# PRACTICAL - 6

**AIM: To apply the concept of Aggregating Data using Group functions**

**THEORY:**

**GROUP BY**

The SQL GROUP BY clause can be used in a SELECT statement to collect data across multiple records and group the results by one or more columns.

**Syntax:**

The syntax for the GROUP BY clause in SQL is:

SELECT expression1,expression2, ...expression-n, aggregate function (aggregate\_expression)FROM tables[WHERE conditions]GROUP BY expression1, expression2, ... expression\_n[ORDER BY expression [ ASC | DESC ]];

Parameters or Arguments

* **expression1, expression2, ...expression-n**: Expressions that are not encapsulated within an aggregate function and must be included in the GROUP BY Clause at the end of the SQL statement.
* **Aggregate function:** This is an aggregate function such as the SUM, COUNT, MIN, MAX, or AVG functions.
* **Aggregate expression:** This is the column or expression that the *aggregate function* will be used on.
* **Tables:** The tables that you wish to retrieve records from. There must be at least one table listed in the FROM clause.
* **WHERE conditions:** Optional. These are conditions that must be met for the records to be selected.
* **ORDER BY expression:** Optional. The expression used to sort the records in the result set. If more than one expression is provided, the values should be comma separated.
* **ASC:** Optional. ASC sorts the result set in ascending order by *expression*. This is the default behavior, if no modifier is provider.
* **DESC:** Optional. DESC sorts the result set in descending order by *expression*.

**Example - Using GROUP BY with the SUM Function**

Let's look at how to use the GROUP BY clause with the SUM function in SQL.

We have a table called *employees* with the following data:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Employee\_number** | **last\_name** | **first\_name** | **salary** | **dept\_id** |
| 1001 | Smith | John | 62000 | 500 |
| 1002 | Anderson | Jane | 57500 | 500 |
| 1003 | Everest | Brad | 71000 | 501 |
| 1004 | Horvath | Jack | 42000 | 501 |

Enter the following SQL statement:

SELECT DEPT\_ID, SUM(SALARY) AS TOTAL\_SALARIESFROM EMPLOYEESGROUP BY DEPT\_ID;

There will be 2 records selected. These are the results that you should see:

|  |  |
| --- | --- |
| **dept\_id** | **total\_salaries** |
| 500 | 119500 |
| 501 | 113000 |

In this example, we've used the SUM function to add up all of the salaries for each *dept\_id* and we've aliased the results of the SUM function as *total\_salaries*. Because the *dept\_id* is not encapsulated in the SUM function, it must be listed in the GROUP BY clause.

**Example - Using GROUP BY with the COUNT function**

We have a table called *products* with the following data:

|  |  |  |
| --- | --- | --- |
| **product\_id** | **product\_name** | **category\_id** |
| 1 | Pear | 50 |
| 2 | Banana | 50 |
| 3 | Orange | 50 |
| 4 | Apple | 50 |
| 5 | Bread | 75 |
| 6 | Sliced Ham | 25 |
| 7 | Kleenex | NULL |

Enter the following SQL statement:

SELECT CATEGORY\_ID, COUNT(\*) AS TOTAL\_PRODUCTSFROM PRODUCTSWHERE CATEGORY\_ID IS NOT NULLGROUP BY CATEGORY\_IDORDER BY CATEGORY\_ID;

There will be 3 records selected. These are the results that you should see:

|  |  |
| --- | --- |
| category\_id | total\_products |
| 25 | 1 |
| 50 | 4 |
| 75 | 1 |

In this example, we've used the COUNT function to calculate the number of products for each *category\_id* and we've aliased the results of the COUNT function as *total\_products*. We've excluded any *category\_id* values that are NULL by filtering them out in the WHERE clause. Because the *category\_id* is not encapsulated in the COUNT function, it must be listed in the GROUP BY clause.

**HAVING**

The SQL HAVING clause is used in combination with the GROUP BY clause to restrict the groups of returned rows to only those whose the condition is TRUE.

**Syntax:**

The syntax for the HAVING clause in SQL is:

SELECT expression1,expression2, ...expression\_n, aggregate\_function (aggregate\_expression)FROM tables[WHERE conditions]GROUP BY expression1, expression2, ... expression\_nHAVING condition;

Parameters or Arguments

* **expression1, expression2, ...expression\_n:** Expressions that are not encapsulated within an aggregate function and must be included in the GROUP BY Clause near the end of the SQL statement.
* **aggregate\_function:** This is an aggregate function such as the SUM, COUNT, MIN, MAX, or AVG functions.
* **aggregate\_expression:** This is the column or expression that the *aggregate\_function* will be used on.
* **Tables:** The tables that you wish to retrieve records from. There must be at least one table listed in the FROM clause.
* **WHERE conditions:** Optional. These are the conditions for the records to be selected.
* **HAVING condition:** This is a further condition applied only to the aggregated results to restrict the groups of returned rows. Only those groups whose condition evaluates to TRUE will be included in the result set.

**Example - Using SUM function:**

You could also use the SQL SUM function to return the name of the department and the total sales (in the associated department). The SQL HAVING clause will filter the results so that only departments with sales greater than $1000 will be returned.

SELECT DEPARTMENT, SUM(SALES) AS "TOTAL SALES"FROM ORDER\_DETAILSGROUP BY DEPARTMENTHAVING SUM(SALES) > 1000;

**Example - Using COUNT function**

You could use the SQL COUNT function to return the name of the department and the number of employees (in the associated department) that make over $25,000 / year. The SQL HAVING clause will filter the results so that only departments with more than 10 employees will be returned.

SELECT department, COUNT(\*) AS "Number of employees "FROM employees WHERE salary > 25000GROUP BY department HAVING COUNT(\*) > 10;

**DIFFERENCE BETWEEN HAVING AND WHERE**

When working with more advanced SQL it can be unclear when it makes sense to use a WHERE versus a HAVING clause.

Though it appears that both clauses do the same thing, they do it in different ways.  In fact, their functions complement each other.

* A WHERE clause is used is filter records from a result.  The filter occurs before any groupings are made.
* A HAVING clause is used to filter values from a group.

**PROGRAM EXECUTION:**

**Queries to be performed:**

(1) List total deposit of customer having account date after 3-jan-96.

**Output:**

|  |
| --- |
|  |

(2) List total deposit of customers living in city Nagpur.

**Output:**

|  |
| --- |
|  |

(3) List maximum deposit of customers living in Bombay.

**Output:**

|  |
| --- |
|  |

(4) Display the highest, lowest, sum, and average salary of all employees. Label the columns Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number.

**Output:**

|  |
| --- |
|  |

(5) Write a query that displays the difference between the highest and lowest salaries. Label the column DIFFERENCE.

**Output:**

|  |
| --- |
|  |

(6) Create a query that will display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998.

**Output:**

|  |
| --- |
|  |

(7) Find the average salaries for each department without displaying the respective department numbers.

**Output:**

|  |
| --- |
|  |

(8) Write a query to display the total salary being paid to each job title, within each department.

**Output:**

|  |
| --- |
|  |

(9) Find the average salaries > 2000 for each department without displaying the respective department numbers.

**Output:**

|  |
| --- |
|  |

|  |
| --- |
| Output: |

(10) Display the job and total salary for each job with a total salary amount exceeding 3000, in which excludes president and sorts the list by the total salary.

(11) List the branches having sum of deposit more than 5000 and located in city Bombay.

**Output:**

|  |
| --- |
|  |

**VIVA QUESTIONS:**

**Q) What is the purpose of the group functions in SQL? Give some examples of group functions.**

**Ans.)** Group functions are built-in functions that operate on group of rows and return one value for the entire group.

There are different functions available like max(), min(), sum(), distinct(), count(), avg() etc.

**Q) Say True or False. Give explanation if False.**

**Group functions cannot be nested.**

**Ans.) False,** Group functions can be nested to a depth level of two.