

LAB 2: Writing Basic SQL SELECT Statements.

Objectives

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

- ✓ List the capabilities of SQL SELECT statements
- ✓ Execute a basic SELECT statement

Lesson Aim

To extract data from the database, you need to use the structured query language (SQL) SELECT statement. You may need to restrict the columns that are displayed. This lesson describes all the SQL statements that you need to perform these actions.

Writing SQL Statements

- SQL statements are not case sensitive.
- SQL statements can be on one or more lines.
- Keywords cannot be abbreviated or split across lines

Basic SELECT Statement

In its simplest form, a **SELECT** statement must include the following:

- A **SELECT** clause, which specifies the columns to be displayed
- A **FROM** clause, which specifies the table containing the columns listed in the **SELECT** clause

In the syntax:

SELECT (is a list of one or more columns) ***** (selects all columns) **DISTINCT** (suppresses duplicates) *column|expression* (selects the named column or the expression *alias* gives selected columns different headings **FROM** *table* specifies the table containing the columns

Note: Throughout this course, the words *keyword*, *clause*, and *statement* are used as follows.

A *keyword* refers to an individual SQL element. For example, **SELECT** and **FROM** are keywords.

A *clause* is a part of a SQL statement. For example, **SELECT** *employee_id, last_name, ...* is a clause.

A *statement* is a combination of two or more clauses. For example, **SELECT * FROM employees** is a SQL statement

Selecting All Columns

```
SELECT * FROM departments;
```

Selecting Specific Columns

```
SELECT department_id, location_id FROM departments;
```

Using Arithmetic Operators

```
SELECT last_name, salary, salary + 300 FROM employees;
```

Operator Precedence

```
SELECT last_name, salary, 12*salary+100 FROM employees;
```

Using Parentheses

```
SELECT last_name, salary, 12*(salary+100) FROM employees;
```

You can override the rules of precedence by using parentheses to specify the order in which operators are executed. The example in the slide displays the last name, salary, and annual compensation of each employee. It calculates the annual compensation as monthly salary plus a monthly bonus of \$100, multiplied by 12. Because of the parentheses, addition takes priority over multiplication.

Defining a Null Value

- A null is a value that is unavailable, unassigned, unknown, or inapplicable.
- A null is not the same as zero or a blank space.

Null Values in Arithmetic Expressions

```
SELECT last_name, 12*salary*commission_pct FROM employees;
```

Using Column Aliases

```
SELECT last_name AS name, commission_pct comm FROM employees;
```

Using the Concatenation Operator

```
SELECT last_name || ' is a ' || job_id AS "Employee Details"
FROM employees;
```

Eliminating Duplicate Rows

```
SELECT DISTINCT department_id
FROM employees;
```

Displaying Table Structure

```
DESCRIBE employees
```

In the result:

Null? Indicates whether a column *must* contain data; NOT NULL indicates that a column must contain data

Type displays the data type for a column The data types are described in the following table:

Data	Type Description
NUMBER (<i>p</i> , <i>s</i>)	Number value having a maximum number of digits <i>p</i> , with <i>s</i> digits to the right of the decimal point
VARCHAR2 (<i>s</i>)	Variable-length character value of maximum size <i>s</i>
DATE	Date and time value between January 1, 4712 B.C., and A.D. December 31, 9999
CHAR (<i>s</i>)	Fixed-length character value of size <i>s</i>

Practice Overview

This practice covers the following topics:

- ✓ Selecting all data from different tables
- ✓ Describing the structure of tables
- ✓ Performing arithmetic calculations and specifying column names

1. The following SELECT statement executes successfully:

```
SELECT last_name, job_id, salary AS Sal
FROM employees;
```

True/False

2. The following SELECT statement executes successfully:

```
SELECT *
FROM job_grades;
```

True/False

3. There are four coding errors in this statement. Can you identify them?

```
SELECT employee_id, last_name
sal x 12 ANNUAL SALARY
FROM employees;
```

4. Show the structure of the DEPARTMENTS table. Select all data from the table.

- ✓ 5. Show the structure of the EMPLOYEES table. Create a query to display the last name, job code, hire date, and employee number for each employee, with employee number appearing first. Save your SQL statement to a file named lab1_7.sql

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(25)
PHONE_NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)

✓ 6. Run your query in the file lab1_7.sql.

EMPLOYEE_ID	LAST_NAME	JOB_ID	StartDate
100	King	AD_PRES	17-JUN-87
101	Kochhar	AD_VP	21-SEP-89
102	De Haan	AD_VP	13-JAN-93
103	Hunold	IT_PROG	03-JAN-90
104	Ernst	IT_PROG	21-MAY-91
107	Lorentz	IT_PROG	07-FEB-99
124	Mourgos	ST_MAN	16-NOV-99
141	Rajs	ST_CLERK	17-OCT-95
142	Davies	ST_CLERK	29-JAN-97
143	Matos	ST_CLERK	15-MAR-98
144	Vargas	ST_CLERK	09-JUL-98
149	Zlotkey	SA_MAN	29-JAN-00

- ✓ 7. Create a query to display unique job codes from the `EMPLOYEES` table.
- ✓ 8. Display the last name concatenated with the job ID, separated by a comma and space, and name the column `Employee and Title`.

Employee and Title
King, AD_PRES
Kochhar, AD_VP
De Haan, AD_VP
Hunold, IT_PROG
Ernst, IT_PROG
Lorentz, IT_PROG
Mourgos, ST_MAN
Rajs, ST_CLERK
Davies, ST_CLERK

If you want an extra challenge, complete the following exercise:

- ✓ 9. Create a query to display all the data from the `EMPLOYEES` table. Separate each column by a comma. Name the column `THE_OUTPUT`.

THE_OUTPUT
100,Steven,King,SKING,515.123.4567,AD_PRES,,17-JUN-87,24000,,90
101,Neena,Kochhar,NKOCHHAR,515.123.4568,AD_VP,100,21-SEP-89,17000,,90
102,Lex,De Haan,LDEHAAN,515.123.4569,AD_VP,100,13-JAN-93,17000,,90
103,Alexander,Hunold,AHUNOLD,590.423.4567,IT_PROG,102,03-JAN-90,9000,,60
104,Bruce,Ernst,BERNST,590.423.4568,IT_PROG,103,21-MAY-91,6000,,60
107,Diana,Lorentz,DLORENTZ,590.423.5567,IT_PROG,103,07-FEB-99,4200,,60
124,Kevin,Mourgos,KMOURGOS,650.123.5234,ST_MAN,100,16-NOV-99,5800,,50
141,Trenna,Rajs,TRAJS,650.121.8009,ST_CLERK,124,17-OCT-95,3500,,50