SQL GROUP BY Clause

What is the SQL Group by Clause?

The **SQL GROUP BY clause** is used to group the result set based on common value present in the result set. The grouping of result set is done after all records are retrieved from tables. The GROUP BY clause can be used with the conjunction of SELECT query to make a group of rows based on the values of a specific column or expression. The SQL AGGREGATE function can be used to get summary information for every group and these are applied to individual group. The main importance of GROUP BY clause is to summarizing of data from database, so these commands are frquently used in the reporting queries. Only one result is returned after the executing the GROUP BY clause & these queries are called as grouped queries.

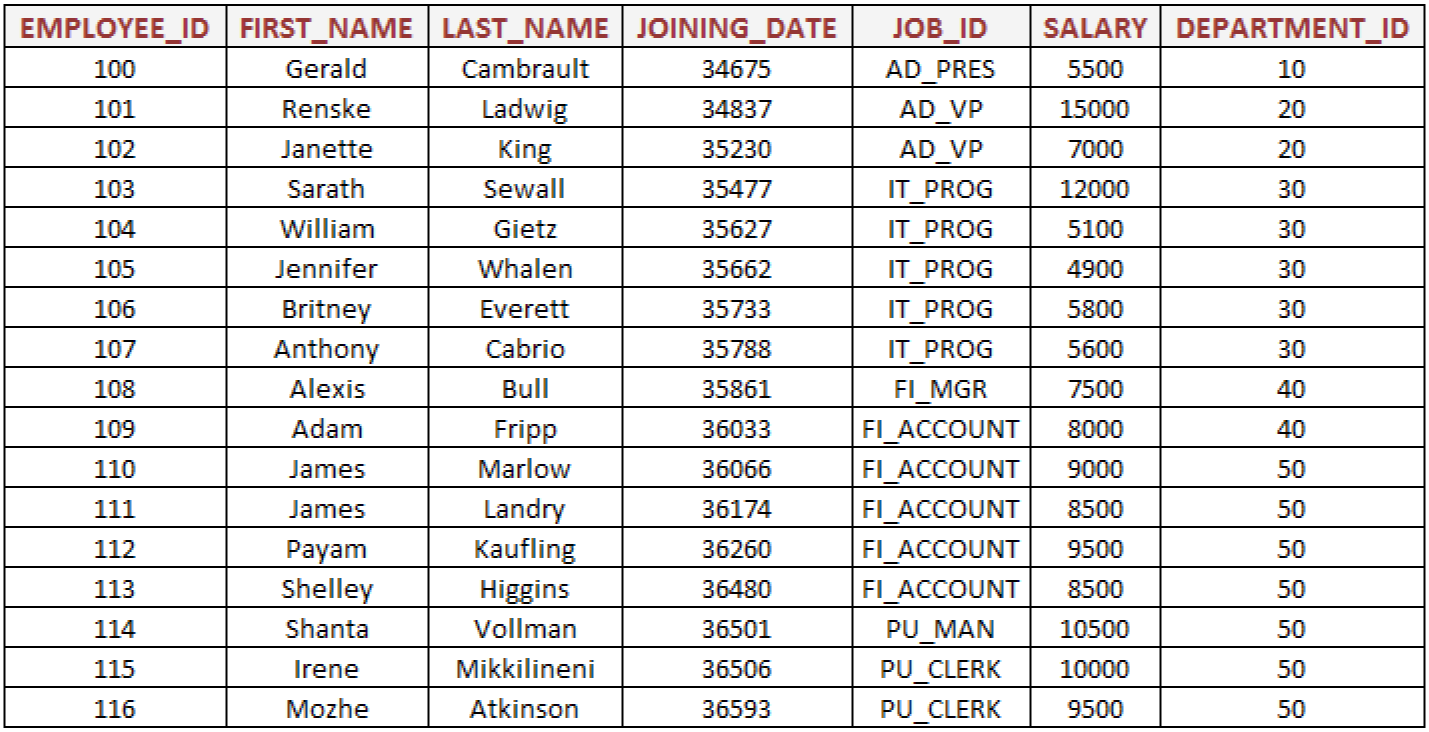
You can use the WHERE clause along with GROUP BY clause, but WHERE condition is used to limit the records from the tables & GROUP BY clause is use to group the retrieved result set based on column name specified. If you want to filter the grouped result set then we can use the HAVING clause. This is used after the GROUP BY clause.

SQL GROUP BY Clause Syntax:

```
SELECT * from Table_Name
GROUP BY Column_Name1,
[Column_Name2,….]
[HAVING CONDITION];
```

- **[SELECT * from Table_Name]** – This is regular SELECT command to get the records from database.
- **GROUP BY Column_Name1** – This is clause is used to group the result set based column name specified. Here you can specify one or more column names.
- **[Column_Name2,….]** – This is optional. In the GROUP BY clause, one column name is mandatory on which you have to group the result set. If you specify more than one column name then result set first group on first column value & then next column(s).
- **[HAVING CONDITION] –** This Clause used to filter out the result data set based on “CONDITION”. You can specify any condition using HAVING clause.

In this tutorial we have created a sample Employees table on which we are executing the different SQL GROUP BY queries,



EMPLOYEE_ID	FIRST_NAME	LAST_NAME	JOINING_DATE	JOB_ID	SALARY	DEPARTMENT_ID
100	Gerald	Cambrault	34675	AD_PRES	5500	10
101	Renske	Ladwig	34837	AD_VP	15000	20
102	Janette	King	35230	AD_VP	7000	20
103	Sarath	Sewall	35477	IT_PROG	12000	30
104	William	Gietz	35627	IT_PROG	5100	30
105	Jennifer	Whalen	35662	IT_PROG	4900	30
106	Britney	Everett	35733	IT_PROG	5800	30
107	Anthony	Cabrio	35788	IT_PROG	5600	30
108	Alexis	Bull	35861	FI_MGR	7500	40
109	Adam	Fripp	36033	FI_ACCOUNT	8000	40
110	James	Marlow	36066	FI_ACCOUNT	9000	50
111	James	Landry	36174	FI_ACCOUNT	8500	50
112	Payam	Kaufling	36260	FI_ACCOUNT	9500	50
113	Shelley	Higgins	36480	FI_ACCOUNT	8500	50
114	Shanta	Vollman	36501	PU_MAN	10500	50
115	Irene	Mikkilineni	36506	PU_CLERK	10000	50
116	Mozhe	Atkinson	36593	PU_CLERK	9500	50

Using GROUP BY with Aggregate Functions:

The use of **AGGREGATE functions** is very much powerful when we use then along with GROUP BY clause. In the SQL GROUP BY queries, the most of times GROUP BY clauses are used with AGGREGATE functions.

Here we are going to see different GROUP BY Clause example with different combinations of the AGGREGATE Functions:

1. SQL GROUP BY with SUM() function
2. SQL GROUP BY with COUNT() function
3. SQL GROUP BY with COUNT() and SUM()
4. SQL GROUP BY on more than one column
5. SQL GROUP BY with WHERE clause
6. SQL GROUP BY with HAVING clause

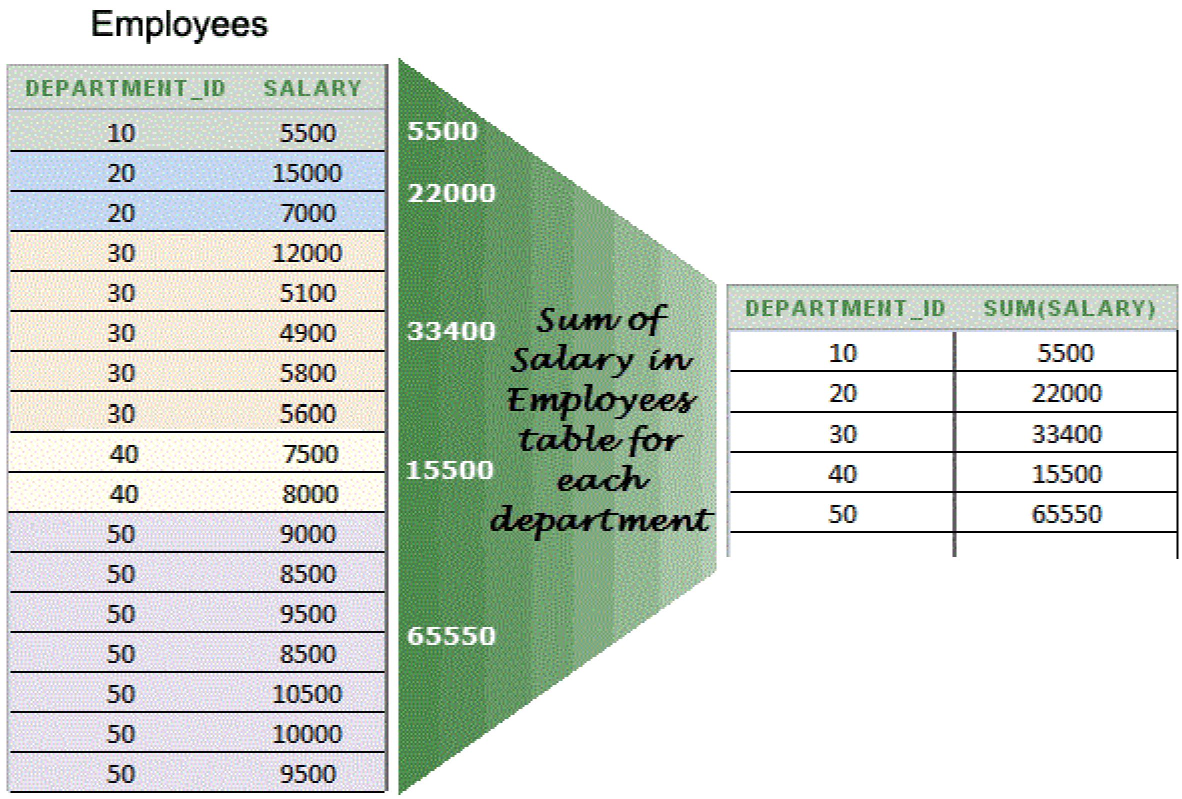
1) SQL GROUP BY with SUM() function:

Suppose user want to get the sum of Salary paid to each department, so in the Employees table we have column called DEPARTMENT_ID & SALARY columns. So we can group the result set using DEPARTMENT_ID & we will use the SUM Aggregate function to get sum of salary by each department.

```
SELECT DEPARTMENT_ID,
SUM(SALARY)  as Total_Salary
```

```
FROM Employees
GROUP BY DEPARTMENT_ID;
```

Following is the result upon executing the above*SQL GROUP BY query*,



Employees

DEPARTMENT_ID	SALARY
10	5500
20	15000
20	7000
30	12000
30	5100
30	4900
30	5800
30	5600
40	7500
40	8000
50	9000
50	8500
50	9500
50	8500
50	10500
50	10000
50	9500

DEPARTMENT_ID	SUM(SALARY)
10	5500
20	22000
30	33400
40	15500
50	65550

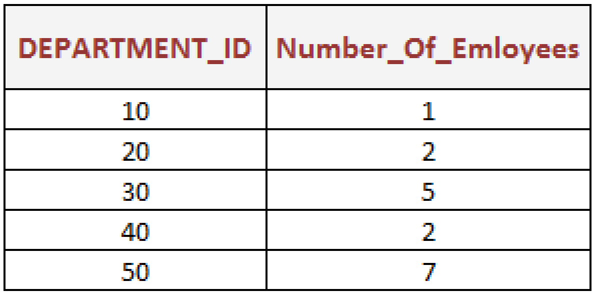
2) SQL GROUP BY with COUNT() function:

Suppose user wants to get the number employees in each department, so in the Employees table we have column called DEPARTMENT_ID. So we can group the result set using DEPARTMENT_ID & we will use the COUNT Aggregate function to get number of employees by each department.

```
SELECT DEPARTMENT_ID,
COUNT(*)  as Number_Of_Emloyees
FROM Employees
```

```
GROUP BY DEPARTMENT_ID;
```

Following is the result upon executing the above*SQL GROUP BY query*,



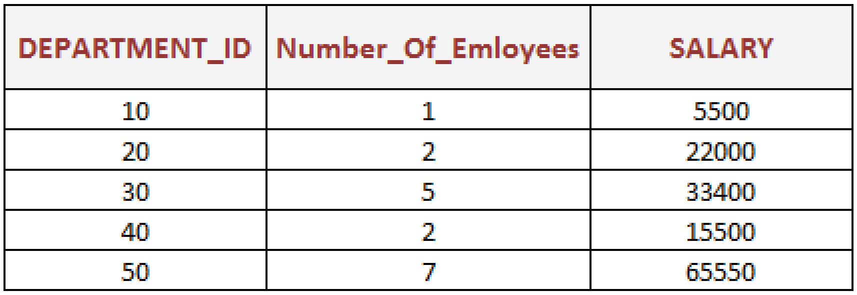
DEPARTMENT_ID	Number_Of_Emloyees
10	1
20	2
30	5
40	2
50	7

3) SQL GROUP BY with COUNT() and SUM() function:

Suppose user wants to get the number employees in each department & sum of Salary paid to each department, so in the Employees table we have column called DEPARTMENT_ID and SALARY. So we can group the result set using DEPARTMENT_ID & we will use COUNT() & SUM() Aggregate functions. COUNT() is used to get the number of employees by each department & SUM() is used to get the sum of

```
SELECT DEPARTMENT_ID,
COUNT(*) as Number_Of_Emloyees,
SUM(SALARY)
FROM Employees
GROUP BY DEPARTMENT_ID;
```

Following is the result upon executing the above SQL GROUP BY query,



DEPARTMENT_ID	Number_Of_Emloyees	SALARY
10	1	5500
20	2	22000
30	5	33400
40	2	15500
50	7	65550

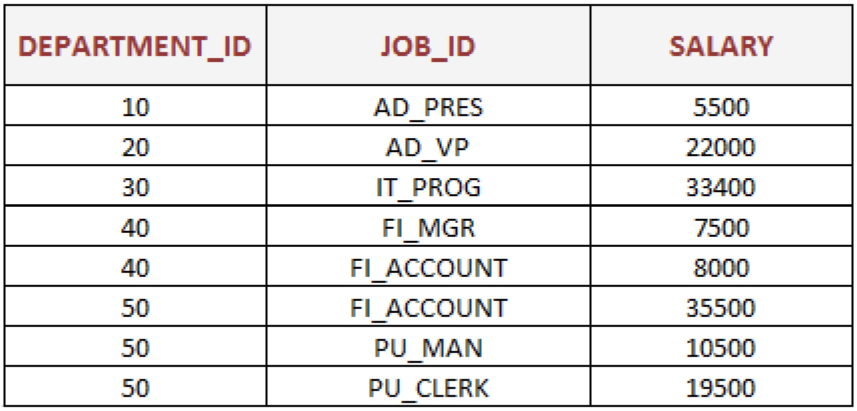
4) SQL GROUP BY on more than one column:

Group by Department id, Job Id & & SUM of salary with department

Suppose user wants to get the number employees in each department & sum of Salary paid to each department by Job ID, so in the Employees table we have column called DEPARTMENT_ID, SALARY and Job_ID. So here first we can group the result set using DEPARTMENT_ID & then by JOB_ID. Also we will use the SUM() Aggregate function to get the sum of salary by each Job_id within each department.

```
SELECT DEPARTMENT_ID,
JOB_ID,
SUM(SALARY)
FROM Employees
GROUP BY DEPARTMENT_ID, JOB_ID;
```

Following is the result upon executing the above SQL GROUP BY query,



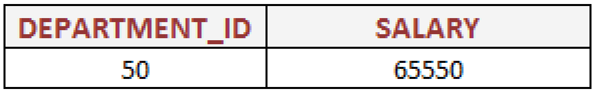
DEPARTMENT_ID	JOB_ID	SALARY
10	AD_PRES	5500
20	AD_VP	22000
30	IT_PROG	33400
40	FI_MGR	7500
40	FI_ACCOUNT	8000
50	FI_ACCOUNT	35500
50	PU_MAN	10500
50	PU_CLERK	19500

5) SQL GROUP BY with WHERE clause:

Suppose user wants to get the sum of Salary paid to DEPARTMENT_ID = 50, so in the Employees table we have column called DEPARTMENT_ID and SALARY. So here first we will use the WHERE condition on DEPARTMENT_ID = 50, this fitters the result set & returns only records from employee table related to DEPARTMENT_ID = 50. Next to WHERE clause, we can group the result set using DEPARTMENT_ID.

```
SELECT DEPARTMENT_ID,
SUM(SALARY)
FROM Employees
WHERE DEPARTMENT_ID = 50
GROUP BY DEPARTMENT_ID;
```

Following is the result upon executing the above SQL GROUP BY query,



DEPARTMENT_ID	SALARY
50	65550

6) SQL GROUP BY with HAVING clause:

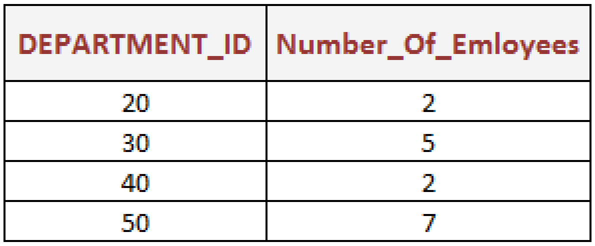
Group by Department id & count of employee with department & HAVING count > 1

Suppose user wants to get the number of employees from each department where at least two employee present in department. So we can group the result set using DEPARTMENT_ID & we will use COUNT()Aggregate function to get the number of employees by each department.

Here we will use HAVING clause to get the employee count > 1

```
SELECT DEPARTMENT_ID,
COUNT(*) as Number_Of_Emloyees
FROM Employees
GROUP BY DEPARTMENT_ID
HAVING COUNT(*) > 1;
```

Following is the result upon executing the above SQL GROUP BY query,



DEPARTMENT_ID	Number_Of_Emloyees
20	2
30	5
40	2
50	7

Conclusion on SQL GROUP BY Clause:

- The SQL GROUP BY clause is used to divide the result set into smaller groups.
- The grouping can happen after retrieves the rows from a table.
- In the GROUP BY Clause the grouping is done using same values present in the result set.
- The GROUP BY Clause is used with the conjunction of SQL SELECT query.
- The WHERE clause is used to retrieve rows based on a certain condition, but it cannot be applied to grouped result.
- The HAVING clause is used to filter the result set of GROUP BY clause. This is used after GROUP BY clause.

.

Source:

http://www.softwaretestingclass.com/sql-group-by-clause/