

2

Restricting and Sorting Data

Objectives

At the end of this lesson, you should be able to:

- **Limit the rows retrieved by a query**
- **Sort the rows retrieved by a query**

Limiting Rows Using a Selection

EMP

EMPNO	ENAME	JOB	...	DEPTNO
7839	KING	PRESIDENT		10
7698	BLAKE	MANAGER		30
7782	CLARK	MANAGER		10
7566	JONES	MANAGER		20
...				

“...retrieve all employees in department 10”



EMP

EMPNO	ENAME	JOB	...	DEPTNO
7839	KING	PRESIDENT		10
7782	CLARK	MANAGER		10
7934	MILLER	CLERK		10

Limiting Rows Selected

- Restrict the rows returned by using the **WHERE** clause.

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

- The **WHERE** clause follows the **FROM** clause.

Using the WHERE Clause

```
SQL> SELECT ename, job, deptno  
2 FROM emp  
3 WHERE job='CLERK';
```

ENAME	JOB	DEPTNO
JAMES	CLERK	30
SMITH	CLERK	20
ADAMS	CLERK	20
MILLER	CLERK	10

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
- Default date format is 'DD-MON-YY'

```
SQL> SELECT   ename, job, deptno
2  FROM      emp
3  WHERE     ename = 'JAMES' ;
```

Comparison Operators

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

Using the Comparison Operators

```
SQL> SELECT  ename, sal, comm
  2  FROM    emp
  3  WHERE   sal<=comm;
```

ENAME	SAL	COMM
MARTIN	1250	1400

A red double-headed arrow is positioned between the SAL and COMM values for MARTIN, indicating that the salary is less than the commission.

Other Comparison Operators

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

Using the BETWEEN Operator

Use the BETWEEN operator to display rows based on a range of values.

```
SQL> SELECT  ename, sal
      2 FROM    emp
      3 WHERE  sal BETWEEN 1000 AND 1500;
```

ENAME	SAL	Lower limit	Higher limit
MARTIN	1250		
TURNER	1500		
WARD	1250		
ADAMS	1100		
MILLER	1300		

Using the IN Operator

Use the IN operator to test for values in a list.

```
SQL> SELECT empno, ename, sal, mgr
2 FROM emp
3 WHERE mgr IN (7902, 7566, 7788);
```

EMPNO	ENAME	SAL	MGR
7902	FORD	3000	7566
7369	SMITH	800	7902
7788	SCOTT	3000	7566
7876	ADAMS	1100	7788

Using the LIKE Operator

- Use the LIKE operator to perform wildcard searches of valid search string values.
- Search conditions can contain either literal characters or numbers.
 - (%) denotes zero or many characters
 - (_) denotes one character

```
SQL> SELECT   ename
2  FROM      emp
3  WHERE     ename LIKE 'S%';
```

Using the LIKE Operator

- You can combine pattern matching characters.

```
SQL> SELECT   ename
      2 FROM     emp
      3 WHERE    ename LIKE '_A%';
```

ENAME

JAMES

WARD

- You can use the ESCAPE identifier to search for “%” or “_”.

Using the IS NULL Operator

Test for null values with the IS NULL operator

```
SQL> SELECT  ename, mgr
      2 FROM    emp
      3 WHERE  mgr IS NULL;
```

ENAME	MGR
-----	-----
KING	

Logical Operators

Operator	Meaning
AND	Returns TRUE if <i>both</i> component conditions are TRUE
OR	Returns TRUE if <i>either</i> component condition is TRUE
NOT	Returns TRUE if the following condition is FALSE

Using the AND Operator

AND requires both conditions to be TRUE.

```
SQL> SELECT empno, ename, job, sal
  2  FROM    emp
  3  WHERE   sal >= 1100
  4  AND     job = 'CLERK';
```

EMPNO	ENAME	JOB	SAL
7876	ADAMS	CLERK	1100
7934	MILLER	CLERK	1300

Using the OR Operator

OR requires either condition to be TRUE.

```
SQL> SELECT empno, ename, job, sal
  2  FROM    emp
  3  WHERE   sal >= 1100
  4  OR     job = 'CLERK' ;
```

EMPNO	ENAME	JOB	SAL
7839	KING	PRESIDENT	5000
7698	BLAKE	MANAGER	2850
7782	CLARK	MANAGER	2450
7566	JONES	MANAGER	2975
7654	MARTIN	SALESMAN	1250

...

14 rows selected.

Using the NOT Operator

```
SQL> SELECT ename, job
  2   FROM emp
  3   WHERE job NOT IN ('CLERK', 'MANAGER', 'ANALYST');
```

ENAME	JOB
-----	-----
KING	PRESIDENT
MARTIN	SALESMAN
ALLEN	SALESMAN
TURNER	SALESMAN
WARD	SALESMAN

Rules of Precedence

Order Evaluated	Operator
1	All comparison operators
2	NOT
3	AND
4	OR

Override rules of precedence by using parentheses.

Rules of Precedence

```
SQL> SELECT ename, job, sal
  2 FROM emp
  3 WHERE job='SALESMAN'
  4 OR job='PRESIDENT'
  5 AND sal>1500;
```

ENAME	JOB	SAL
KING	PRESIDENT	5000
MARTIN	SALESMAN	1250
ALLEN	SALESMAN	1600
TURNER	SALESMAN	1500
WARD	SALESMAN	1250

Rules of Precedence

Use parentheses to force priority.

```
SQL> SELECT      ename, job, sal
  2  FROM          emp
  3  WHERE         (job='SALESMAN'
  4  OR           job='PRESIDENT')
  5  AND          sal>1500;
```



ENAME	JOB	SAL
-----	-----	-----
KING	PRESIDENT	5000
ALLEN	SALESMAN	1600

ORDER BY Clause

- Sort rows with the ORDER BY clause
 - ASC: ascending order, default
 - DESC: descending order
- The ORDER BY clause comes last in the SELECT statement.

```
SQL> SELECT      ename, job, deptno, hiredate
  2  FROM          emp
  3  ORDER BY hiredate;
```

ENAME	JOB	DEPTNO	HIREDATE
SMITH	CLERK	20	17-DEC-80
ALLEN	SALESMAN	30	20-FEB-81
...			

14 rows selected.

Sorting in Descending Order

```
SQL> SELECT      ename, job, deptno, hiredate
  2  FROM          emp
  3  ORDER BY hiredate DESC;
```

ENAME	JOB	DEPTNO	HIREDATE
ADAMS	CLERK	20	12-JAN-83
SCOTT	ANALYST	20	09-DEC-82
MILLER	CLERK	10	23-JAN-82
JAMES	CLERK	30	03-DEC-81
FORD	ANALYST	20	03-DEC-81
KING	PRESIDENT	10	17-NOV-81
MARTIN	SALESMAN	30	28-SEP-81
...			

14 rows selected.

Sorting by Column Alias

```
SQL> SELECT empno, ename, sal*12 annsal  
2 FROM emp  
3 ORDER BY annsal;
```

EMPNO	ENAME	ANNSAL
7369	SMITH	9600
7900	JAMES	11400
7876	ADAMS	13200
7654	MARTIN	15000
7521	WARD	15000
7934	MILLER	15600
7844	TURNER	18000

...

14 rows selected.

Sorting by Multiple Columns

- The order of ORDER BY list is the order of sort.

```
SQL> SELECT  ename, deptno, sal
      2  FROM    emp
      3  ORDER BY deptno, sal DESC;
```

ENAME	DEPTNO	SAL
-----	-----	-----
KING	10	5000
CLARK	10	2450
MILLER	10	1300
FORD	20	3000

...

14 rows selected.

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

Summary

```
SELECT      [DISTINCT] {*, column [alias], ...}  
FROM        table  
[WHERE      condition(s)]  
[ORDER BY  {column, expr, alias} [ASC|DESC]];
```

Practice Overview

- **Selecting data and changing the order of rows displayed**
- **Restricting rows by using the WHERE clause**
- **Using the double-quotation-marks in column aliases**

Course Overview

**<Enter course-overview information
here>**