GROUP BY Clause

- Groupby clause is used to create groups of values using the group functions.

```
SELECT expression1, expression2, ..., expression_n,
       aggregate_function(aggregate_expression)
FROM table
[WHERE condition]
GROUP BY expression1, expression2, ..., expression_n
[ORDER BY order_expression]
```

Above is the expression for using Group By clause. In the below points, we are going to learn how group-by can be used in various use cases.

- We can use more than one expression -or column- in a GROUP BY clause.

```
SELECT hire_date, job_id, COUNT(*)
FROM employees
GROUP BY hire_date, job_id
ORDER BY hire_date
```

Here we have used hire_date and job_id to group the data and then use aggregate functions on the grouped data. Let's see the difference when we use only one column to group the data and then use two columns.

```
SELECT hire_date, job_id, COUNT(*)
FROM employees
GROUP BY hire_date
ORDER BY COUNT(*) DESC
```

The output for this query is shown below. You can see that the data is grouped by hire_date and the number of employees is counted based on the hire_date. And there are 4 employees whose hire date is 1994-06-07.

hire_date	job_id	COUNT(*)
1994-06-07	HR_REP	4
1994-08-16	FI_ACCOUNT	1
1994-08-17	FI_MGR	1
1994-12-07	PU_MAN	1
1995-05-01	ST_MAN	1

Now, let's group the data based on two columns.

```
SELECT hire_date, job_id, COUNT(*)
FROM employees
GROUP BY hire_date, job_id
ORDER BY hire_date
```

The output of this query is shown below. It can be seen that for each hire_date the data is grouped based on the job_id. It is clearly shown using the hire_date = 1994-06-07.

hire_date	job_id	COUNT(*)
1994-06-07	AC_ACCOUNT	1
1994-06-07	AC_MGR	1
1994-06-07	HR_REP	1
1994-06-07	PR_REP	1
1994-08-16	FI_ACCOUNT	1
1994-08-17	FI_MGR	1

Important Points

- The **SELECT** clause cannot have any other individual columns than what is used with the **GROUP BY** clause. In the case of MYSQL if any column name is used in the select statement which is not present in the group by clause, then it shows the first entry for that column, but for PostgreSQL, it will throw an error.

- We don't need to use all the columns used with the GROUP BY clause in the SELECT statement.

```
SELECT job_id AS "Job ID",
       SUM(salary) AS "Sum of Salary",
       MAX(hire_date) AS "Latest Hire Date",
       COUNT(*) AS "Number of Employees"
FROM employees
GROUP BY department_id, job_id;
```

In the above query, it is grouped based on the department_id and job_id, but in the select statement, only job_id is used.

- In the SELECT statements, we can use the group functions with different columns than GROUP BY has. As shown in the query above the group functions are used on the salary and hire_date columns which are not in the group by clause.
- We can use as many group functions as we want which can also be seen in the above-shown query.