|  |  |  |  |
| --- | --- | --- | --- |
| TE-COMP Roll number : | | | |
| Experiment no. : 4 Date of Implementation : | | | |
| Aim : To implement simple SQL commands | | | |
| Tool Used : PostgreSQL | | | |
| Related Course outcome : At the end of the course, Students will be able to Use  SQL : Standard language of relational database | | | |
| **Rubrics for assessment of Experiment:**   |  |  |  |  | | --- | --- | --- | --- | | Indicator | Poor | Average | Good | | Timeliness   * Maintains assignment deadline (3) | Assignment not done (0) | One or More than One week late (1-2) | Maintains deadline (3) | | Completeness and neatness   * Complete all parts of QUERY assignment(3) | N/A | < 80% complete (1-2) | 100% complete (3) | | Originality   * Extent of plagiarism(2) | Copied it from someone else(0) | At least few questions have been done without copying(1) | Assignment has been solved completely without copying (2) | | Knowledge   * In depth knowledge of the QUERY assignment(2) | Unable to answer 2 questions(0) | Unable to answer 1 question (1) | Able to answer 2 questions (2) | | | | |
| **Assessment Marks :**   |  |  | | --- | --- | | Timeliness |  | | Completeness and neatness |  | | Originality |  | | Knowledge |  | | Total |  | | | | |
| **Total : (Out of 10)** | | | |
| **Teacher's Sign :** | | | |
| EXPERIMENT 4 | Basic SQL Commands |
| Aim | To implement simple SQL commands |
| Tools | PostgreSQL |
| Theory | **SELECT:** SELECT statement returns a result set of records from one or more tables.  The select statement has optional clauses:   * WHERE specifies which rows to retrieve * GROUP BY groups rows sharing a property so that an aggregate function can be applied to each group having group. * HAVING selects among the groups defined by the GROUP BY clause. * ORDER BY specifies an order in which to return the rows.   Syntax:  SELECT<attribute list>  FROM<table list>  WHERE<condition>  Where   * Attribute list is a list of attribute name whose values to be retrieved by the query. * Table list is a list of table name required to process query. * Condition is a Boolean expression that identifies the tuples to be retrieved by query.   **SQL Aggregate Functions**  SQL aggregate functions return a single value, calculated from values in a column.  Useful aggregate functions:   * AVG() - Returns the average value * COUNT() - Returns the number of rows * FIRST() - Returns the first value * LAST() - Returns the last value * MAX() - Returns the largest value * MIN() - Returns the smallest value * SUM() - Returns the sum   **The SQL ORDER BY Keyword**  The ORDER BY keyword is used to sort the result-set by one or more columns.  The ORDER BY keyword sorts the records in ascending order by default. To sort the records in a descending order, you can use the DESC keyword.  **SQL ORDER BY Syntax**  SELECT column\_name, column\_name FROM table\_name ORDER BY column\_name ASC|DESC, column\_name ASC|DESC; |
| Procedure | TASK 1:  1. Create following table:  Table name : sales\_order   |  |  |  | | --- | --- | --- | | Column Name | Data type | Size | | order\_no | varchar | 6 | | Order\_date | date |  | | Client\_no | varchar | 6 | | Dely\_addr | varchar | 25 | | Salesman\_no | varchar | 6 | | Dely\_type | char | 1 | | Billed\_yn | char | 1 | | Dely\_date | Date |  | | Order\_status | varchar | 10 |   2. Insert 5-6 records in table.  3. Find the names of all clients having ‘a’ as the second letter in their names.  4. Find out the clients who stay in a city whose second letter is ‘a’  5. Find the list of all clients who stay in ‘mumbai’ ordered by their names  6. Print the list of clients whose bal\_due is greater than value 10000  7. Print the information from sales\_order table for orders placed in the month of January  8. Display the order information for client\_no C001 and C002  9. Find the products whose selling price is greater than 2000 and less than or equal to 5000  10. Find the products whose selling price is more than 1500. Calculate new selling price as original selling price \* 1.5. Rename the new column in the above query as new\_price  11. Count the total number of orders  12. Calculate the average price of all the product  13. Determine minimum and maximum product prices  14. count the number of products having price greater than or equal to 1500  15. Display the order number and day on which clients placed their order  16. Display the order\_date in the format ‘dd-month-yy’  17. Display the month (in alphabets) and date when the order must be delivered  18. Find the date, 15 days after today’s date  19. Find the no. of days elapsed between today’s date and the delivery date of orders placed by the clients.  Task2: Use select with where statement with SQL aggregate functions for the tables created in Expt. no. 3 |
|  |
| **Post Lab Questions:** | 1. Write a short note on DBA 2. Explain system structure of DBMS 3. Write different date functions 4. Differentiate between group by and having with example 5. Give different string functions |