

1

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**
- **Differentiate between SQL statements and *iSQL*Plus* commands**

Capabilities of SQL SELECT Statements

Projection

Table 1

Selection

Table 1

Table 1

Join



Table 2

Basic SELECT Statement

```
SELECT    * | { [DISTINCT] column | expression [alias] , ... }  
FROM      table;
```

- **SELECT** identifies *what* columns
- **FROM** identifies *which* table

Selecting All Columns

```
SELECT *  
FROM departments;
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

8 rows selected.

Selecting Specific Columns

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

DEPARTMENT_ID	LOCATION_ID
10	1700
20	1800
50	1500
60	1400
80	2500
90	1700
110	1700
190	1700

8 rows selected.

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.**
- **Clauses are usually placed on separate lines.**
- **Indents are used to enhance readability.**

Column Heading Defaults

- ***i*SQL*Plus:**
 - Default heading justification: Center
 - Default heading display: Uppercase
- **SQL*Plus:**
 - Character and Date column headings are left-justified
 - Number column headings are right-justified
 - Default heading display: Uppercase

Arithmetic Expressions

Create expressions with number and date data by using arithmetic operators.

Operator	Description
+	Add
-	Subtract
*	Multiply
/	Divide

Using Arithmetic Operators

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

LAST_NAME	SALARY	SALARY+300
King	24000	24300
Kochhar	17000	17300
De Haan	17000	17300
Hunold	9000	9300
Ernst	6000	6300
...		
Hartstein	13000	13300
Fay	6000	6300
Higgins	12000	12300
Gietz	8300	8600

20 rows selected.

Operator Precedence



- **Multiplication and division take priority over addition and subtraction.**
- **Operators of the same priority are evaluated from left to right.**
- **Parentheses are used to force prioritized evaluation and to clarify statements.**

Operator Precedence

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

LAST_NAME	SALARY	12*SALARY+100
King	24000	288100
Kochhar	17000	204100
De Haan	17000	204100
Hunold	9000	108100
Ernst	6000	72100

...

Hartstein	13000	156100
Fay	6000	72100
Higgins	12000	144100
Gietz	8300	99700

20 rows selected.

Using Parentheses

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

LAST_NAME	SALARY	12*(SALARY+100)
King	24000	289200
Kochhar	17000	205200
De Haan	17000	205200
Hunold	9000	109200
Ernst	6000	73200

...

Hartstein	13000	157200
Fay	6000	73200
Higgins	12000	145200
Gietz	8300	100800

20 rows selected.

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.

```
SELECT last_name, job_id, salary, commission_pct  
FROM employees;
```

LAST_NAME	JOB_ID	SALARY	COMMISSION_PCT
King	AD_PRES	24000	
Kochhar	AD_VP	17000	

...

Zlotkey	SA_MAN	10500	.2
Abel	SA_REP	11000	.3
Taylor	SA_REP	8600	.2

...

Gietz	AC_ACCOUNT	8300	
-------	------------	------	--

20 rows selected.

Null Values in Arithmetic Expressions

Arithmetic expressions containing a null value evaluate to null.

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

Kochhar	
King	
LAST_NAME	12*SALARY*COMMISSION_PCT

...

Zlotkey	25200
Abel	39600
Taylor	20640

...

Gietz	
-------	--

20 rows selected.

Defining a Column Alias

A column alias:

- **Renames a column heading**
- **Is useful with calculations**
- **Immediately follows the column name - there can also be the optional AS keyword between the column name and alias**
- **Requires double quotation marks if it contains spaces or special characters or is case sensitive**

Using Column Aliases

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

NAME	COMM
King	
Kochhar	
De Haan	

...

20 rows selected.

```
SELECT last_name "Name", salary*12 "Annual Salary"
FROM employees;
```

Name	Annual Salary
King	288000
Kochhar	204000
De Haan	204000

...

20 rows selected.

Concatenation Operator

A concatenation operator:

- **Concatenates columns or character strings to other columns**
- **Is represented by two vertical bars (||)**
- **Creates a resultant column that is a character expression**

Using the Concatenation Operator

```
SELECT    last_name||job_id AS "Employees"  
FROM      employees;
```

Employees	
KingAD_PRES	
KochharAD_VP	
De HaanAD_VP	
HunoldIT_PROG	
ErnstIT_PROG	
LorentzIT_PROG	
MourgosST_MAN	
RajsST_CLERK	

...

20 rows selected.

Literal Character Strings

- A literal is a character, a number, or a date included in the `SELECT` list.
- Date and character literal values must be enclosed within single quotation marks.
- Each character string is output once for each row returned.

Using Literal Character Strings

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

Employee Details	
King is a AD_PRE	S
Kochhar is a AD_V	P
De Haan is a AD_V	P
Hunold is a IT_PROG	
Ernst is a IT_PROG	
Lorentz is a IT_PROG	
Mourgos is a ST_MAN	
Rajs is a ST_CLERK	

...

20 rows selected.

Duplicate Rows

The default display of queries is all rows, including duplicate rows.

```
SELECT department_id  
FROM   employees;
```

DEPARTMENT_ID	
	90
	90
	90
	60
	60
	60
	50
	50
	50

...

20 rows selected.

Eliminating Duplicate Rows

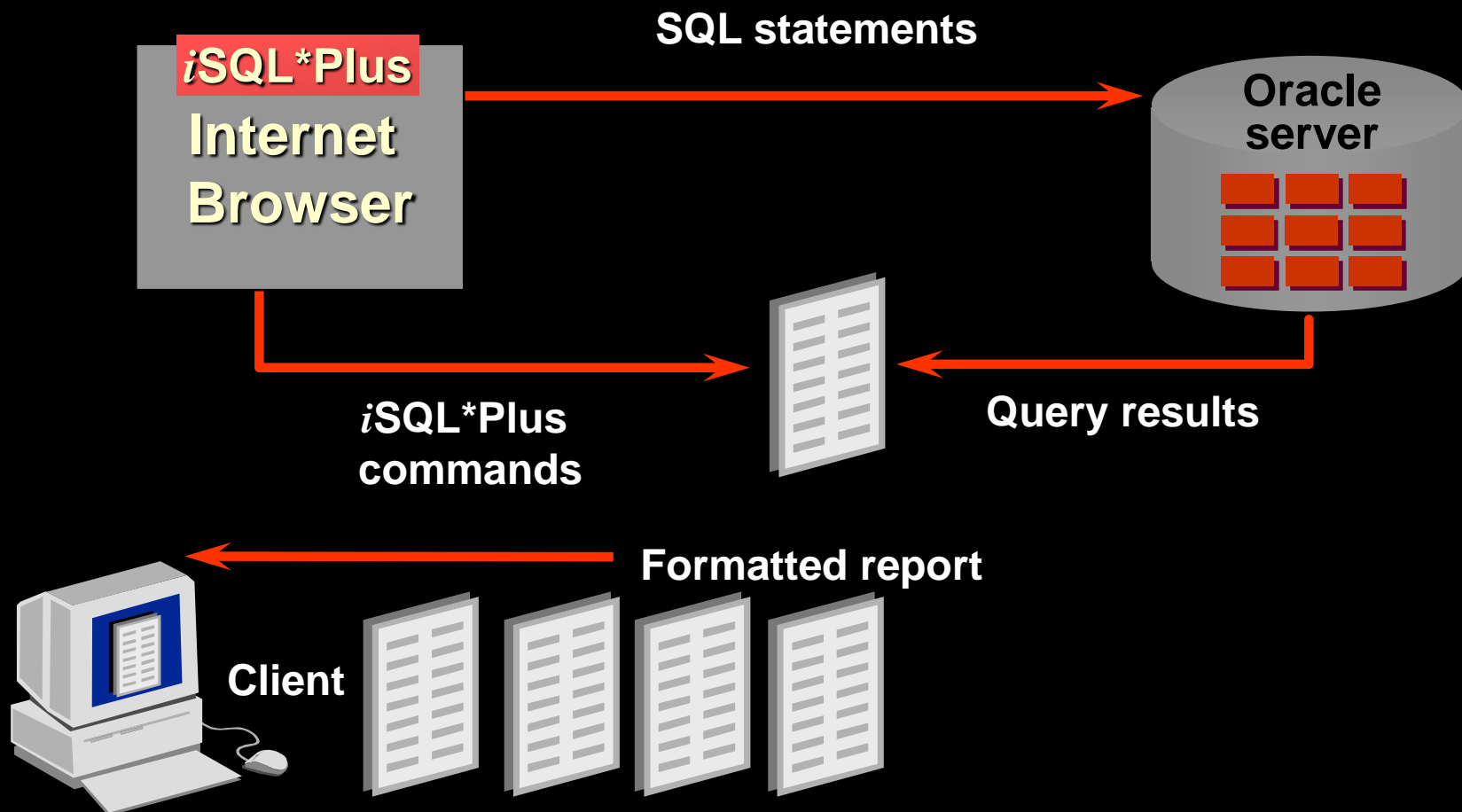
Eliminate duplicate rows by using the **DISTINCT** keyword in the **SELECT** clause.

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

DEPARTMENT_ID	
	10
	20
	50
	60
	80
	90
	110

8 rows selected.

SQL and *i*SQL*Plus Interaction



SQL Statements Versus *i*SQL*Plus Commands

SQL

- A language
- ANSI standard
- Keyword cannot be abbreviated
- Statements manipulate data and table definitions in the database

**SQL
statements**

*i*SQL*Plus

- An environment
- Oracle proprietary
- Keywords can be abbreviated
- Commands do not allow manipulation of values in the database
- Runs on a browser
- Centrally loaded, does not have to be implemented on each machine

***i*SQL*Plus
commands**

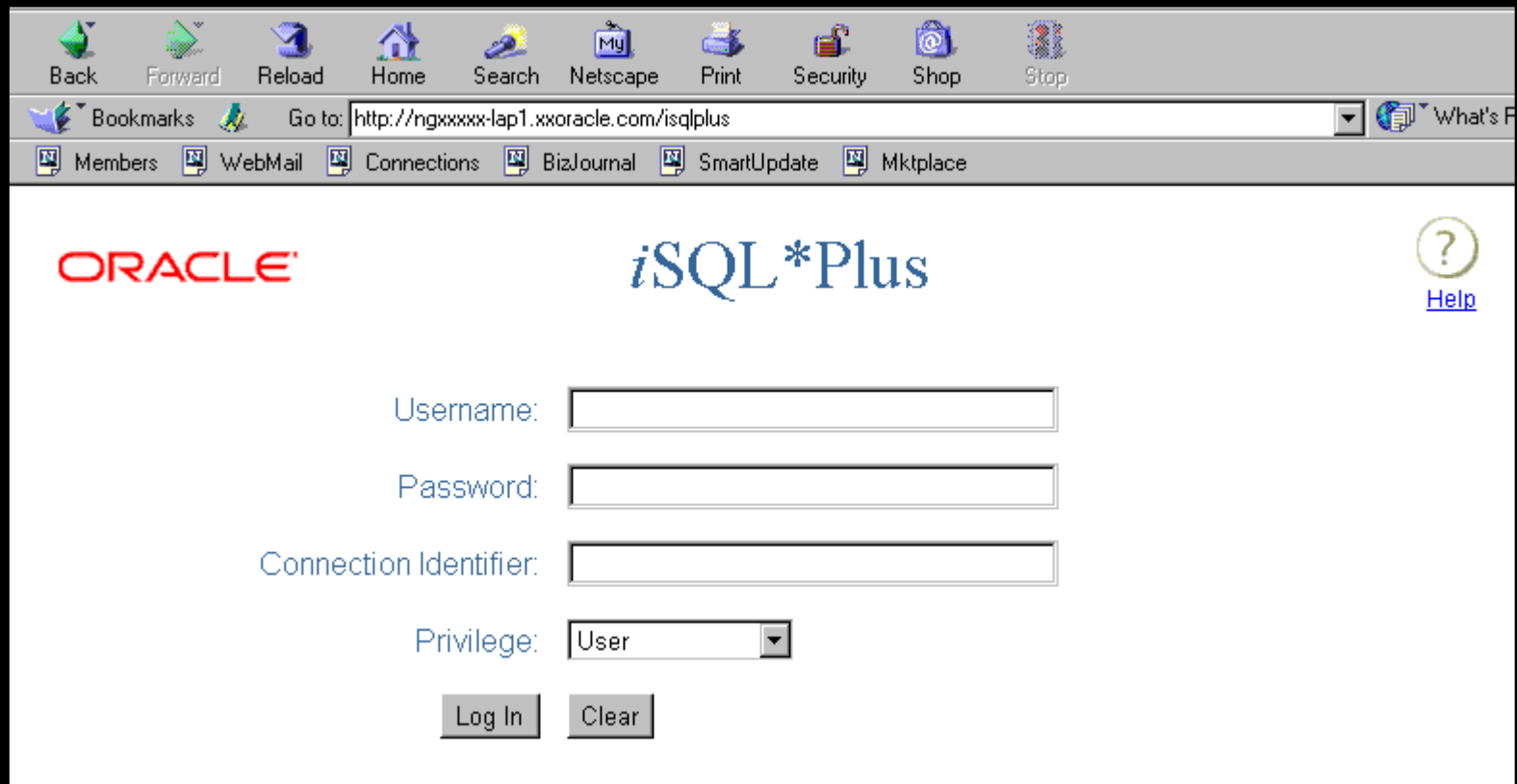
Overview of *iSQL*Plus*

After you log into *iSQL*Plus*, you can:

- Describe the table structure
- Edit your SQL statement
- Execute SQL from *iSQL*Plus*
- Save SQL statements to files and append SQL statements to files
- Execute statements stored in saved files
- Load commands from a text file into the *iSQL*Plus* Edit window

Logging In to *iSQL*Plus*

From your Windows browser environment:



The screenshot shows a Netscape browser window with the address bar set to `http://ngxxxx-lap1.xxoracle.com/isqlplus`. The browser's toolbar includes buttons for Back, Forward, Reload, Home, Search, Netscape, Print, Security, Shop, and Stop. Below the toolbar is a bookmarks bar with links to Members, WebMail, Connections, BizJournal, SmartUpdate, and Mktplace. The main content area displays the Oracle logo on the left and the *iSQL*Plus* title in the center. On the right, there is a help icon (a question mark in a circle) with a [Help](#) link below it. The login form consists of four labeled input fields: Username, Password, and Connection Identifier, each followed by a text box. Below these is a Privilege dropdown menu currently showing 'User'. At the bottom of the form are two buttons: 'Log In' and 'Clear'.

ORACLE[®] *iSQL*Plus* [Help](#)

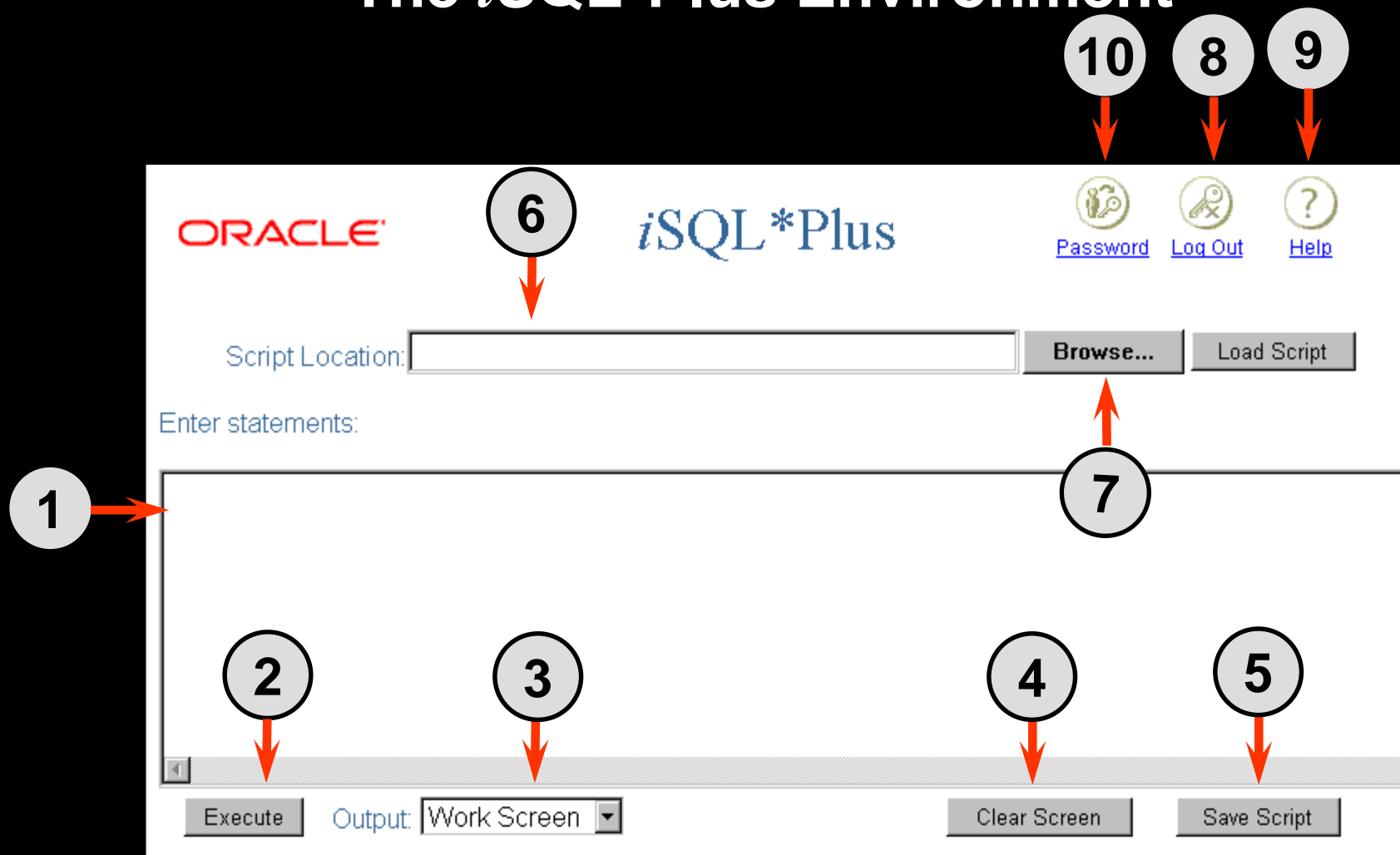
Username:

Password:

Connection Identifier:

Privilege:

The *i*SQL*Plus Environment



Displaying Table Structure

Use the *iSQL*Plus* DESCRIBE command to display the structure of a table.

```
DESC[RIBE] tablename
```

Displaying Table Structure

```
DESCRIBE employees
```

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)

Interacting with Script Files

The screenshot displays the iSQL*Plus web interface. At the top left is the ORACLE logo. In the center is the text 'iSQL*Plus'. On the top right are three circular icons with links: 'Password', 'Log Out', and 'Help'. Below the top bar, there is a 'Script Location:' label followed by a text input field, a 'Browse...' button, and a 'Load Script' button. Underneath is the label 'Enter statements:'. A large text area contains the SQL script: `SELECT last_name, hire_date, salary` on the first line and `FROM employees;` on the second line. A red arrow labeled with a circled '1' points to the end of the first line of the script. Below the script area is a horizontal scrollbar. At the bottom, there is a row of buttons: 'Execute', 'Output: Work Screen' (with a dropdown arrow), 'Clear Screen', and 'Save Script'. A red arrow labeled with a circled '2' points to the 'Save Script' button.

ORACLE[®] iSQL*Plus

[Password](#) [Log Out](#) [Help](#)

Script Location:

Enter statements:

```
SELECT last_name, hire_date, salary
FROM employees;
```

Output:

Interacting with Script Files

The screenshot shows the iSQL*Plus web interface. At the top left is the ORACLE logo. To its right is a circled '1' with a red arrow pointing to the 'Script Location' text box. The text box contains 'D:\temp\emp_sql.htm'. To the right of the text box are 'Browse...' and 'Load Script' buttons. Further right are three icons: a key for 'Password', a key for 'Log Out', and a question mark for 'Help'. Below the 'Script Location' section is the 'Enter statements:' label. Below that is a large text area containing the SQL query: 'SELECT last_name, hire_date, salary FROM employees;'. To the right of this text area is a circled '2' with a red arrow pointing to the 'Load Script' button. At the bottom left of the text area is a circled '3' with a red arrow pointing to the 'Execute' button. At the bottom of the interface are four buttons: 'Execute', 'Output: Work Screen' (with a dropdown arrow), 'Clear Screen', and 'Save Script'.

ORACLE

iSQL*Plus

[Password](#) [Log Out](#) [Help](#)

Script Location:

Enter statements:

```
SELECT last_name, hire_date, salary
FROM employees;
```

[Output:](#)

Interacting with Script Files

The screenshot shows the Oracle iSQL*Plus web interface. At the top, the Oracle logo is on the left, the text 'iSQL*Plus' is in the center, and three circular icons (a person, a key, and a question mark) are on the right, each with a corresponding link: 'Password', 'Log Out', and 'Help'. Below the header, there is a 'Script Location:' label followed by a text input field, a 'Browse...' button, and a 'Load Script' button. Underneath is the 'Enter statements:' label. The main area contains a text editor with the following SQL script: `DESCRIBE employees`, `SELECT first_name, last_name, job_id`, and `FROM employees;`. Three red arrows with circular numbers point to specific elements: arrow 1 points to the end of the `SELECT` statement, arrow 2 points to the 'Execute' button, and arrow 3 points to the 'Output:' dropdown menu. At the bottom, there is an 'Execute' button, an 'Output:' dropdown menu currently set to 'Work Screen', a 'Clear Screen' button, and a 'Save Script' button.

ORACLE[®] iSQL*Plus

[Password](#) [Log Out](#) [Help](#)

Script Location:

Enter statements:

```
DESCRIBE employees
SELECT first_name, last_name, job_id
FROM employees;
```

Output:

Summary

In this lesson, you should have learned how to:

- Write a **SELECT** statement that:
 - Returns all rows and columns from a table
 - Returns specified columns from a table
 - Uses column aliases to give descriptive column headings
- Use the *iSQL*Plus* environment to write, save, and execute SQL statements and *iSQL*Plus* commands.

```
SELECT      * | { [DISTINCT]  column | expression [alias] , ... }  
FROM        table;
```

Practice 1 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**
- **Using *iSQL*Plus***

