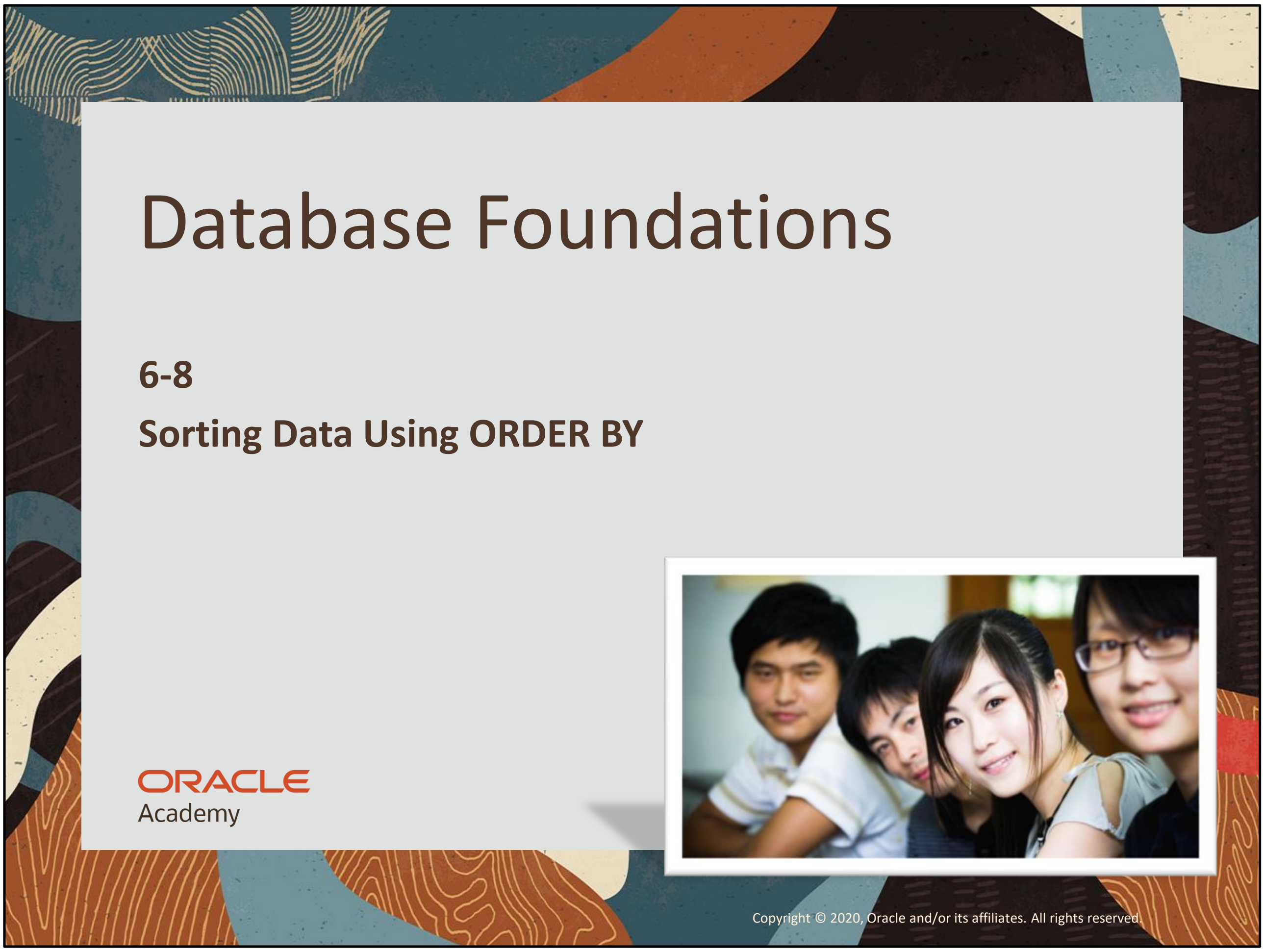
ORACLE

Academy



Database Foundations

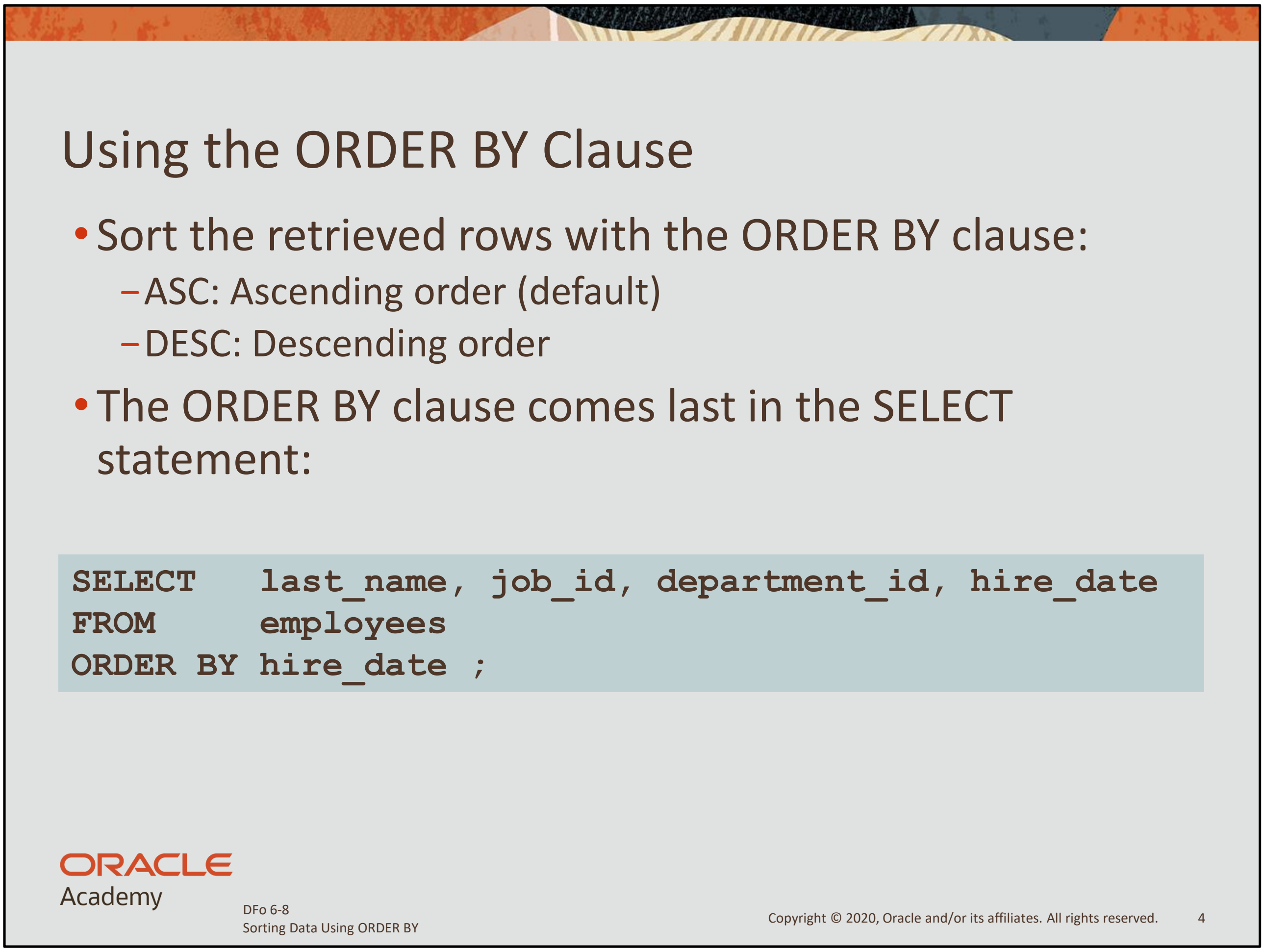
6-8

Sorting Data Using ORDER BY

ORACLE
Academy

Objectives

- This lesson covers the following objectives:
  - Use the ORDER BY clause to sort SQL query results
  - Identify the correct placement of the ORDER BY clause within a SELECT statement
  - Using ROWNUM for Top-N-Analysis
  - Using substitution variables in WHERE clause



Using the ORDER BY Clause

- Sort the retrieved rows with the ORDER BY clause:
  - ASC: Ascending order (default)
  - DESC: Descending order
- The ORDER BY clause comes last in the SELECT statement:

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

ORACLE
Academy

Syntax:

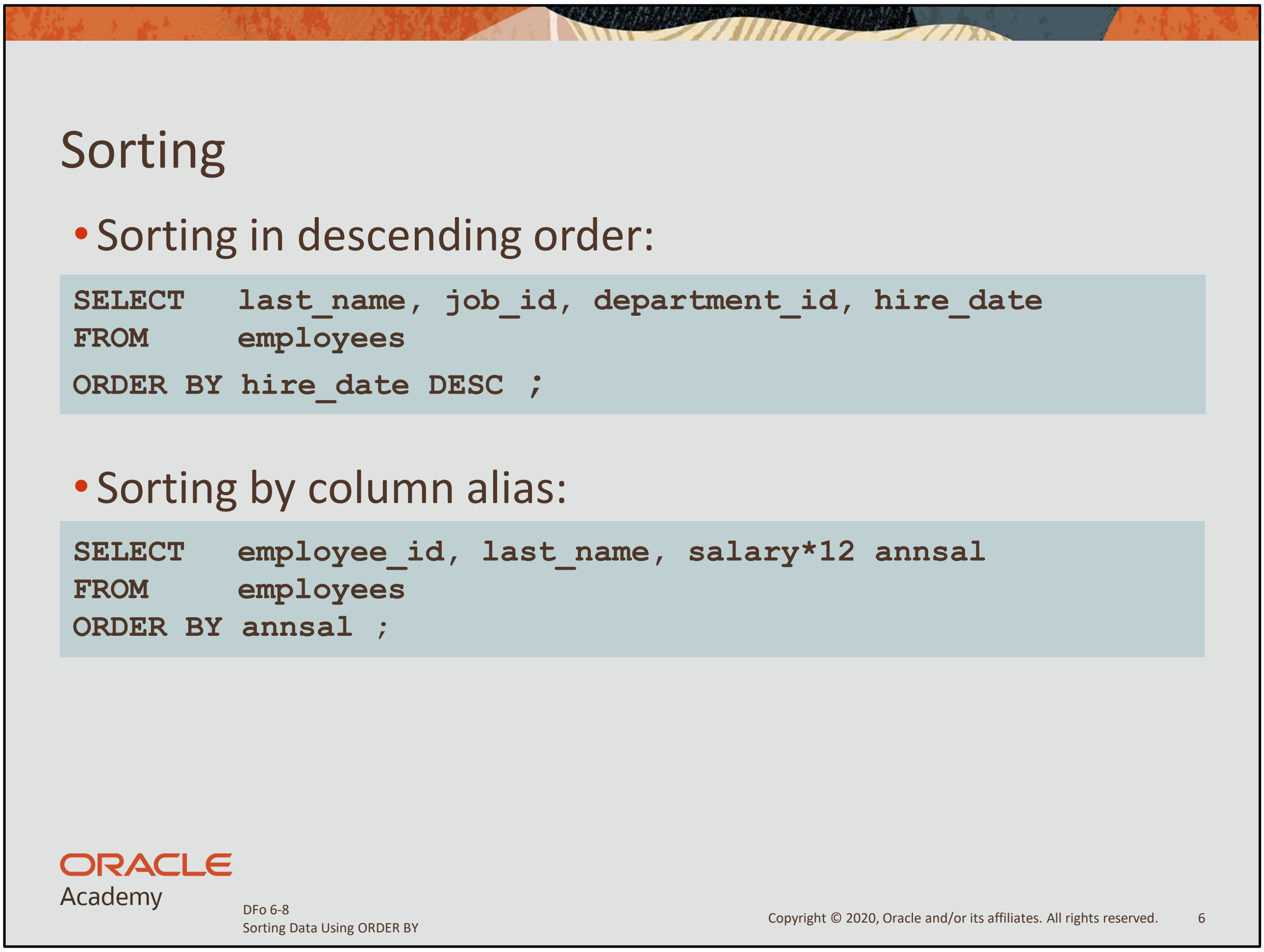
```
SELECT      expr
FROM        table
[WHERE condition(s)]
[ORDER BY {column, expr, numeric_position} [ASC|DESC]];
```

In the syntax:

- ORDER BY specifies the order in which the retrieved rows are displayed.
- ASC orders the rows in ascending order. (This is the default order.)
- DESC orders the rows in descending order.
  - If the ORDER BY clause is not used, the sort order is undefined, and the Oracle server may not fetch rows in the same order for the same query twice. Use the ORDER BY clause to display the rows in a specific order.
  - Use the NULLS FIRST or NULLS LAST keywords to specify whether returned rows containing null values should appear first or last in the ordering sequence.

ORDER BY Clause

- Numeric values are displayed lowest to highest
- Date values are displayed with the earliest value first
- Character values are displayed in alphabetical order
- Null values are displayed last in ascending order and first in descending order
- NULLS FIRST specifies that NULL values should be returned before non-NULL values
- NULLS LAST specifies that NULL values should be returned after non-NULL values



Sorting

- 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 ;
```

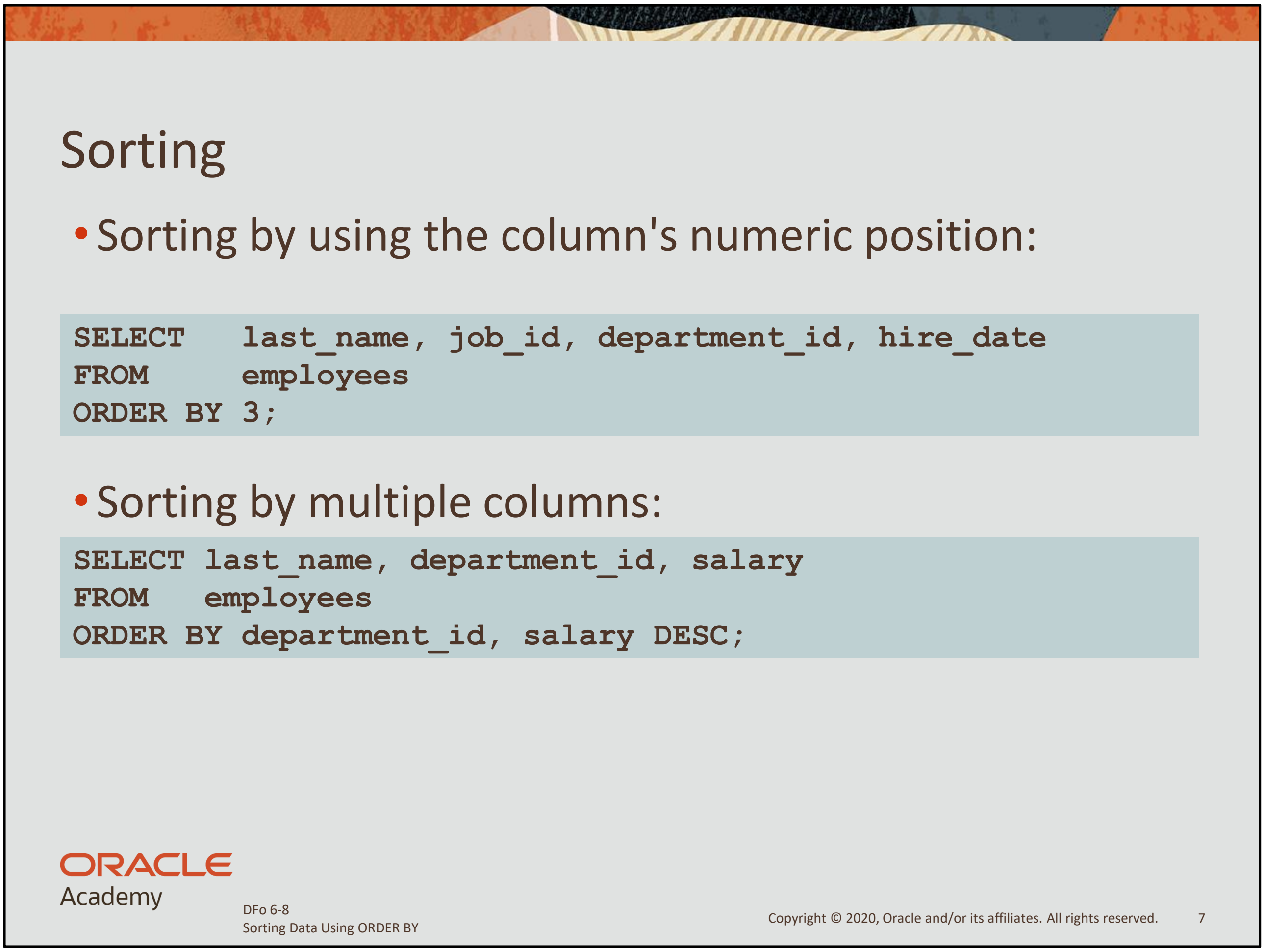
ORACLE
Academy

The default sort order is ascending:

- Numeric values are displayed with the lowest values first (for example, 1 to 999).
- Date values are displayed with the earliest value first (for example, 01-Jan-1992 before 01-Jan-1995).
- Character values are displayed in alphabetical order (for example, "A" first and "Z" last).
- By default, null values are displayed last for ascending sequences and first for descending sequences. You can change this by adding a NULLS FIRST or NULLS LAST option.
- You can also sort by a column that is not in the SELECT list.

You can use a column alias in the ORDER BY clause. The second slide example sorts the data by annual salary.

Note: The DESC keyword used here for sorting in descending order should not be confused with the DESC keyword used to describe table structures.



Sorting

- Sorting by using the column's numeric position:

```
SELECT   last_name, job_id, department_id, hire_date
FROM     employees
ORDER BY 3;
```

- Sorting by multiple columns:

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

Order of Execution

- The order of execution of a SELECT statement is as follows:
  - FROM clause:
    - locates the table that contains the data
  - WHERE clause:
    - restricts the rows to be returned
  - SELECT clause:
    - selects from the reduced data set the columns requested
  - ORDER BY clause:
    - orders the result set

Project Exercise 1

- DFo_6_8_1_Project
  - Oracle Baseball League Store Database
  - Use the ORDER BY Clause to Sort SQL Results

ORACLE
Academy

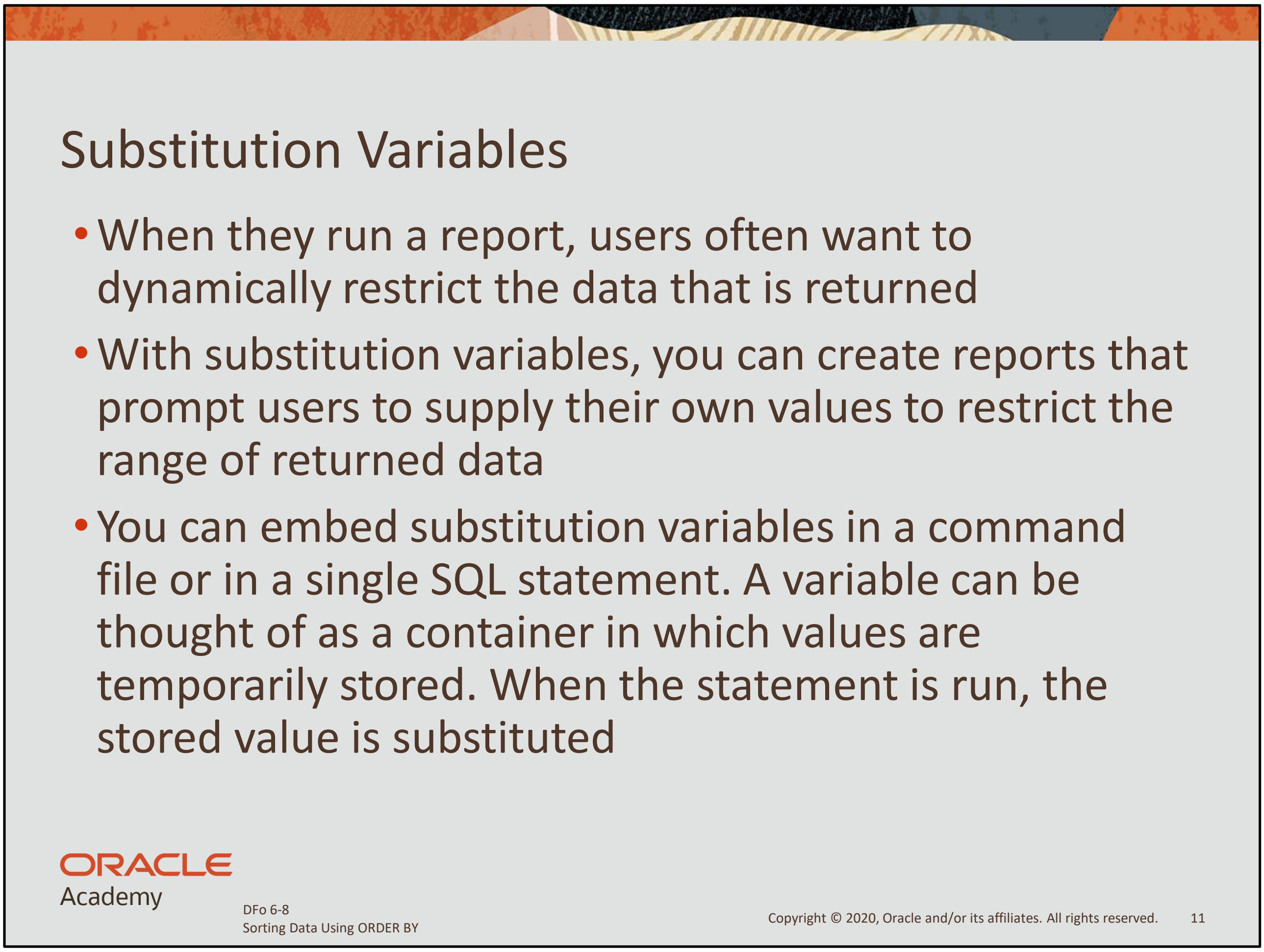
TOP-N-ANALYSIS

- Top-n-analysis is a SQL operation used to rank results

```
SELECT ROWNUM As "Longest Employed",
       last_name,hire_date
FROM
  (SELECT last_name, hire_date
   FROM employees
   ORDER BY hire_date)
WHERE ROWNUM <=3;
```

In APEX, Top-N queries are performed using ROWNUM

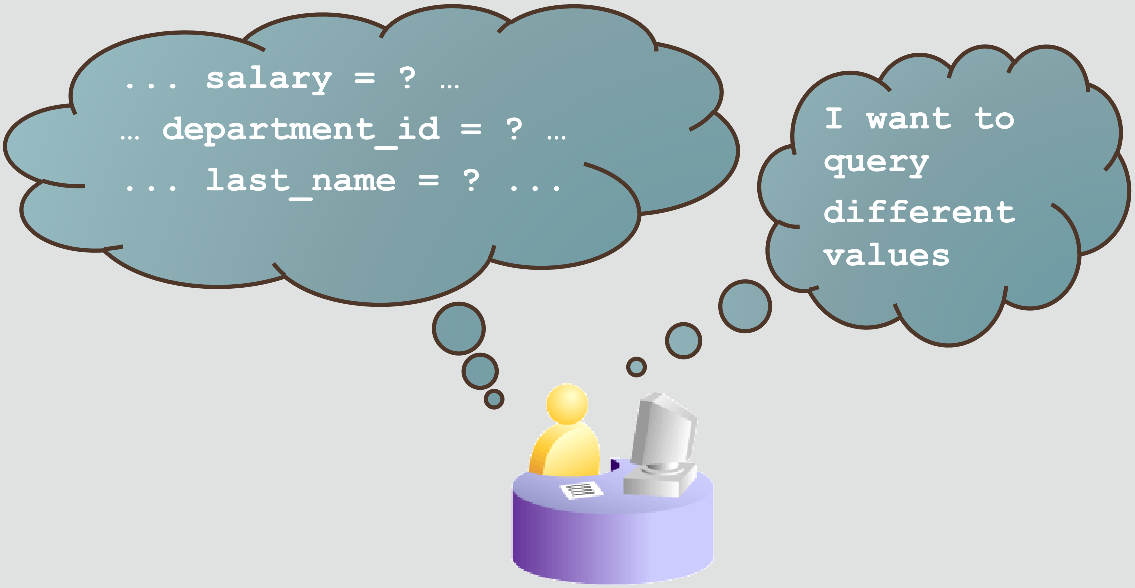
Longest Employed	LAST_NAME	HIRE_DATE
1	King	17-Jun-1987
2	Whalen	17-Sep-1987
3	Kochhar	21-Sep-1989



Substitution Variables

- When they run a report, users often want to dynamically restrict the data that is returned
- With substitution variables, you can create reports that prompt users to supply their own values to restrict the range of returned data
- You can embed substitution variables in a command file or in a single SQL statement. A variable can be thought of as a container in which values are temporarily stored. When the statement is run, the stored value is substituted

Substitution Variables



```
. . . salary = ? …
… department_id = ? …
. . . last_name = ? . . .
```

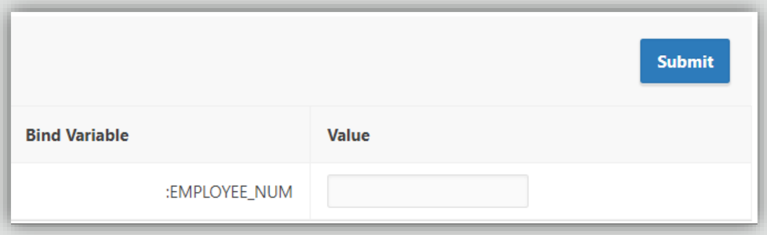
I want to query different values

ORACLE
Academy

Using a Substitution Variable

- Use substitution variables to prompt for values
- Use a variable prefixed with an colon(:) to prompt the user for a value:

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



Bind Variable	Value
:EMPLOYEE_NUM	

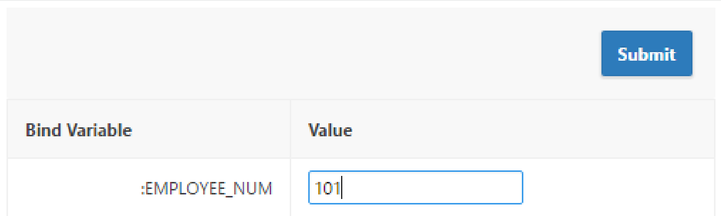
**Note : you must allow pop-ups in your browser

ORACLE
Academy

SQL*Plus or SQL Developer provides this flexibility with an ampersand (&) to identify each variable in your SQL statement.

Using a Substitution Variable

- When APEX detects that the SQL statement contains a substitution variable, you are prompted to enter a value for it
- After you enter a value and click Submit, the results are displayed on the Results tab of your APEX session



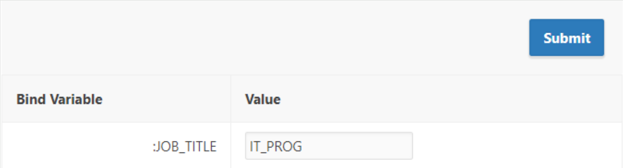
Bind Variable	Value
:EMPLOYEE_NUM	101

EMPLOYEE_ID	LAST_NAME	SALARY	DEPARTMENT_ID
101	Kochhar	17000	90

Character and Date Values with Substitution Variables

- You can also use date or character values:

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



Bind Variable	Value
:JOB_TITLE	IT_PROG

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

You do not need to enclose the literals in quotes when using substitution variables in APEX.

Project Exercise 2

- DFo_6_8_2_Project
  - Oracle Baseball League Store Database
  - Top-N-Analysis
  - Using a Substitution Variable

Summary

- In this lesson, you should have learned how to:
  - Use the ORDER BY clause to sort SQL query results
  - Identify the correct placement of the ORDER BY clause within a SELECT statement
  - Using ROWNUM for Top-N-Analysis
  - Using substitution variables in WHERE clause

ORACLE

Academy