

LAB 3: Restricting and Sorting Data.

Objectives

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

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

Lesson Aim

While retrieving data from the database, you may need to restrict the rows of data that are displayed or specify the order in which the rows are displayed. This lesson explains the SQL statements that you use to perform these actions.

Limiting the Rows Selected

You can restrict the rows returned from the query by using the `WHERE` clause. A `WHERE` clause contains a condition that must be met, and it directly follows the `FROM` clause. If the condition is true, the row meeting the condition is returned.

In the syntax:

`WHERE` restricts the query to rows that meet a condition

condition is composed of column names, expressions, constants, and a comparison operator

The `WHERE` clause can compare values in columns, literal values, arithmetic expressions, or functions. It consists of three elements:

- Column name
- Comparison condition
- Column name, constant, or list of values

Using the `WHERE` Clause

```
SELECT employee_id, last_name, job_id, department_id
FROM employees
WHERE department_id = 90;
```

Character Strings and Dates

Character strings and dates in the `WHERE` clause must be enclosed in single quotation marks (' ').

Number constants, however, should not be enclosed in single quotation marks.

Oracle databases store dates in an internal numeric format, representing the century, year, month, day, hours, minutes, and seconds. The default date display is DD-MON-RR.

Comparison Conditions : `=, >, >=, <, <=, <>, !=, ^=`

Using Comparison Conditions

```
SELECT last_name, salary
FROM employees
WHERE salary <= 3000;
```

Using the `BETWEEN` Condition

```
SELECT last_name, salary
FROM employees
WHERE salary BETWEEN 2500 AND 3500;
```

Using the `IN` Condition

```
SELECT employee_id, last_name, salary, manager_id
FROM employees
WHERE manager_id IN (100, 101, 201);
```

Using the LIKE Condition

```
SELECT first_name
FROM employees
WHERE first_name LIKE 'S%';
```

Using the NULL Conditions

```
SELECT last_name, manager_id
FROM employees
WHERE manager_id IS NULL;
```

Logical Conditions: AND, OR, NOT**Using the AND Operator**

```
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE salary >= 10000
AND job_id LIKE '%MAN%';
```

Using the OR Operator

```
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE salary >= 10000
OR job_id LIKE '%MAN%';
```

Using the NOT Operator

```
SELECT last_name, job_id
FROM employees
WHERE job_id NOT IN ('IT_PROG', 'ST_CLERK', 'SA_REP');
```

ORDER BY Clause

```
SELECT last_name, job_id, department_id, hire_date
FROM employees
ORDER BY hire_date;
```

Sorting in Descending Order

```
SELECT last_name, job_id, department_id, hire_date
FROM employees
ORDER BY hire_date DESC;
```

Sorting by Column Alias

```
SELECT employee_id, last_name, salary*12 annsal
FROM employees
ORDER BY annsal;
```

Sorting by Multiple Columns

```
SELECT last_name, department_id, salary
FROM employees
ORDER BY department_id, salary DESC;
```

Practice

1. Create a query to display the last name and salary of employees earning more than \$12,000. Place your SQL statement in a text file named lab2_1.sql. Run your query.

LAST_NAME	SALARY
King	24000
Kochhar	17000
De Haan	17000
Hartstein	13000

2. Create a query to display the employee last name and department number for employee no= 176.

LAST_NAME	DEPARTMENT_ID
Taylor	80

3. Modify lab2_1.sql to display the last 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.

LAST_NAME	SALARY
King	24000
Kochhar	17000
De Haan	17000
Lorentz	4200
Rajs	3500
Davies	3100
Matos	2600
Vargas	2500
Whalen	4400
Hartstein	13000

10 rows selected.

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

LAST_NAME	JOB_ID	HIRE_DATE
Matos	ST_CLERK	15-MAR-98
Taylor	SA_REP	24-MAR-98

5. Display the last name and department number of all employees in departments 20 and 50 in alphabetical order by name.

LAST_NAME	DEPARTMENT_ID
Davies	50
Fay	20
Hartstein	20
Matos	50
Mourgos	50
Rajs	50
Vargas	50

7 rows selected.

6. Modify lab2_3.sql to list the last 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.

Employee	Monthly Salary
Mourgos	5800
Fay	6000

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

LAST_NAME	HIRE_DATE
Higgins	07-JUN-94
Gietz	07-JUN-94

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

LAST_NAME	JOB_ID
King	AD_PRES

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

LAST_NAME	SALARY	COMMISSION_PCT
Abel	11000	.3
Zlotkey	10500	.2
Taylor	8600	.2
Grant	7000	.15

Employee	Monthly Salary	COMMISSION_PCT
Zlotkey	10500	.2
Taylor	8600	.2