

Lab 4

Built in Functions in SQL, Group by clause, Having clause

Group By clause:

The SQL GROUP BY clause is used in collaboration with the SELECT statement to arrange identical data into groups. The GROUP BY clause follows the WHERE clause in a SELECT statement and precedes the ORDER BY clause.

Syntax:

The basic syntax of GROUP BY clause is given below. The GROUP BY clause must follow the conditions in the WHERE clause and must precede the ORDER BY clause if one is used.

```
SELECT column1, column2
```

```
FROM table_name
```

```
WHERE [ conditions ]
```

```
GROUP BY column1, column2
```

```
ORDER BY column1, column2
```

Example:

```
SELECT NAME, SUM(SALARY) FROM CUSTOMERS
```

```
GROUP BY NAME;
```

ORDER BY Clause

The SQL ORDER BY clause is used to sort the data in ascending or descending order, based on one or more columns. Some database sorts query results in ascending order by default.

Syntax:

```
SELECT column-list
```

```
FROM table_name
```

```
[WHERE condition]
```

```
[ORDER BY column1, column2, .. columnN] [ASC | DESC];
```

You can use more than one column in the ORDER BY clause. Make sure whatever column you are using to sort, that column should be in column-list.

Example:

```
SELECT * FROM CUSTOMERS  
  
ORDER BY NAME DESC;
```

Note: By default the records are ordered in ascending order.

TOP Clause

The SQL TOP clause is used to fetch a TOP N number or X percent records from a table. Note: All the databases do not support TOP clause. For example MySQL supports LIMIT clause to fetch limited number of records and Oracle uses ROWNUM to fetch limited number of records.

Syntax:

```
SELECT TOP number|percent column_name(s)  
  
FROM table_name  
  
WHERE [condition]
```

Example:

```
SELECT TOP 3 * FROM CUSTOMERS;
```

HAVING CLAUSE

The HAVING clause enables you to specify conditions that filter which group results appear in the final results. The WHERE clause places conditions on the selected columns, whereas the HAVING clause places conditions on groups created by the GROUP BY clause.

The HAVING clause must follow the GROUP BY clause in a query and must also precedes the ORDER BY clause if used. The following is the syntax of the SELECT statement, including the HAVING clause:

Syntax:

```
SELECT column1, column2  
  
FROM table1, table2  
  
WHERE [ conditions ]  
  
GROUP BY column1, column2  
  
HAVING [ conditions ]  
  
ORDER BY column1, column2
```

Example: display record for which similar age count would be more than or equal to 2:

```
SELECT *  
  
FROM CUSTOMERS  
  
GROUP BY age  
  
HAVING COUNT(age) >= 2;
```

Built-in functions

SQL has many built-in functions for performing processing on string or numeric data. Following is the list of all useful SQL built-in functions:

- ☐ SQL COUNT Function - The SQL COUNT aggregate function is used to count the number of rows in a database table.
- ☐ SQL MAX Function - The SQL MAX aggregate function allows us to select the highest (maximum) value for a certain column.
- ☐ SQL MIN Function - The SQL MIN aggregate function allows us to select the lowest (minimum) value for a certain column.
- ☐ SQL AVG Function - The SQL AVG aggregate function selects the average value for certain table column.
- ☐ SQL SUM Function - The SQL SUM aggregate function allows selecting the total for a numeric column.
- ☐ SQL SQRT Functions - This is used to generate a square root of a given number.
- ☐ SQL RAND Function - This is used to generate a random number using SQL command.
- ☐ SQL CONCAT Function - This is used to concatenate any string inside any SQL command.
- ☐ SQL Numeric Functions - Complete list of SQL functions required to manipulate numbers in SQL.
- ☐ SQL String Functions - Complete list of SQL functions required to manipulate strings in SQL.

SQL COUNT Function

SQL COUNT function is the simplest function and very useful in counting the number of records, which are expected to be returned by a SELECT statement.

Example:

```
SQL>SELECT COUNT(*) FROM employee_tbl ;
```

Similarly, if you want to count the number of records for Nepal, then it can be done as follows:

```
SQL>SELECT COUNT(*) FROM employee_tbl WHERE name= 'Nepal';
```

SQL MAX Function

SQL MAX function is used to find out the record with maximum value among a record set.

Example:

```
SQL> SELECT * FROM employee_tbl;
```

id	name	work_date	daily_typing_pages
1	John	2007-01-24	250
2	Ram	2007-05-27	220
3	Jack	2007-05-06	170
3	Jack	2007-04-06	100
4	Jill	2007-04-06	220
5	Zara	2007-06-06	300
5	Zara	2007-02-06	350

output:

```
SQL> SELECT id, name, MAX(daily_typing_pages)
-> FROM employee_tbl GROUP BY name;
```

id	name	MAX(daily_typing_pages)
3	Jack	170
4	Jill	220

1	John	250
2	Ram	220
5	Zara	350

5 rows in set (0.00 sec)

MIN Function

SQL MIN function is used to find out the record with minimum value among a record set.

```
SQL> SELECT id, name, work_date, MIN(daily_typing_pages)
-> FROM employee_tbl GROUP BY name;
```

id	name	MIN(daily_typing_pages)
3	Jack	100
4	Jill	220
1	John	250

AVG Function

SQL AVG function is used to find out the average of a field in various records.

```
SQL> SELECT AVG(daily_typing_pages)
-> FROM employee_tbl;
```

AVG(daily_typing_pages)
230.0000

1 row in set (0.03 sec)

SUM Function

SQL SUM function is used to find out the sum of a field in various records.

```
SQL> SELECT SUM(daily_typing_pages)
-> FROM employee_tbl;
+-----+
| SUM(daily_typing_pages) |
+-----+
|           1610         |
+-----+
1 row in set (0.00 sec)
```

SQRT Function

SQL SQRT function is used to find out the square root of any number. You can Use SELECT statement to find out square root of any number.

```
SQL> select SQRT(16);
+-----+
| SQRT(16) |
+-----+
| 4.000000 |
+-----+
1 row in set (0.00 sec)
```

ABS(X)

The ABS() function returns the absolute value of X.

```
SQL> SELECT ABS(-2);
+-----+
| ABS(2) |
+-----+
| 2      |
+-----+
1 row in set (0.00 sec)
```

SQRT(X)

This function returns the non-negative square root of X

```
SQL>SELECT SQRT(49);
+-----+
| SQRT(49) |
+-----+
| 7        |
+-----+
1 row in set (0.00 sec)
```

Lab Task:

1. Sort the employee records in descending order.
2. Sort name and publication name in ascending order.
3. Display top three records from teachers relation.
4. Display the sum of salaries of all employees.
5. Display the minimum salary of employee.
6. Display the average price of book written by same author.
7. Display publication name and number of books published by it from book list relation publication wise.
8. Display the bid and bname of books whose price is greater than average prices of book.
9. Find the bid , bname and author in ascending order where author name is started by "s".
10. Find the teachers name and book taken by him. The teacher's salary who takes the book should be the maximum salary.
11. Find the authors name who have written more than one book.