



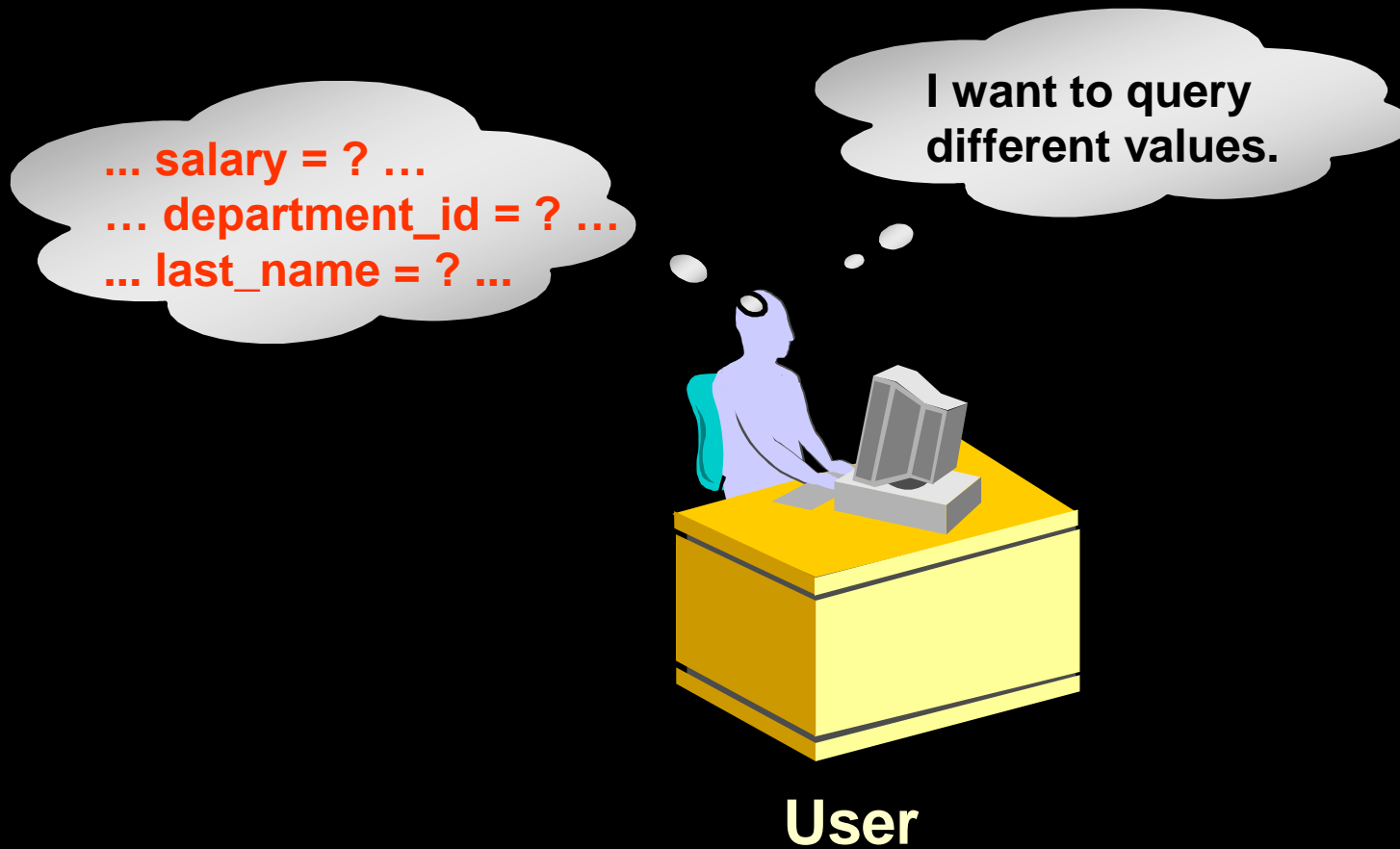
# Producing Readable Output with *i*SQL\*Plus

# Objectives

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

- **Produce queries that require a substitution variable**
- **Customize the *iSQL\*Plus* environment**
- **Produce more readable output**
- **Create and execute script files**

# Substitution Variables



# Substitution Variables

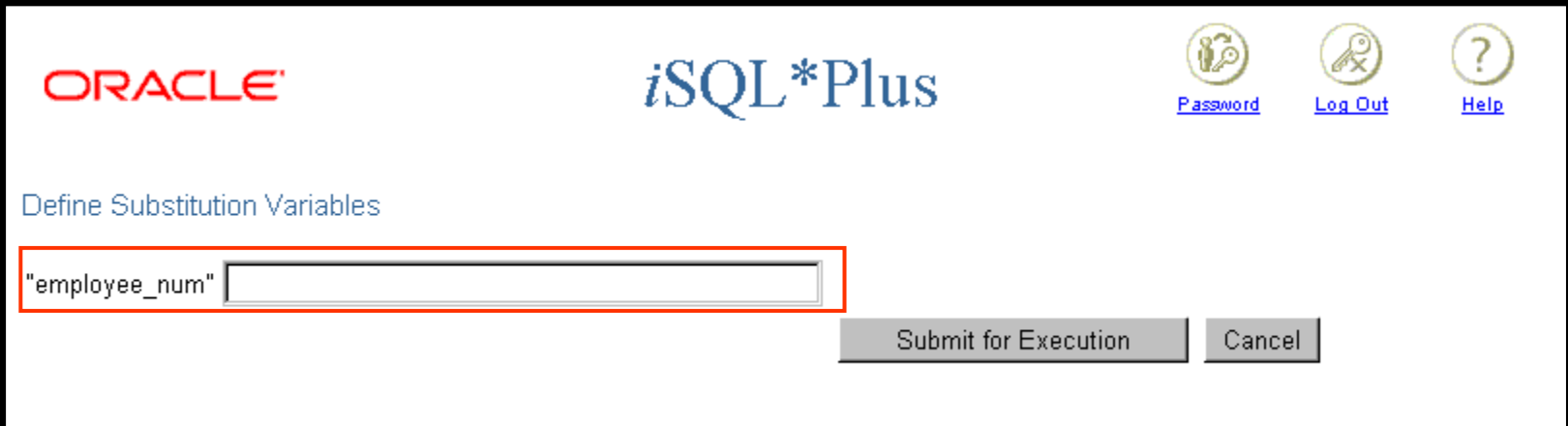
Use *iSQL\*Plus* substitution variables to:

- Temporarily store values
  - Single ampersand (&)
  - Double ampersand (&&)
  - DEFINE command
- Pass variable values between SQL statements
- Dynamically alter headers and footers

# Using the & Substitution Variable



Use a variable prefixed with an ampersand (&) to prompt the user for a value.




```
SELECT    employee_id, last_name, salary, department_id
FROM      employees
WHERE     employee_id = &employee_num ;
```



The screenshot shows the iSQL\*Plus web interface. At the top, there are logos for ORACLE and iSQL\*Plus, and links for Password, Log Out, and Help. Below the logos, the text 'Define Substitution Variables' is displayed. A text input field contains the variable name 'employee\_num'. To the right of the input field are two buttons: 'Submit for Execution' and 'Cancel'.

# Using the & Substitution Variable



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

Define Substitution Variables

"employee\_num"  1

2

old 3: WHERE employee\_id = &employee\_num  
new 3: WHERE employee\_id = 101

| EMPLOYEE_ID | LAST_NAME | SALARY | DEPARTMENT_ID |
|-------------|-----------|--------|---------------|
| 101         | Kochhar   | 17000  | 90            |

# Character and Date Values with Substitution Variables

Use single quotation marks for date and character values.

```
SELECT last_name, department_id, salary*12
FROM   employees
WHERE  job_id = '&job_title' ;
```

Define Substitution Variables

"job\_title"

Submit for Execution

Cancel

| LAST_NAME | DEPARTMENT_ID | SALARY*12 |
|-----------|---------------|-----------|
| Hunold    | 60            | 108000    |
| Ernst     | 60            | 72000     |
| Lorentz   | 60            | 50400     |

# Specifying Column Names, Expressions, and Text

Use substitution variables to supplement the following:

- **WHERE conditions**
- **ORDER BY clauses**
- **Column expressions**
- **Table names**
- **Entire SELECT statements**



# Specifying Column Names, Expressions, and Text

```
SELECT      employee_id, last_name, job_id,  
            &column_name  
FROM        employees  
WHERE       &condition  
ORDER BY    &order_column ;
```

## Define Substitution Variables

"column\_name" salary

"condition" salary > 15000

"order\_column" last\_name

Submit for Execution

Cancel

| EMPLOYEE_ID | LAST_NAME | JOB_ID  | SALARY |
|-------------|-----------|---------|--------|
| 102         | De Haan   | AD_VP   | 17000  |
| 100         | King      | AD_PRES | 24000  |
| 101         | Kochhar   | AD_VP   | 17000  |

# Defining Substitution Variables

- You can predefine variables using the *iSQL\*Plus* **DEFINE** command.

**DEFINE** *variable* = *value* creates a user variable with the CHAR data type.

- If you need to predefine a variable that includes spaces, you must enclose the value within single quotation marks when using the **DEFINE** command.
- A defined variable is available for the session

# DEFINE and UNDEFINE Commands

- A variable remains defined until you either:
  - Use the UNDEFINE command to clear it
  - Exit *iSQL\*Plus*
- You can verify your changes with the DEFINE command.

```
DEFINE job_title = IT_PROG
DEFINE job_title
DEFINE JOB_TITLE           = "IT_PROG" (CHAR)
```

```
UNDEFINE job_title
DEFINE job_title
SP2-0135: symbol job_title is UNDEFINED
```

# Using the DEFINE Command with & Substitution Variable

- Create the substitution variable using the DEFINE command.

```
DEFINE employee_num = 200
```

- Use a variable prefixed with an ampersand (&) to substitute the value in the SQL statement.

```
SELECT employee_id, last_name, salary, department_id  
FROM employees  
WHERE employee_id = &employee_num ;
```

| EMPLOYEE_ID | LAST_NAME | SALARY | DEPARTMENT_ID |
|-------------|-----------|--------|---------------|
| 200         | Whalen    | 4400   | 10            |

# Using the && Substitution Variable

Use the double-ampersand (&&) if you want to reuse the variable value without prompting the user each time.

```
SELECT    employee_id, last_name, job_id, &&column_name
FROM      employees
ORDER BY  &column_name;
```

Define Substitution Variables

"column\_name"

Submit for Execution

Cancel

| EMPLOYEE_ID | LAST_NAME | JOB_ID  | DEPARTMENT_ID |
|-------------|-----------|---------|---------------|
| 200         | Whalen    | AD_ASST | 10            |
| 201         | Hartstein | MK_MAN  | 20            |

...

20 rows selected.

ORACLE

# Using the VERIFY Command

Use the VERIFY command to toggle the display of the substitution variable, before and after *iSQL\*Plus* replaces substitution variables with values.

```
SET VERIFY ON
```

```
SELECT employee_id, last_name, salary, department_id  
FROM   employees  
WHERE  employee_id = &employee_num;
```

"employee\_num" 200

```
old   3: WHERE  employee_id = &employee_num  
new   3: WHERE  employee_id = 200
```

# Customizing the *iSQL\*Plus* Environment

- Use SET commands to control current session.

```
SET system_variable value
```

- Verify what you have set by using the SHOW command.

```
SET ECHO ON
```

```
SHOW ECHO  
echo ON
```

# SET Command Variables

- `ARRAYSIZE {20 | n}`
- `FEEDBACK {6 | n | OFF | ON}`
- `HEADING {OFF | ON}`
- `LONG {80 | n} | ON | text}`

`SET HEADING OFF`

`SHOW HEADING`

**`HEADING OFF`**



# *i*SQL\*Plus Format Commands

- COLUMN [*column option*]
- TTITLE [*text* | OFF | ON]
- BTITLE [*text* | OFF | ON]
- BREAK [ON *report\_element*]

# The COLUMN Command

## Controls display of a column:

```
COL[UMN] [{column|alias} [option]]
```

- CLE[AR]: Clears any column formats
- HEA[DING] *text*: Sets the column heading
- FOR[MAT] *format*: Changes the display of the column using a format model
- NOPRINT | PRINT
- NULL

# Using the COLUMN Command

- **Create column headings.**

```
COLUMN last_name HEADING 'Employee|Name'  
COLUMN salary JUSTIFY LEFT FORMAT $99,990.00  
COLUMN manager FORMAT 999999999 NULL 'No manager'
```

- **Display the current setting for the LAST\_NAME column.**

```
COLUMN last_name
```

- **Clear settings for the LAST\_NAME column.**

```
COLUMN last_name CLEAR
```

# COLUMN Format Models

| Element | Description                   | Example | Result  |
|---------|-------------------------------|---------|---------|
| 9       | Single zero-suppression digit | 999999  | 1234    |
| 0       | Enforces leading zero         | 099999  | 001234  |
| \$      | Floating dollar sign          | \$9999  | \$1234  |
| L       | Local currency                | L9999   | L1234   |
| .       | Position of decimal point     | 9999.99 | 1234.00 |
| ,       | Thousand separator            | 9,999   | 1,234   |

# Using the BREAK Command

**Use the BREAK command to suppress duplicates.**

```
BREAK ON job_id
```

# Using the TTITLE and BTITLE Commands

- Display headers and footers.

```
TTI[TLE] [text|OFF|ON]
```

- Set the report header.

```
TTITLE 'Salary|Report'
```

- Set the report footer.

```
BTITLE 'Confidential'
```

# Using the TTITLE and BTITLE Commands

- Display headers and footers.

```
TTI[TLE] [text|OFF|ON]
```

- Set the report header.

```
TTITLE 'Salary|Report'
```

- Set the report footer.

```
BTITLE 'Confidential'
```

# Creating a Script File to Run a Report

1. Create and test the SQL `SELECT` statement.
2. Save the `SELECT` statement into a script file.
3. Load the script file into an editor.
4. Add formatting commands before the `SELECT` statement.
5. Verify that the termination character follows the `SELECT` statement.



# Creating a Script File to Run a Report

6. Clear formatting commands after the `SELECT` statement.
7. Save the script file.
8. Load the script file into the *iSQL\*Plus* text window, and click the Execute button.

# Sample Report

Fri Sep 28

Employee  
Report

page 1

| Job<br>Category | Employee  | Salary      |
|-----------------|-----------|-------------|
| AC_ACCOUNT      | Gietz     | \$8,300.00  |
| AC_MGR          | Higgins   | \$12,000.00 |
| AD_ASST         | Whalen    | \$4,400.00  |
| IT_PROG         | Ernst     | \$6,000.00  |
|                 | Hunold    | \$9,000.00  |
|                 | Lorentz   | \$4,200.00  |
| MK_MAN          | Hartstein | \$13,000.00 |
| MK_REP          | Fay       | \$6,000.00  |
| SA_MAN          | Zlotkey   | \$10,500.00 |
| SA_REP          | Abel      | \$11,000.00 |
|                 | Grant     | \$7,000.00  |
|                 | Taylor    | \$8,600.00  |

Confidential

...

# Sample Report

Fri Sep 28

Employee  
Report

page 1

| Job<br>Category | Employee  | Salary      |
|-----------------|-----------|-------------|
| AC_ACCOUNT      | Gietz     | \$8,300.00  |
| AC_MGR          | Higgins   | \$12,000.00 |
| AD_ASST         | Whalen    | \$4,400.00  |
| IT_PROG         | Ernst     | \$6,000.00  |
|                 | Hunold    | \$9,000.00  |
|                 | Lorentz   | \$4,200.00  |
| MK_MAN          | Hartstein | \$13,000.00 |
| MK_REP          | Fay       | \$6,000.00  |
| SA_MAN          | Zlotkey   | \$10,500.00 |
| SA_REP          | Abel      | \$11,000.00 |
|                 | Grant     | \$7,000.00  |
|                 | Taylor    | \$8,600.00  |

Confidential

...

# Summary

In this lesson, you should have learned how to:

- Use *iSQL\*Plus* substitution variables to store values temporarily
- Use **SET** commands to control the current *iSQL\*Plus* environment
- Use the **COLUMN** command to control the display of a column
- Use the **BREAK** command to suppress duplicates and divide rows into sections
- Use the **TTITLE** and **BTITLE** commands to display headers and footers

# Practice 7 Overview

**This practice covers the following topics:**

- **Creating a query to display values using substitution variables**
- **Starting a command file containing variables**