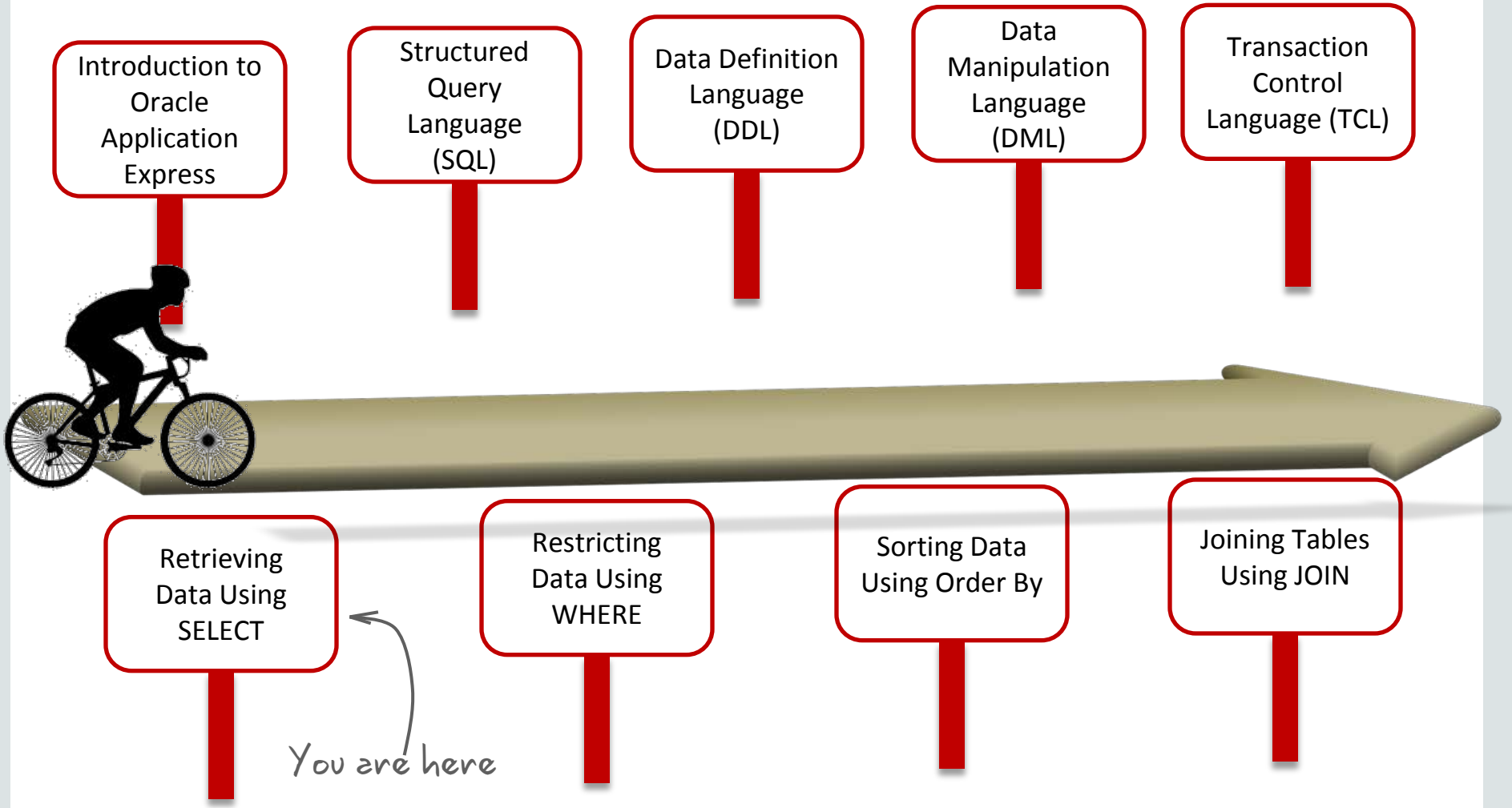

ORACLE
ACADEMY

Database Foundations

6-6
Retrieving Data Using SELECT

ORACLE ACADEMY

Roadmap



Objectives

This lesson covers the following objectives:

- List the capabilities of SQL SELECT statements
- Write and execute a SELECT statement that:
  - Returns all rows and columns from a table
  - Returns specific columns from a table
  - Uses column aliases to display descriptive column headings
  - Uses arithmetic and concatenation operators
  - Uses literal character strings
  - Eliminates duplicate rows
- Describe the structure of a table

Basic SELECT Statement

- SELECT identifies the columns to be displayed.
- FROM identifies the table that contains those columns.

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

Selecting All Columns

```
SELECT *
FROM   departments;
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
40	Human Resources	203	2400
50	Shipping	121	1500
60	IT	103	1400
70	Public Relations	204	2700

Selecting Specific Columns

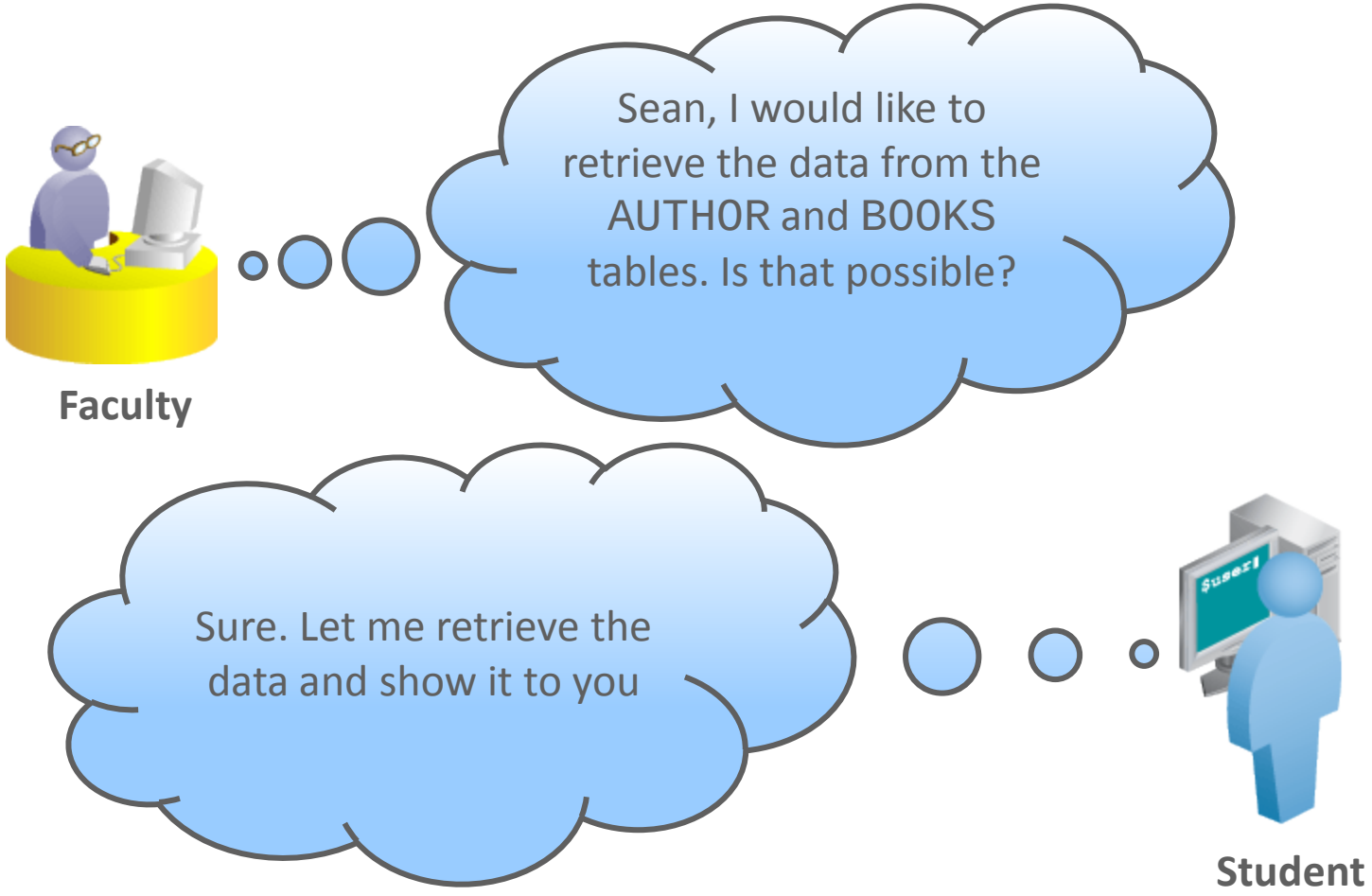
```
SELECT department_id, location_id
FROM   departments;
```

DEPARTMENT_ID	LOCATION_ID
10	1700
20	1800
30	1700
40	2400
50	1500
60	1400
70	2700

Writing SQL Statements

- SQL statements are not case-sensitive.
- SQL statements can be entered on one or more lines.
- Keywords cannot be abbreviated or split across lines and are typically spelled with uppercase letters.
- Clauses are usually placed on separate lines.
- Indents are used to enhance readability.
- In SQL Developer, SQL statements can be terminated by a semicolon (;). Semicolons are required when you execute multiple SQL statements.
- In SQL*Plus, you are required to end each SQL statement with a semicolon (;).

Case Scenario: Retrieving Data



Case Scenario: Retrieving Data

```
SELECT AUTHOR_ID, AUTHOR_NAME
FROM AUTHOR;
```

```
AN0001    Oliver Goldsmith
AN0002    Oscar Wilde
AN0003    George Bernard Shaw
AN0004    Leo Tolstoy
AN0005    Percy Shelley
AN0006    Lord Byron
AN0007    John Keats
AN0008    Rudyard Kipling
AN0009    P. G. Wodehouse

9 rows selected
```

Here is the information

```
SELECT *
FROM BOOKS;
```

```
BN0001  Florentine Tragedy
BN0002  A Vision
BN0003  Citizen of the World
BN0004  The Complete Poetical Works of Oliver Goldsmith
BN0005  Androcles and the Lion
BN0006  An Unsocial Socialist
BN0007  A Thing of Beauty is a Joy Forever
BN0008  Beyond the Pale
BN0009  The Clicking of Cuthbert
BN0010  Bride of Frankenstein
BN0011  Shelley Poetry and Prose
BN0012  War and Peace
```

Column Heading Defaults

- SQL Developer:
  - Default heading alignment: Left-aligned
  - Default heading display: Uppercase
- SQL*Plus:
  - Character and Date column headings: Left-aligned
  - Number column headings: 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
Austin	4800	5100
Pataballa	4800	5100

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

```
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
Hunold	9000	109200

Defining a Null Value

- Null is a value that is unavailable, unassigned, unknown, or inapplicable.
- 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	-
De Haan	AD_VP	17000	-
Pataballa	IT_PROG	4800	-

...

Tucker	SA_REP	10000	.3
Bernstein	SA_REP	9500	.25

Null Values in Arithmetic Expressions

Arithmetic expressions containing a null value to evaluate to null.

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

LAST_NAME	12*SALARY*COMMISSION_PCT
King	-
Kochhar	-
De Haan	-
Hunold	-

...

Russell	67200
Partners	48600
Errazuriz	43200
Cambrault	39600

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 the 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	-
Hunold	-

...

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

Name	Annual Salary
King	288000
Kochhar	204000
De Haan	204000
Hunold	108000
Ernst	72000

Concatenation Operator

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

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

Employees
AbelSA_REP
AndeSA_REP
AtkinsonST_CLERK
AustinIT_PROG
BaerPR_REP
BaidaPU_CLERK
BandaSA_REP

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 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
Abel is a SA_REP
Ande is a SA_REP
Atkinson is a ST_CLERK
Austin is a IT_PROG
Baer is a PR_REP
Baida is a PU_CLERK
Banda is a SA_REP

Alternative Quote (q) Operator

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

Department and Manager
Administration Department's Manager Id: 200
Marketing Department's Manager Id: 201
Purchasing Department's Manager Id: 114
Human Resources Department's Manager Id: 203
Shipping Department's Manager Id: 121
IT Department's Manager Id: 103

Case Scenario: Using the Column Alias

Faculty

Sean, I would like to see the different locations where the members are located.

I can create a simple query using the SELECT statement and display that information.

Student

Case Scenario: Using the SELECT Statement

Here the concatenation operator as well as the column alias has been used.

```
SELECT FIRST_NAME || ' IS LOCATED IN '|| CITY
AS "MEMBER LOCATION"
FROM MEMBERS;
```

```
MEMBER LOCATION
VelasquezCarmen IS LOCATED IN Seattle
Ngao LaDoris IS LOCATED IN Bratislava
Nagayama Midori IS LOCATED IN Sao Paolo
Quick-To-See Mark IS LOCATED IN Lagos
Ropeburn Audry IS LOCATED IN Hong Kong
Urguhart Molly IS LOCATED IN Quebec
Menchu Roberta IS LOCATED IN Brussels
Biri Ben IS LOCATED IN Columbus

8 rows selected
```

Successful retrieval of data

Duplicate Rows

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

1

```
SELECT department_id
FROM   employees;
```

DEPARTMENT_ID
90
90
90
60
60

2

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

DEPARTMENT_ID
100
30
-
90
20
70

Displaying the Table Structure

- Use the DESCRIBE command to display the structure of a table.
- Or, select the table in the Connections tree and use the Columns tab to view the table structure.

```
DESC[RIBE] tablename
```

Using the DESCRIBE Command

```
DESCRIBE employees
```

```
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)
```

Summary

In this lesson, you should have learned how to:

- List the capabilities of SQL SELECT statements
- Write and execute a SELECT statement that:
  - Returns all rows and columns from a table
  - Returns specific columns from a table
  - Uses column aliases to display descriptive column headings
  - Uses arithmetic and concatenation operators
  - Uses literal character strings
  - Eliminates duplicate rows
- Describe the structure of a table


ORACLE
ACADEMY