|  |
| --- |
| **Part A** |
| **Aim:** commands:   1. To perform join operations 2. Use set operators |
| **Prerequisite:** SQL Server. |
| **Outcome:** Understanding and use of join operations. |
| **Theory:** SQL JOIN An SQL JOIN clause is used to combine rows from two or more tables, based on a common field between them. SQL INNER JOIN The INNER JOIN keyword selects all rows from both tables as long as there is a match between the columns in both tables. Syntax SELECT *column\_name(s)* FROM *table1* INNER JOIN *table2* ON *table1.column\_name*=*table2.column\_name*; SQL LEFT JOIN The LEFT JOIN keyword returns all rows from the left table (table1), with the matching rows in the right table (table2). The result is NULL in the right side when there is no match. Syntax SELECT *column\_name(s)* FROM *table1* LEFT JOIN *table2* ON *table1.column\_name*=*table2.column\_name*; SQL RIGHT JOIN The RIGHT JOIN keyword returns all rows from the right table (table2), with the matching rows in the left table (table1). The result is NULL in the left side when there is no match. Syntax SELECT *column\_name(s)* FROM *table1* RIGHT JOIN *table2* ON *table1.column\_name*=*table2.column\_name*; SQL FULL OUTER JOIN The FULL OUTER JOIN keyword returns all rows from the left table (table1) and from the right table (table2).  The FULL OUTER JOIN keyword combines the result of both LEFT and RIGHT joins. Syntax SELECT *column\_name(s)* FROM *table1* FULL OUTER JOIN *table2* ON *table1.column\_name*=*table2.column\_name*;  **SQL UNION :**  The SQL UNION operator combines the result of two or more SELECT statements.  **Syntax:**  SELECT *column\_name(s)* FROM *table1* UNION SELECT *column\_name(s)* FROM *table2*;  **SQL INTERSECT:**  The SQL Intersect operator returns all the results which are common in two or more SELECT statements.  **Syntax:**  SELECT *column\_name(s)* FROM *table1* Intersect SELECT *column\_name(s)* FROM *table2*;  **SQL EXCEPT:**  The SQL Except operator returns all the results which are in the result of first but not in the result of second SELECT statement (Set-Difference).  **Syntax:**  SELECT *column\_name(s)* FROM *table1* Except SELECT *column\_name(s)* FROM *table2*; |
| **Procedure:**   1. Formulate the query for given problem. 2. Write the SQL query with proper input. 3. Execute the query. |
| **Practice Exercise:**   1. Display the common jobs from department number 10 and 20. 2. Display the jobs found in department number 10 and 20 eliminate duplicate jobs. 3. Display the jobs which are in dept no 10 but not in 20. 4. Display those employees who are working in the same dept where his manager is working. 5. Delete those employees who joined the company before 31-dec-82 while there dept location is ‘NEW YORK’ or ‘CHICAGO’. 6. Display employees name for the dept Accounting or Sales but job is not clerk, while joined the company before 31-dec-82. 7. Display employee name, job, deptname, location for all who are working as managers. 8. Display those employees whose manager names is Jones, and also display there manager name. 9. Display emp number and salary of ford if his Sal is equal to highest Sal of his department. 10. List out all the employees name, job, and salary grade and department name for every one in the company except ‘CLERK’. Sort on salary. 11. Display employees who are without manager. 12. Display the name of those employees who are getting highest salary. 13. Display the name of those employees who are getting second highest salary. 14. Display those employees whose salary is equal to average of maximum and minimum. 15. Display count of employees in each department where count greater than 3. 16. Display dname where at least 3 are working and display only dname. 17. Display name of those managers name whose salary is more than average salary of company. 18. Find out the top 5 earner of company. 19. Find out the last 5(least) earner of the company? 20. Display employee name, his job, his dept name, his manager name, his sal and arrange it based on salary under department wise. |
| **Instructions:**   1. Write and execute the query in Oracle SQL server. 2. Paste the snapshot of the output in input & output section. |
| **Part B** |
| **CREATING TABLE DEPARTMENT** |
| **INSERTING VALUES INTO THE DEPARTMENT TABLE** |
| **CREATING AND INSERTING VALUES INTO THE TABLE EMP** |
| **DISPLAYING THE DATA OF DEPARTMENT** |
| **DISPLAYING THE DATA OF EMP** |
| **1** |
| **2** |
| **3** |
| **4** |
| **5** |
| **6** |
| **7** |
| **8** |
| **9** |
| **10** |
| **11** |
| **12** |
| **13** |
| **14** |
| **15** |
| **16** |
| **17** |
| **18** |
| **19** |
| **20** |
| **Observation & Learning:**  From this experiment, we observed and learned how the following **SQL** commands are used to perform **SQL** join operations and set operations in the **ORACLE DATABASE.** |
| **Conclusion:**  In this experiment, the following **SQL commands** are executed to perform **SQL** join operations and set operationsin the queries in the **ORACLE DATABASE** and the Outputs are obtained as per queries. |
| **Questions:**   1. Explain self-join with example query and output. 2. The SQL **SELF JOIN** is used to join a table to itself as if the table were two tables; temporarily renaming at least one table in the SQL statement |