SQL GROUP BY clause

- The SQL **GROUP BY** clause is used in collaboration with the SELECT statement to arrange identical data into groups i.e., to make a group of rows based on the values of a specific column or expression.
- The usage of SQL GROUP BY clause is, to divide the rows in a table into smaller groups.
- The GROUP BY clause is used with the SQL SELECT statement.
- The grouping can happen after retrieves the rows from a table.

```
SELECT column1, function name(column2)

FROM table_name

WHERE condition

GROUP BY column1, column2

[HAVING] <condition>;
```

function_name: Name of the function used for example, SUM() , AVG().

table_name: Name of the table.

condition: Condition used.

```
SELECT <column_list>
FROM < table name >
WHERE <condition>
GROUP BY <columns>
 [HAVING] <condition>;
```

Name	Description
table_name	Name of the table.
column_list	Name of the columns of the table.
columns	Name of the columns which will participate in grouping..

Sample Table

Student

SUBJECT	YEAR	NAME
English	1	Harsh
English	1	Pratik
English	1	Ramesh
English	2	Ashish
English	2	Suresh
Mathematics	1	Deepak
Mathematics	1	Sayan

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
100	Steven	King	SKING	515.123.4567	6/17/1987	AD_PRES	24000			90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	6/18/1987	AD_VP	17000		100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	6/19/1987	AD_VP	17000		100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	6/20/1987	IT_PROG	9000		102	60
104	Bruce	Ernst	BERNST	590.423.4568	6/21/1987	IT_PROG	6000		103	60
105	David	Austin	DAUSTIN	590.423.4569	6/22/1987	IT_PROG	4800		103	60
106	Valli	Pataballa	VPATABAL	590.423.4560	6/23/1987	IT_PROG	4800		103	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	6/24/1987	IT_PROG	4200		103	60
108	Nancy	Greenberg	NGREENBE	515.124.4569	6/25/1987	FI_MGR	12000		101	100
109	Daniel	Faviet	DFAVIET	515.124.4169	6/26/1987	FI_ACCOUNT	9000		108	100
110	John	Chen	JCHEN	515.124.4269	6/27/1987	FI_ACCOUNT	8200		108	100
111	Ismael	Sciarra	ISCIARRA	515.124.4369	6/28/1987	FI_ACCOUNT	7700		108	100
112	Jose Manue	Urman	JMURMAN	515.124.4469	6/29/1987	FI_ACCOUNT	7800		108	100
113	Luis	Popp	LPOPP	515.124.4567	6/30/1987	FI_ACCOUNT	6900		108	100
114	Den	Raphaely	DRAPHEAL	515.127.4561	7/1/1987	PU_MAN	11000		100	30
115	Alexander	Khoo	AKHOO	515.127.4562	7/2/1987	PU_CLERK	3100		114	30
116	Shelli	Baida	SBAIDA	515.127.4563	7/3/1987	PU_CLERK	2900		114	30
117	Sigal	Tobias	STOBIAS	515.127.4564	7/4/1987	PU_CLERK	2800		114	30
118	Guy	Himuro	GHIMURO	515.127.4565	7/5/1987	PU_CLERK	2600		114	30
119	Karen	Colmenares	KCOLMENA	515.127.4566	7/6/1987	PU_CLERK	2500		114	30
120	Matthew	Weiss	MWEISS	650.123.1234	7/7/1987	ST_MAN	8000		100	50

180	Winston	Taylor	WTAYLOR	650.507.9876	9/5/1987	SH_CLERK	3200	120	50
181	Jean	Fleaur	JFLEAUR	650.507.9877	9/6/1987	SH_CLERK	3100	120	50
182	Martha	Sullivan	MSULLIVA	650.507.9878	9/7/1987	SH_CLERK	2500	120	50
183	Girard	Geoni	GGEONI	650.507.9879	9/8/1987	SH_CLERK	2800	120	50
184	Nandita	Sarchand	NSARCHAN	650.509.1876	9/9/1987	SH_CLERK	4200	121	50
185	Alexis	Bull	ABULL	650.509.2876	9/10/1987	SH_CLERK	4100	121	50
186	Julia	Dellinger	JDELLING	650.509.3876	9/11/1987	SH_CLERK	3400	121	50
187	Anthony	Cabrio	ACABRIO	650.509.4876	9/12/1987	SH_CLERK	3000	121	50
188	Kelly	Chung	KCHUNG	650.505.1876	9/13/1987	SH_CLERK	3800	122	50
189	Jennifer	Dilly	JDILLY	650.505.2876	9/14/1987	SH_CLERK	3600	122	50
190	Timothy	Gates	TGATES	650.505.3876	9/15/1987	SH_CLERK	2900	122	50
191	Randall	Perkins	RPERKINS	650.505.4876	9/16/1987	SH_CLERK	2500	122	50
192	Sarah	Bell	SBELL	650.501.1876	9/17/1987	SH_CLERK	4000	123	50
193	Britney	Everett	BEVERETT	650.501.2876	9/18/1987	SH_CLERK	3900	123	50
194	Samuel	McCain	SMCCAIN	650.501.3876	9/19/1987	SH_CLERK	3200	123	50
195	Vance	Jones	VJONES	650.501.4876	9/20/1987	SH_CLERK	2800	123	50
196	Alana	Walsh	AWALSH	650.507.9811	9/21/1987	SH_CLERK	3100	124	50
197	Kevin	Feeney	KFEENEY	650.507.9822	9/22/1987	SH_CLERK	3000	124	50
198	Donald	OConnell	DOCONNEL	650.507.9833	9/23/1987	SH_CLERK	2600	124	50
199	Douglas	Grant	DGRANT	650.507.9844	9/24/1987	SH_CLERK	2600	124	50
200	Jennifer	Whalen	JWHALEN	515.123.4444	9/25/1987	AD_ASST	4400	101	10
201	Michael	Hartstein	MHARTSTE	515.123.5555	9/26/1987	MK_MAN	13000	100	20
202	Pat	Fay	PFAY	603.123.6666	9/27/1987	MK_REP	6000	201	20
203	Susan	Mavris	SMAVRIS	515.123.7777	9/28/1987	HR_REP	6500	101	40
204	Hermann	Baer	HBAER	515.123.8888	9/29/1987	PR_REP	10000	101	70
205	Shelley	Higgins	SHIGGINS	515.123.8080	9/30/1987	AC_MGR	12000	101	110
206	William	Gietz	WGIETZ	515.123.8181	10/1/1987	AC_ACCOUNT	8300	205	110

Queries.....

- Display number of employees work in each department.
- Display total salary paid to employees work in each department.
- Display number of employees, total salary paid to employees work in each department.

- Display number of employees work in each department.

```
SELECT department_id "Department Code", COUNT(*) "No of
Employees"
FROM employees
GROUP BY department_id;
```

Department Code	No of Employees
100	6
30	6
	1
90	3
20	2
70	1
110	2
50	45
80	34
40	1
60	5
10	1

SQL GROUP BY with SUM() function

Display total salary paid to employees work in each department.

```
SELECT department_id, SUM(salary)

FROM  employees

GROUP BY  department_id;
```

DEPARTMENT_ID	SUM(SALARY)
100	51608
30	24900
	7000
90	58000
20	19000
70	10000
110	20308
50	156400
80	304500
40	6500
60	28800
10	4400

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

- Display number of employees, total salary paid to employees work in each department.

```
SELECT department_id "Department Code",
COUNT(*) "No of Employees",
SUM(salary) "Total Salary"
FROM  employees
GROUP BY  department_id;
```

Department Code	No of Employees	Total Salary
100	6	51608
30	6	24900
	1	7000
90	3	58000
20	2	19000
70	1	10000
110	2	20308
50	45	156400
80	34	304500
40	1	6500
60	5	28800
10	1	4400

SQL GROUP BY on more than one columns

- Display the department code, job id, total salary paid to employees group by department_id, job_id.

```
SELECT department_id "Department Code", job_id,
SUM(salary) "Total Salary"
FROM  employees
GROUP BY  department_id,job_id;
```

Department Code	JOB_ID	Total Salary
110	AC_ACCOUNT	8300
90	AD_VP	34000
50	ST_CLERK	55700
80	SA_REP	243500
50	ST_MAN	36400
80	SA_MAN	61000
110	AC_MGR	12008
90	AD_PRES	24000
60	IT_PROG	28800
100	FI_MGR	12008
30	PU_CLERK	13900
50	SH_CLERK	64300
20	MK_MAN	13000
100	FI_ACCOUNT	39600
	SA_REP	7000
70	PR_REP	10000
30	PU_MAN	11000
10	AD_ASST	4400
20	MK_REP	6000
40	HR_REP	6500

SQL GROUP BY with WHERE clause

Displays the department code, total salary paid to employees group by department_id and manager_id=103.

```
SELECT department_id "Department Code",
SUM(salary) "Total Salary"
FROM  employees
WHERE MANAGER_ID = 103
GROUP BY  department_id;
```

Department Code	Total Salary
60	19800

HAVING Clause

- The WHERE clause is used to place conditions on columns but what if we want to place conditions on groups?
- This is where HAVING clause comes into use. **We can use HAVING clause to place conditions to decide which group will be the part of final result-set.**
- Also we can not use the aggregate functions like SUM(), COUNT() etc. with WHERE clause. So we have to use HAVING clause if we want to use any of these functions in the conditions.

SQL GROUP BY with HAVING clause

- Display the department id, number of employees of those groups that have more than 2 employees.

```
SELECT department_id, count(*) "No. of Employee"
FROM employees
GROUP BY  department_id
HAVING count(*)>2;
```

DEPARTMENT_ID	No. of Employee
100	6
30	6
90	3
50	45
80	34
60	5