ORACLE

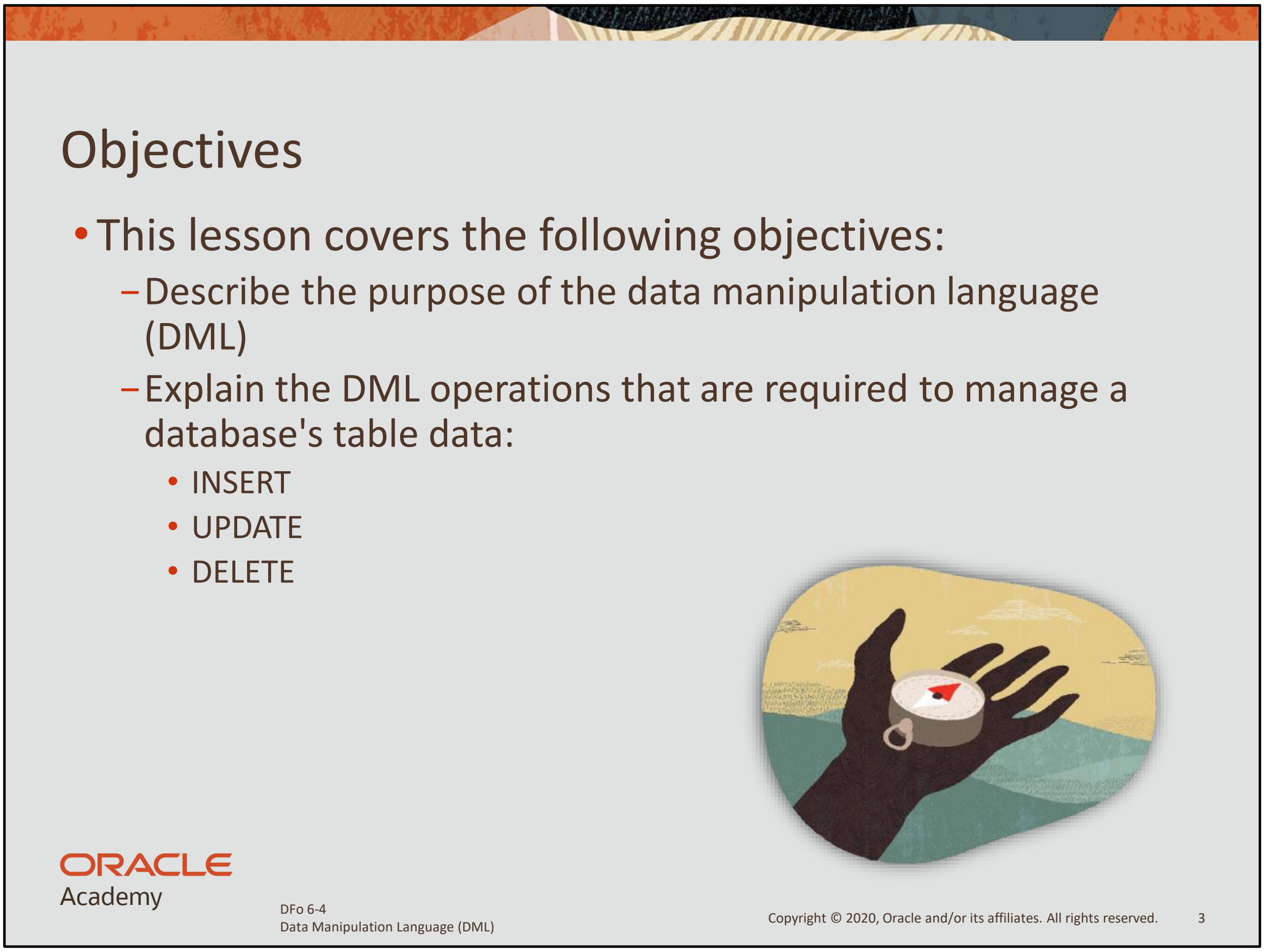
Academy

Database Foundations

6-4

Data Manipulation Language (DML)

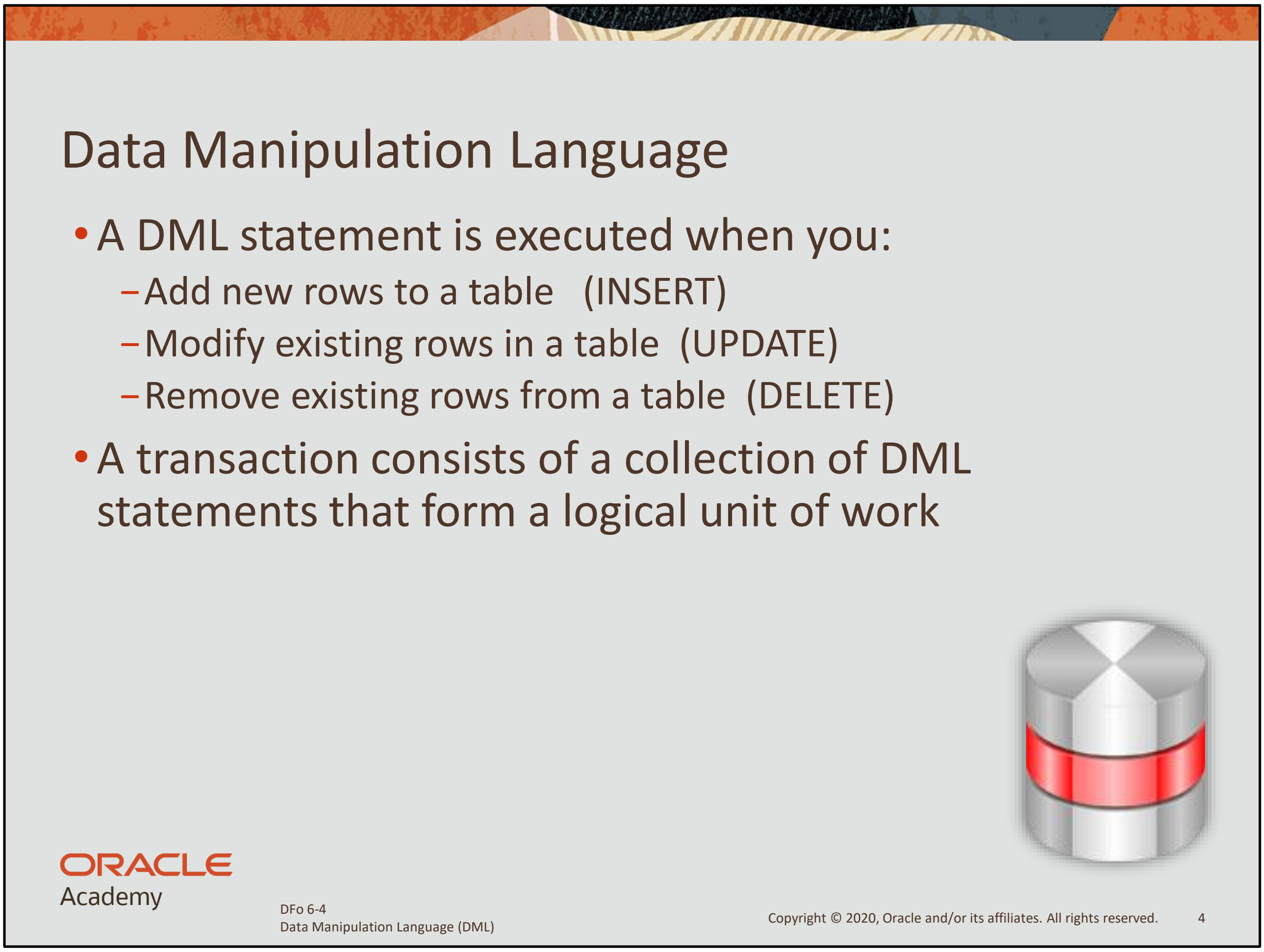
ORACLE
Academy



Objectives

- This lesson covers the following objectives:
  - Describe the purpose of the data manipulation language (DML)
  - Explain the DML operations that are required to manage a database's table data:
    - INSERT
    - UPDATE
    - DELETE

ORACLE
Academy



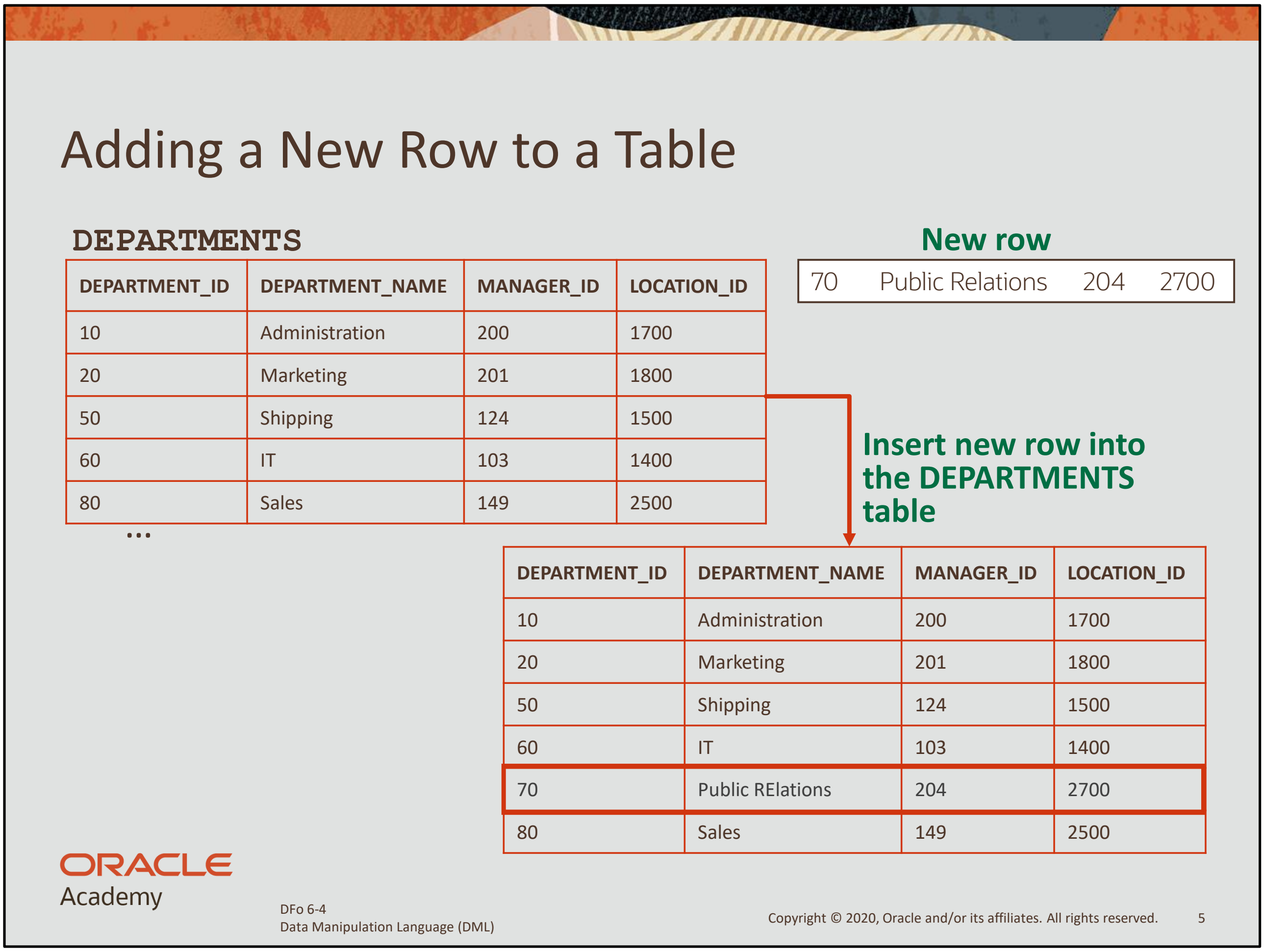
Data Manipulation Language

- A DML statement is executed when you:
  - Add new rows to a table (INSERT)
  - Modify existing rows in a table (UPDATE)
  - Remove existing rows from a table (DELETE)
- A transaction consists of a collection of DML statements that form a logical unit of work

ORACLE
Academy

DFo 6-4
Data Manipulation Language (DML)

Consider a banking database. When a bank customer transfers money from a savings account to a checking account, the transaction might consist of three separate operations: decreasing the savings account, increasing the checking account, and recording the transaction in the transaction journal. The Oracle server must guarantee that all the three SQL statements are performed to maintain the accounts in proper balance. When something prevents one of the statements in the transaction from executing, the other statements of the transaction must be undone.



Adding a New Row to a Table

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

…

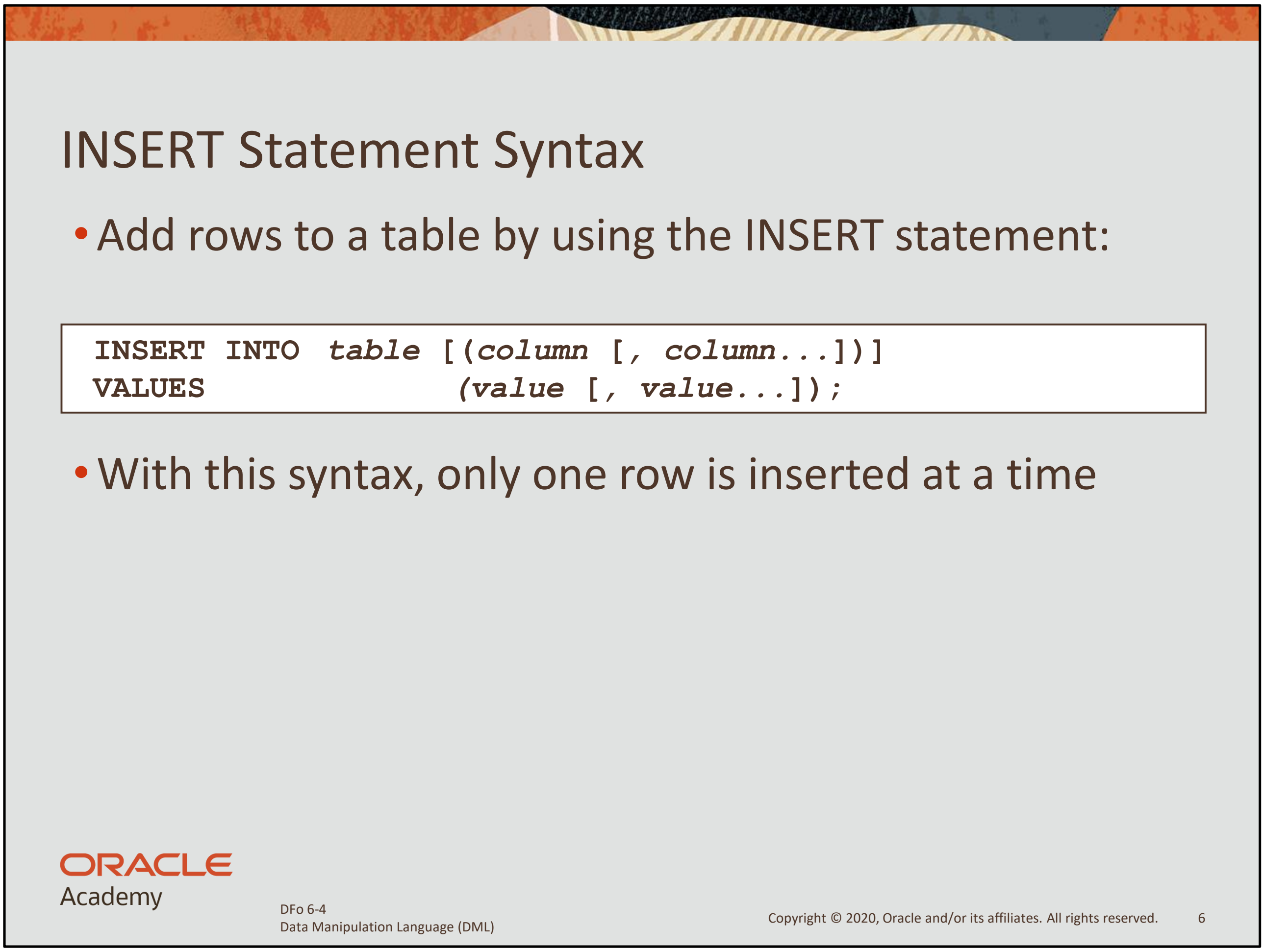
New row

70 Public Relations 204 2700

Insert new row into the DEPARTMENTS table

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
70	Public RElations	204	2700
80	Sales	149	2500

ORACLE
Academy



INSERT Statement Syntax

- Add rows to a table by using the INSERT statement:

```
INSERT INTO  table [(column [, column...])]
VALUES                 (value [, value...]);
```

- With this syntax, only one row is inserted at a time

In the syntax:

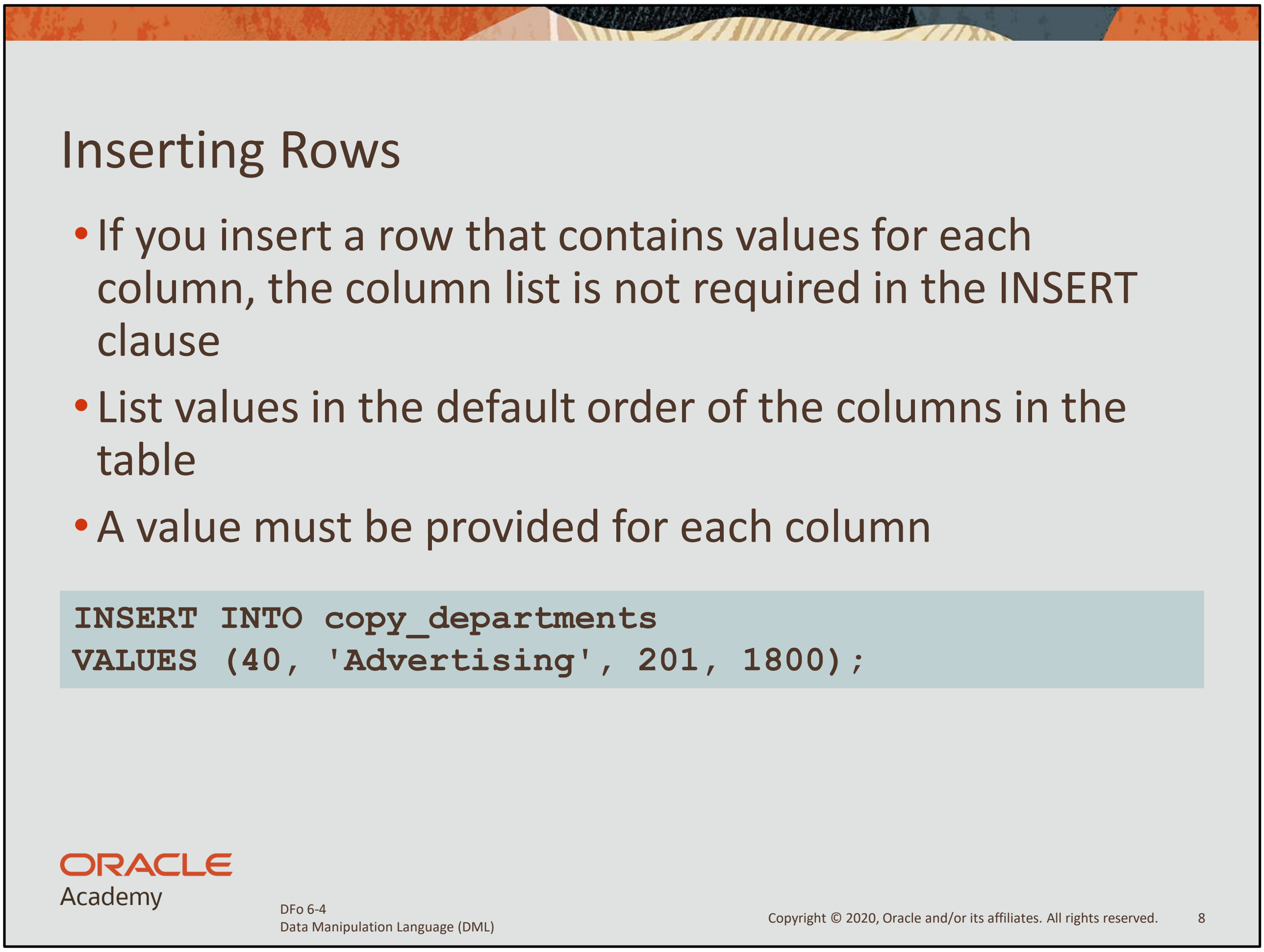
- table is the name of the table.
- column is the name of the column in the table that you want to populate.
- value is the corresponding value for the column.

INSERT Statement Syntax

- Before any DML operations are executed on a table make a copy of the table:

```
CREATE TABLE copy_departments
AS (SELECT * FROM departments);
```

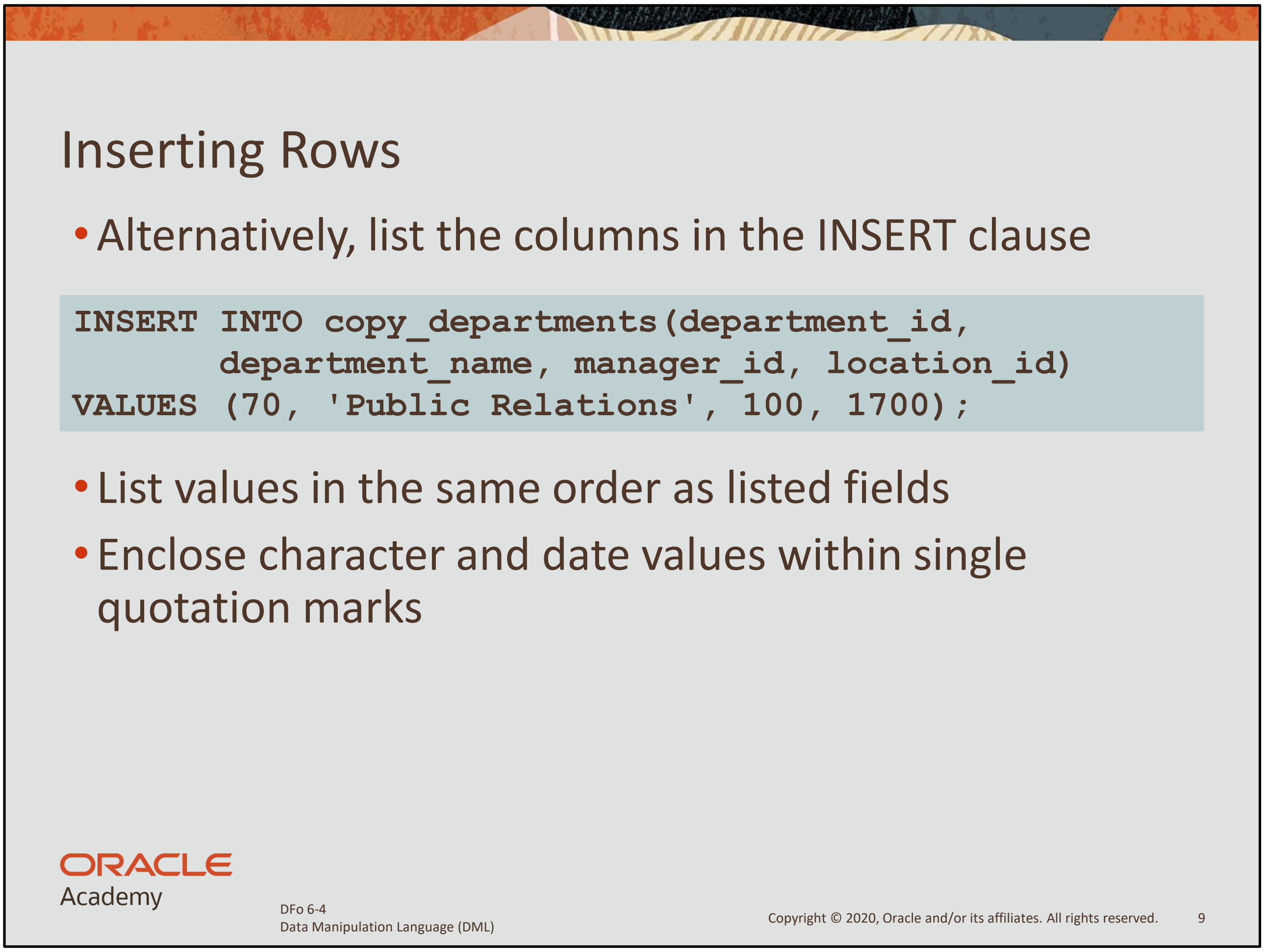
- Be aware that copying a table in this manner does not copy all constraints (only NOT NULL)



Inserting Rows

- If you insert a row that contains values for each column, the column list is not required in the INSERT clause
- List values in the default order of the columns in the table
- A value must be provided for each column

```
INSERT INTO copy_departments
VALUES (40, 'Advertising', 201, 1800);
```



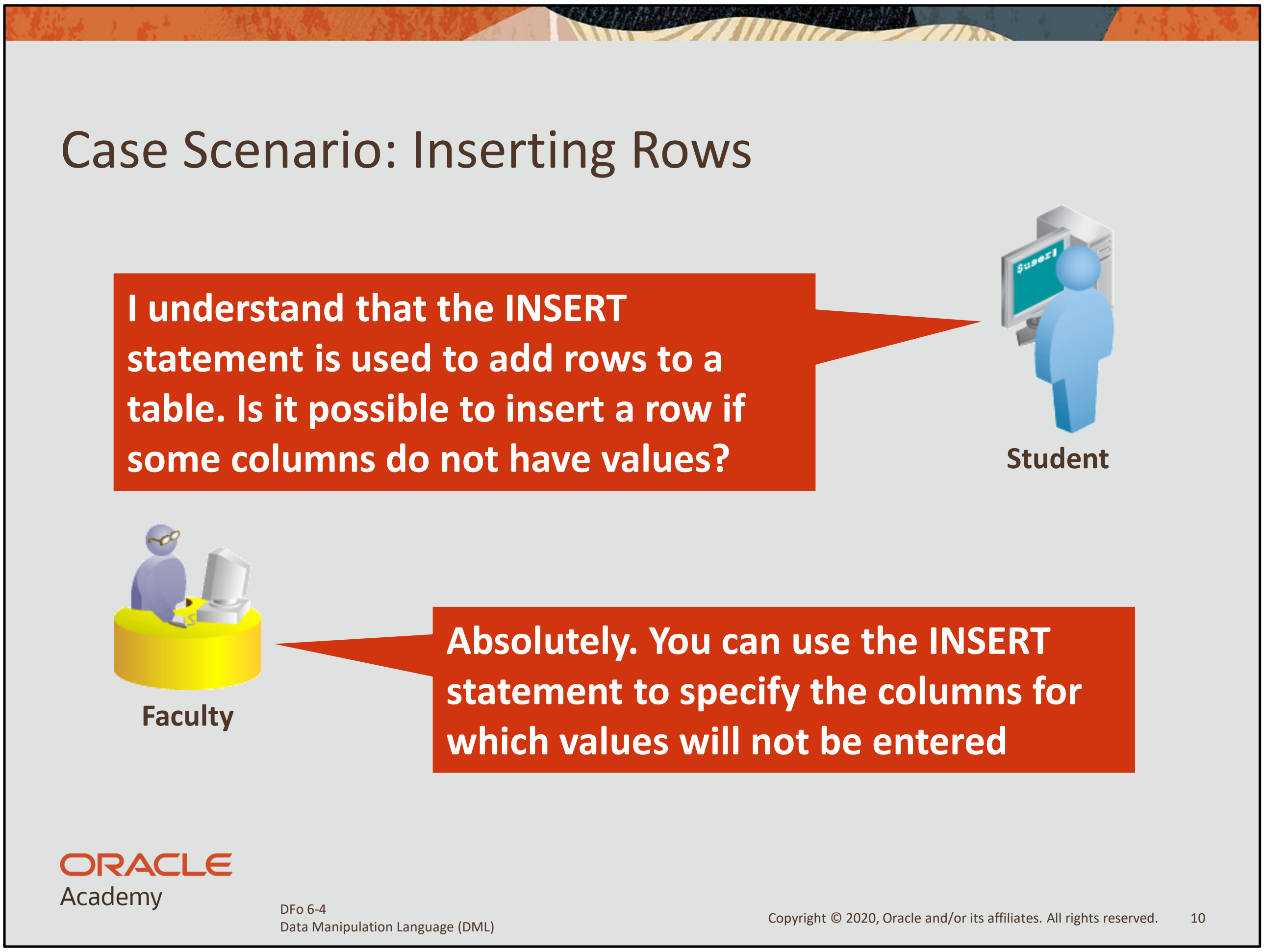
Inserting Rows

- Alternatively, list the columns in the INSERT clause

```
INSERT INTO copy_departments(department_id,
       department_name, manager_id, location_id)
VALUES (70, 'Public Relations', 100, 1700);
```

- List values in the same order as listed fields
- Enclose character and date values within single quotation marks

ORACLE
Academy



Case Scenario: Inserting Rows

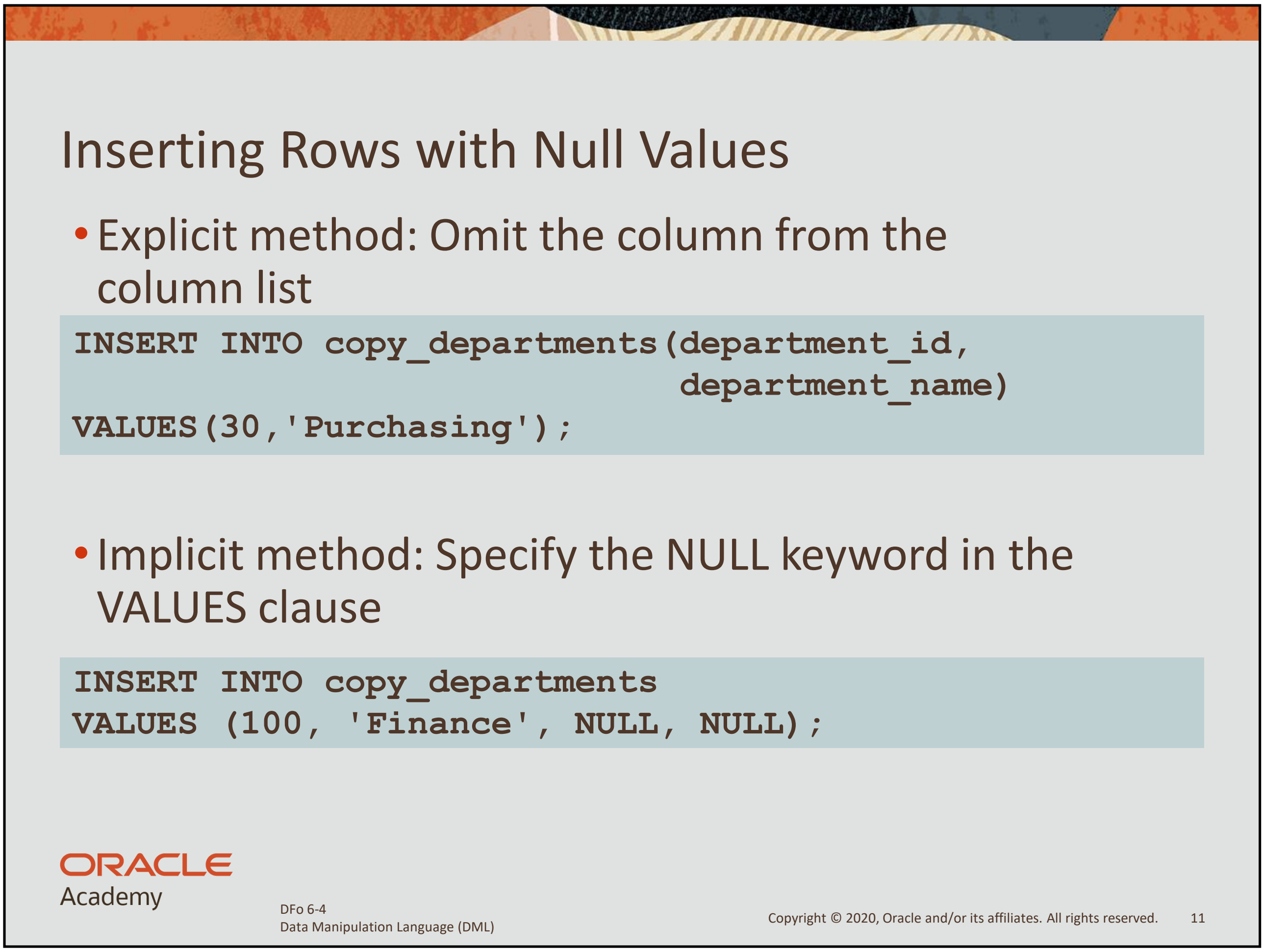
I understand that the INSERT statement is used to add rows to a table. Is it possible to insert a row if some columns do not have values?

Student

Absolutely. You can use the INSERT statement to specify the columns for which values will not be entered

Faculty

ORACLE
Academy



Inserting Rows with Null Values

- Explicit method: Omit the column from the column list

```
INSERT INTO copy_departments(department_id,
                             department_name)
VALUES(30,'Purchasing');
```

- Implicit method: Specify the NULL keyword in the VALUES clause

```
INSERT INTO copy_departments
VALUES (100, 'Finance', NULL, NULL);
```

ORACLE
Academy

DFo 6-4
Data Manipulation Language (DML)

Be sure that you can use null values in the targeted column by verifying the Null status with the DESCRIBE command.

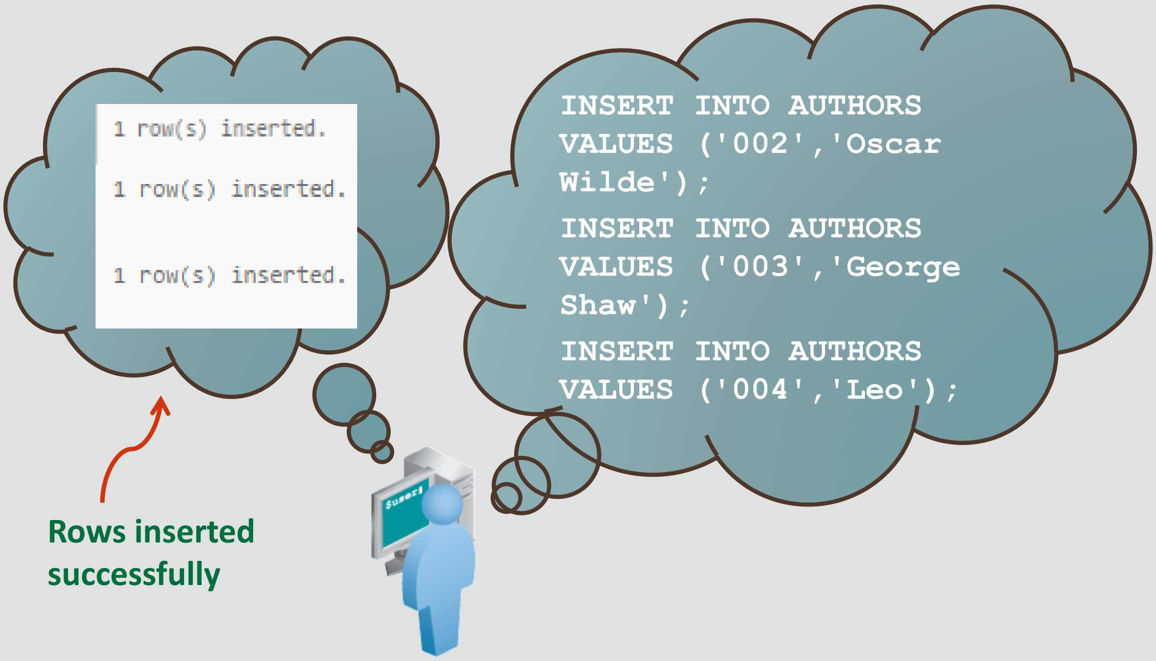
The Oracle server automatically enforces all data types, data ranges, and data integrity constraints. Any column that is not listed explicitly obtains a null value in the new row unless there are default values for the missing columns that are used.

Common errors that can occur during user input are checked in the following order:

- A mandatory value is missing for a NOT NULL column.
- A duplicate value violates any unique or primary key constraint.
- The Any value violates a CHECK constraint.
- A foreign key violates the referential integrity constraint.
- Data type mismatches or values are too wide to fit in the column.

Note: Use of the column list is recommended because it makes the INSERT statement more readable and reliable and less prone to mistakes.

Case Scenario: Inserting Rows



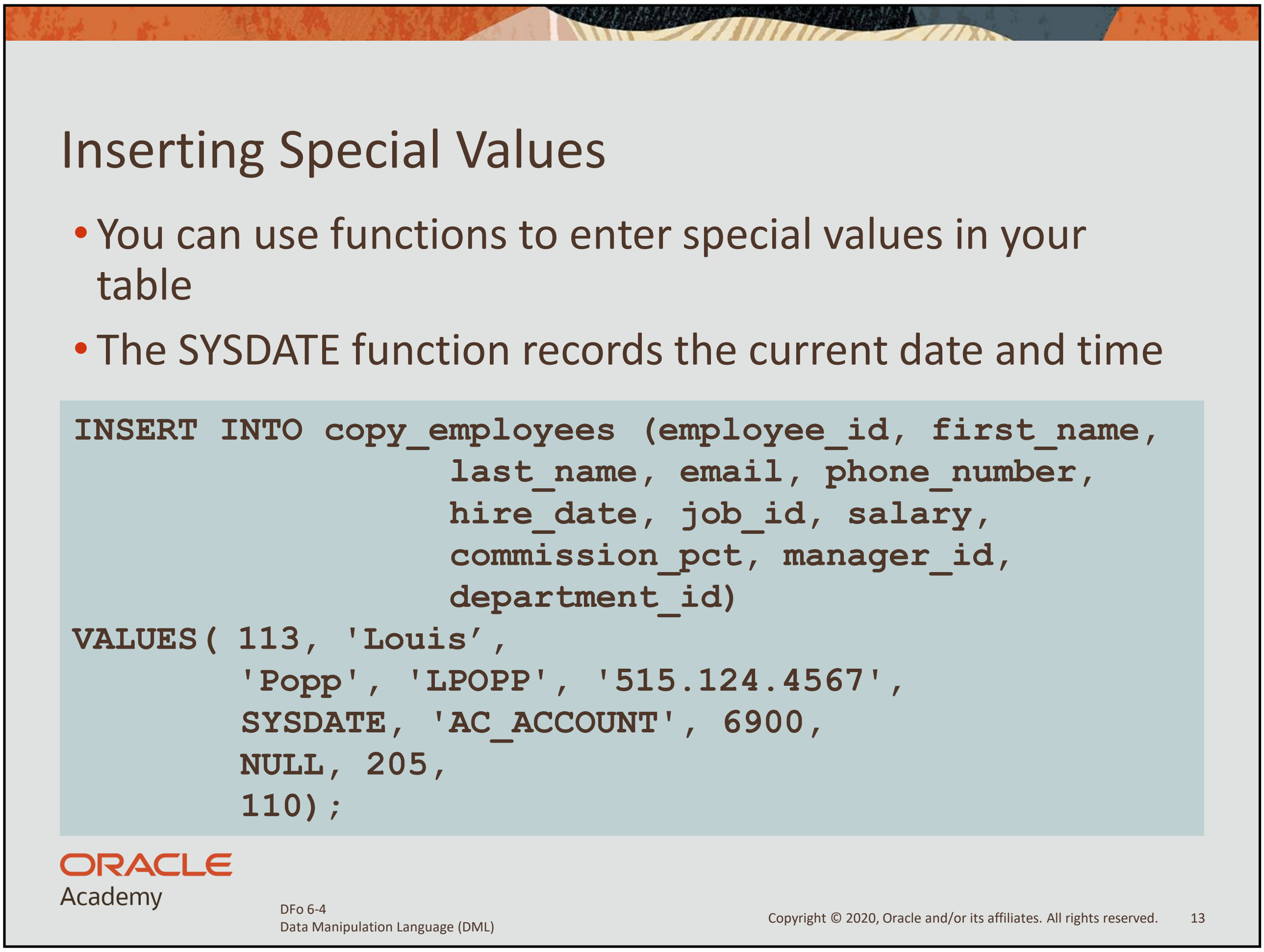
```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

```
INSERT INTO AUTHORS
VALUES ('002','Oscar
Wilde');
INSERT INTO AUTHORS
VALUES ('003','George
Shaw');
INSERT INTO AUTHORS
VALUES ('004','Leo');
```

Rows inserted successfully



Inserting Special Values

- You can use functions to enter special values in your table
- The SYSDATE function records the current date and time

```
INSERT INTO copy_employees (employee_id, first_name,
                    last_name, email, phone_number,
                    hire_date, job_id, salary,
                    commission_pct, manager_id,
                    department_id)
VALUES( 113, 'Louis’,
        'Popp', 'LPOPP', '515.124.4567',
        SYSDATE, 'AC_ACCOUNT', 6900,
        NULL, 205,
        110);
```

ORACLE Academy

DFo 6-4
Data Manipulation Language (DML)

Note : First create the copy_employees table : CREATE copy_employees AS (SELECT * FROM employees);

The slide example records information for employee Popp in the EMPLOYEES table. It supplies the current date and time in the HIRE_DATE column. It uses the SYSDATE function to return the current date and time of the database server.

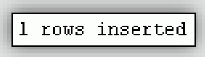
You can also use the CURRENT_DATE function to return the current date in the session time zone.

When you insert rows in a table, you can also use the USER function to record the current username.

Inserting Specific Date and Time Values

- Add an employee

```
INSERT INTO copy_employees
VALUES(114, 'Den', 'Raphealy', 'DRAPHEAL’,
       '515.127.4561’,
        TO_DATE('Dec 7, 2002', 'MON DD, YYYY'),
       'SA_REP', 11000, 0.2, 100, 60, NULL);
```



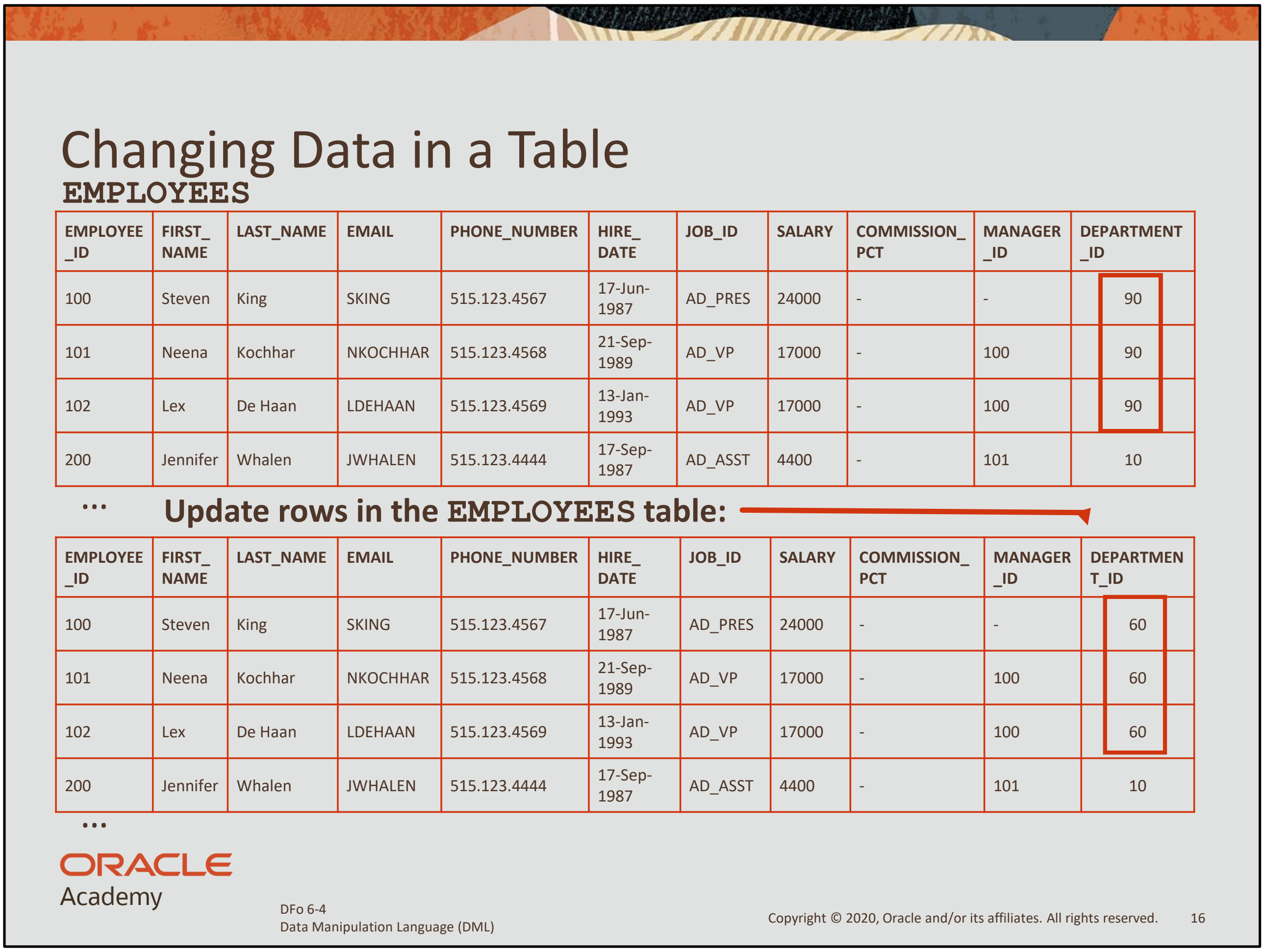
```
1 rows inserted
```

- Verify your addition

114	Den	Raphealy	DRAPHEAL	515.127.4561	07-Dec-2002

Project Exercise 1

- DFo_6_4_1_Project
  - Oracle Baseball League Store Database
  - Using DML operations to manage database tables : Inserting records



Changing Data in a Table

EMPLOYEES

EMPLOYEE _ID	FIRST_ NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_ DATE	JOB_ID	SALARY	COMMISSION_ PCT	MANAGER _ID	DEPARTMENT _ID
100	Steven	King	SKING	515.123.4567	17-Jun-1987	AD_PRES	24000	-	-	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-Sep-1989	AD_VP	17000	-	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	13-Jan-1993	AD_VP	17000	-	100	90
200	Jennifer	Whalen	JWHALEN	515.123.4444	17-Sep-1987	AD_ASST	4400	-	101	10

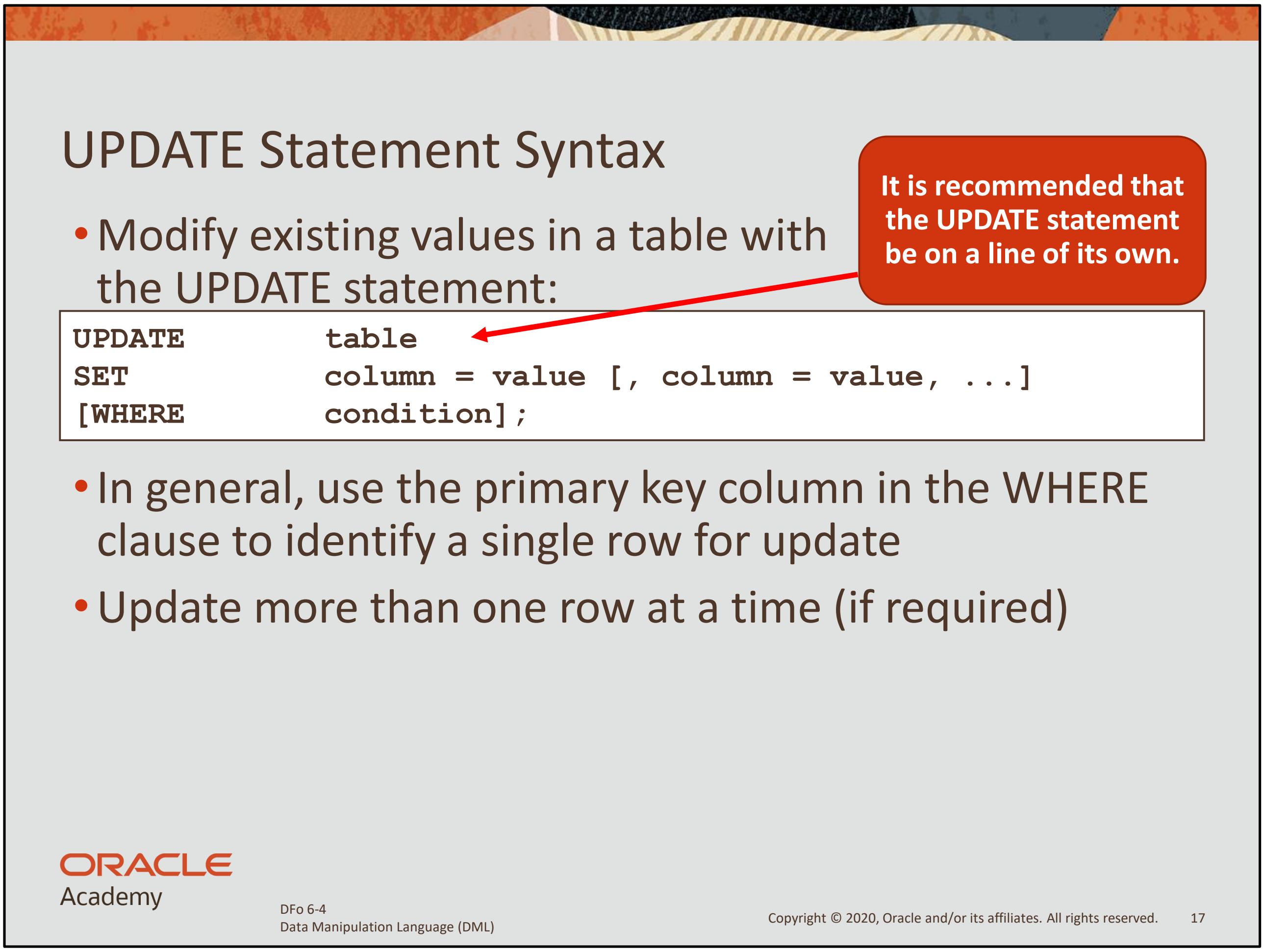
…

Update rows in the EMPLOYEES table:

EMPLOYEE _ID	FIRST_ NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_ DATE	JOB_ID	SALARY	COMMISSION_ PCT	MANAGER _ID	DEPARTMEN T_ID
100	Steven	King	SKING	515.123.4567	17-Jun-1987	AD_PRES	24000	-	-	60
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-Sep-1989	AD_VP	17000	-	100	60
102	Lex	De Haan	LDEHAAN	515.123.4569	13-Jan-1993	AD_VP	17000	-	100	60
200	Jennifer	Whalen	JWHALEN	515.123.4444	17-Sep-1987	AD_ASST	4400	-	101	10

…

ORACLE
Academy



UPDATE Statement Syntax

- Modify existing values in a table with the UPDATE statement:

```
UPDATE        table
SET           column = value [, column = value, ...]
[WHERE        condition];
```

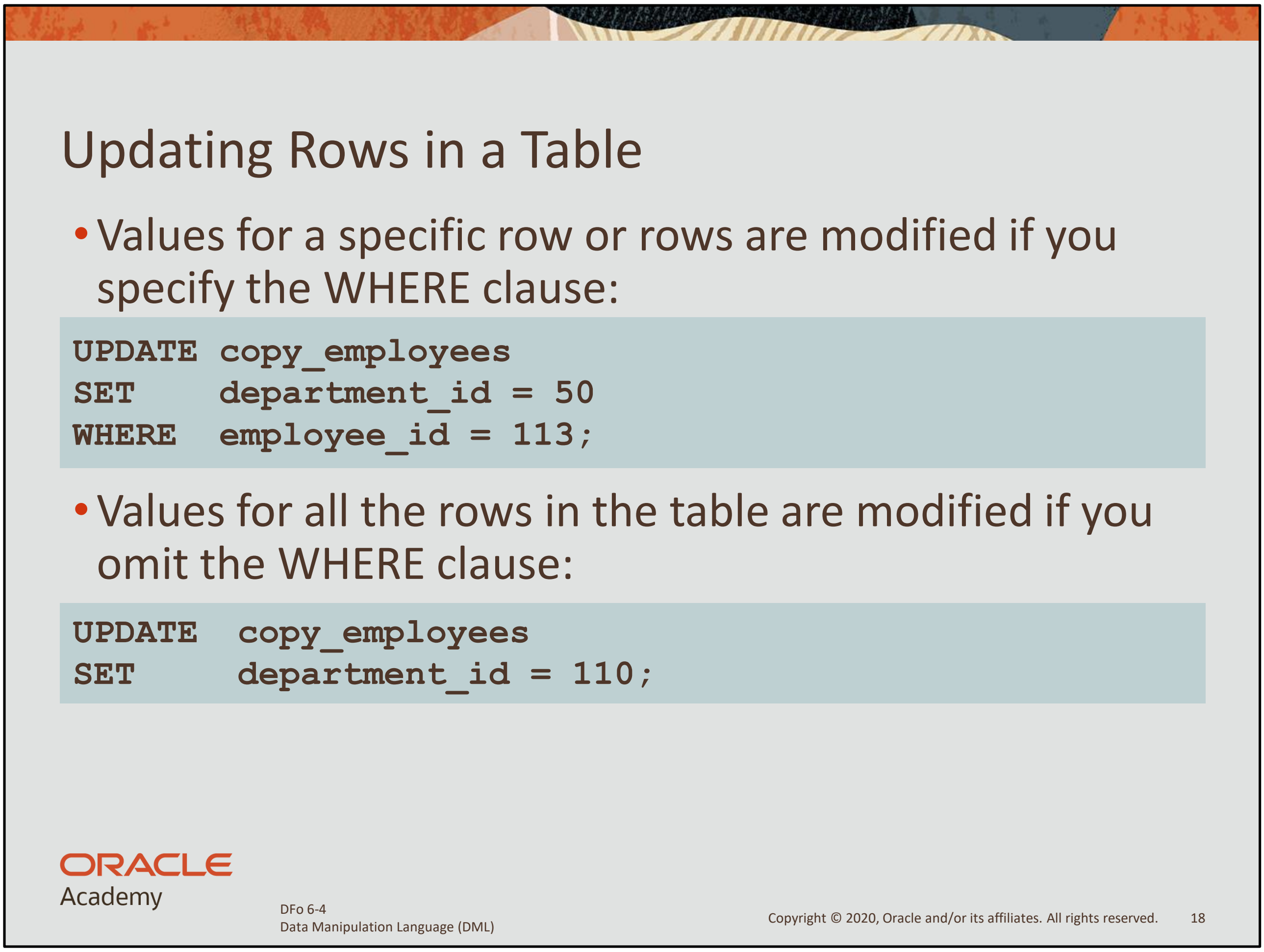
It is recommended that the UPDATE statement be on a line of its own.

- In general, use the primary key column in the WHERE clause to identify a single row for update
- Update more than one row at a time (if required)

ORACLE
Academy

In the syntax:

- table is the name of the table.
- column is the name of the column in the table to populate.
- value is the corresponding value or subquery for the column.
- condition identifies the rows to be updated and consists of column names, expressions, constants, subqueries, and comparison operators.



Updating Rows in a Table

- Values for a specific row or rows are modified if you specify the WHERE clause:

```
UPDATE copy_employees
SET    department_id = 50
WHERE  employee_id = 113;
```

- Values for all the rows in the table are modified if you omit the WHERE clause:

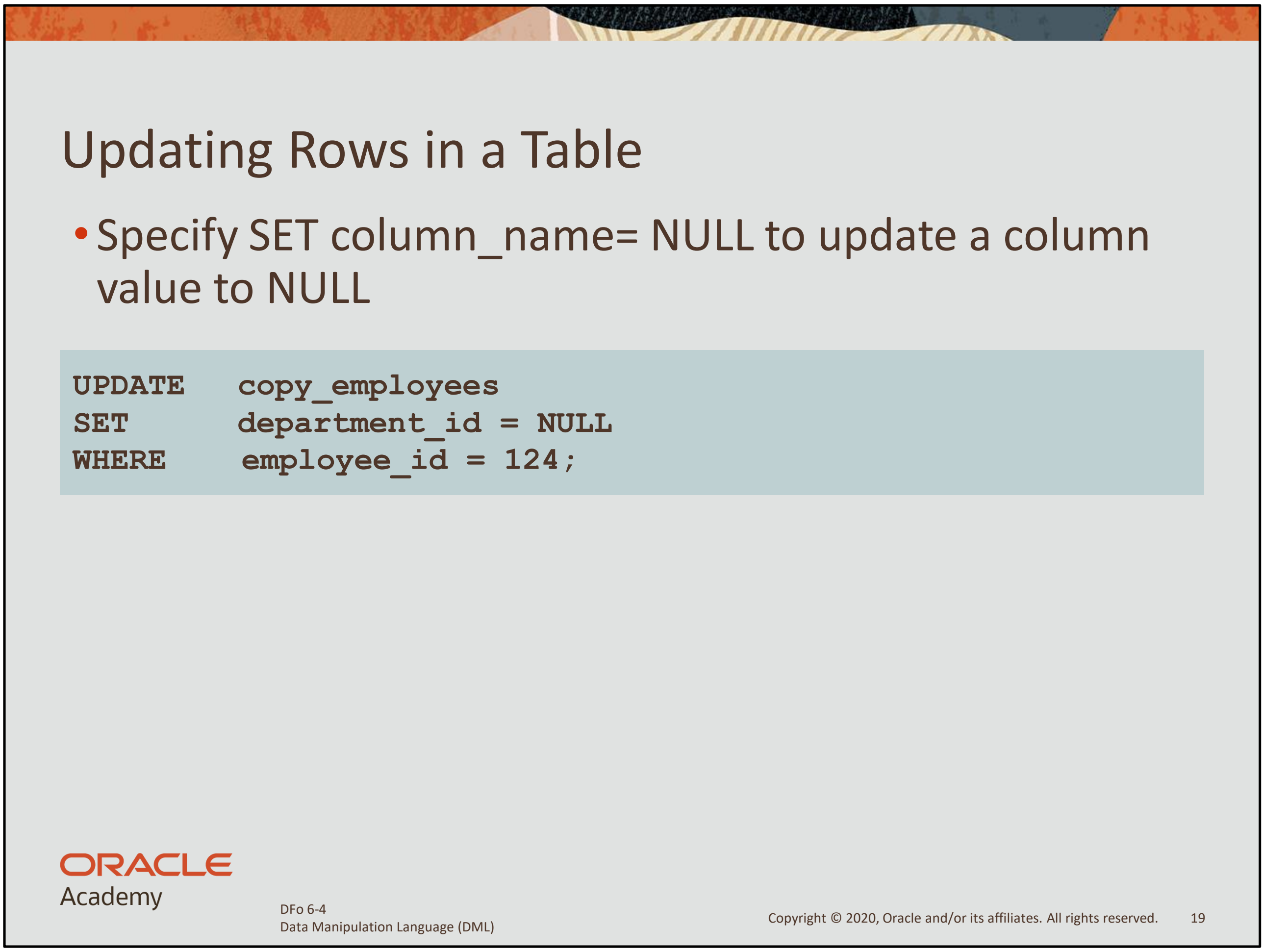
```
UPDATE  copy_employees
SET     department_id = 110;
```

ORACLE
Academy

For example, an employee who was an SA_REP has changed his job to an IT_PROG. Therefore, his JOB_ID needs to be updated, and the commission field needs to be set to NULL.

```
          UPDATE copy_employees
          SET job_id = 'IT_PROG', commission_pct = NULL
     ...
          WHERE employee_id = 114;
```

Note: The copy_employees table has the same data as the EMPLOYEES table.



Updating Rows in a Table

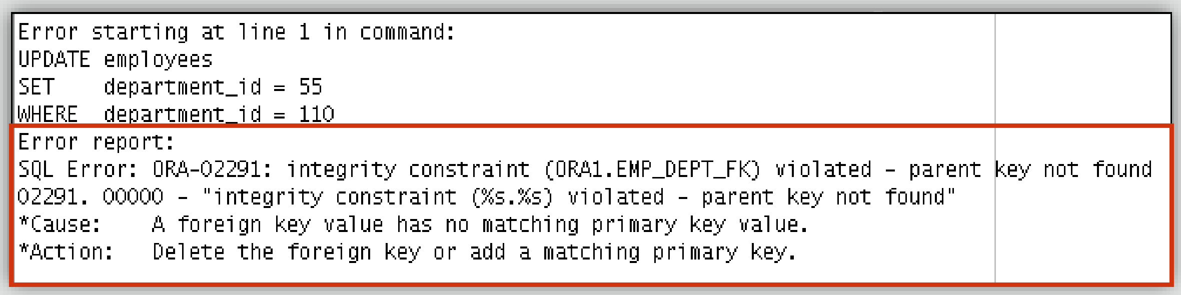
- Specify SET column_name= NULL to update a column value to NULL

```
UPDATE   copy_employees
SET      department_id = NULL
WHERE    employee_id = 124;
```

ORACLE
Academy

Violating Constraints

```
UPDATE   employees
SET      department_id = 55
WHERE    department_id = 110;
```



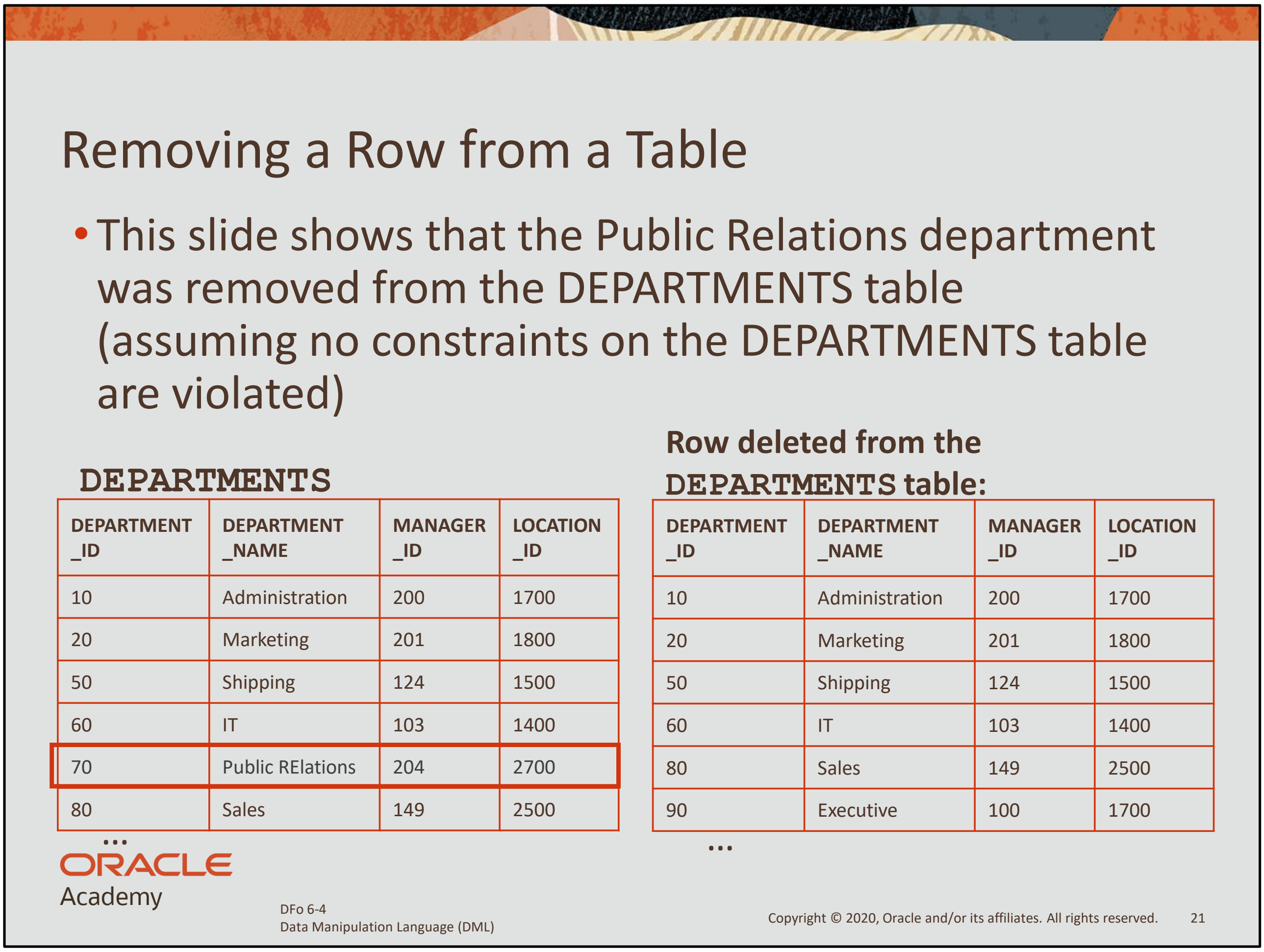
```
Error starting at line 1 in command:
UPDATE employees
SET    department_id = 55
WHERE  department_id = 110
Error report:
SQL Error: ORA-02291: integrity constraint (ORA1.EMP_DEPT_FK) violated - parent key not found
02291. 00000 - "integrity constraint (%s.%s) violated - parent key not found"
*Cause:    A foreign key value has no matching primary key value.
*Action:   Delete the foreign key or add a matching primary key.
```

- Department 55 does not exist on the parent table – DEPARTMENTS

Note : notice the query is using the employees table which has integrity constraints in effect (copy_employees – does not have these constraints in effect).

When constraints are in place on columns, an error is returned if you try to violate the constraint rule. For example, an error is returned if you try to update a record with a value that is tied to an integrity constraint.

In the slide example, department 55 does not exist in the parent table, DEPARTMENTS, and so you receive the "parent key not found" ORA-02291 violation.



Removing a Row from a Table

- This slide shows that the Public Relations department was removed from the DEPARTMENTS table (assuming no constraints on the DEPARTMENTS table are violated)

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
70	Public RElations	204	2700
80	Sales	149	2500

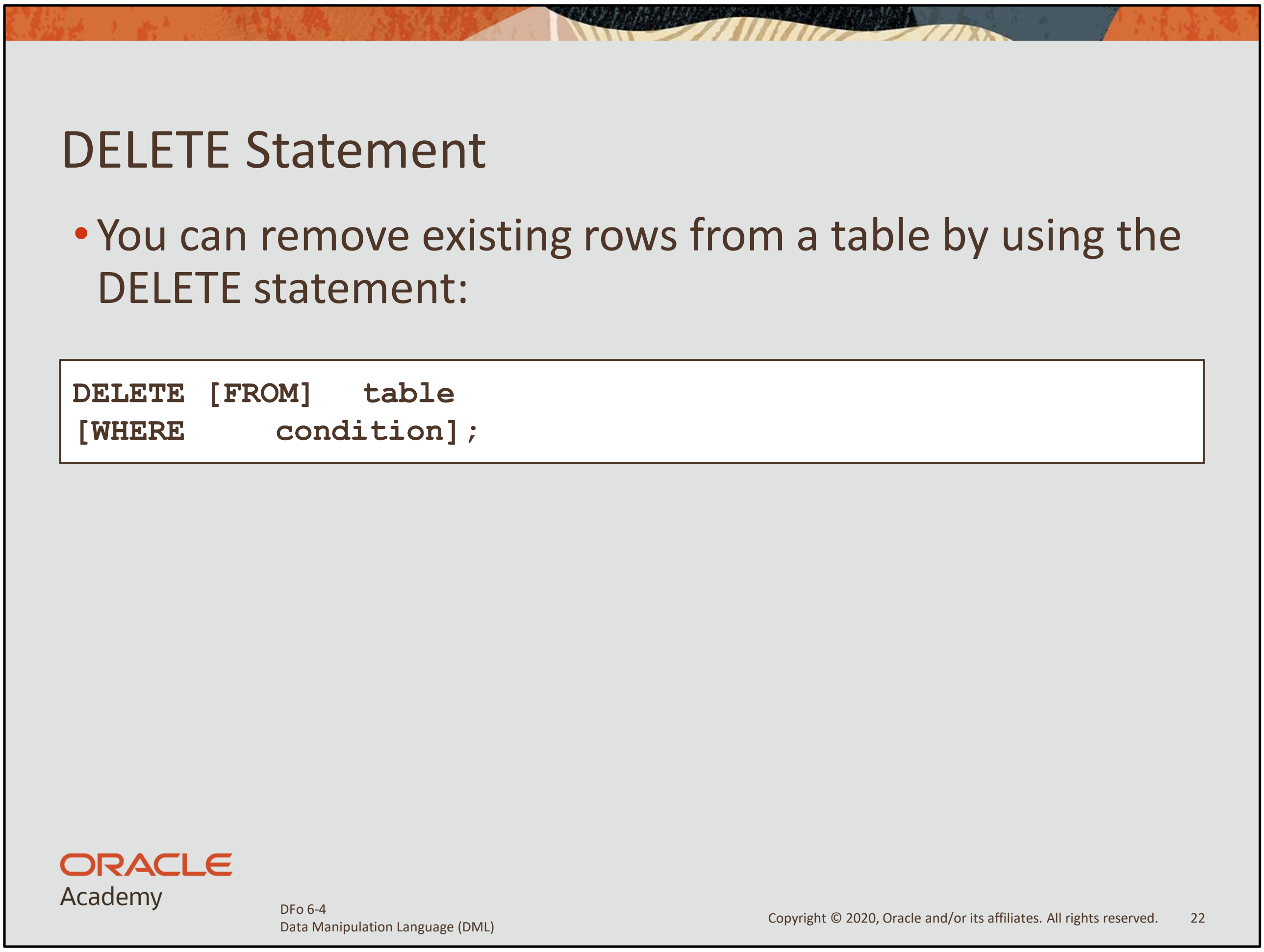
…

Row deleted from the DEPARTMENTS table:

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

…

ORACLE Academy



DELETE Statement

- You can remove existing rows from a table by using the DELETE statement:

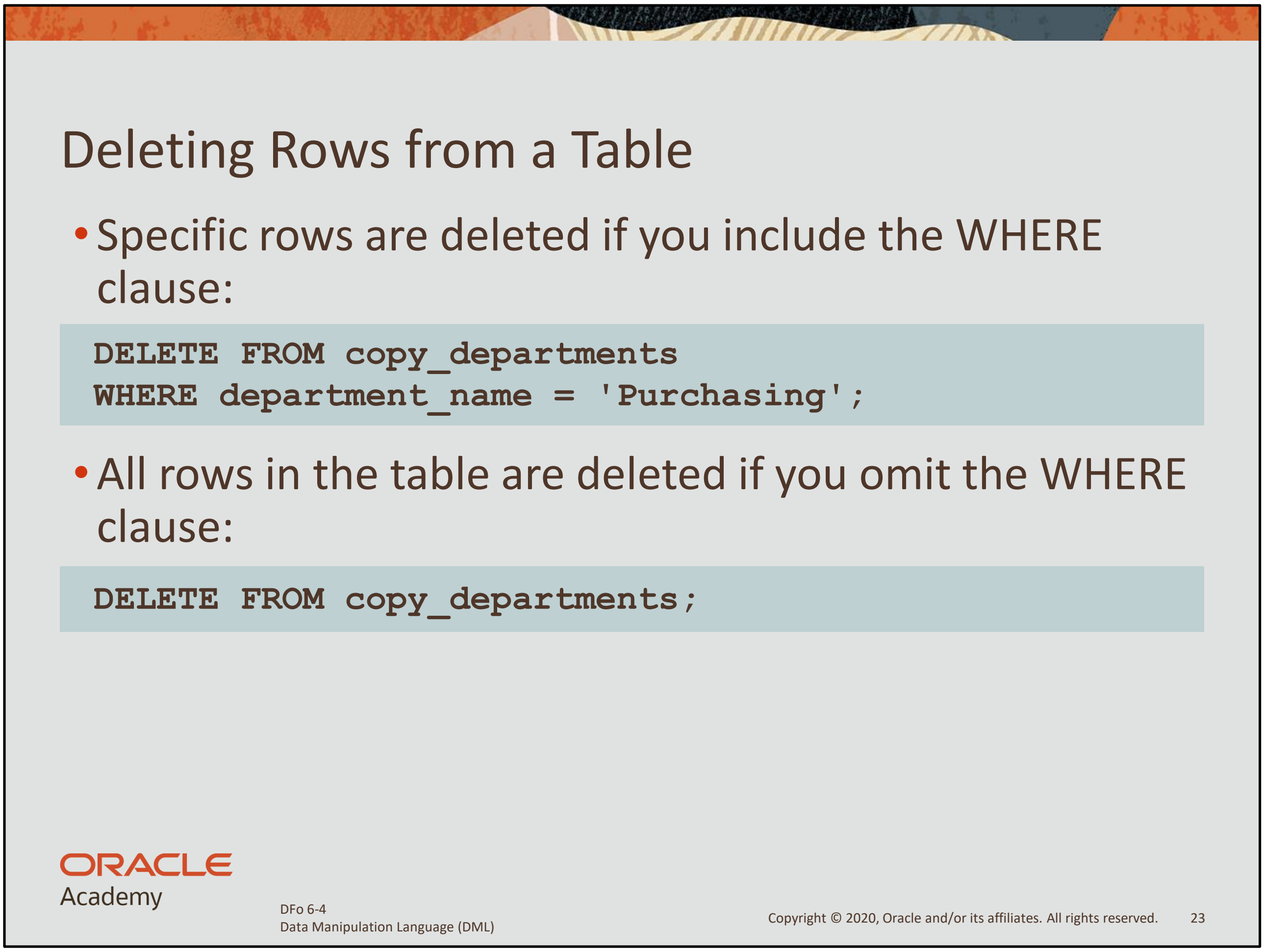
```
DELETE [FROM]   table
[WHERE      condition];
```

ORACLE Academy

In the syntax:

- table is the name of the table.
- condition identifies the rows to be deleted, and consists of column names, expressions, constants, subqueries, and comparison operators.

Note: If no rows are deleted, the "0 rows deleted" message is returned.



Deleting Rows from a Table

- Specific rows are deleted if you include the WHERE clause:

```
DELETE FROM copy_departments
WHERE department_name = 'Purchasing';
```

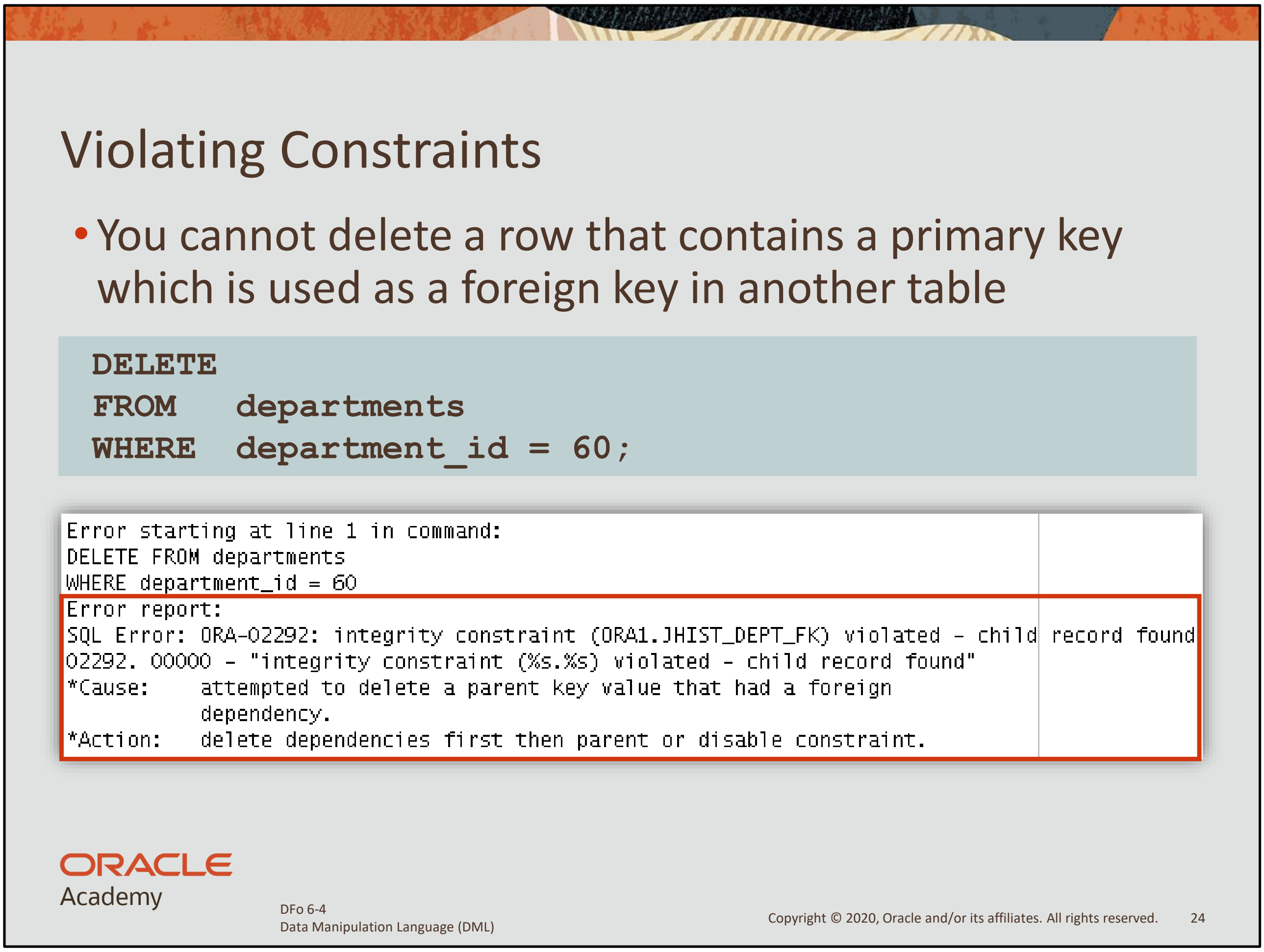
- All rows in the table are deleted if you omit the WHERE clause:

```
DELETE FROM copy_departments;
```

ORACLE Academy

Here is another example of removing rows identified in the WHERE clause:

```
DELETE FROM copy_departments WHERE department_id IN (30, 40);
```



Violating Constraints

- You cannot delete a row that contains a primary key which is used as a foreign key in another table

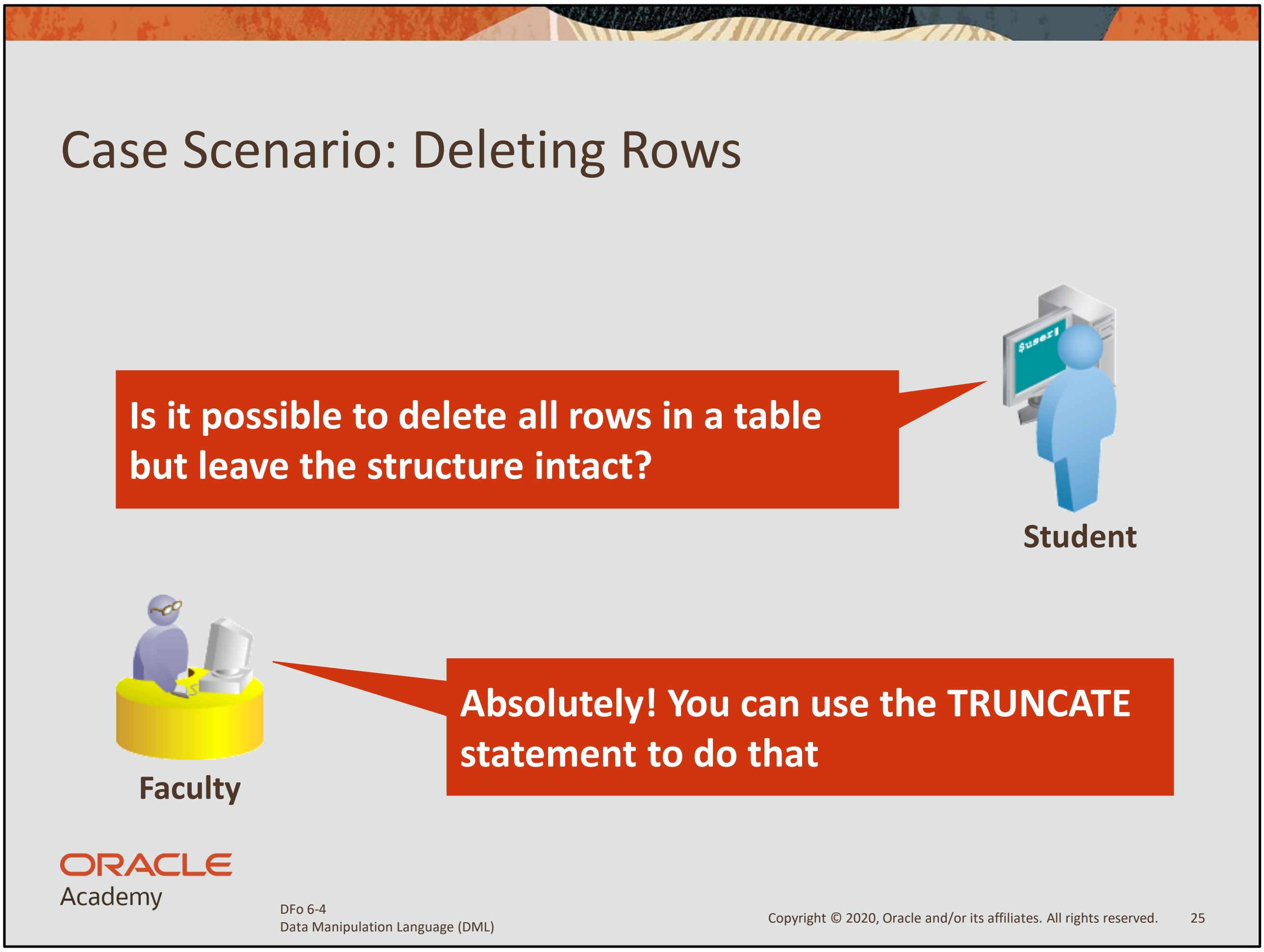
```
DELETE
FROM   departments
WHERE  department_id = 60;
```

```
Error starting at line 1 in command:
DELETE FROM departments
WHERE department_id = 60
Error report:
SQL Error: ORA-02292: integrity constraint (ORA1.JHIST_DEPT_FK) violated - child record found
02292. 00000 - "integrity constraint (%s.%s) violated - child record found"
*Cause:    attempted to delete a parent key value that had a foreign
           dependency.
*Action:   delete dependencies first then parent or disable constraint.
```

Note : notice the query is using the departments table which has integrity constraints in effect (copy_departments – does not have these constraints in effect).

If you attempt to delete a record with a value that is tied to an integrity constraint, an error is returned.

The slide example shows an attempt to delete department 60 from the DEPARTMENTS table, but it results in an error because that department number is used as a foreign key in the EMPLOYEES table. If the parent record that you attempt to delete has child records, you receive the "child record found" ORA-02292 violation.

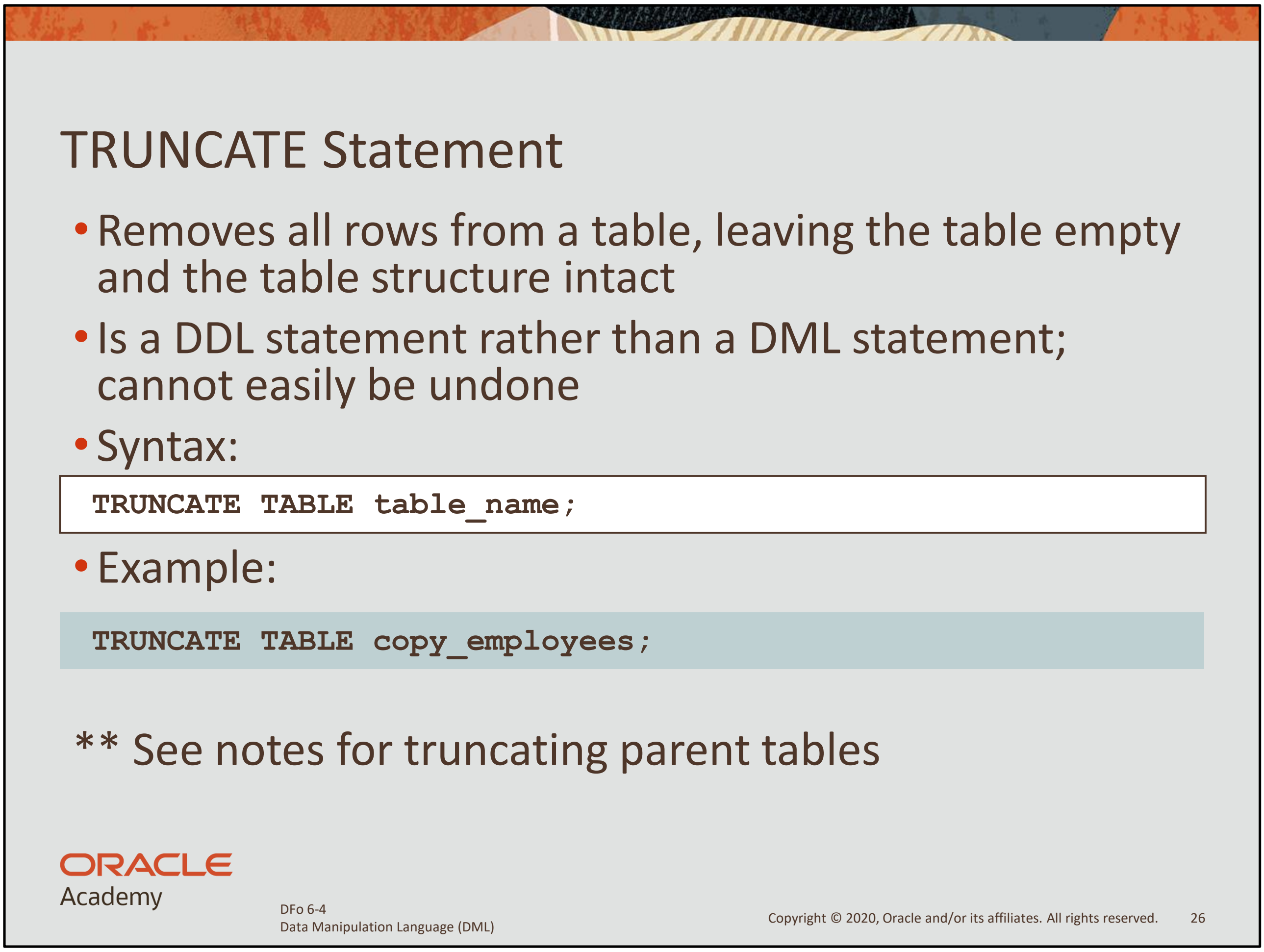


Case Scenario: Deleting Rows

Is it possible to delete all rows in a table but leave the structure intact?

Absolutely! You can use the TRUNCATE statement to do that

ORACLE
Academy



TRUNCATE Statement

- Removes all rows from a table, leaving the table empty and the table structure intact
- Is a DDL statement rather than a DML statement; cannot easily be undone
- Syntax:

```
TRUNCATE TABLE table_name;
```

- Example:

```
TRUNCATE TABLE copy_employees;
```

** See notes for truncating parent tables

ORACLE
Academy

DFo 6-4
Data Manipulation Language (DML)

The TRUNCATE statement is a more efficient method to remove all rows from a table or cluster.

Removing rows with the TRUNCATE statement is faster than removing them with the DELETE statement for the following reasons:

- The TRUNCATE statement is a DDL statement and generates no rollback information. Rollback information is covered later in this lesson.
- Truncating a table does not fire the delete triggers of the table.

If the table is the parent of a referential integrity constraint, you cannot truncate the table. You need to disable the constraint before issuing the TRUNCATE statement. Disabling constraints is covered in the lesson titled "Introduction to DDL Statements."

Project Exercise 2

- DFo_6_4_2_Project
  - Oracle Baseball League Store Database
  - Using DML operations to manage database tables : Updating and deleting records

ORACLE
Academy

Summary

- In this lesson, you should have learned how to:
  - Describe the purpose of DML
  - List the DML operations that are required to manage a database's table data:
    - INSERT
    - UPDATE
    - DELETE

ORACLE
Academy

ORACLE

Academy