|  |
| --- |
| **Part A** |
| **Aim:**SQL commands:   1. To perform SQL Aggregate Functions 2. Group by Clause 3. Having Clause |
| **Prerequisite:**Oracle. |
| **Outcome:** Understanding and use of various Oracle functions. |
| **Theory:**  **Aggregate Functions**  **AVG:** returns average value  Avg(<ColumnName>)  **MIN:** returns minimum value  min(<ColumnName>)  **COUNT:** returns no of rows where expression is not NULL  count(<ColumnName>)  **COUNT(\*):** returns no of rows in the table including duplicates and those with NULL  count(\*)  **MAX:** returns maximum value  max(<ColumnName>)  **SUM:** returns sum of the values  sum(<ColumnName>)  **Group by clause:** this optional clause tells Oracle to group rows based on distinct values that exists for specified columns.  Select <columnname 1><columnname 2>...<columnname n>,  Aggregate\_function(<expression>) from tablename Where <condition>  Group by <columnname 1><columnname 2>...<columnname n>;  **Having clause:** imposes a condition on group by clause.  Select <columnname 1><columnname 2>...<columnname n>,  Aggregate\_function(<expression>) from tablename Where <condition>  Group by <columnname 1><columnname 2>...<columnname n>  Having <condition>;  Example |
| **Procedure:**   1. Formulate the query for given problem. 2. Write the SQL query with proper input. 3. Execute the query. |
| **Practice Exercise:**   1. Give one example query to demonstrate each function. 2. Display the total expenditure of company on the salary of employees. 3. Find average salary of clerks. 4. Find average salary of managers and salesman. 5. Find employees with maximum annual income. 6. Find the employee with minimum monthly income. 7. Find the number of employees earning more than the average salary of employees. 8. List the details of the department where maximum number of emps are working. 9. Find the total salary department wise. 10. Find total salary average salary Job wise. 11. Find the name of department taking maximum salary. 12. Find name of department taking minimum salary. |
| **Instructions:**   1. Write and execute the query in Oracle SQL server. 2. Paste the snapshot of the output in input & output section. |
| **Part B** |
| **Code and Output:**  **1.** |
| **2.** |
| **3.** |
| **4.** |
| **5.** |
| **6.** |
| **7.** |
| **8.** |
| **9.** |
| **10.** |
| **11.** |
| **12.** |
| **Observation & Learning:**  From this experiment, we observed and learned how the following **SQL** commands SQL Aggregate Functions, Group by Clause and Having clause are used to perform in the **ORACLE DATABASE** |
| **Conclusion:**  In this experiment, the following **SQL commands** are executed to perform SQL Aggregate Functions, Group by Clause and Having clause in the **ORACLE DATABASE** and the Outputs are obtained as per queries. |
| **Questions:**   1. **What is the use of aggregate function?** 2. An **aggregate function** performs a calculation one or more values and returns a single value. The aggregate function is often used with the **GROUP BY** clause and **HAVING** clause of the **SELECT** statement.   **a)** The **AVG** aggregate function calculates the average of **non-NULL** values in a set  **b)** The **COUNT** function returns the total number of values in the specified field. It works on both numeric and non-numeric data types.  **c)COUNT (\*)** is a special implementation of the **COUNT** function that returns the count of all the rows in a specified table. **COUNT (\*)** also considers Nulls and duplicates.  **d)**The **MIN** function returns the smallest value in the specified table field.  **e)** The **MAX** function returns the largest value from the specified table field.  **f)** The **SUM** function which returns the sum of all the values in the specified column. **SUM** works on numeric fields only. **Null** values are excluded from the result returned.   1. **How different number of rows can be counted?** 2. The **SQL COUNT ()** function returns the number of rows in a table satisfying the criteria specified in the WHERE clause. It sets the number of rows or **non -NULL** column values. **COUNT ()** returns **0** if there were no matching rows. To calculate the different number of rows we use the **DISTINCT** keyword in **COUNT ()** function which returns the **distinct or different** rows. 3. **What is the difference between having and where clause?** 4. **1)**The **WHERE** clause is used in the selection of rows according to given conditions whereas the **HAVING** clause is used in column operations and is applied to aggregated rows or groups. 5. If **GROUP BY** is used then it is executed after the **WHERE** clause is executed in the query. It means it selects the rows before grouping is done or aggregate calculations are performed. That’s why, **WHERE** clause is also called Pre-filter. 6. But, **GROUP BY** is executed before the execution of the **HAVING** clause. It means it selects the rows after aggregate calculations are performed. That’s why, **HAVING** clause is also called as Post-filter. 7. We cannot use the **HAVING** clause without **SELECT** statement whereas the **WHERE** clause can be used with **SELECT, UPDATE, DELETE** etc. 8. We can use aggregate functions like **sum, min, max, avg**, etc with the **HAVING** clause but they can never be used with **WHERE** clause. 9. **Does *WHERE* clause work with aggregate functions?** 10. **WHERE** clause does not work with aggregate functions. An **aggregate function** may not appear in the **WHERE** clause unless it is in a subquery contained in a **HAVING** clause or a select list, and the column being aggregated is an outer reference |