQL, The Group By statement is used for organizing similar data into groups. The data is further organized with the help of equivalent function. It means, if different rows in a precise column have the same values, it will arrange those rows in a group.

1. The **SELECT** statement is used with the **GROUP BY** clause in the **SQL** query.
2. **WHERE** clause is placed before the **GROUP BY** clause in **SQL**.
3. **ORDER BY** clause is placed after the **GROUP BY** clause in **SQL**.

Syntax

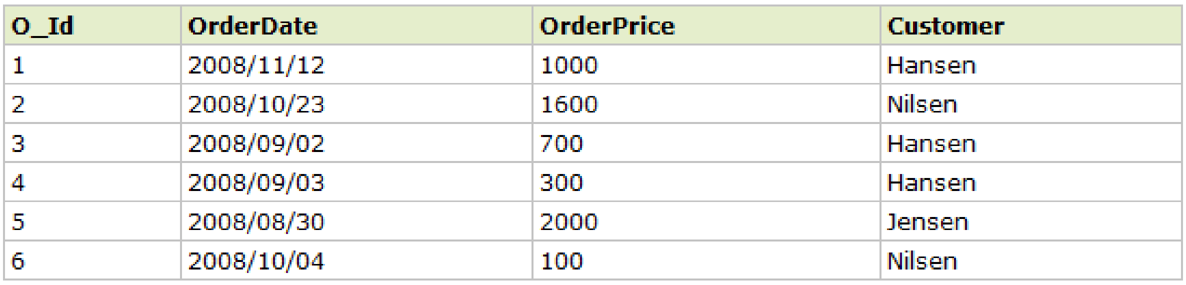
SELECT column_1, function_name(column_2)

FROM table_name

WHERE condition

GROUP BY column_name;

Sample Table:



O_Id	OrderDate	OrderPrice	Customer
1	2008/11/12	1000	Hansen
2	2008/10/23	1600	Nilsen
3	2008/09/02	700	Hansen
4	2008/09/03	300	Hansen
5	2008/08/30	2000	Jensen
6	2008/10/04	100	Nilsen

Example:

The GROUP BY statement is often used with aggregate functions (COUNT(), MAX(), MIN(), SUM(), AVG()) to group the result-set by one or more columns.

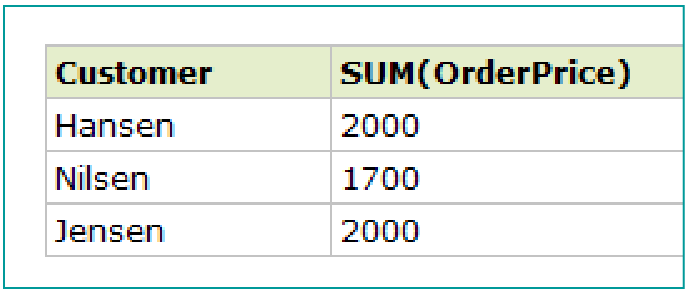
Group By single column:

Group By single column is used to place all the rows with the same value. These values are of that specified column in one group. It signifies that all rows will put an equal amount through a single column, which is of one appropriate column in one group.

Consider the below query:

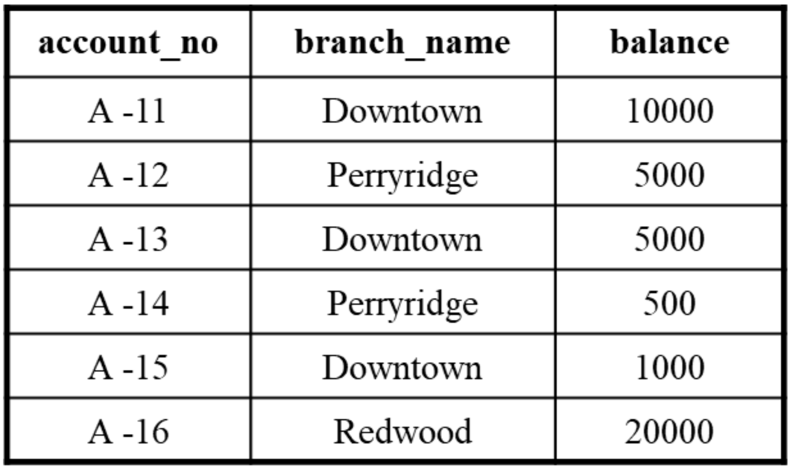
```
SELECT Customer, SUM(OrderPrice)
FROM Orders
GROUP BY Customer
```

The output of the query is:



Customer	SUM(OrderPrice)
Hansen	2000
Nilsen	1700
Jensen	2000

Input: Account Relation



account_no	branch_name	balance
A -11	Downtown	10000
A -12	Perryridge	5000
A -13	Downtown	5000
A -14	Perryridge	500
A -15	Downtown	1000
A -16	Redwood	20000

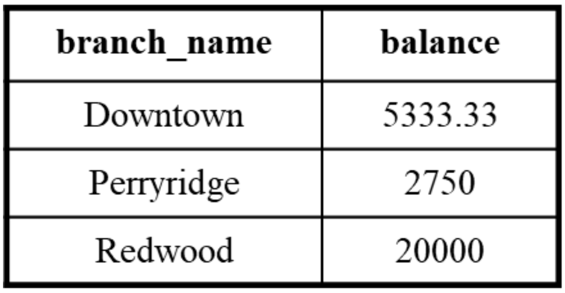
Find the average account balance at each branch:

```
SELECT branch_name, avg(balance)

FROM account

GROUP BY branch_name
```

Output:



branch_name	balance
Downtown	5333.33
Perryridge	2750
Redwood	20000

A new Table:

Employee

S.no	Name	AGE	Salary
1	**John**	**24**	**25000**
2	**Nick**	22	22000
3	**Amara**	25	15000
4	**Nick**	22	22000
5	**John**	24	25000

Continue..

Student

SUBJECT	YEAR	NAME
C language	**2**	**John**
C language	**2**	Ginny
C language	**2**	Jasmeen
C language	**3**	Nick
C language	**3**	Amara
Java	**1**	Sifa
Java	**1**	dolly

Consider the below query:

SELECT SUBJECT, YEAR, Count (*)

FROM Student

Group BY SUBJECT, YEAR;

Output:

SUBJECT	YEAR	Count
C language	2	3
C language	3	2
Java	1	2

HAVING Clause

WHERE clause is used for deciding purpose. It is used to place conditions on the columns to determine the part of the last result-set of the group. Here, we are not required to use the combined functions like COUNT (), SUM (), etc. with the WHERE clause. After that, we need to use a HAVING clause.

Having clause Syntax:

SELECT column_name, aggregate_function(column_name)
FROM table_name
WHERE column_name operator value
GROUP BY column_name
HAVING aggregate_function(column_name) operator value

Example:

SELECT NAME, **SUM**(SALARY) FROM Employee

GROUP BY NAME

HAVING SUM(SALARY)>23000;

Output:

Name	SUM(SALARY)
John	50000

According to the above output, only one name in the NAME column has been listed in the result because there is only one data in the database whose sum of salary is more than 50000.

It should be placed on groups, not on the columns.

Points:

- The **GROUP BY** Clause is used to group the rows, which have the same values.
- The **SELECT** statement in SQL is used with the GROUP BY clause.
- In the **Group BY** clause, the SELECT statement can use **constants, aggregate functions, expressions,** and **column names**.
- The **GROUP BY** Clause is called when the HAVING clause is used to reduce the results.