

**Database Management System**

**(DBMS – 204)**

Experiment # 02

***Writing Basic SQL SELECT* Restricting and Sorting Data**

|  |
| --- |
| Student Name: Muhammad Faraz Ansar |
| Roll Number: SE-061 |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Maximum Marks** | **Performance = 05** | **Viva = 05** | **Total = 10** |
| **Marks Obtained** |  |  |  |
| **Remarks (if any)** |  | | |
|  | | | |
| **Experiment evaluated by** | | | |
| Instructor Name: Engr. Adiba Jafar | | | |
| Signature and Date: | | | |



***OUTCOMES***

**Restricting and Sorting Data Objectives**

After completing this lesson, you should be able to do the following:

1. Limit the rows retrieved by a query
2. Sort the rows retrieved by a query

**THOERY**

**Limiting Rows Using a Selection**

1. EMPLOYEES “retrieve all employees in department 90”
2. Limiting Rows Using a Selection
3. Limiting the Rows Selected

**Restrict the rows returned by using the WHERE clause. The WHERE clause follows the FROM clause.**

SELECT \*|{[DISTINCT] column expression [ alias ],...} FROM table [WHERE condition(s) ];

SELECT empno, ename, job, deptno FROM emp WHERE deptno = 90;

**Character Strings and Dates**

• Character strings and date values are enclosed in single quotation marks.

• Character values are case sensitive, and date values are format sensitive.

• The default date format is DD-MON-YR.

SELECT ename, job, deptno FROM emp WHERE ename = 'Goyal';

**Comparison Conditions**

|  |
| --- |
| Operator Meaning |
| = Equal to |
| > Greater than |
| >= Greater than or equal to |
| < Less than |
| <= Less than or equal to |
| <> Not equal to |
|  |

**Example**

1. WHERE hire\_date='01-JAN-95'
2. WHERE salary>=6000
3. WHERE ename='Smith'

NOTE:

An alias cannot be used in the WHERE clause.

The symbol != and ^= can also represent the not equal to condition.

**Using Comparison Conditions**

SELECT ename, sal FROM emp WHERE sal <= 3000;

**Using the Comparison Conditions Other Comparison Conditions**

|  |
| --- |
| Operator Meaning |
| BETWEEN Between two values (inclusive) |
| ...AND... |
| IN(set) Match any of a list of values |
| LIKE Match a character pattern |
| IS NULL Is a null value |
|  |

**Using the BETWEEN Condition**

Use the BETWEEN condition to display rows based on a range of values. (Lower and Upper limit)

**SELECT ename, sal FROM emp WHERE sal BETWEEN 2500 AND 3500;**

**Using the IN Condition**

Use the IN membership condition to test for values in a list.

SELECT empno, ename, sal, mgr FROM emp WHERE mgr IN (7902,7698,7839);

**Using the LIKE Condition**

Use the LIKE condition to perform wildcard searches of valid search string values.

Search conditions can contain either literal ,characters or numbers:

1. % denotes zero or many characters.
2. \_ (underscore) denotes one character.

**SELECT ename FROM emp WHERE ename LIKE 'S%';**

**You can combine pattern-matching characters.**

**SELECT ename FROM emp WHERE ename LIKE '\_A%';**

|  |
| --- |
| You can use the ESCAPE identifier to search for the Actual % and \_ symbols. |
| **Where sal LIKE ‘200%’;** |
| **Where sal LIKE %200%’;** |
| **Where sal LIKE ‘\_00%’;** |
| **Where sal LIKE ‘2\_%\_%’;** |
| **Where sal LIKE ‘%2’;** |
| **Where sal LIKE \_2%3’;** |
|  |

**Using the NULL Conditions**

Test for nulls with the IS NULL operator.

**SELECT ename, mgr FROM emp where mgr IS NULL;**

**Logical Conditions**

|  |  |
| --- | --- |
| Meaning | Operator |
| AND | Returns TRUE if *both* component  conditions are true |
| OR | Returns TRUE If *either* component  condition is true |
| NOT | Returns TRUE if the following  condition is false |

**Using the AND Operator**

**SELECT empno, ename, job, sal FROM emp WHERE sal >=10000**

**AND job LIKE '%MAN%';**

**Using the OR Operator**

**SELECT empno, ename, job, sal FROM emp WHERE salary >= 10000**

**OR job LIKE '%MAN%';**

**Using the NOT Operator**

**SELECT ename, job FROM emp**

**WHERE job NOT IN ('ANALYST', 'ST\_CLERK', 'MANAGER');**

**Rules of Precedence**

**Order Evaluated Operator**

**1** Arithmetic operators

2 Concatenation operators

3 Comparison conditions

4 IS [NOT] NULL , LIKE , [NOT] IN

5 [NOT] BETWEEN

6 NOT logical condition

7 AND logical condition

8 OR logical condition

Override rules of precedence by using parentheses.

S**ELECT ename, job, sal FROM emp WHERE job = 'CLERK'**

**OR job = 'MANAGER' AND sal > 15000;**

Use parentheses to force priority.

**SELECT ename, job, sal**

**FROM empl**

**WHERE (job = 'CLERK' OR job = 'MANAGER') AND salary > 15000;**

**ORDER BY Clause**

**Syntax**

SELECT *expr* FROM *table* [WHERE *condition(s)* ] [ORDER BY {*column*,

*Expr* } [ASC|DESC]];

1. **Sort rows with the ORDER BY clause**

**ASC : ascending order (the default order)**

**DESC : descending order**

1. **The ORDER BY clause comes last in the SELECT**

**statement.**

**SELECT ename, job, deptno, hiredate FROM emp ORDER BY hiredate;**

**Sorting in Descending Order**

**SELECT ename, job, deptno, hiredate FROM emp ORDER BY hiredate DESC;**

**Sorting by Column Alias**

**SELECT empno, ename, sal\*12 annsal FROM emp ORDER BY annsal;**

**Sorting by Column Aliases**

You can use a column alias in the ORDER BY clause. The slide example

sorts the data by annual salary.

**Sorting by Multiple Columns**

1. **The order of ORDER BY list is the order of sort.**

**SELECT ename, deptno, sal FROM emp**

**ORDER BY deptno, sal DESC;**

1. **You can sort by a column that is not in the SELECT list.**

**Simple Tasks**

1. Create a query to display the name and salary of employees earning more than $4000.

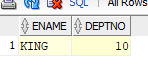
Place your SQL statement in a text file named lab2\_1.sql . Run your query.





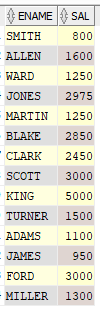
2. Create a query to display the employee name and department number for employee number 7839.





3. Modify lab2\_1.sql to display the name and salary for all employees whose salary is not in the range of $5,000 and $12,000. Place your SQL statement in a text file named lab2\_3.sql





4. Display the employee name, job , and hiredate of employees hired between February 20, 1998, and May 1, 1998. Order the query in ascending order by hiredate.



5. Display the name and department number of all employees in departments 20 and 30 in alphabetical order by name.



6. Modify lab2\_3.sql to list the name and salary of employees who earn between $5,000 and $12,000, and are in department 20 or 50. Label the columns Employee and Monthly Salary , respectively. Resave lab2\_3.sql as lab2\_6.sql . Run the statement in lab2\_6.sql .

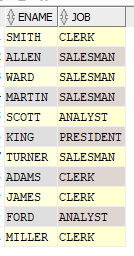


7. Display the name and hire date of every employee who was hired in 1994.

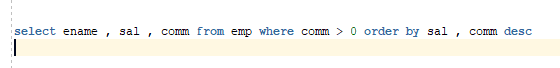


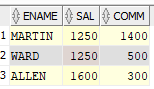
8. Display the name and job title of all employees who do not have a manager.





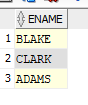
9. Display the name, salary, and commission for all employees who earn commissions. Sort data in descending order of salary and commissions.



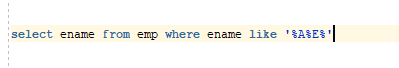


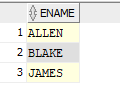
10. Display the names of all employees where the third letter of the name is an *a.*





11. Display the name of all employees who have an *a* and an *e* in their last name.





12. Display the employee name, job, and salary for all employees whose job is salesman or clerk and whose salary is not equal to $2,500, $3,500, or $800.



