

1

Retrieving Data Using the SQL `SELECT` Statement

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 `SQL*Plus` commands**

Capabilities of SQL `SELECT` Statements

Projection

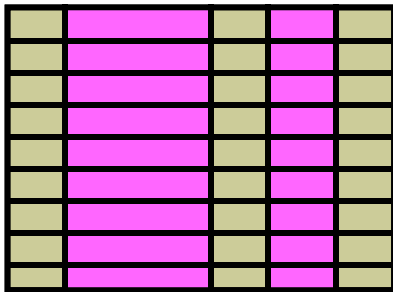


Table 1

Selection

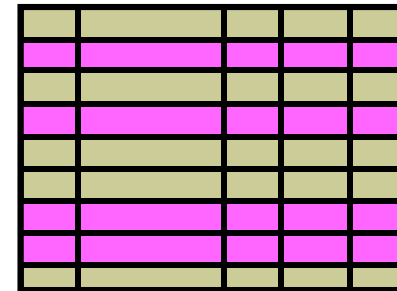


Table 1

Join

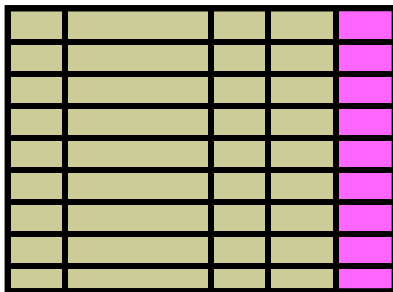


Table 1

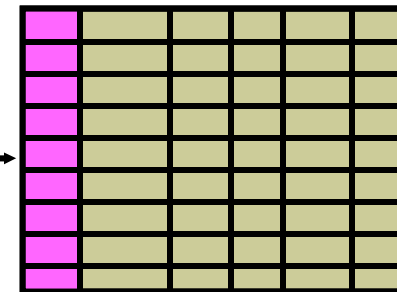


Table 2

Basic SELECT Statement

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

- **SELECT** identifies the columns to be displayed
- **FROM** identifies the table containing those columns

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.
- In **SQL*Plus**, SQL statements can optionally be terminated by a semicolon (;). Semicolons are required if you execute multiple SQL statements.
- In **SQL*plus**, you are required to end each SQL statement with a semicolon (;).

Column Heading Defaults

- **SQL*Plus:**
 - Default heading alignment: Center
 - Default heading display: Uppercase
- **SQL*Plus:**
 - Character and Date column headings are left-aligned
 - Number column headings are right-aligned
 - 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

■ ■ ■

20 rows selected.

Operator Precedence

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

1

LAST_NAME	SALARY	12*SALARY+100
King	24000	288100
Kochhar	17000	204100
De Haan	17000	204100

■ ■ ■
20 rows selected.

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

2

LAST_NAME	SALARY	12*(SALARY+100)
King	24000	289200
Kochhar	17000	205200
De Haan	17000	205200

■ ■ ■
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 a 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;
```

LAST_NAME	12*SALARY*COMMISSION_PCT
King	
Kochhar	
...	
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 if it 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

important

A concatenation operator:

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

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

Employees	
KingAD_PRES	
KochharAD_VP	
De HaanAD_VP	
...	

20 rows selected.

Literal Character Strings

- A literal is a character, a number, or a date that is included in the `SELECT` statement.
- **Date and character literal values must be enclosed by 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_PRES
Kochhar is a AD_VP
De Haan is a AD_VP
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.

Alternative Quote (q) Operator

- Specify your own quotation mark delimiter
- Choose any delimiter
- Increase readability and usability

```
SELECT department_name ||  
       q'[, it's assigned Manager Id: ]'  
       || manager_id  
       AS "Department and Manager"  
FROM departments;
```

Department and Manager
Administration, it's assigned manager ID: 200
Marketing, it's assigned manager ID: 201
Shipping, it's assigned manager ID: 124

...

8 rows selected.

Duplicate Rows

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

```
SELECT department_id
FROM   employees;
```

1

DEPARTMENT_ID	
	90
	90
	90

...

20 rows selected.

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

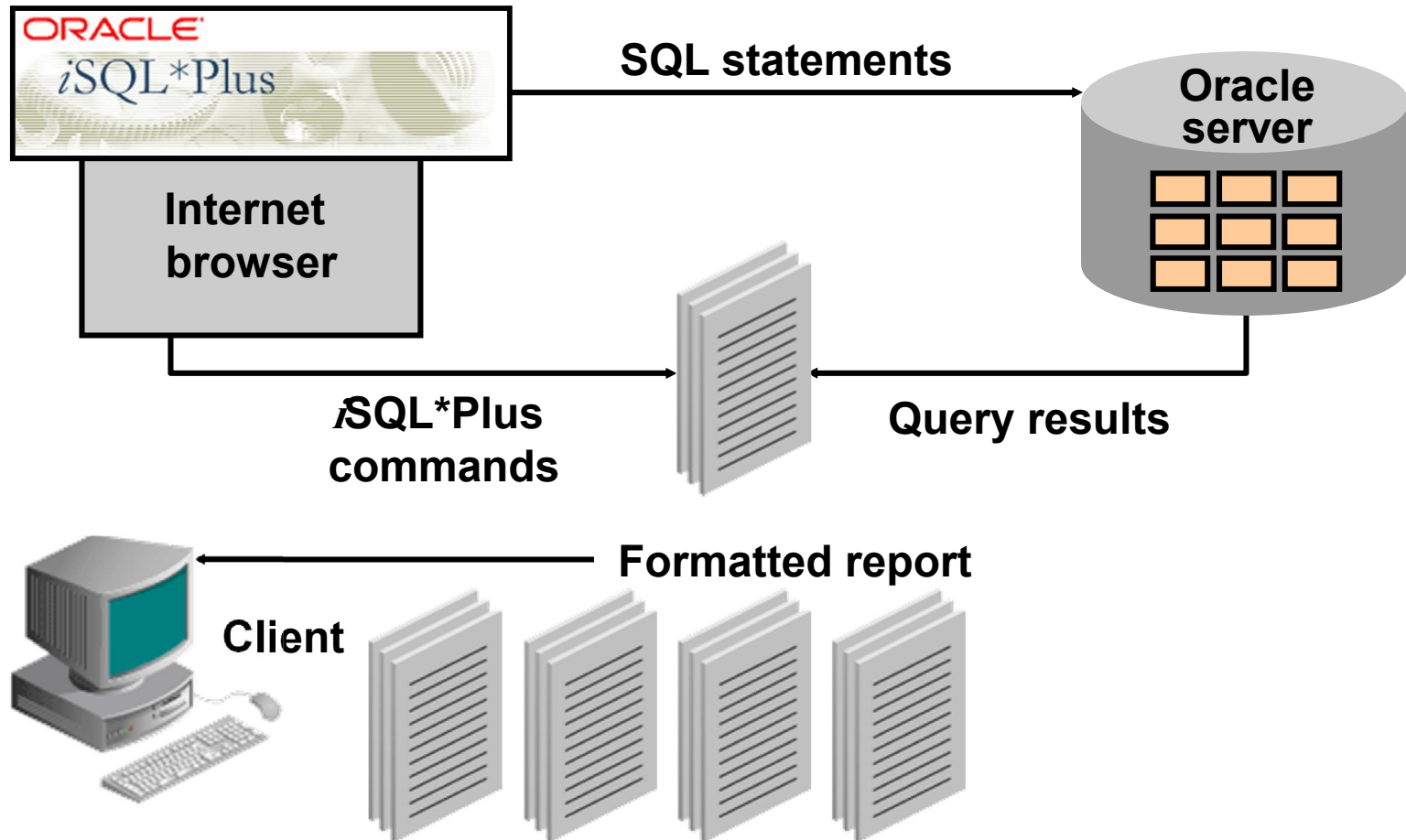
2

DEPARTMENT_ID	
	10
	20
	50

...

8 rows selected.

SQL and iSQL*Plus Interaction



SQL Statements Versus SQL*Plus Commands

SQL

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

**SQL
statements**

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

**SQL*Plus
commands**

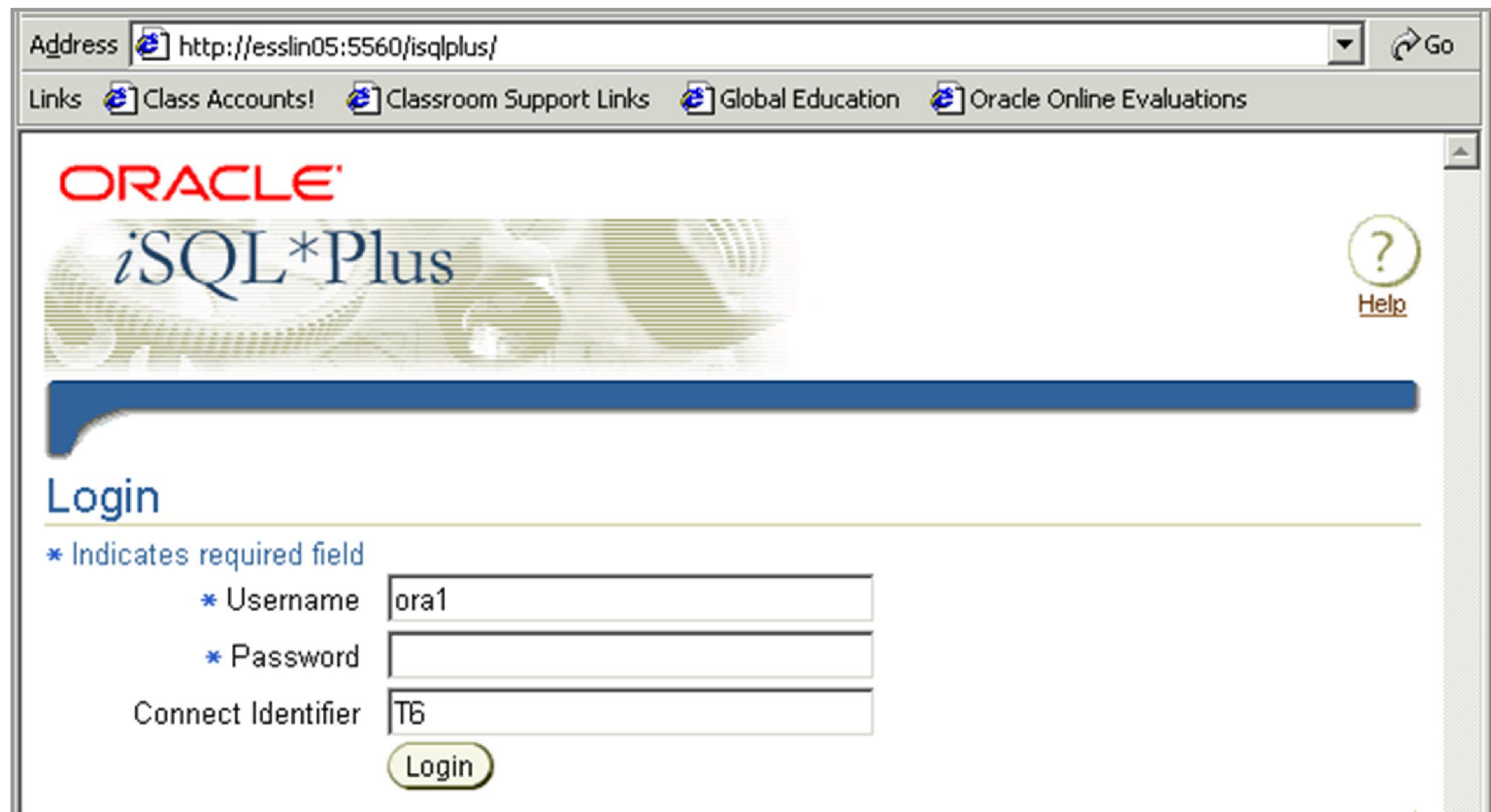
Overview of SQL*Plus

After you log in to SQL*Plus, you can:

- **Describe table structures**
- **Enter, execute, and edit SQL statements**
- **Save or append SQL statements to files**
- **Execute or edit statements that are stored in saved script files**

Logging In to iSQL*Plus

From your browser environment:



The screenshot shows a web browser window with the address bar containing `http://esslin05:5560/isqlplus/`. Below the address bar is a "Links" section with four items: "Class Accounts!", "Classroom Support Links", "Global Education", and "Oracle Online Evaluations". The main content area features the "ORACLE" logo in red and "iSQL*Plus" in blue. A "Help" link with a question mark icon is in the top right. A blue horizontal bar separates the header from the "Login" section. Under "Login", there is a note: "* Indicates required field". The login form includes three fields: "Username" with the value "ora1", "Password" (empty), and "Connect Identifier" with the value "T6". A "Login" button is positioned below the "Connect Identifier" field.

Address <http://esslin05:5560/isqlplus/> Go

Links [Class Accounts!](#) [Classroom Support Links](#) [Global Education](#) [Oracle Online Evaluations](#)

ORACLE
iSQL*Plus

[Help](#)

Login

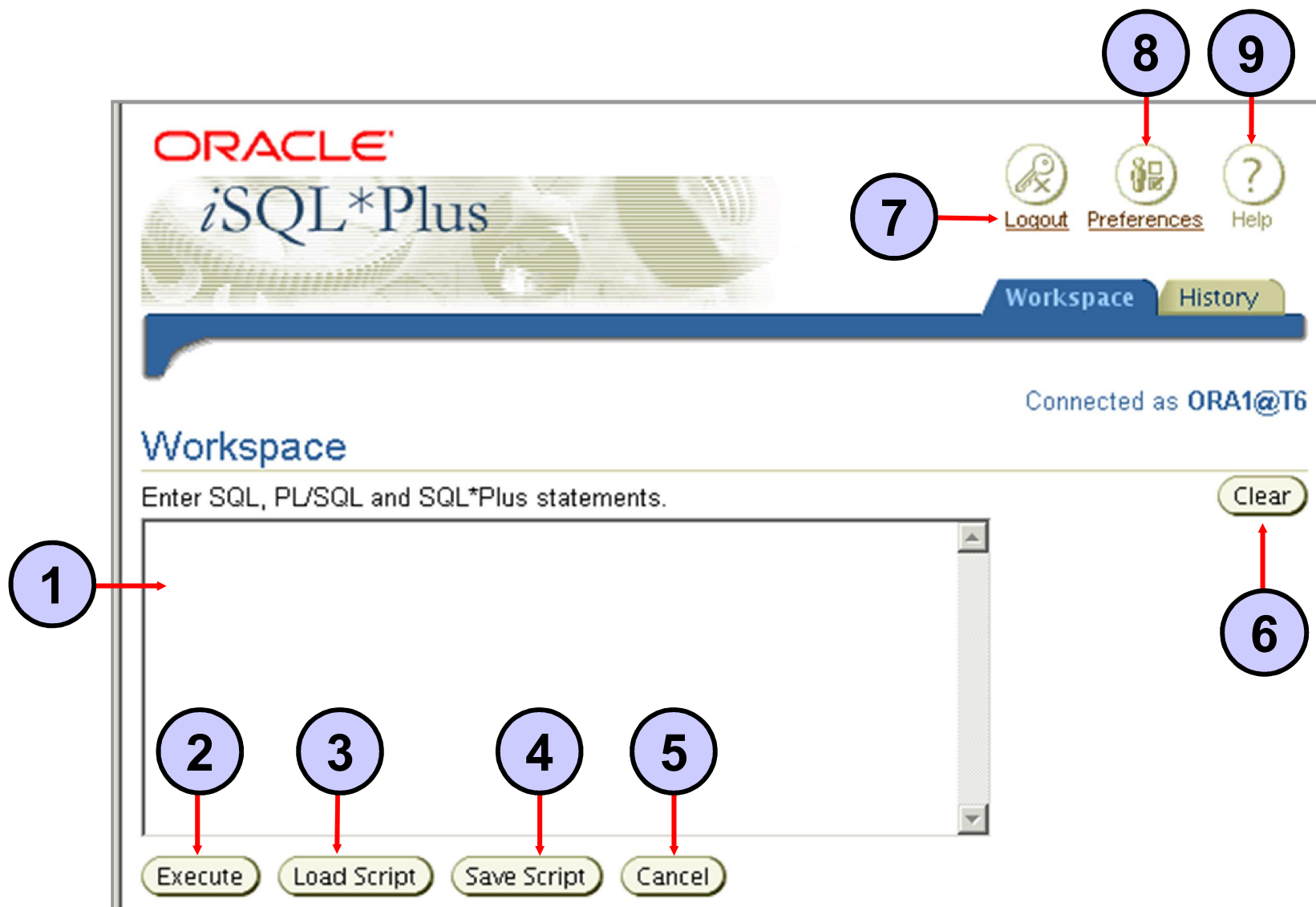
* Indicates required field

* Username

* Password

Connect Identifier

iSQL*Plus Environment



Displaying Table Structure

Use the **SQL*Plus DESCRIBE** command to display the structure of a table:

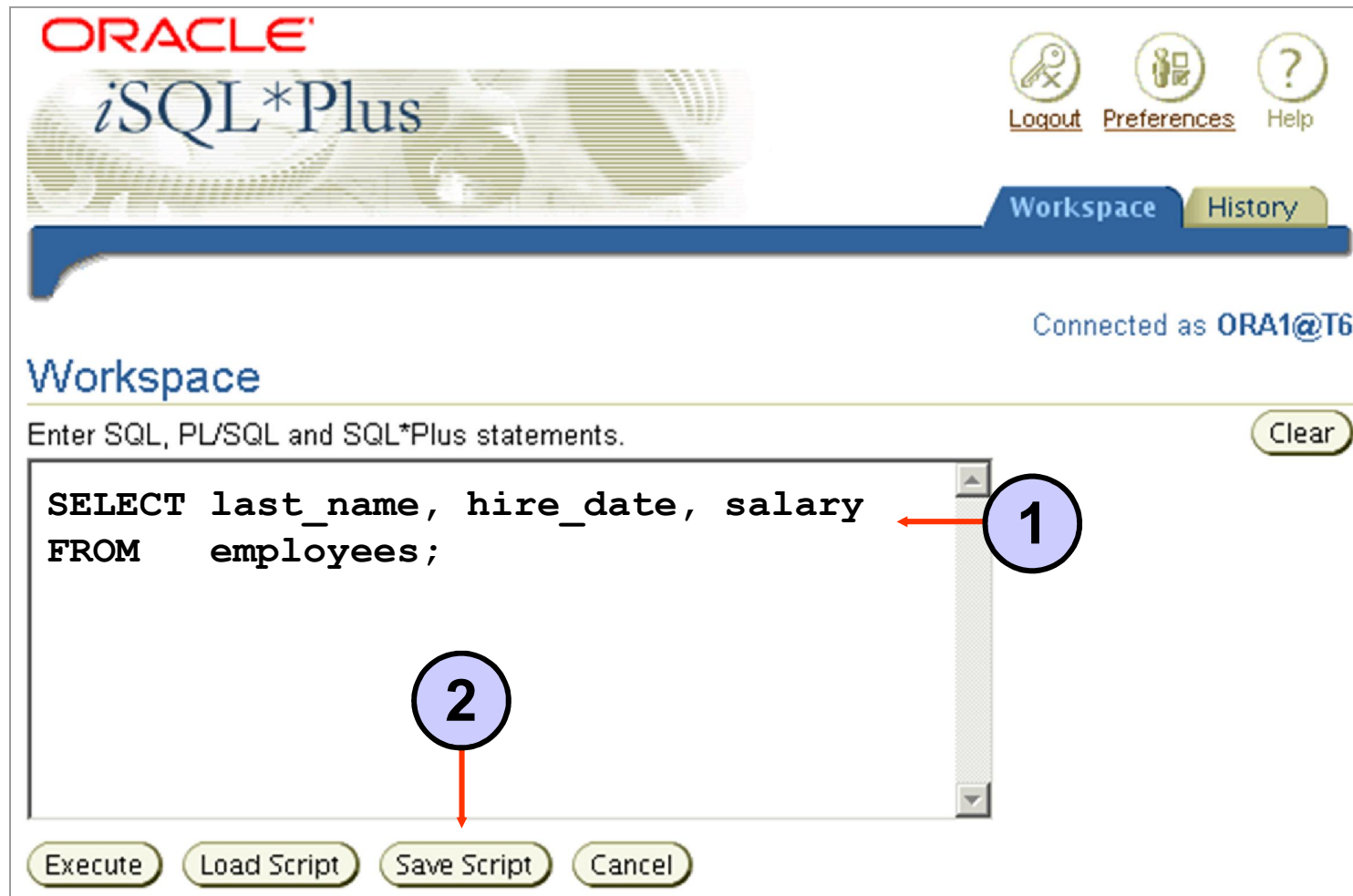
```
DESC[RIBE] tablename
```

Displaying Table Structure

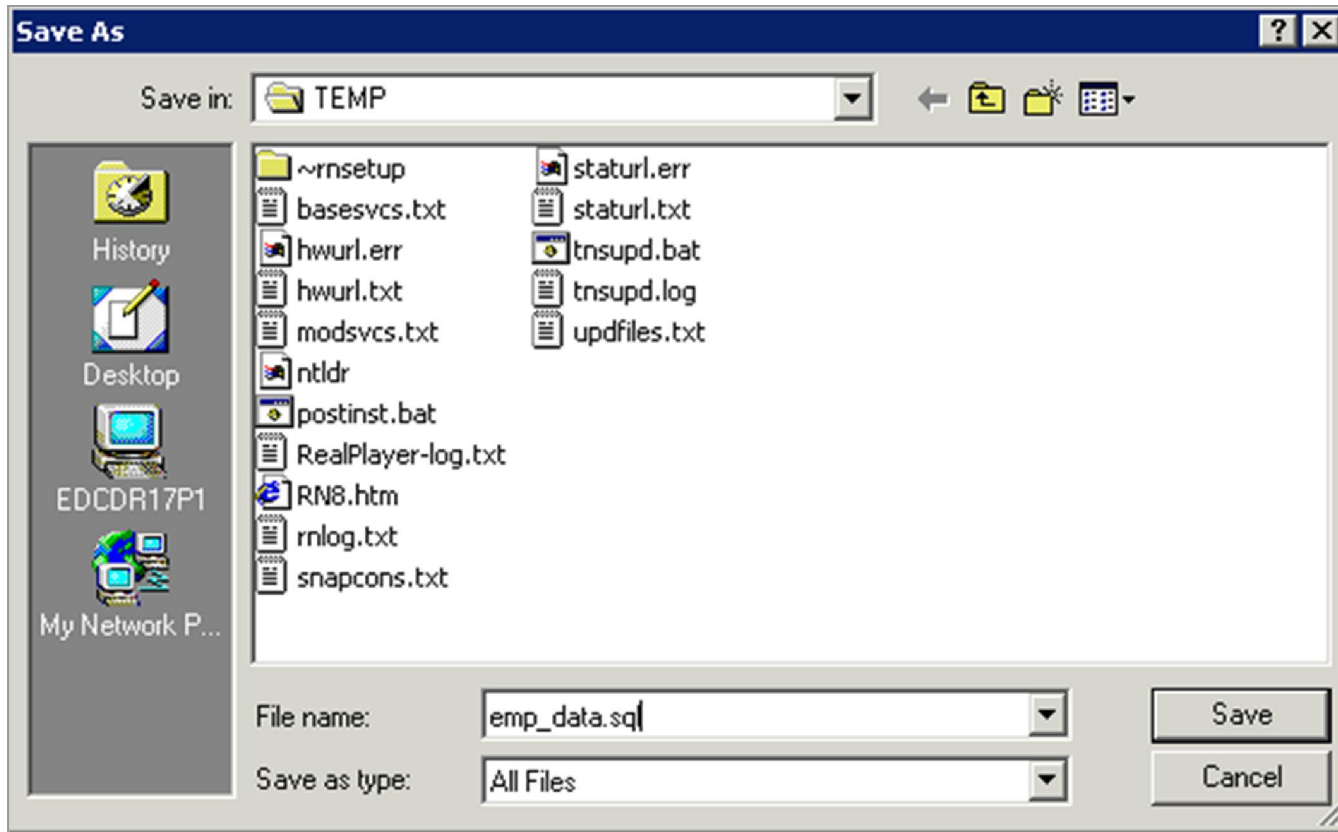
```
DESCRIBE employees
```

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)

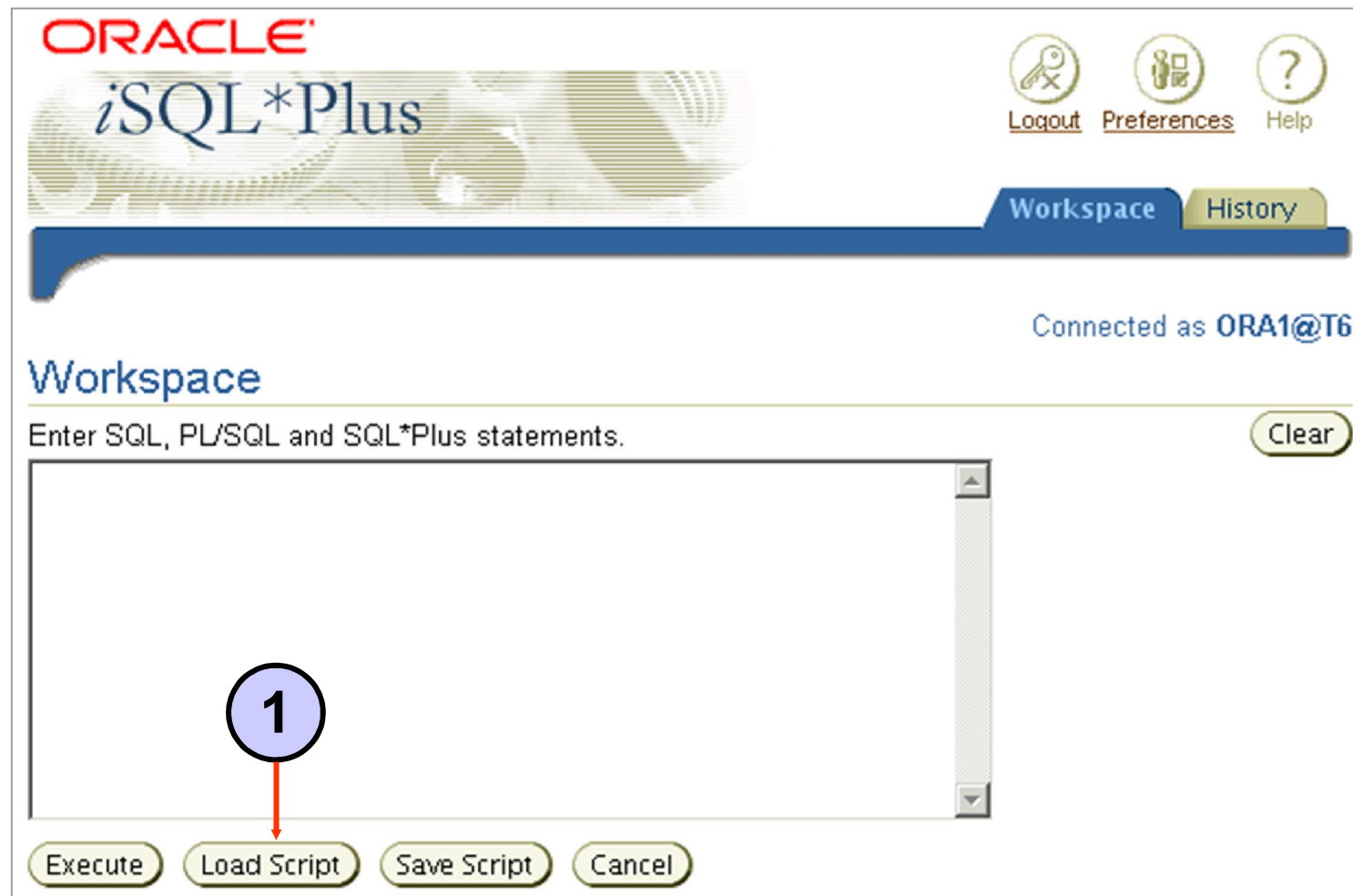
Interacting with Script Files



Interacting with Script Files



Interacting with Script Files



Interacting with Script Files

ORACLE[®]
iSQL*Plus

[Logout](#) [Preferences](#) [Help](#)

Workspace **History**

Connected as **ORA1@T6**

Load Script

Enter a URL, or a path and file name of the script to load.

URL

File [Browse...](#)

[Cancel](#) [Load](#)

[Cancel](#) [Load](#)

2

Workspace | [History](#) | [Logout](#) | [Preferences](#) | [Help](#)

Copyright © 2004 Oracle. All rights reserved.

3

SQL*Plus History Page

The screenshot shows the SQL*Plus History Page. At the top, there are tabs for 'Workspace' and 'History', with 'History' being the active tab (callout 3). Below the tabs, it says 'Connected as ORA1@T6'. The main heading is 'History'. A message states: 'The scripts listed are for the current session. Script history is not available for previous sessions.' Below this, there is a bar with 'Select scripts and ...', 'Delete', and 'Load' buttons (callout 2). Under 'Select scripts and ...', there are links for 'Select All' and 'Select None'. The main section is titled 'Select Script' and contains a list of SQL scripts with checkboxes. Callout 1 points to the first checkbox. The scripts are:

- ☐ `SELECT DISTINCT department_id FROM employees;`
- ☐ `SELECT department_id FROM employees;`
- ☐ `SELECT department_name || ' ' || q'X it's assigned manager ID: X' || manager`
- ☐ `SELECT last_name || ' is a ' || job_id AS "Employee Details" FROM employees;`
- ☐ `SELECT last_name || job_id AS "Employees" FROM employees;`
- ☒ `SELECT last_name "Name", 12 * salary "Annual Salary" FROM employees;`
- ☐ `SELECT last_name AS name, commission_pct AS comm FROM employees;`
- ☒ `SELECT last_name, 12 * salary * commission_pct FROM employees;`
- ☐ `SELECT last_name, job_id, salary, commission_pct FROM employees;`
- ☐ `SELECT last_name, salary, 12 * (salary + 100) FROM employees;`

iSQL*Plus History Page

ORACLE[®]
iSQL*Plus

[Logout](#) [Preferences](#) [Help](#)

3 [Workspace](#) [History](#)

Connected as **ORA1@T6**

Workspace

Enter SQL, PL/SQL and SQL*Plus statements. [Clear](#)

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

4

[Execute](#) [Load Script](#) [Save Script](#) [Cancel](#)

Setting iSQL*Plus Preferences

ORACLE[®]
iSQL*Plus

[Logout](#) [Preferences](#) [Help](#)

Workspace History

1

- Interface Configuration**
- System Configuration
 - [Script Formatting](#)
 - [Script Execution](#)
 - [Database Administration](#)
- [Change Password](#)

2

Interface Configuration

Configure settings that affect the iSQL*Plus user interface.

[Cancel](#) [Apply](#)

History Size

Set the number of scripts displayed in the script history.

Scripts

Input Area Size

Set the size of the script input area.

Width Height

3

Output Location

Setting the Output Location Preference

Interface Configuration

Configure settings that affect the iSQL*Plus user interface.

History Size

Set the number of scripts displayed in the script history.

Scripts

Input Area Size

Set the size of the script input area.

Width

Height

Output Location

Set where script output is displayed.

☐ Below Input Area

☒ Save to HTML File

☐ Printable output in new browser window

☐ Printable output in same browser window

Interface Configuration

- **Interface Configuration**
- System Configuration
 - [Script Formatting](#)
 - [Script Execution](#)
 - [Database Administration](#)
- [Change Password](#)

Cancel **Apply**

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 display more descriptive column headings
- Use the **SQL*Plus** environment to write, save, and execute SQL statements and **SQL*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 *SQL*Plus***

