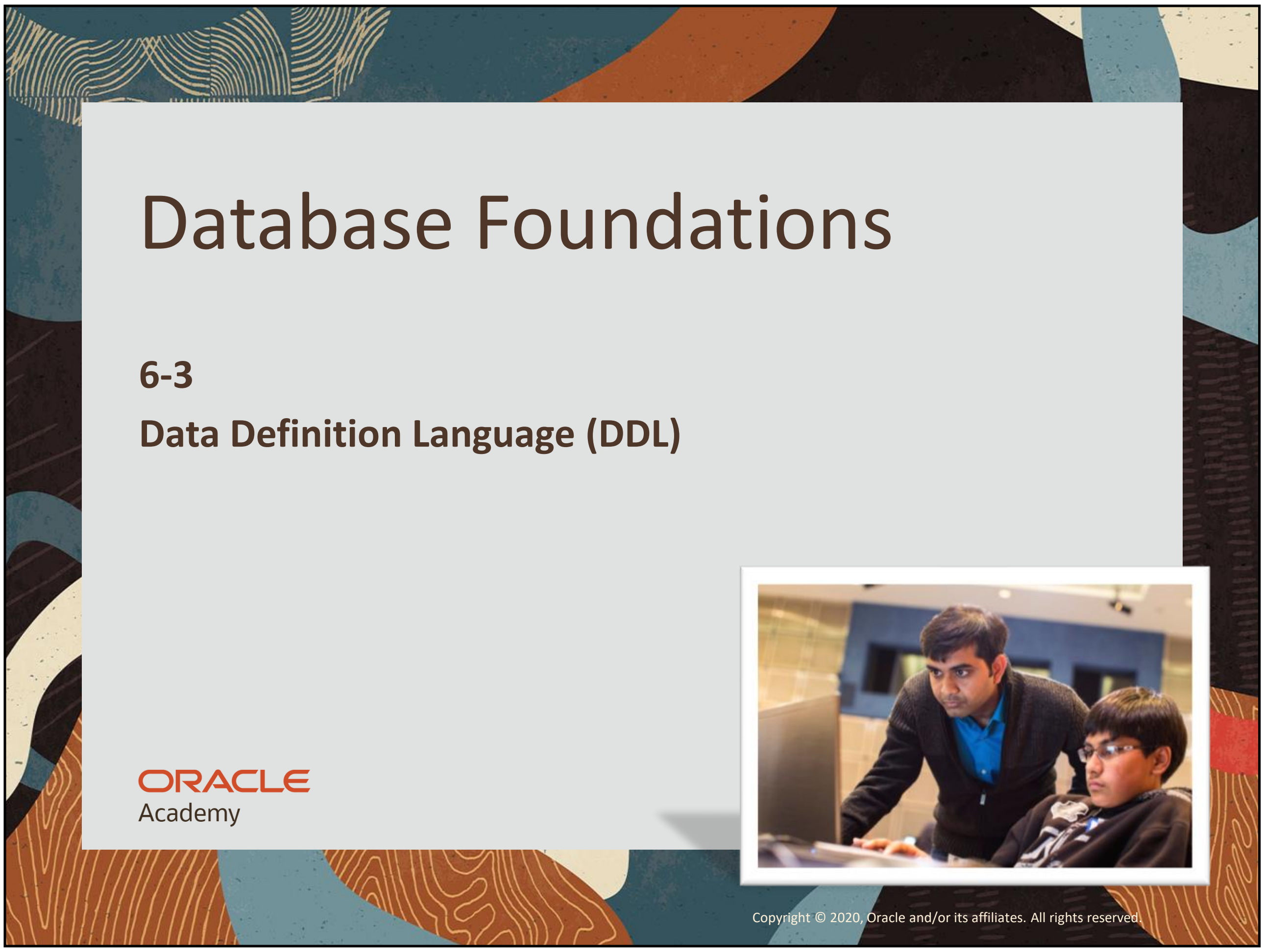
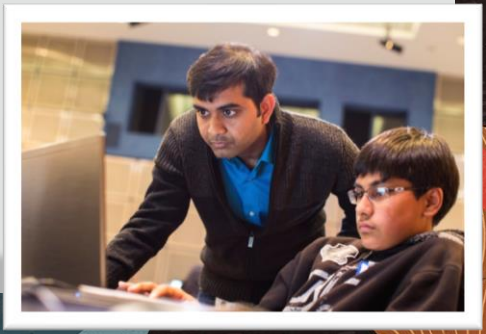
ORACLE
Academy



Database Foundations

6-3

Data Definition Language (DDL)

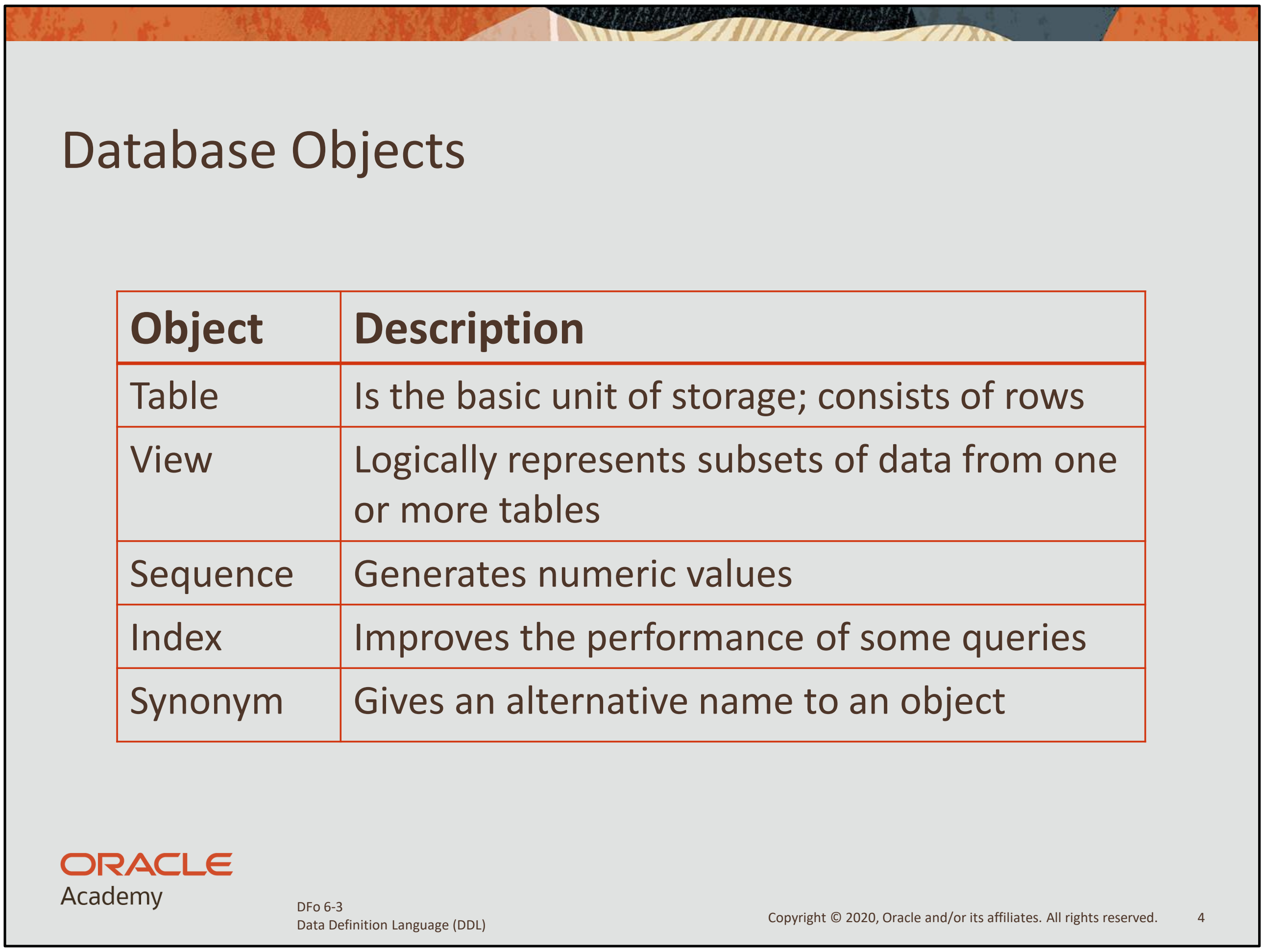


ORACLE
Academy

Objectives

- This lesson covers the following objectives:
  - Identify the steps needed to create database tables
  - Describe the purpose of the data definition language (DDL)
  - List the DDL operations needed to build and maintain a database's tables

ORACLE
Academy

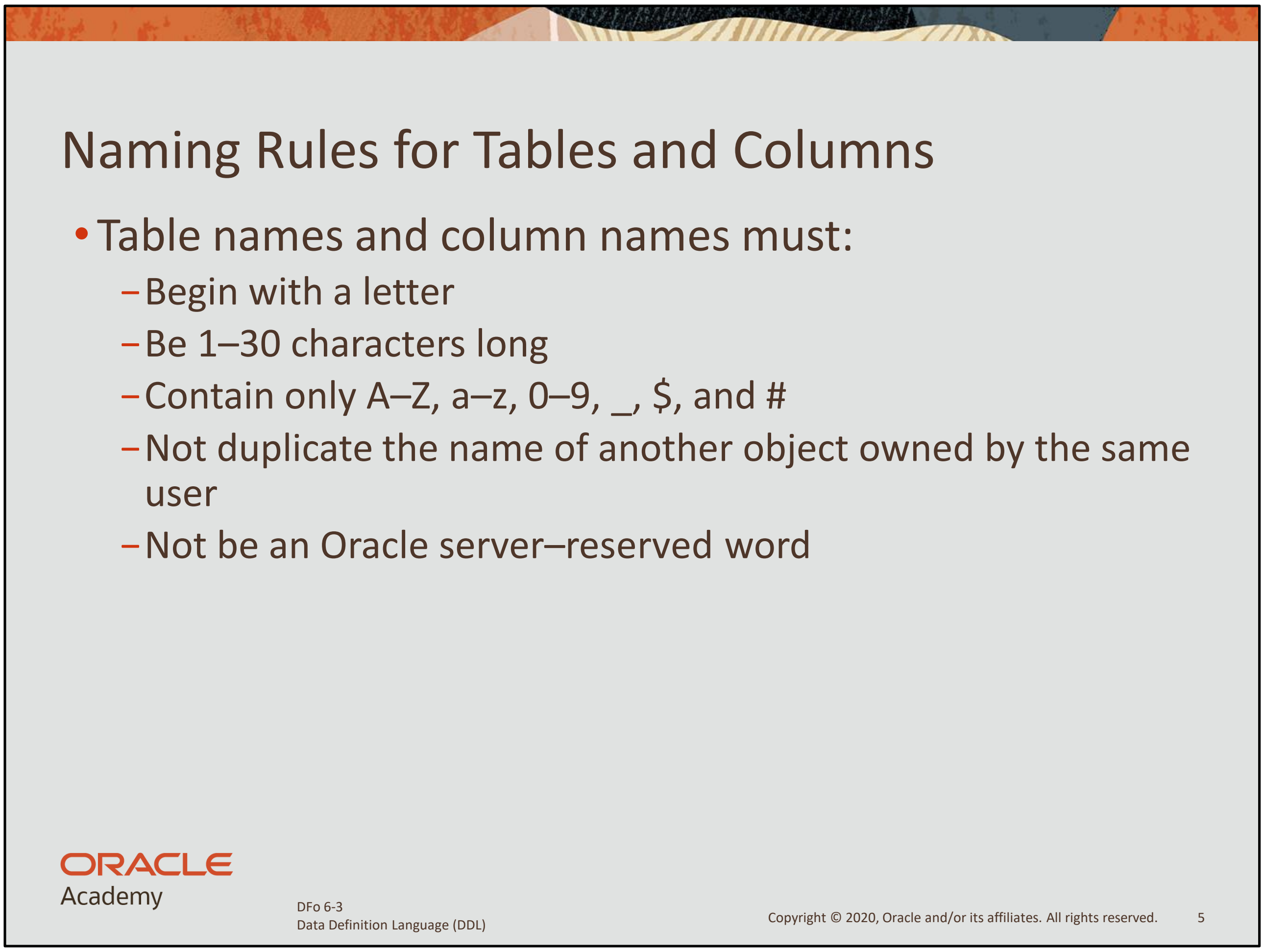


Database Objects

Object	Description
Table	Is the basic unit of storage; consists of rows
View	Logically represents subsets of data from one or more tables
Sequence	Generates numeric values
Index	Improves the performance of some queries
Synonym	Gives an alternative name to an object

ORACLE Academy

Note: In this course, we will create and retrieve information from the basic unit of storage, tables. More database objects than those listed are available, but they are not covered in this course.



Naming Rules for Tables and Columns

- Table names and column names must:
  - Begin with a letter
  - Be 1–30 characters long
  - Contain only A–Z, a–z, 0–9, _, $, and #
  - Not duplicate the name of another object owned by the same user
  - Not be an Oracle server–reserved word

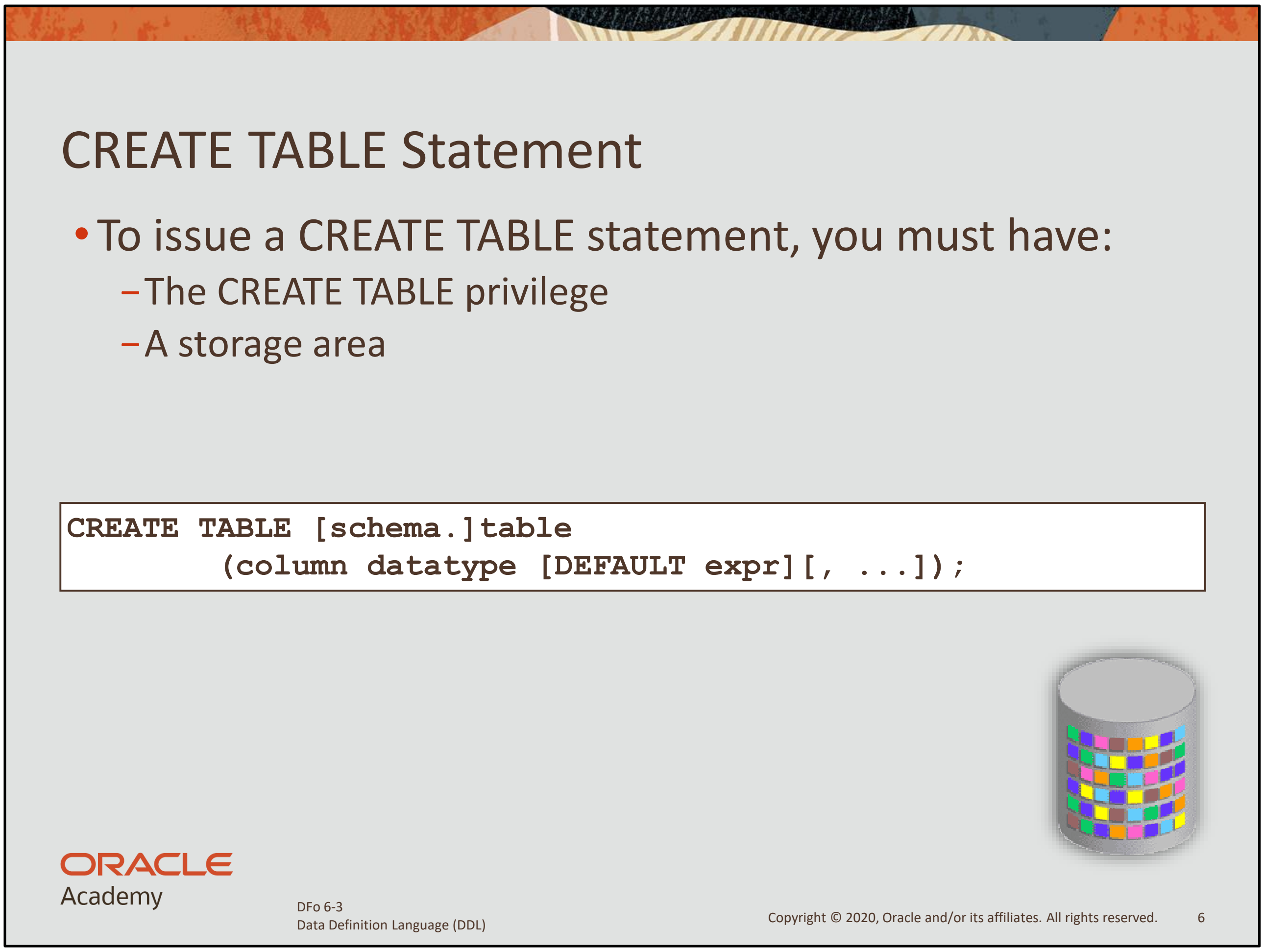
ORACLE Academy

DFo 6-3
Data Definition Language (DDL)

5

Note: Names are not case-sensitive. For example, EMPLOYEES is treated the same as eMPloyees or eMpLOYEES. However, quoted identifiers are case-sensitive.

For a complete list of reserved words see :
https://docs.oracle.com/cd/B28359_01/appdev.111/b31231/appb.htm#CJHIIICD



CREATE TABLE Statement

- To issue a CREATE TABLE statement, you must have:
  - The CREATE TABLE privilege
  - A storage area

```
CREATE TABLE [schema.]table
         (column datatype [DEFAULT expr][, ...]);
```

ORACLE Academy

To create a table, a user must have the CREATE TABLE privilege and a storage area in which to create objects. The database administrator (DBA) uses data control language (DCL) statements to grant privileges to users.

In the syntax:

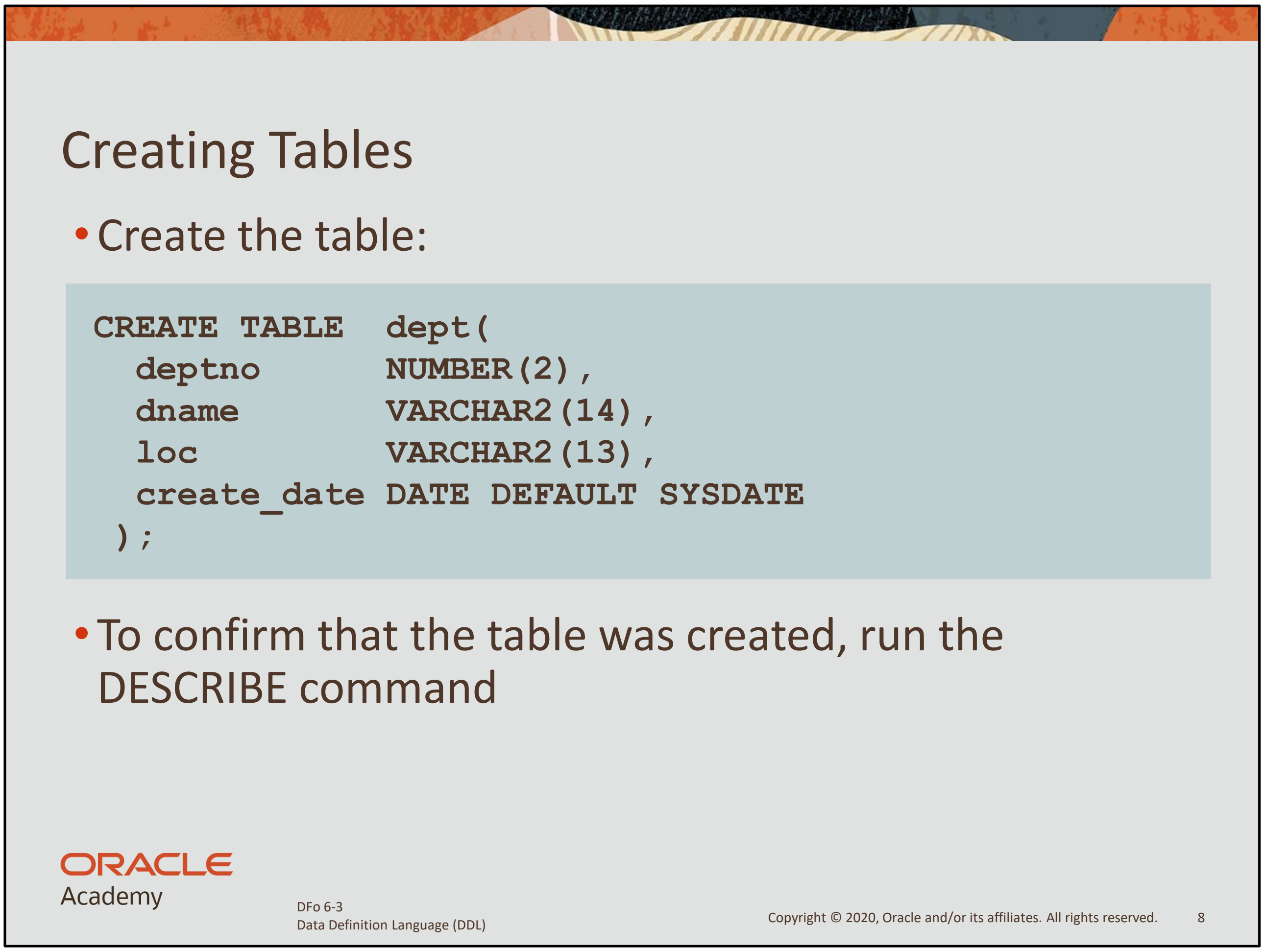
- schema is the same as the owner's name.
- table is the name of the table.
- DEFAULT expr specifies a default value if a value is omitted in the INSERT statement.
- column is the name of the column.
- datatype is the column's data type and length.

Note: The CREATE ANY TABLE privilege is needed to create a table in any schema other than the user's schema.

CREATE TABLE Statement

- Specify in the statement:
  - Table name
  - Column name, column data type, column size
  - Integrity constraints (optional)
  - Default values (optional)

```
CREATE TABLE [schema.]table
        (column datatype [DEFAULT expr][, ...]);
```



Creating Tables

- Create the table:

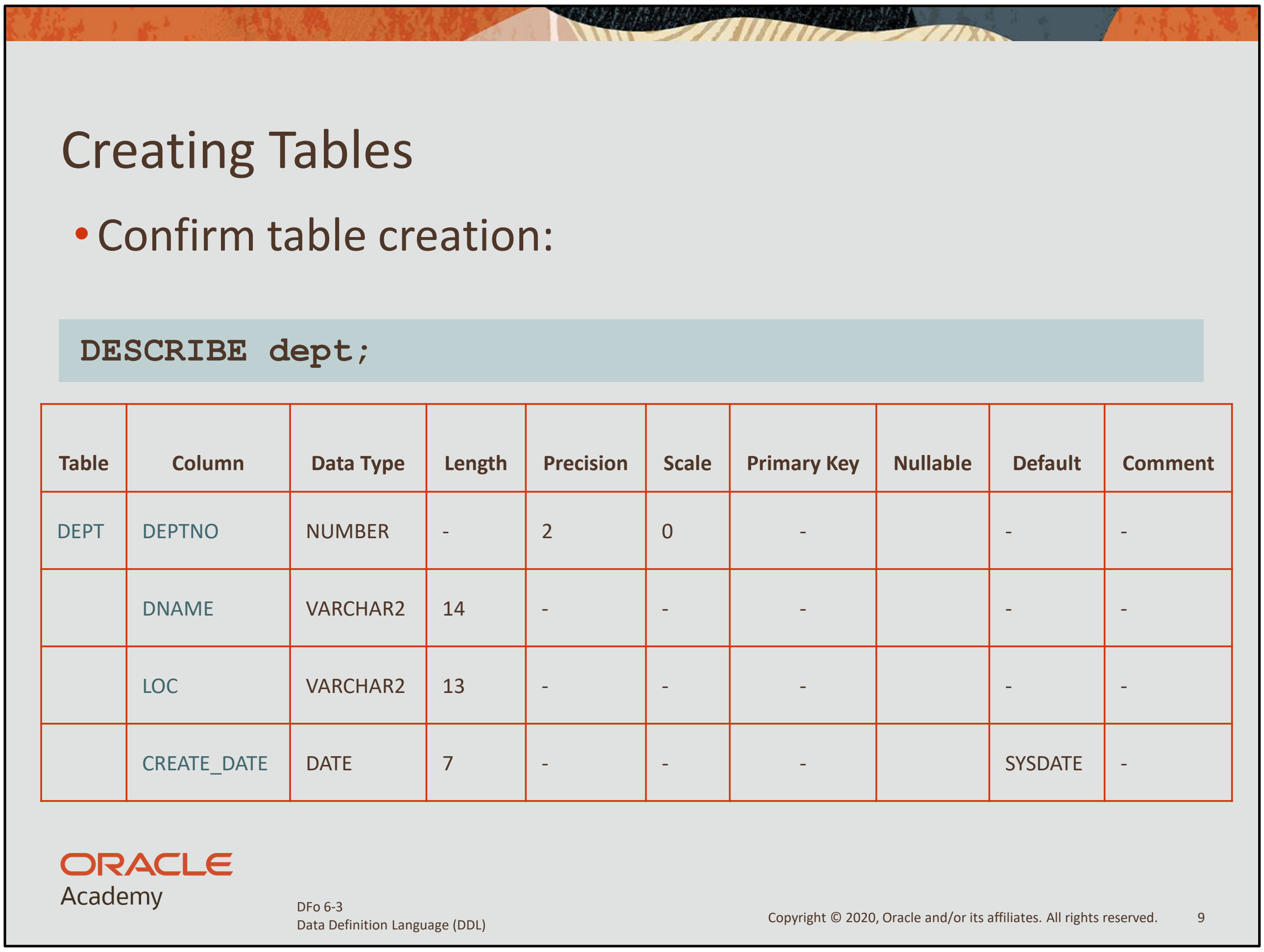
```
CREATE TABLE  dept(
  deptno      NUMBER(2),
  dname       VARCHAR2(14),
  loc         VARCHAR2(13),
  create_date DATE DEFAULT SYSDATE
 );
```

- To confirm that the table was created, run the DESCRIBE command

ORACLE Academy

Note: You can view the list of tables that you own by querying the data dictionary. For example, select table_name from user_tables;

For more information data dictionary tables see :
https://docs.oracle.com/database/121/GMSWN/apc.htm#GMSWN600



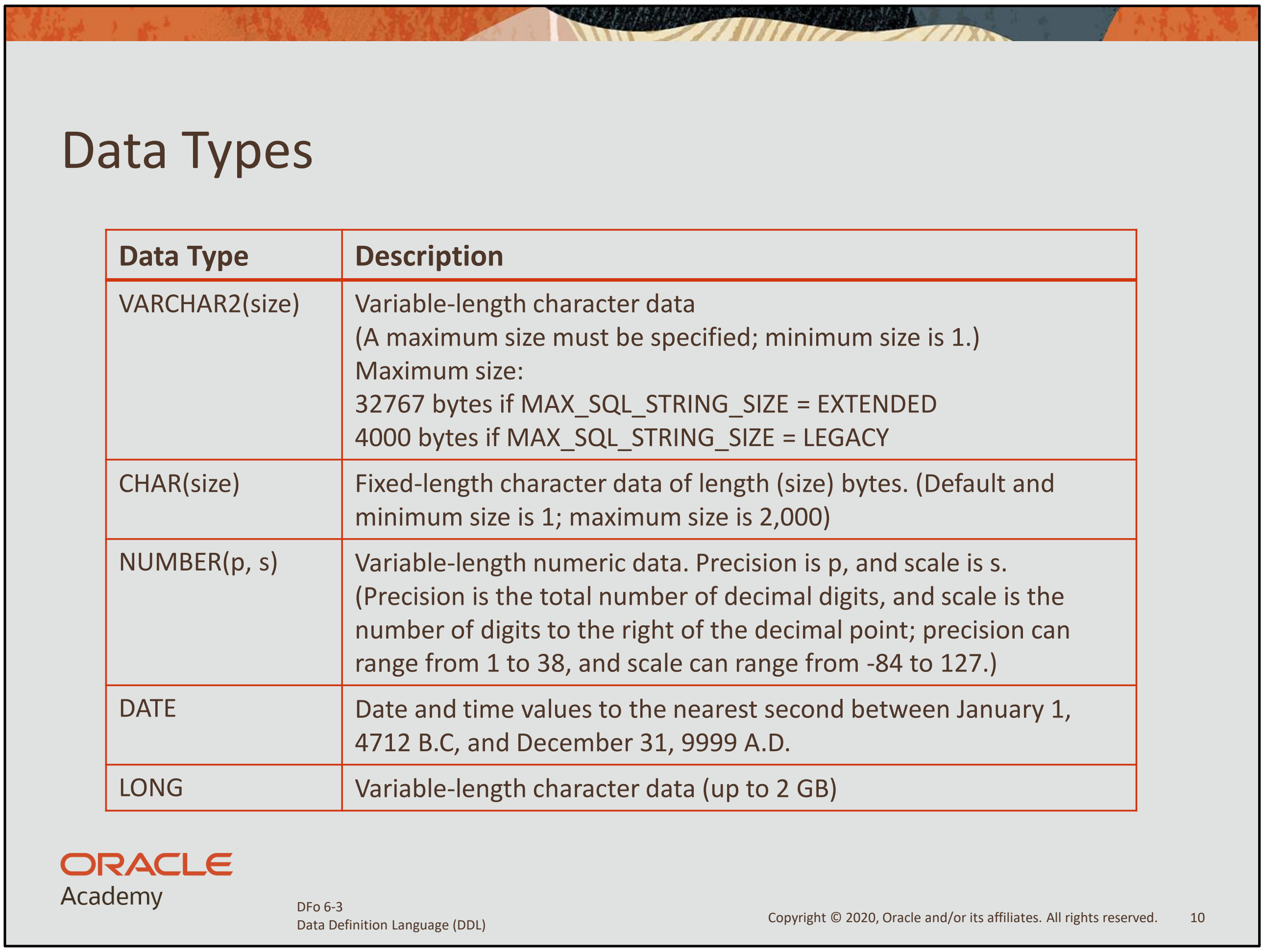
Creating Tables

- Confirm table creation:

```
DESCRIBE dept;
```

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DEPT	DEPTNO	NUMBER	-	2	0	-		-	-
	DNAME	VARCHAR2	14	-	-	-		-	-
	LOC	VARCHAR2	13	-	-	-		-	-
	CREATE_DATE	DATE	7	-	-	-		SYSDATE	-

ORACLE Academy



Data Types

Data Type	Description
VARCHAR2(size)	Variable-length character data (A maximum size must be specified; minimum size is 1.) Maximum size: 32767 bytes if MAX_SQL_STRING_SIZE = EXTENDED 4000 bytes if MAX_SQL_STRING_SIZE = LEGACY
CHAR(size)	Fixed-length character data of length (size) bytes. (Default and minimum size is 1; maximum size is 2,000)
NUMBER(p, s)	Variable-length numeric data. Precision is p, and scale is s. (Precision is the total number of decimal digits, and scale is the number of digits to the right of the decimal point; precision can range from 1 to 38, and scale can range from -84 to 127.)
DATE	Date and time values to the nearest second between January 1, 4712 B.C, and December 31, 9999 A.D.
LONG	Variable-length character data (up to 2 GB)

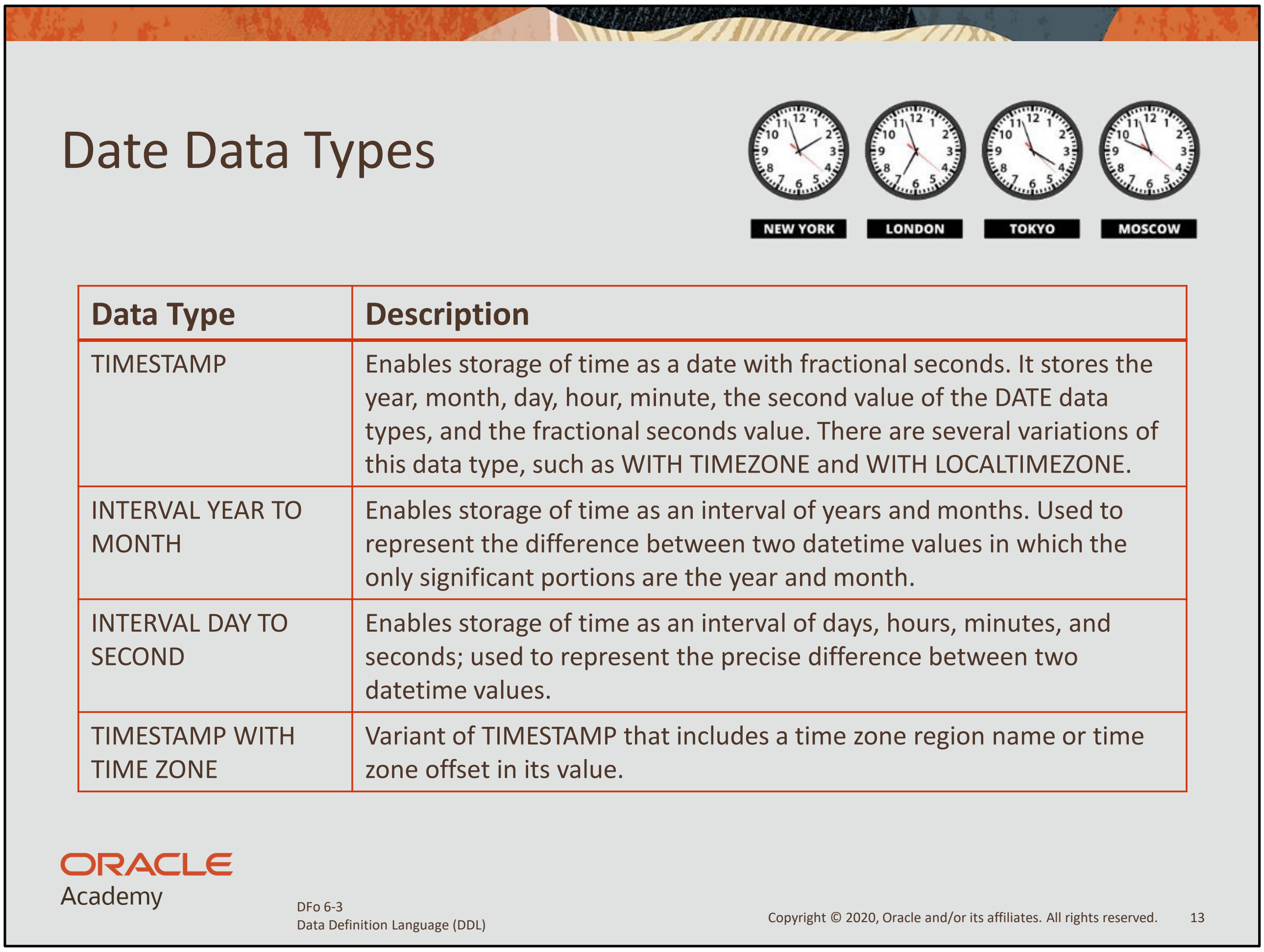
ORACLE
Academy

Data Types

Data Type	Description
CLOB	A character large object (CLOB) containing single-byte or multibyte characters. Maximum size is (4 GB - 1) * (DB_BLOCK_SIZE); stores national character set data.
NCLOB	A CLOB containing Unicode characters. Both fixed-width and variable-width character sets are supported, both using the database national character set. Maximum size is (4 GB - 1) * (database block size); stores national character set data.
RAW (Size)	Raw binary data of length size bytes. You must specify size for a RAW value. Maximum size: 32767 bytes if MAX_SQL_STRING_SIZE = EXTENDED 4000 bytes if MAX_SQL_STRING_SIZE = LEGACY
LONG RAW	Raw binary data of variable length up to 2 GB.
BLOB	A binary large object. Maximum size is (4 GB - 1) * (DB_BLOCK_SIZE initialization parameter (8 TB to 128 TB)).
BFILE	Binary data stored in an external file (up to 4 GB).
ROWID	Base 64 string representing the unique address of a row in its table. This data type is primarily for values returned by the ROWID pseudocolumn

Example: Creating a Table with Different Data Types

```
CREATE TABLE   print_media(
  product_id   NUMBER(6),
  id           NUMBER(6),
  desc         VARCHAR2(100),
  composite    BLOB,
  msourcetext  CLOB,
  finaltext    CLOB,
  photo        BLOB,
  graphic      BFILE
);
```



Date Data Types

Data Type	Description
TIMESTAMP	Enables storage of time as a date with fractional seconds. It stores the year, month, day, hour, minute, the second value of the DATE data types, and the fractional seconds value. There are several variations of this data type, such as WITH TIMEZONE and WITH LOCALTIMEZONE.
INTERVAL YEAR TO MONTH	Enables storage of time as an interval of years and months. Used to represent the difference between two datetime values in which the only significant portions are the year and month.
INTERVAL DAY TO SECOND	Enables storage of time as an interval of days, hours, minutes, and seconds; used to represent the precise difference between two datetime values.
TIMESTAMP WITH TIME ZONE	Variant of TIMESTAMP that includes a time zone region name or time zone offset in its value.

ORACLE
Academy

You can use several date data types.

TIMESTAMP WITH TIMEZONE example:

```
CREATE TABLE table_tstz (c_id NUMBER, c_tstz TIMESTAMP WITH
TIME ZONE);
INSERT INTO table_tstz VALUES(1, '01-JAN-2003 2:00:00 AM –
07:00');
```

Examples: Date Data Types

- Example of TIMESTAMP data type:

```
CREATE TABLE table_ts(
   c_id NUMBER(6),
   c_ts TIMESTAMP
);
```

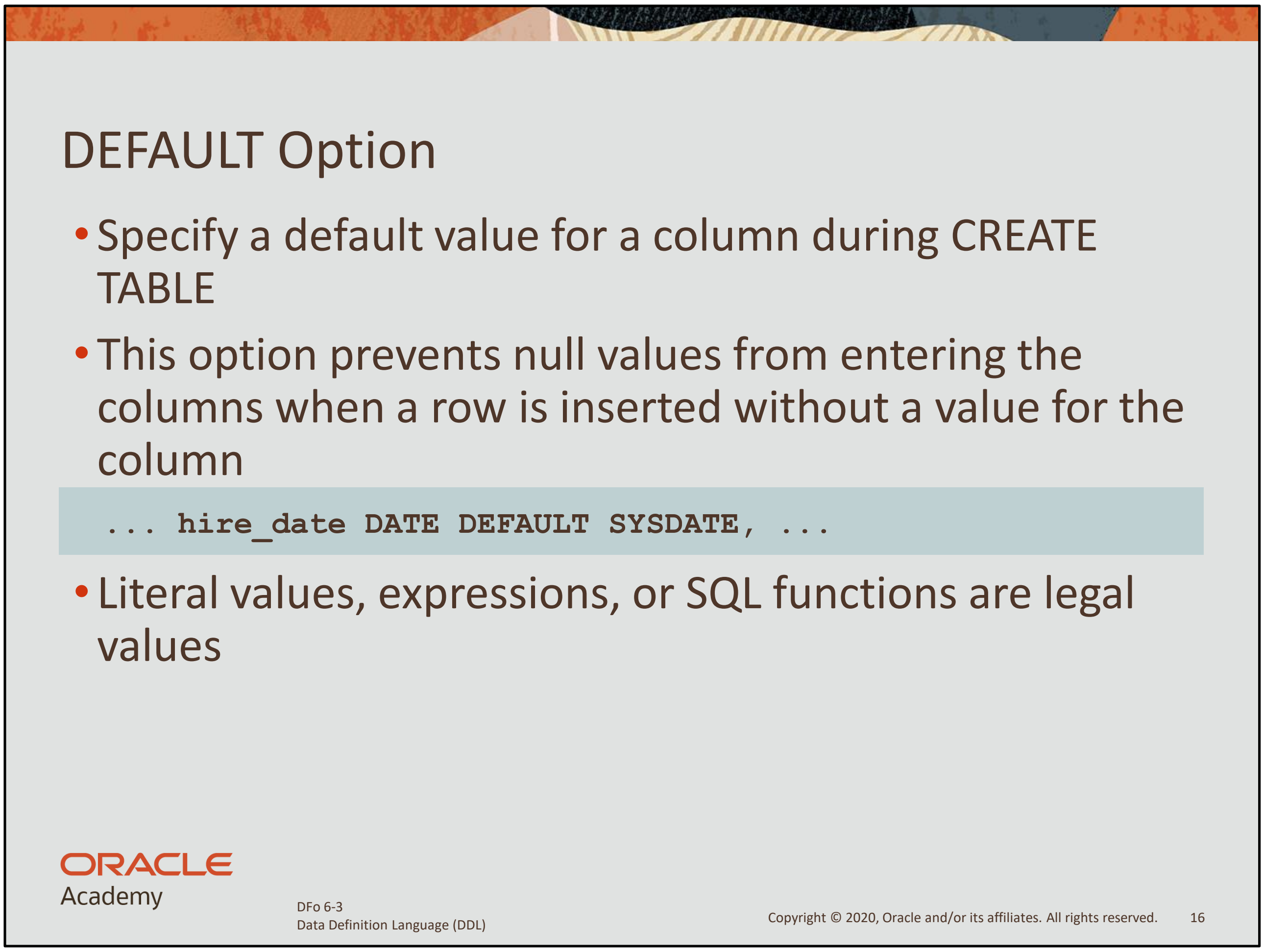
```
INSERT INTO table_ts
VALUES(1, '01-JAN-2003 2:00:00');
```

Examples: Date Data Types

- Example of a table with TIMESTAMP, INTERVAL YEAR TO MONTH and INTERVAL DAY TO SECOND columns:

```
CREATE TABLE time_table(
   start_time    TIMESTAMP,
   duration_1    INTERVAL DAY (6) TO SECOND (5),
   duration_2    INTERVAL YEAR TO MONTH
);
```

ORACLE
Academy



DEFAULT Option

- Specify a default value for a column during CREATE TABLE
- This option prevents null values from entering the columns when a row is inserted without a value for the column

```
... hire_date DATE DEFAULT SYSDATE, ...
```

- Literal values, expressions, or SQL functions are legal values

ORACLE Academy

DFo 6-3
Data Definition Language (DDL)

16

Consider the following example, where the statement inserts the NULL value rather than the default value:

INSERT INTO hire_dates values(45, NULL);

In the next example, the statement inserts the SYSDATE for the HIRE_DATE column because it is the DEFAULT value:

INSERT INTO hire_dates(id) values(35);

DEFAULT Option

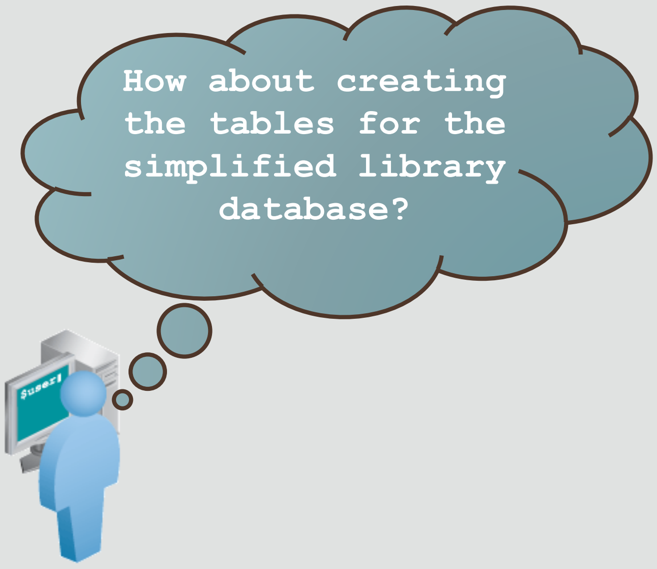
- Another column's name or a pseudocolumn are illegal values
- The default data type must match the column data type

```
CREATE TABLE  hire_dates(
  id          NUMBER(8),
  hire_date   DATE DEFAULT SYSDATE
);
```

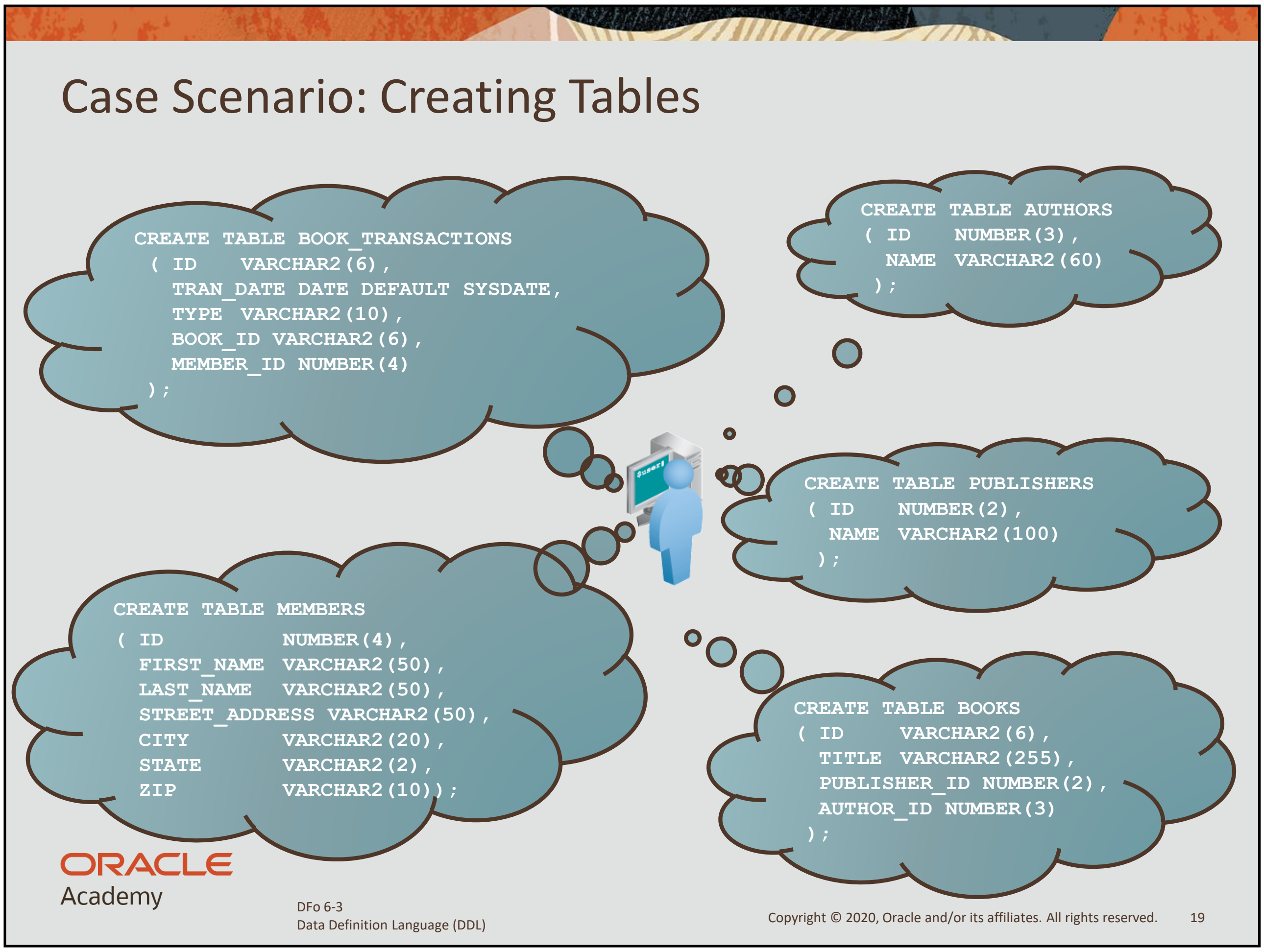
```
Table created.
```

ORACLE
Academy

Case Scenario: Creating Tables


How about creating the tables for the simplified library database?
$user1

ORACLE
Academy



Case Scenario: Creating Tables

```
CREATE TABLE BOOK_TRANSACTIONS
 ( ID    VARCHAR2(6),
   TRAN_DATE DATE DEFAULT SYSDATE,
   TYPE  VARCHAR2(10),
   BOOK_ID VARCHAR2(6),
   MEMBER_ID NUMBER(4)
 );
```

```
CREATE TABLE AUTHORS
 ( ID    NUMBER(3),
   NAME  VARCHAR2(60)
 );
```

```
CREATE TABLE PUBLISHERS
 ( ID    NUMBER(2),
   NAME  VARCHAR2(100)
 );
```

```
CREATE TABLE MEMBERS
( ID            NUMBER(4),
  FIRST_NAME    VARCHAR2(50),
  LAST_NAME     VARCHAR2(50),
  STREET_ADDRESS VARCHAR2(50),
  CITY          VARCHAR2(20),
  STATE         VARCHAR2(2),
  ZIP           VARCHAR2(10));
```

```
CREATE TABLE BOOKS
( ID     VARCHAR2(6),
  TITLE  VARCHAR2(255),
  PUBLISHER_ID NUMBER(2),
  AUTHOR_ID NUMBER(3)
 );
```

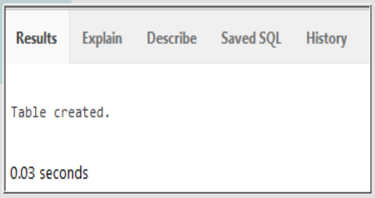
ORACLE
Academy

Case Scenario: Creating Tables

Creating Tables

```
CREATE TABLE authors(
 id    NUMBER(3),
 name  VARCHAR2(60)
);

CREATE TABLE members(
 id             NUMBER(4),
 first_name     VARCHAR2(50),
 last_name      VARCHAR2(50),
 street_address VARCHAR2(50),
 city           VARCHAR2(20),
 state          VARCHAR2(2),
 zip            VARCHAR2(10)
);
```

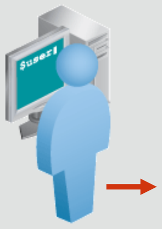
Successful creation of tables



```
Table created.

0.03 seconds
```

ORACLE Academy

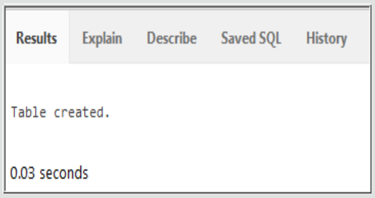
Case Scenario: Creating Tables



Creating Tables

```
CREATE TABLE publishers(
  id   NUMBER(2),
  name VARCHAR2(100) NOT NULL
);

CREATE TABLE books(
 id           VARCHAR2(6),
 title        VARCHAR2(255)NOT NULL,
 publisher_id NUMBER(2),
 author_id    NUMBER(3)
);
```

Successful creation of tables



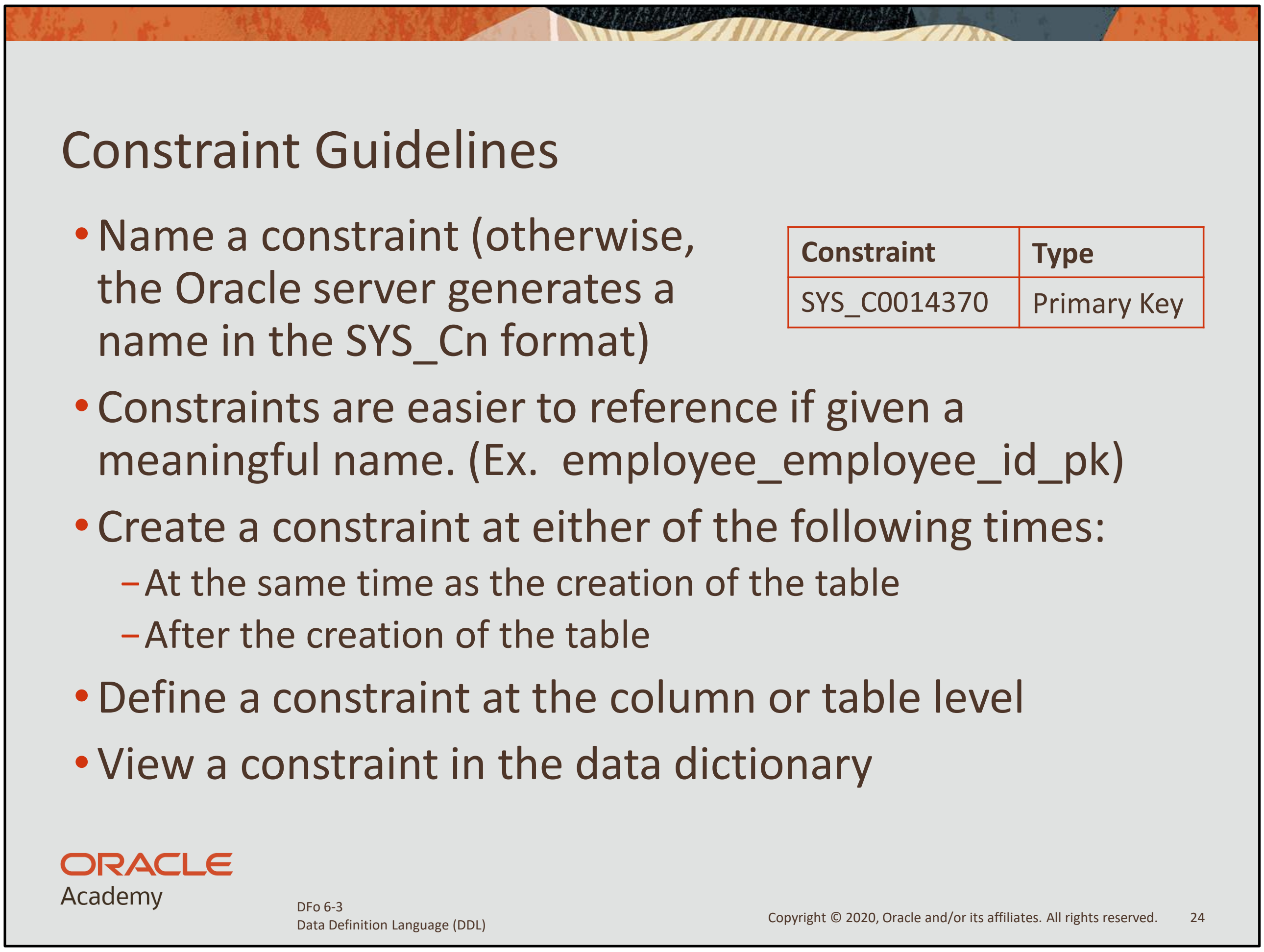
```
Table created.

0.03 seconds
```

Including Constraints

- Constraints enforce rules at the table level
- Constraints ensure the consistency and integrity of the database
- The following constraint types are valid:
  - NOT NULL
  - UNIQUE
  - PRIMARY KEY
  - FOREIGN KEY
  - CHECK

Data Integrity Constraints

Constraints	Description
NOT NULL	The column cannot contain a null value
UNIQUE	The values for a column or a combination of columns must be unique for all rows in the table
PRIMARY KEY	The column (or a combination of columns) must contain the unique AND IS NOT NULL value for all rows
FOREIGN KEY	The column (or a combination of columns) must establish and enforce a reference to a column or a combination of columns in another (or the same) table
CHECK	A condition must be true



Constraint Guidelines

- Name a constraint (otherwise, the Oracle server generates a name in the SYS_Cn format)

Constraint	Type
SYS_C0014370	Primary Key

- Constraints are easier to reference if given a meaningful name. (Ex. employee_employee_id_pk)
- Create a constraint at either of the following times:
  - At the same time as the creation of the table
  - After the creation of the table
- Define a constraint at the column or table level
- View a constraint in the data dictionary

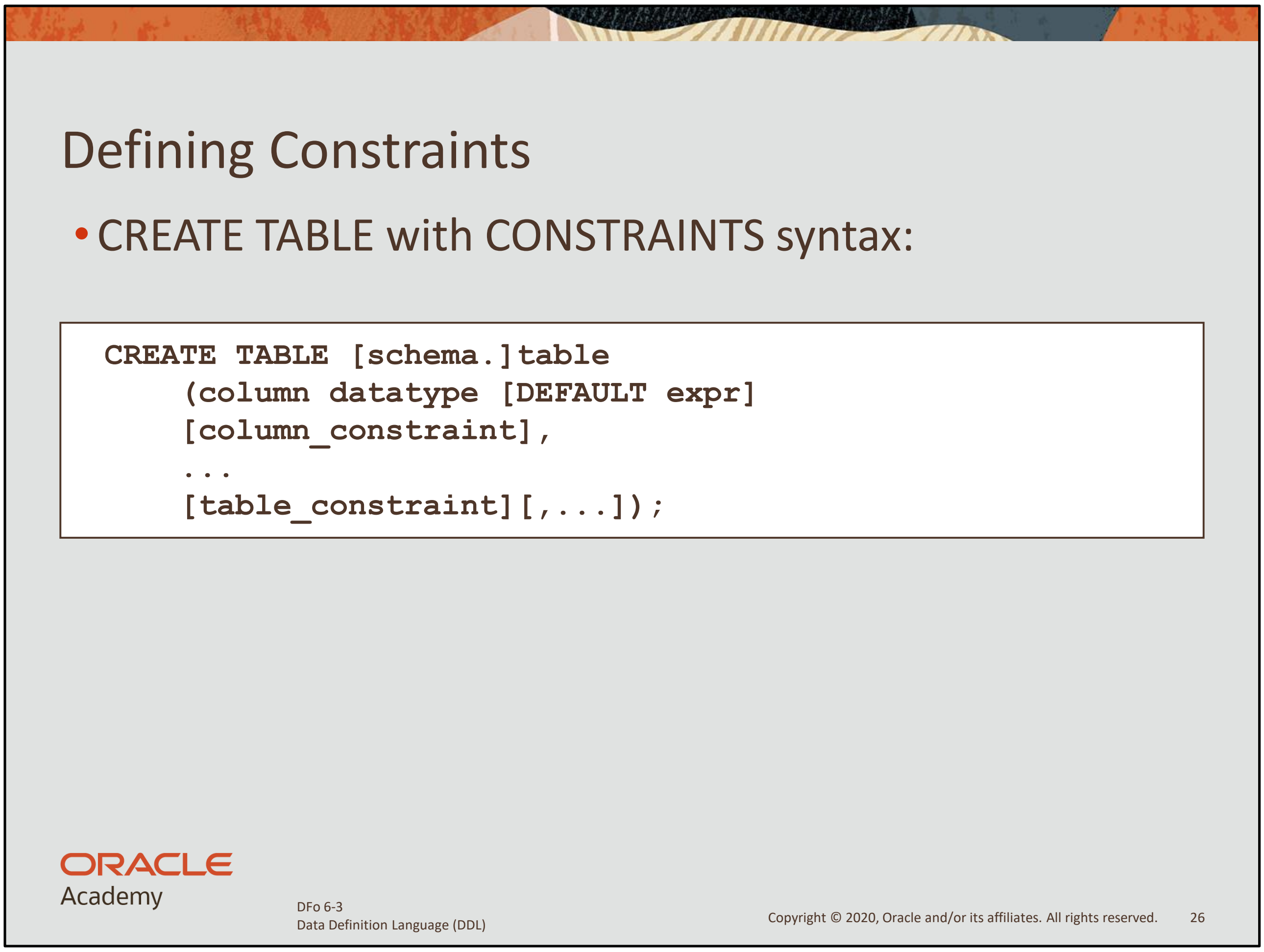
For example, when you create a table if you specify a column to be the primary key without using the "CONSTRAINT“ reserved word, Oracle generates a constraint name, as shown here:

CREATE TABLE DEPT_SAMPLE(DEPT_ID NUMBER(2) PRIMARY KEY, DEPARTMENT_ID VARCHAR2(50));

Constraint Guidelines

- Column-level constraints are included when the column is defined
- Table-level constraints are defined at the end of the table definition, and must refer to the column or columns on which the constraint pertains
- Functionally, a column-level constraint is the same as a table-level constraint
- NOT NULL constraints can be defined only at the column level
- Constraints that apply to more than one column must be defined at the table level

ORACLE
Academy



Defining Constraints

- CREATE TABLE with CONSTRAINTS syntax:

```
CREATE TABLE [schema.]table
    (column datatype [DEFAULT expr]
    [column_constraint],
    ...
    [table_constraint][,...]);
```

ORACLE Academy

DFo 6-3
Data Definition Language (DDL)

26

In the syntax:

- schema is the same as the owner's name.
- table is the name of the table.
- DEFAULT expr specifies a default value to be used if a value is omitted in the INSERT statement.
- column is the name of the column.
- datatype is the column's data type and length.
- column_constraint is an integrity constraint as part of the column definition.
- table_constraint is an integrity constraint as part of the table definition.

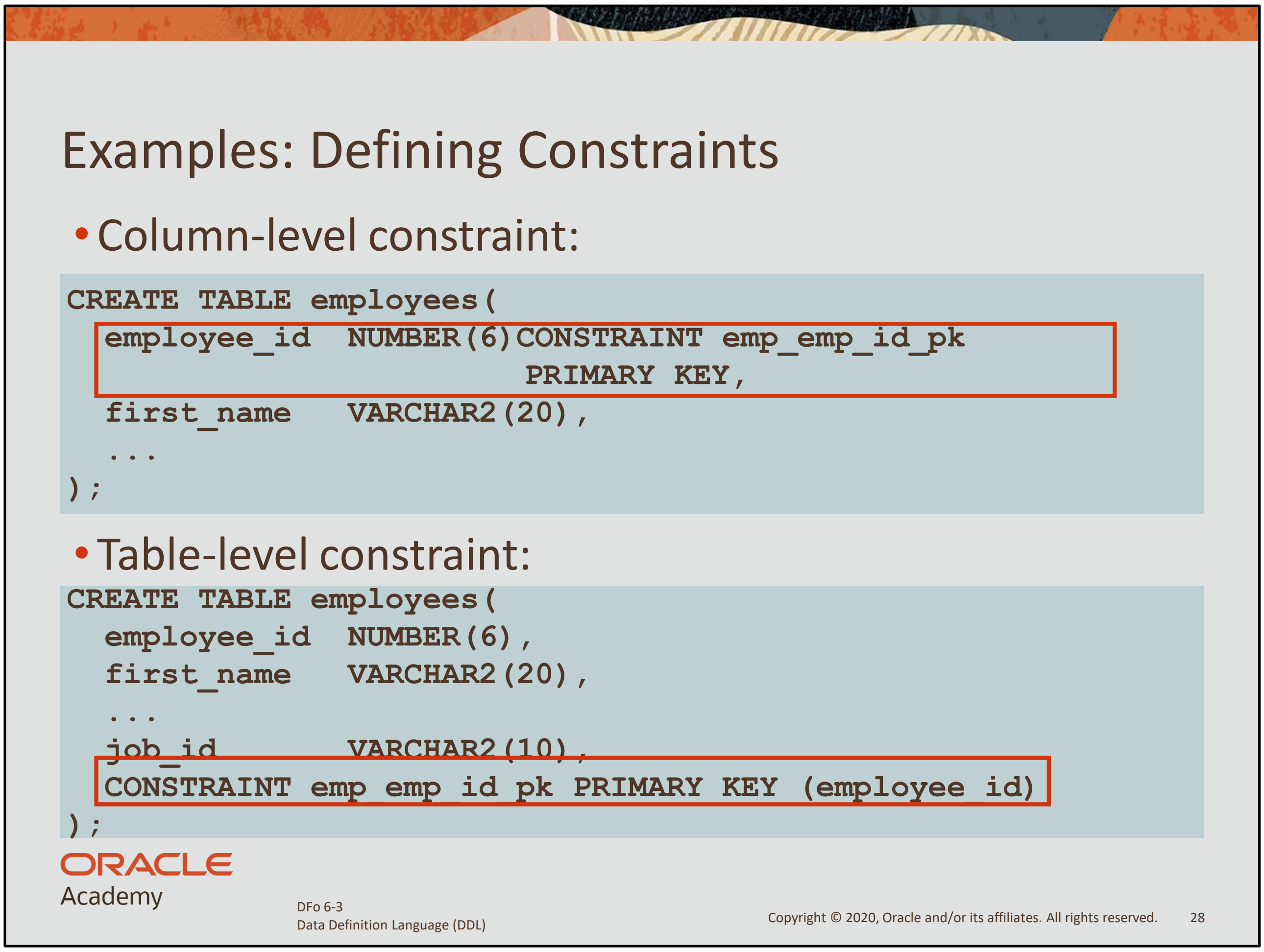
Defining Constraints

- Column-level constraint syntax:

```
column [CONSTRAINT constraint_name] constraint_type,
```

- Table-level constraint syntax:

```
column,...
 [CONSTRAINT constraint_name] constraint_type
 (column, ...),
```



Examples: Defining Constraints

- Column-level constraint:

```
CREATE TABLE employees(
  employee_id  NUMBER(6)CONSTRAINT emp_emp_id_pk
                           PRIMARY KEY,
  first_name   VARCHAR2(20),
  ...
);
```

- Table-level constraint:

```
CREATE TABLE employees(
  employee_id  NUMBER(6),
  first_name   VARCHAR2(20),
  ...
  job_id       VARCHAR2(10),
  CONSTRAINT emp_emp_id_pk PRIMARY KEY (employee_id)
);
```

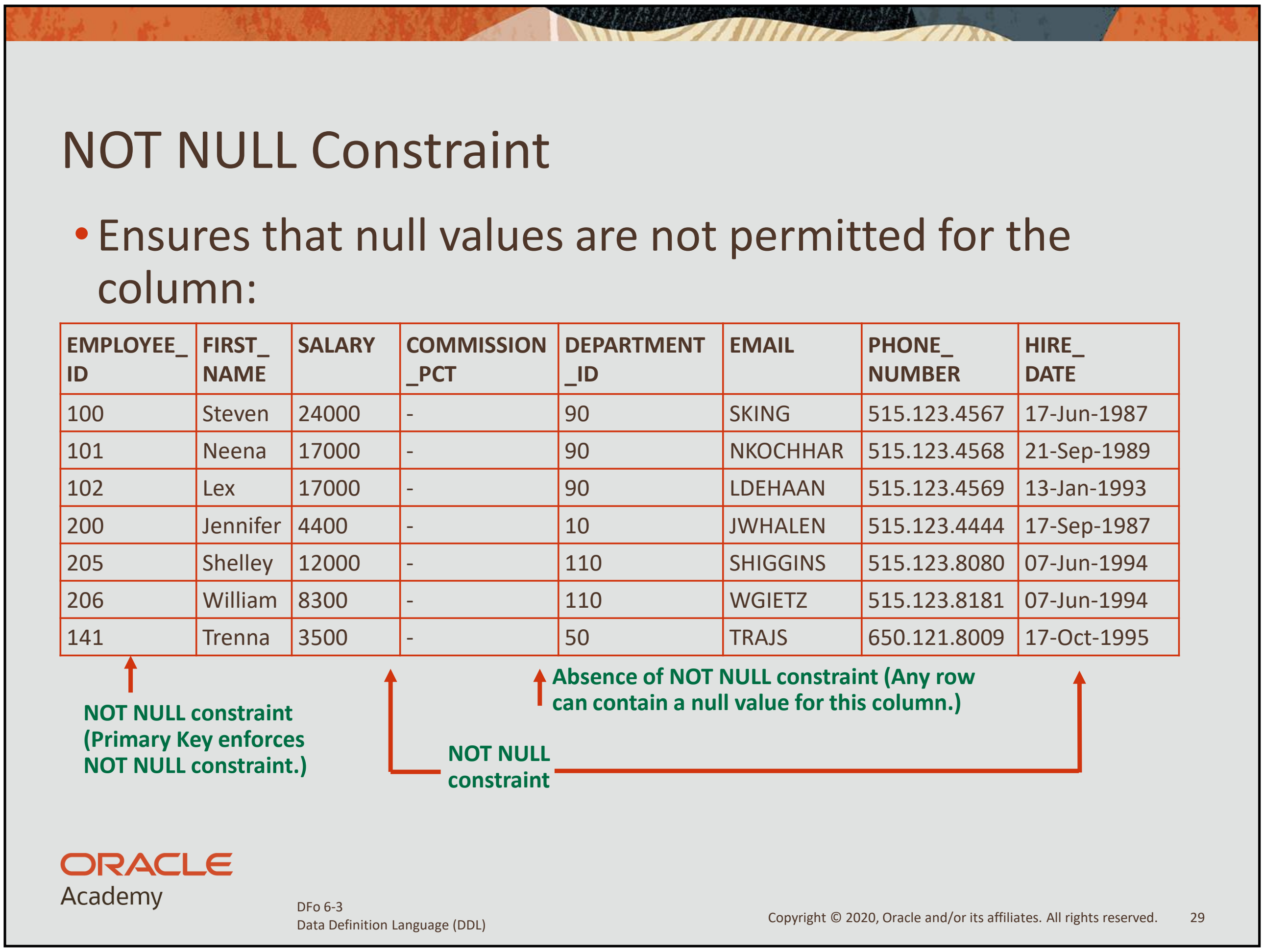
ORACLE
Academy

DFo 6-3
Data Definition Language (DDL)

28

In this example the primary key constraint uses the UID designated for that entity and creates the primary key – this can be created at the column or table level – more details on primary key constraints to follow in slides 33-34.

Note : Examples on this **and following slides** show only a portion of the code used to create the employees table and therefore cannot be run as is.



NOT NULL Constraint

- Ensures that null values are not permitted for the column:

EMPLOYEE_ ID	FIRST_ NAME	SALARY	COMMISSION _PCT	DEPARTMENT _ID	EMAIL	PHONE_ NUMBER	HIRE_ DATE
100	Steven	24000	-	90	SKING	515.123.4567	17-Jun-1987
101	Neena	17000	-	90	NKOCHHAR	515.123.4568	21-Sep-1989
102	Lex	17000	-	90	LDEHAAN	515.123.4569	13-Jan-1993
200	Jennifer	4400	-	10	JWHALEN	515.123.4444	17-Sep-1987
205	Shelley	12000	-	110	SHIGGINS	515.123.8080	07-Jun-1994
206	William	8300	-	110	WGIETZ	515.123.8181	07-Jun-1994
141	Trenna	3500	-	50	TRAJS	650.121.8009	17-Oct-1995

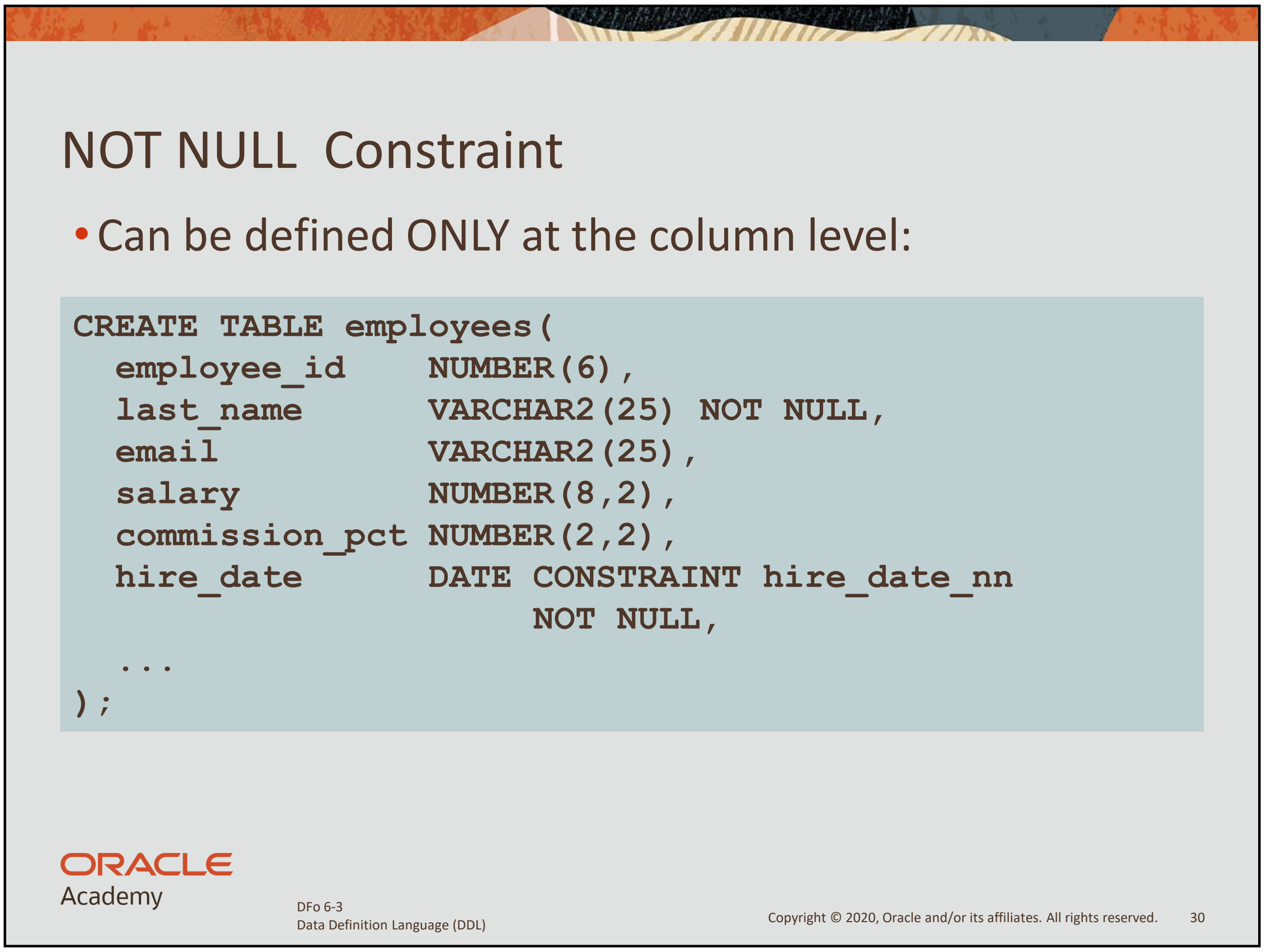
NOT NULL constraint (Primary Key enforces NOT NULL constraint.)

Absence of NOT NULL constraint (Any row can contain a null value for this column.)

NOT NULL constraint

ORACLE Academy

Creating NOT NULL constraints enforce the mandatory attributes in the design.

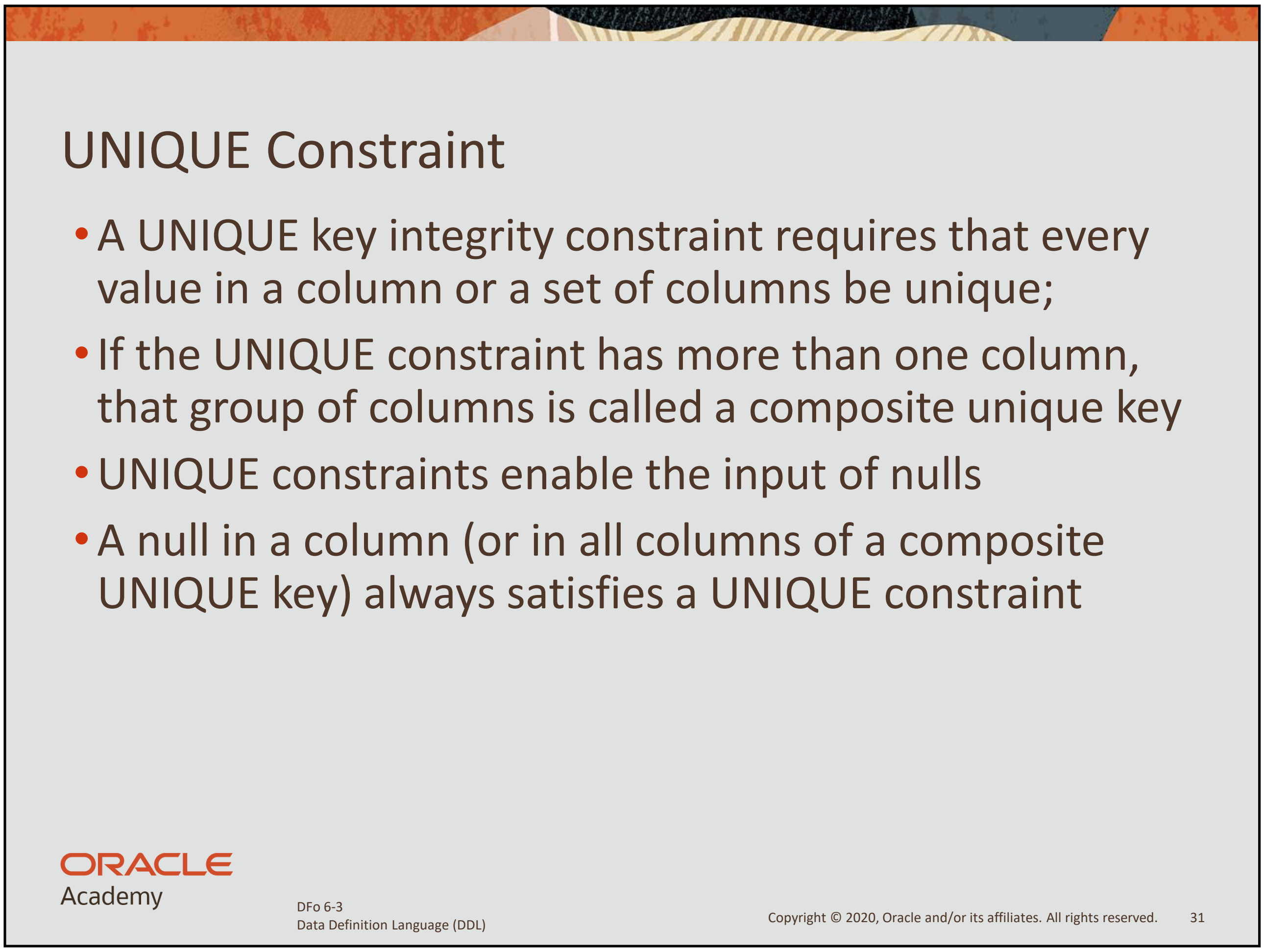


NOT NULL Constraint

- Can be defined ONLY at the column level:

```
CREATE TABLE employees(
  employee_id    NUMBER(6),
  last_name      VARCHAR2(25) NOT NULL,
  email          VARCHAR2(25),
  salary         NUMBER(8,2),
  commission_pct NUMBER(2,2),
  hire_date      DATE CONSTRAINT hire_date_nn
                      NOT NULL,
  ...
);
```

ORACLE
Academy



UNIQUE Constraint

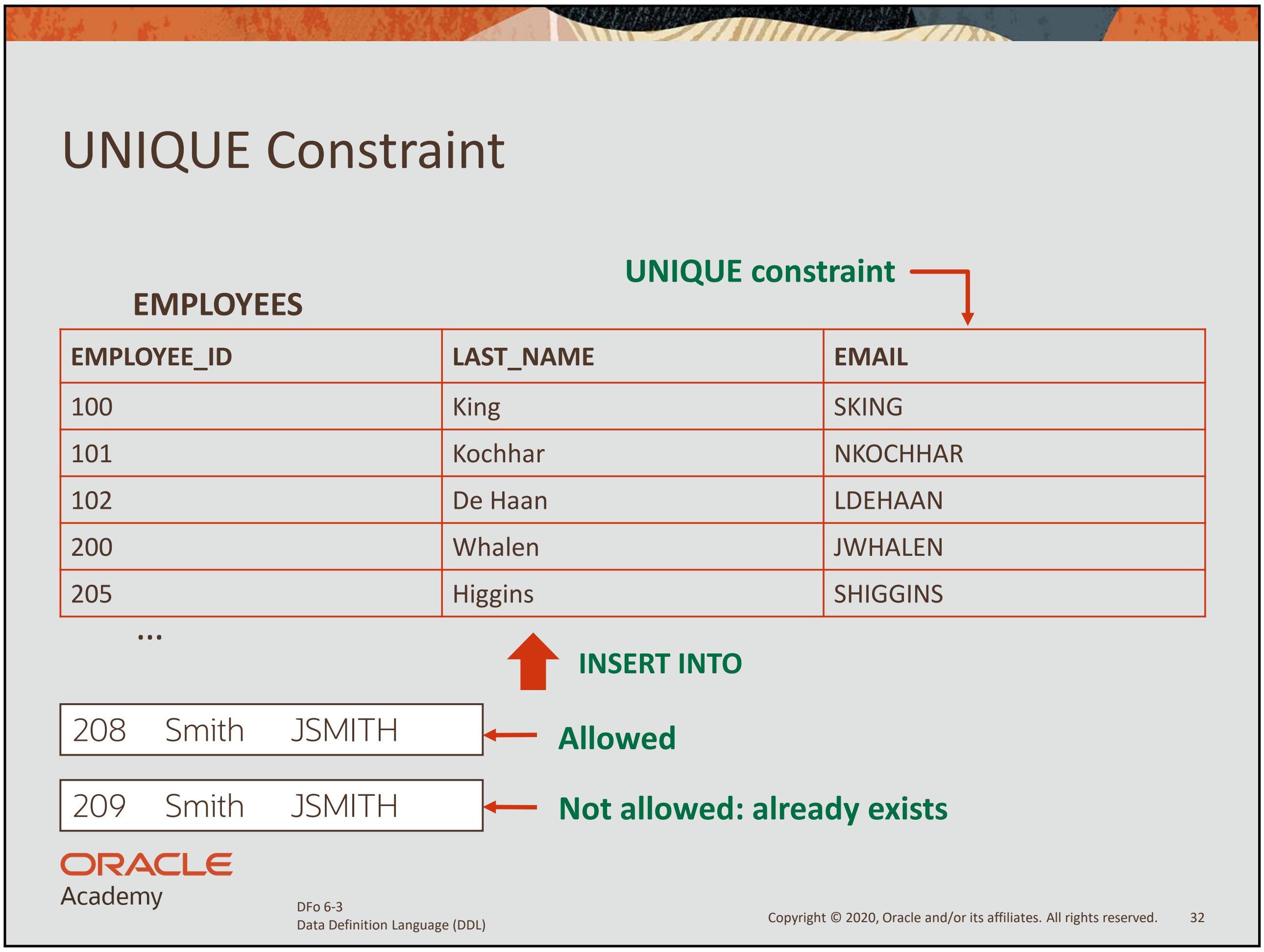
- A UNIQUE key integrity constraint requires that every value in a column or a set of columns be unique;
- If the UNIQUE constraint has more than one column, that group of columns is called a composite unique key
- UNIQUE constraints enable the input of nulls
- A null in a column (or in all columns of a composite UNIQUE key) always satisfies a UNIQUE constraint

ORACLE Academy

DFo 6-3
Data Definition Language (DDL)

31

Note: Because of the search mechanism for the UNIQUE constraints on more than one column, you cannot have identical values in the non-null columns of a partially null composite UNIQUE key constraint.



UNIQUE Constraint

UNIQUE constraint

EMPLOYEES

EMPLOYEE_ID	LAST_NAME	EMAIL
100	King	SKING
101	Kochhar	NKOCHHAR
102	De Haan	LDEHAAN
200	Whalen	JWHALEN
205	Higgins	SHIGGINS

…

INSERT INTO

208	Smith	JSMITH

Allowed

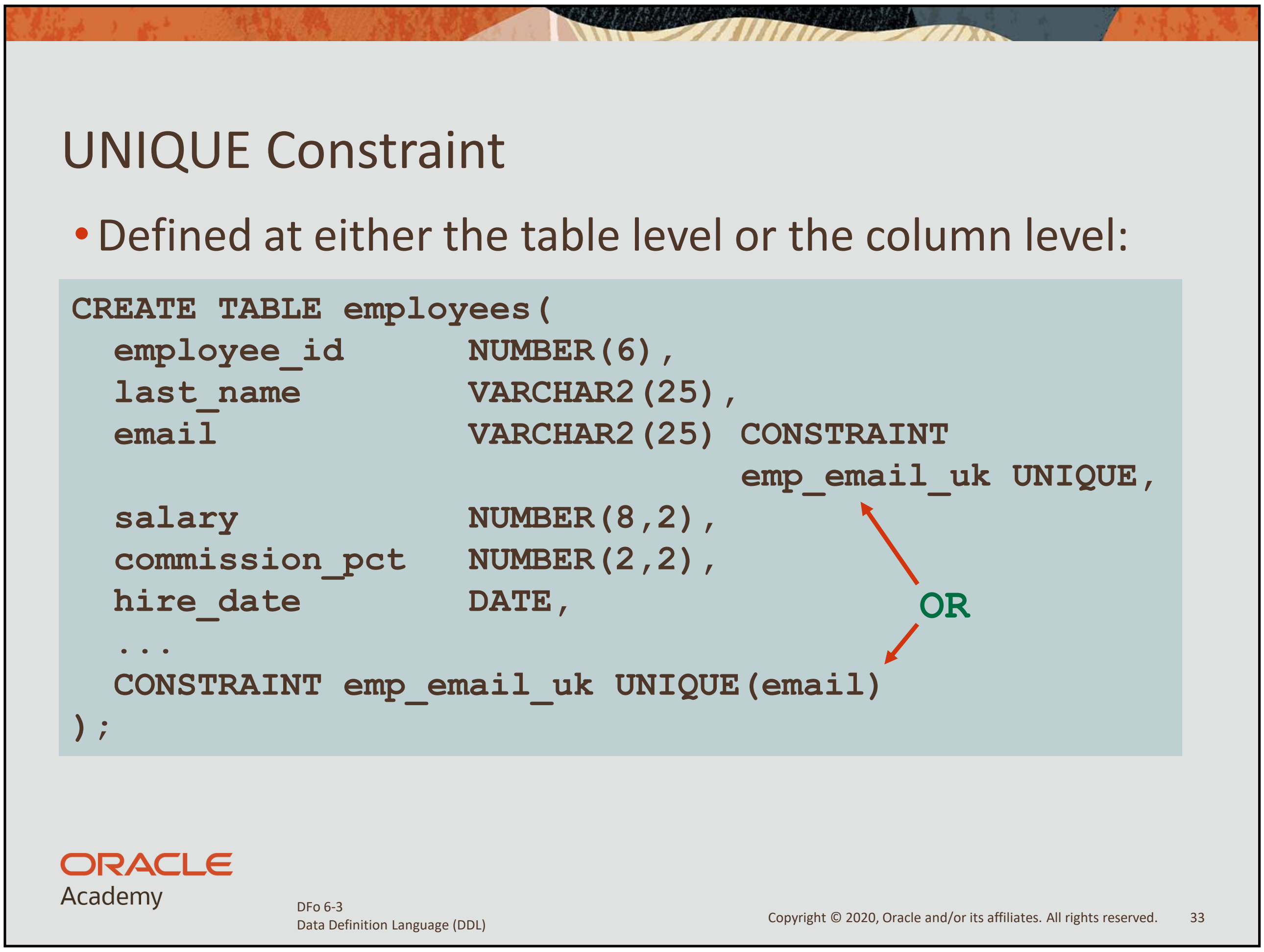
209	Smith	JSMITH

Not allowed: already exists

ORACLE Academy

DFo 6-3
Data Definition Language (DDL)

32



UNIQUE Constraint

- Defined at either the table level or the column level:

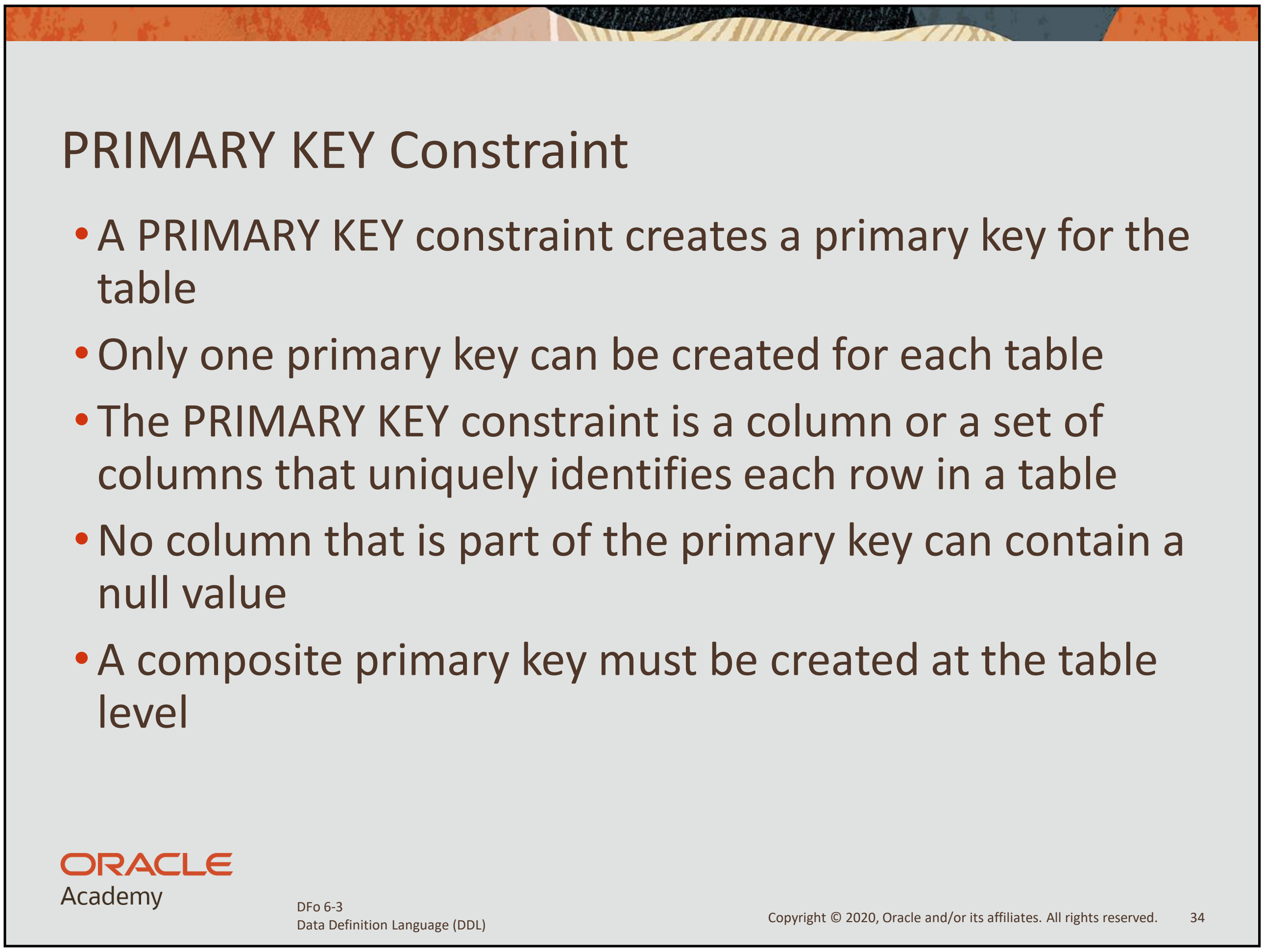
```
CREATE TABLE employees(
  employee_id      NUMBER(6),
  last_name        VARCHAR2(25),
  email            VARCHAR2(25) CONSTRAINT
                                emp_email_uk UNIQUE,
  salary           NUMBER(8,2),
  commission_pct   NUMBER(2,2),
  hire_date        DATE,                        OR
  ...
  CONSTRAINT emp_email_uk UNIQUE(email)
);
```

ORACLE
Academy

A composite unique key is defined at the table level. For example:

ALTER TABLE DEPT_SAMPLE ADD CONSTRAINT unq_dept_det UNIQUE (DEPT_ID, DEPARTMENT_NAME) ;

Note: The Oracle server enforces the UNIQUE constraint by implicitly creating a unique index on the unique key column or columns.



PRIMARY KEY Constraint

- A PRIMARY KEY constraint creates a primary key for the table
- Only one primary key can be created for each table
- The PRIMARY KEY constraint is a column or a set of columns that uniquely identifies each row in a table
- No column that is part of the primary key can contain a null value
- A composite primary key must be created at the table level

ORACLE Academy

DFo 6-3
Data Definition Language (DDL)

34

For example:

```
create table dept(
dept_id number(8),
dept_name varchar2(30),
loc_id number(4),
constraint pk_dept primary key(dept_id,loc_id));
```

Note: Because uniqueness is part of the primary key constraint definition, the Oracle server enforces the uniqueness by implicitly creating a unique index on the primary key column or columns.

PRIMARY KEY Constraint

DEPARTMENTS

PRIMARY KEY

DEPARTMENT_ID	**DEPARTMENT_NAME**	**MANAGER_ID**	**LOCATION_ID**
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400

…

Not allowed (null value)

INSERT INTO

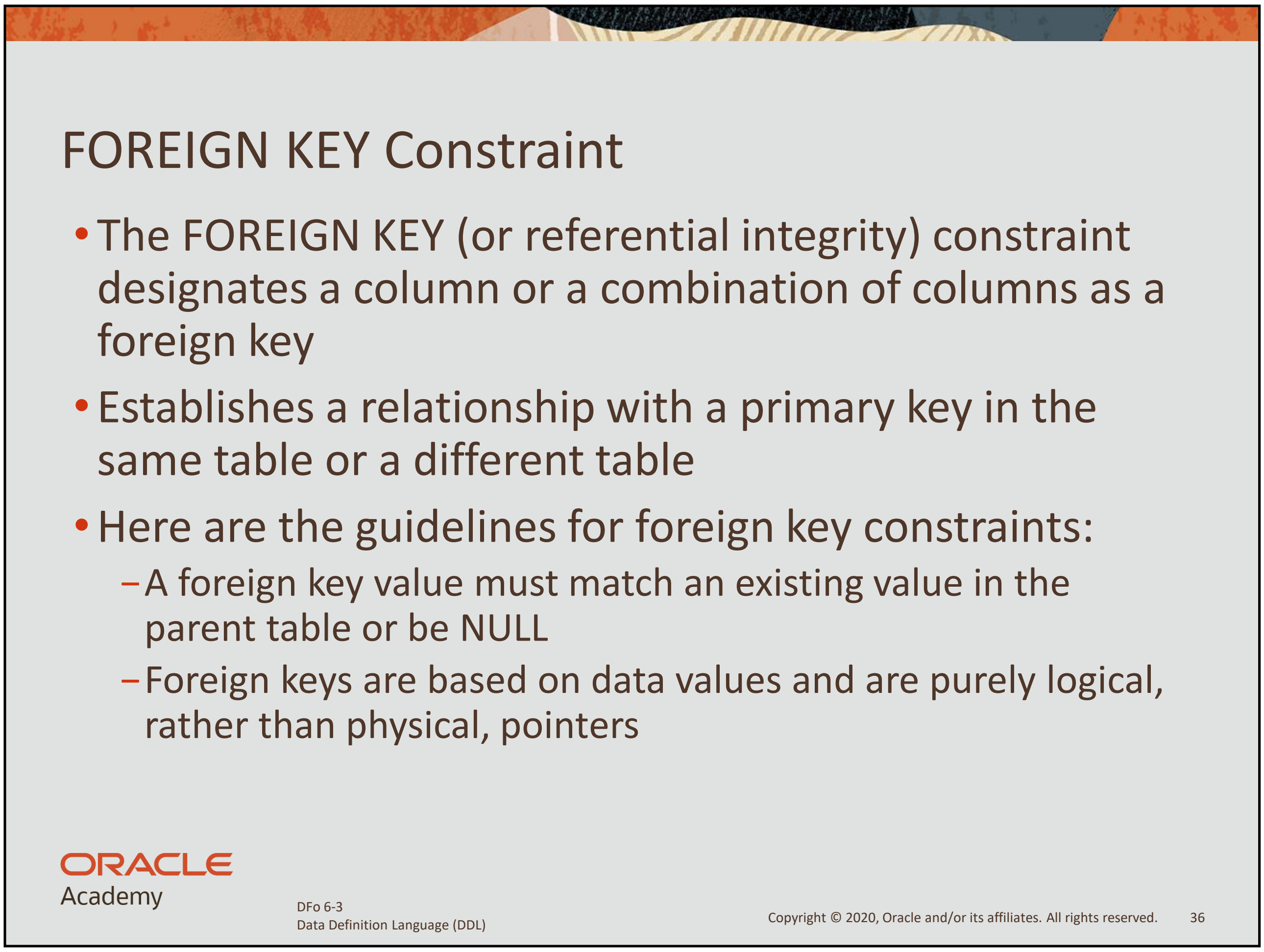
NULL	Public Accounting	124	2500

50	Finance	124	1500

Not allowed (50 already exists)

- Note: See Slide 27 for primary key constraint coding examples

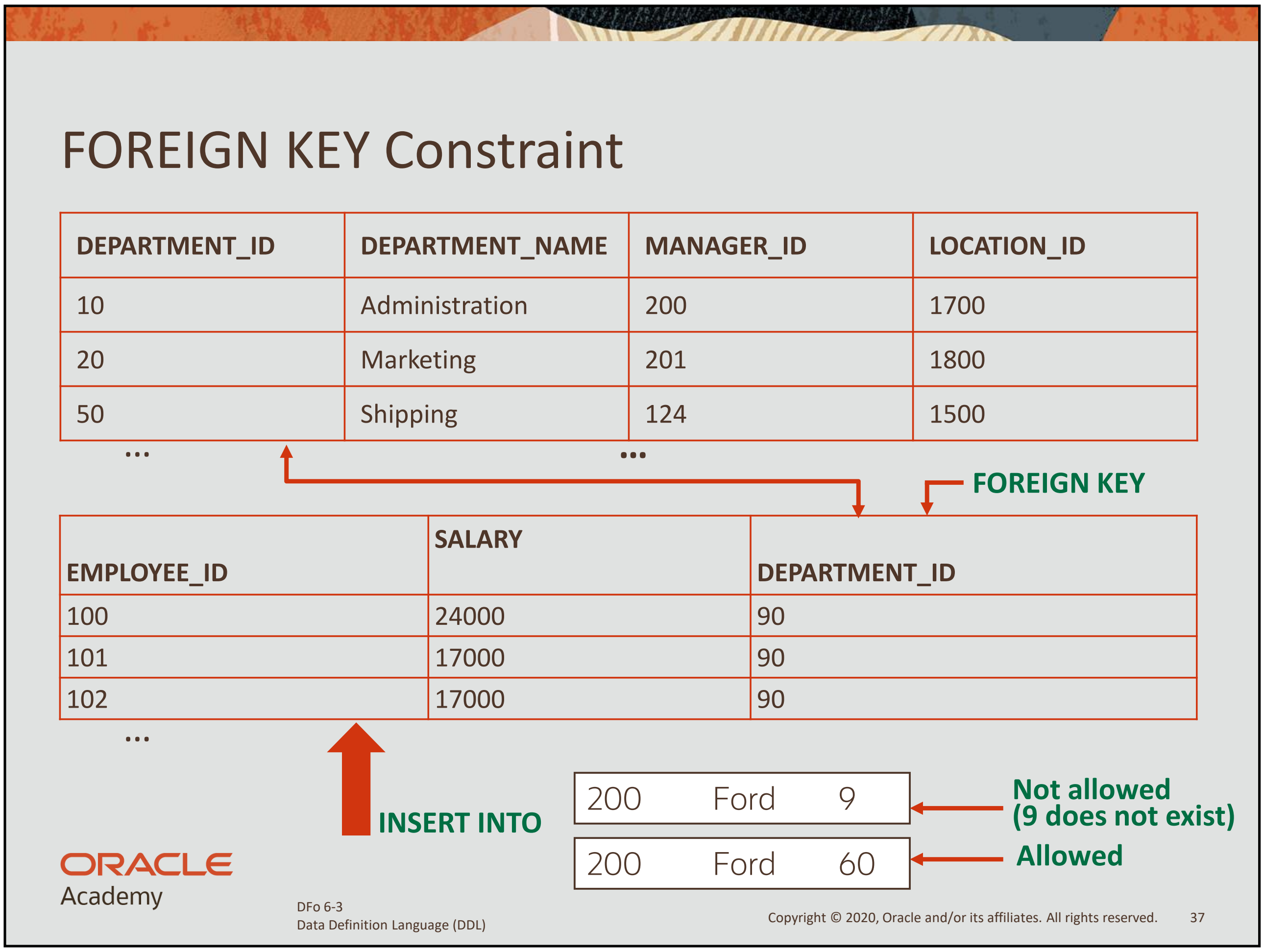
ORACLE Academy



FOREIGN KEY Constraint

- The FOREIGN KEY (or referential integrity) constraint designates a column or a combination of columns as a foreign key
- Establishes a relationship with a primary key in the same table or a different table
- Here are the guidelines for foreign key constraints:
  - A foreign key value must match an existing value in the parent table or be NULL
  - Foreign keys are based on data values and are purely logical, rather than physical, pointers

ORACLE
Academy



FOREIGN KEY Constraint

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500

…   …

FOREIGN KEY

EMPLOYEE_ID	SALARY	DEPARTMENT_ID
100	24000	90
101	17000	90
102	17000	90

…

INSERT INTO

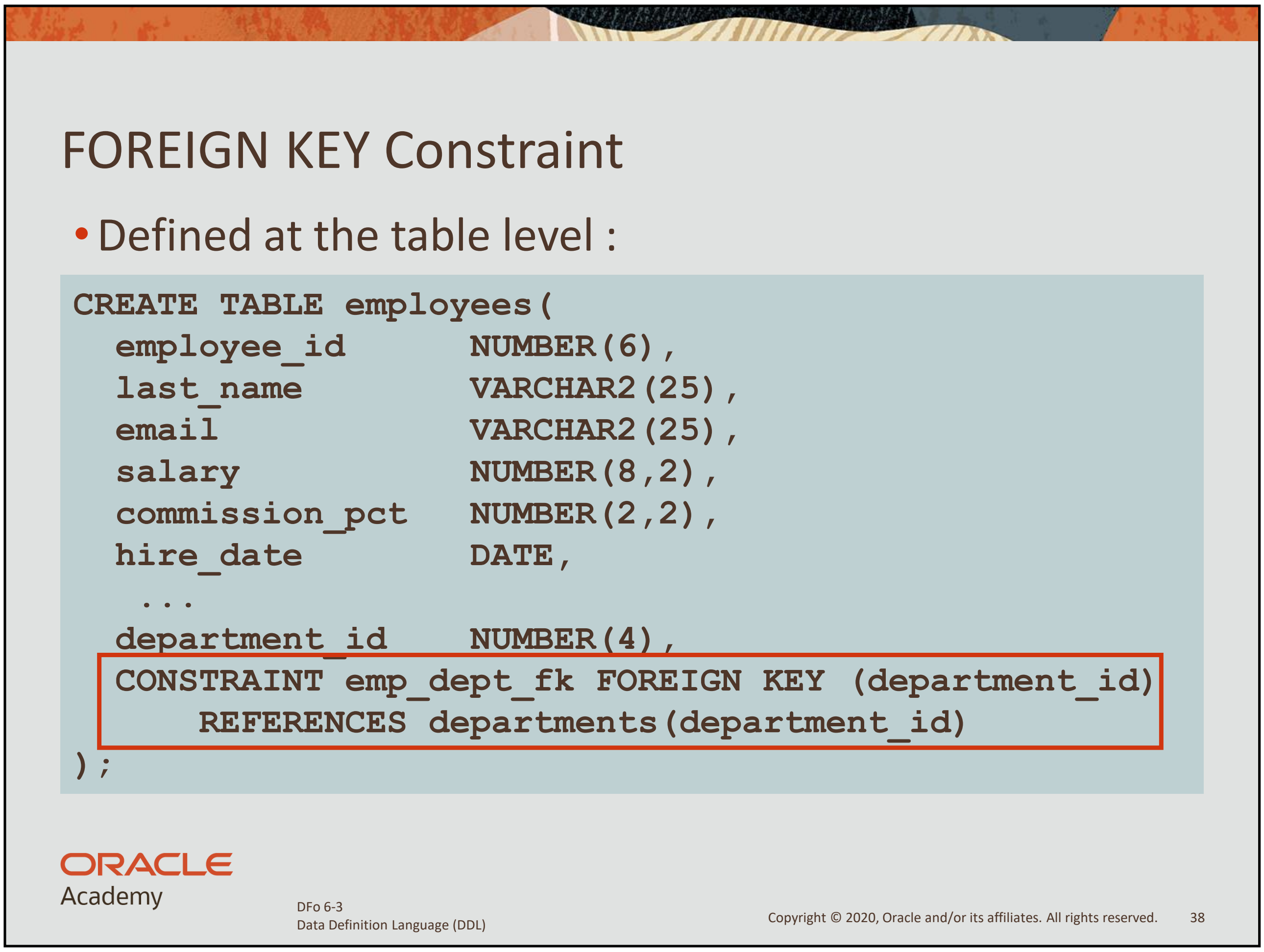
200	Ford	9

Not allowed
(9 does not exist)

200	Ford	60

Allowed

ORACLE
Academy



FOREIGN KEY Constraint

- Defined at the table level :

```
CREATE TABLE employees(
  employee_id         NUMBER(6),
  last_name           VARCHAR2(25),
  email               VARCHAR2(25),
  salary              NUMBER(8,2),
  commission_pct      NUMBER(2,2),
  hire_date           DATE,
   ...
  department_id       NUMBER(4),
  CONSTRAINT emp_dept_fk FOREIGN KEY (department_id)
      REFERENCES departments(department_id)
);
```

ORACLE Academy

DFo 6-3
Data Definition Language (DDL)

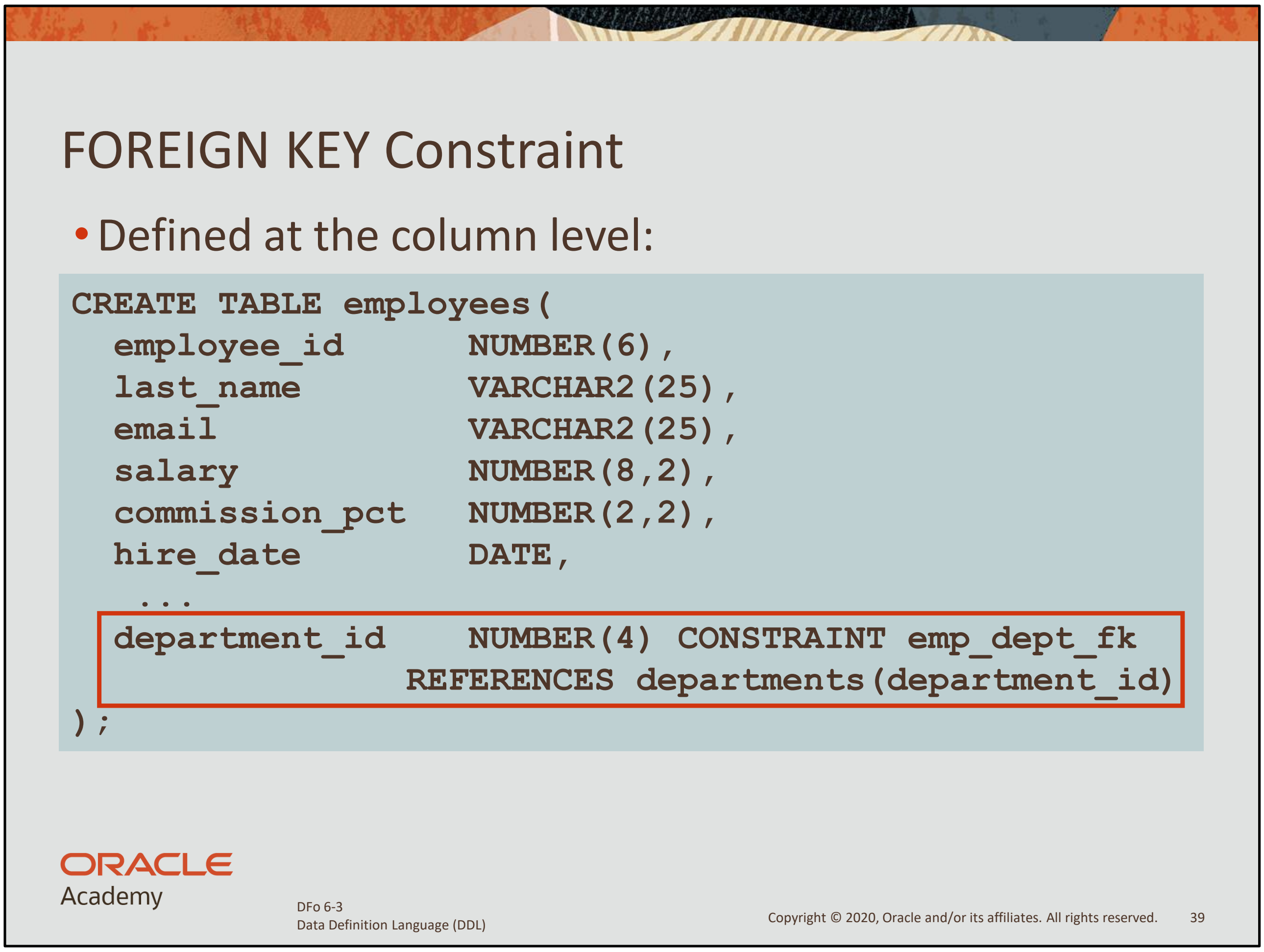
38

A composite foreign key must be created by using the table-level definition.

In the slide, the example defines a FOREIGN KEY constraint on the DEPARTMENT_ID column of the EMPLOYEES table, using table-level syntax. The name of the constraint is EMP_DEPT_FK.

The foreign key can also be defined at the column level, provided that the constraint is based on a single column. The syntax differs in that the FOREIGN KEY keywords do not appear. For example:

```
CREATE TABLE employees
(...
department_id NUMBER(4) CONSTRAINT emp_deptid_fk
REFERENCES departments(department_id),
...
)
```



FOREIGN KEY Constraint

- Defined at the column level:

```
CREATE TABLE employees(
  employee_id         NUMBER(6),
  last_name           VARCHAR2(25),
  email               VARCHAR2(25),
  salary              NUMBER(8,2),
  commission_pct      NUMBER(2,2),
  hire_date           DATE,
   ...
  department_id       NUMBER(4)  CONSTRAINT emp_dept_fk
                   REFERENCES departments(department_id)
);
```

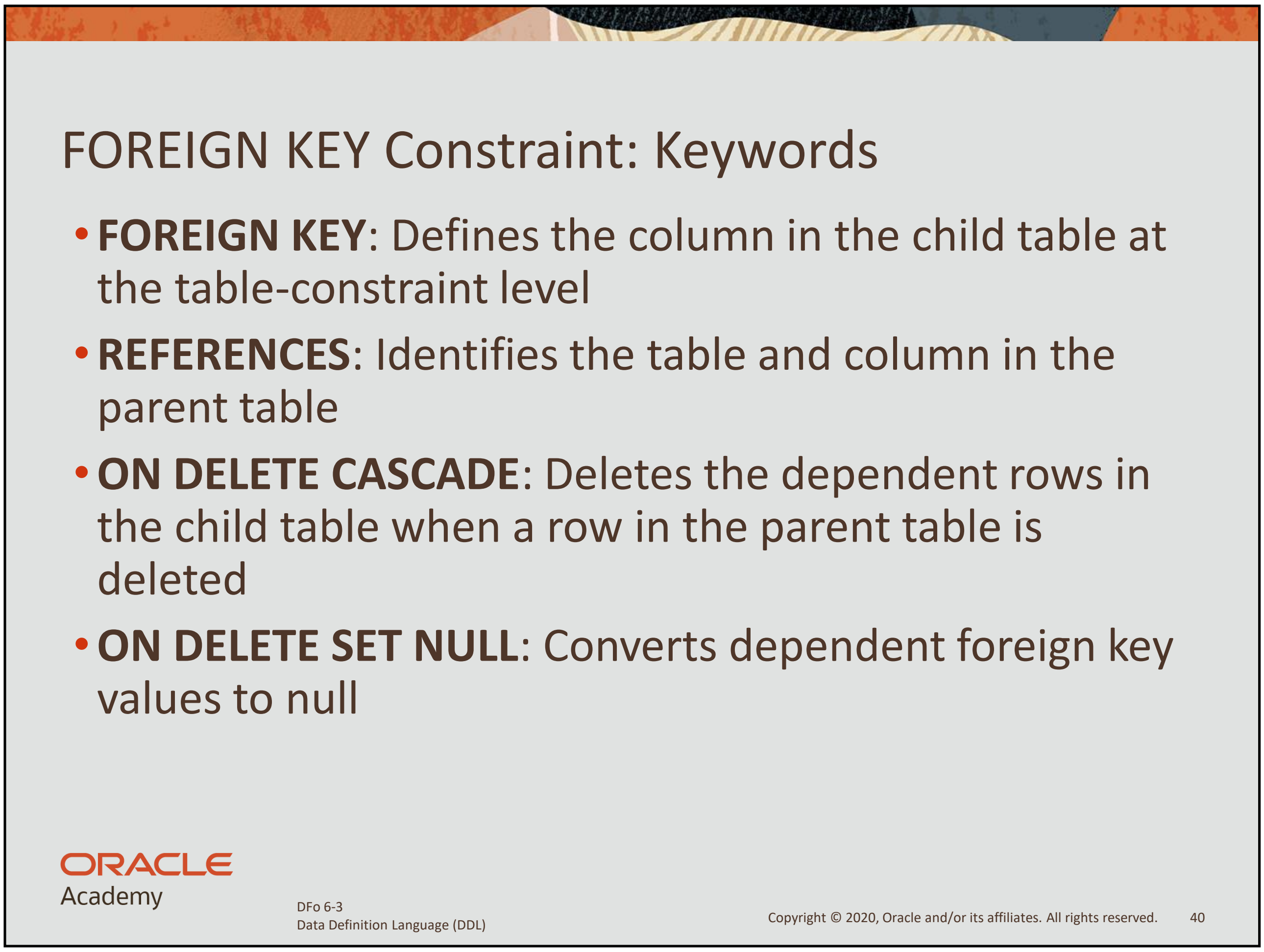
ORACLE Academy

DFo 6-3
Data Definition Language (DDL)

39

A composite foreign key must be created at the table level; for example:

```
CREATE TABLE supplier
( sup_id numeric(15) not null,
  sup_name varchar2(45) not null,
  contact_name varchar2(45),
  CONSTRAINT sup_pk PRIMARY KEY (sup_id, sup_name)
);
```



FOREIGN KEY Constraint: Keywords

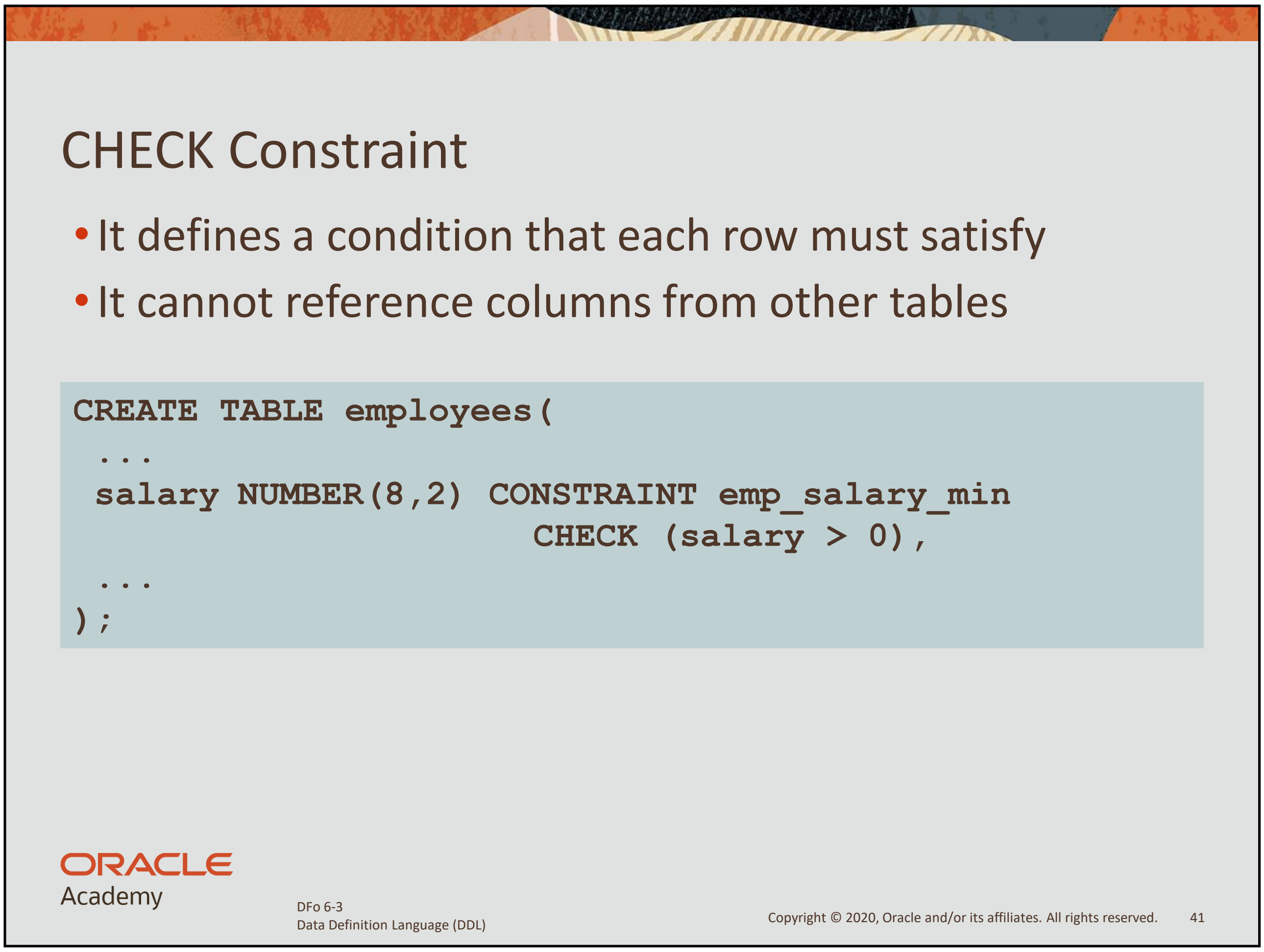
- **FOREIGN KEY**: Defines the column in the child table at the table-constraint level
- **REFERENCES**: Identifies the table and column in the parent table
- **ON DELETE CASCADE**: Deletes the dependent rows in the child table when a row in the parent table is deleted
- **ON DELETE SET NULL**: Converts dependent foreign key values to null

ORACLE
Academy

DFo 6-3
Data Definition Language (DDL)

40

Without the ON DELETE CASCADE or the ON DELETE SET NULL options, the row in the parent table cannot be deleted if it is referenced in the child table. And these keyword cannot be used in column-level syntax.



CHECK Constraint

- It defines a condition that each row must satisfy
- It cannot reference columns from other tables

```
CREATE TABLE employees(
 ...
 salary NUMBER(8,2) CONSTRAINT emp_salary_min
                            CHECK (salary > 0),
 ...
);
```

ORACLE
Academy

To satisfy the constraint, each row in the table must make the condition either TRUE or unknown (due to a null).

A single column can have multiple CHECK constraints that refer to the column in its definition. There is no limit to the number of CHECK constraints that you can define on a column.

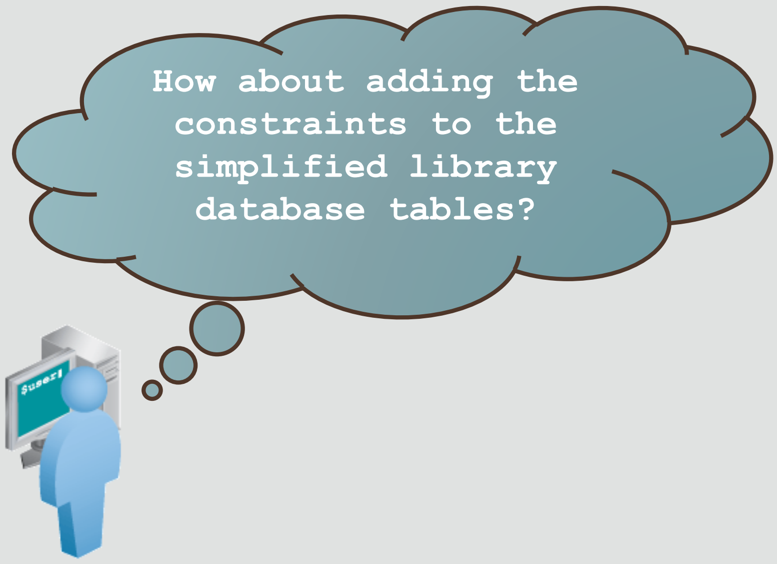
CHECK constraints can be defined at the column level or table level.

CREATE TABLE: CHECK Constraint Example

```
CREATE TABLE employees(
    employee_id       NUMBER(6),
    last_name         VARCHAR2(25),
    email             VARCHAR2(25),
    salary            NUMBER(8,2),
    commission_pct    NUMBER(2,2),
    hire_date         DATE,
    ...
    CONSTRAINT hire_date_min CHECK
                          (hire_date > '01-JAN-2018')
);
```

ORACLE
Academy

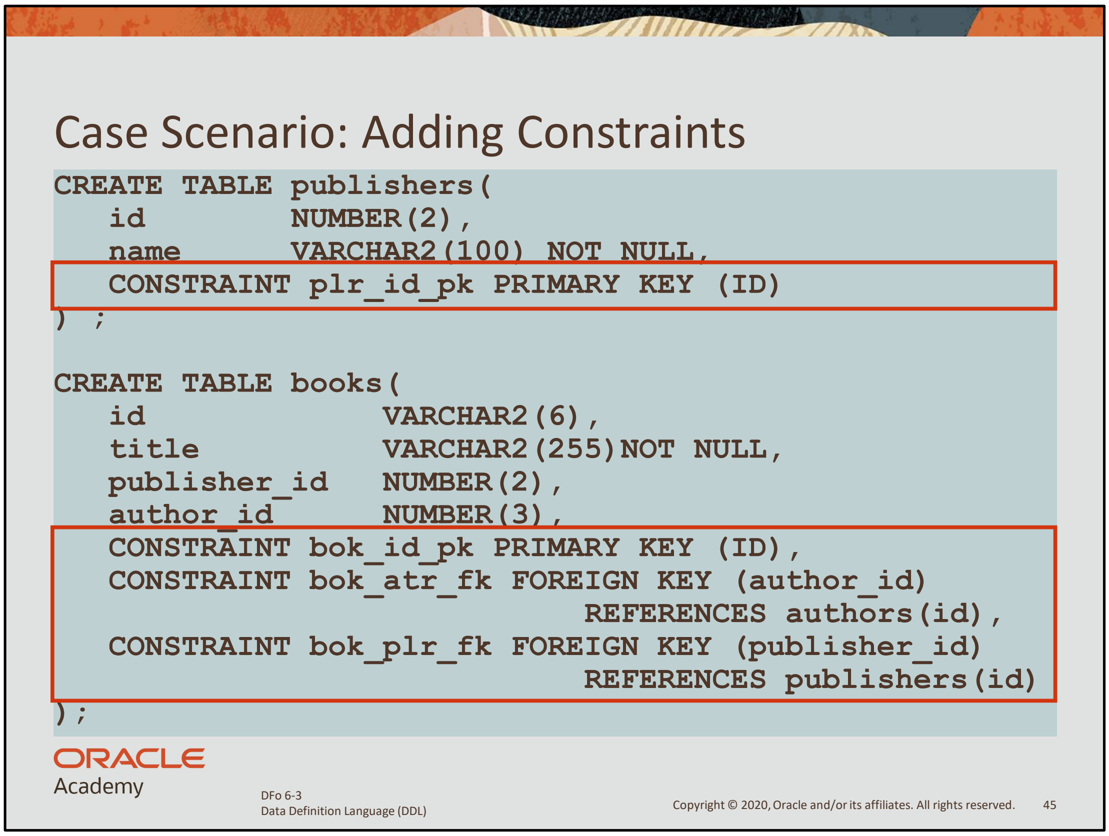
Case Scenario: Creating Tables


How about adding the constraints to the simplified library database tables?
$user1

Case Scenario: Adding Constraints

```
CREATE TABLE authors(
   id         NUMBER(3),
   name       VARCHAR2(60),
   CONSTRAINT  atr_id_pk PRIMARY KEY (ID)
);

CREATE TABLE members(
   id                 NUMBER(4),
   first_name         VARCHAR2(50),
   last_name          VARCHAR2(50),
   street_address     VARCHAR2(50),
   city               VARCHAR2(20),
   state              VARCHAR2(2),
   zip                VARACHAR2(10),
   CONSTRAINT mbr_id_pk PRIMARY KEY (ID)
);
```

ORACLE
Academy



Case Scenario: Adding Constraints

```
CREATE TABLE publishers(
   id         NUMBER(2),
   name       VARCHAR2(100) NOT NULL,
   CONSTRAINT plr_id_pk PRIMARY KEY (ID)
) ;

CREATE TABLE books(
   id              VARCHAR2(6),
   title           VARCHAR2(255)NOT NULL,
   publisher_id    NUMBER(2),
   author_id       NUMBER(3),
   CONSTRAINT bok_id_pk PRIMARY KEY (ID),
   CONSTRAINT bok_atr_fk FOREIGN KEY (author_id)
                            REFERENCES authors(id),
   CONSTRAINT bok_plr_fk FOREIGN KEY (publisher_id)
                            REFERENCES publishers(id)
);
```

ORACLE
Academy

Case Scenario: Adding Constraints

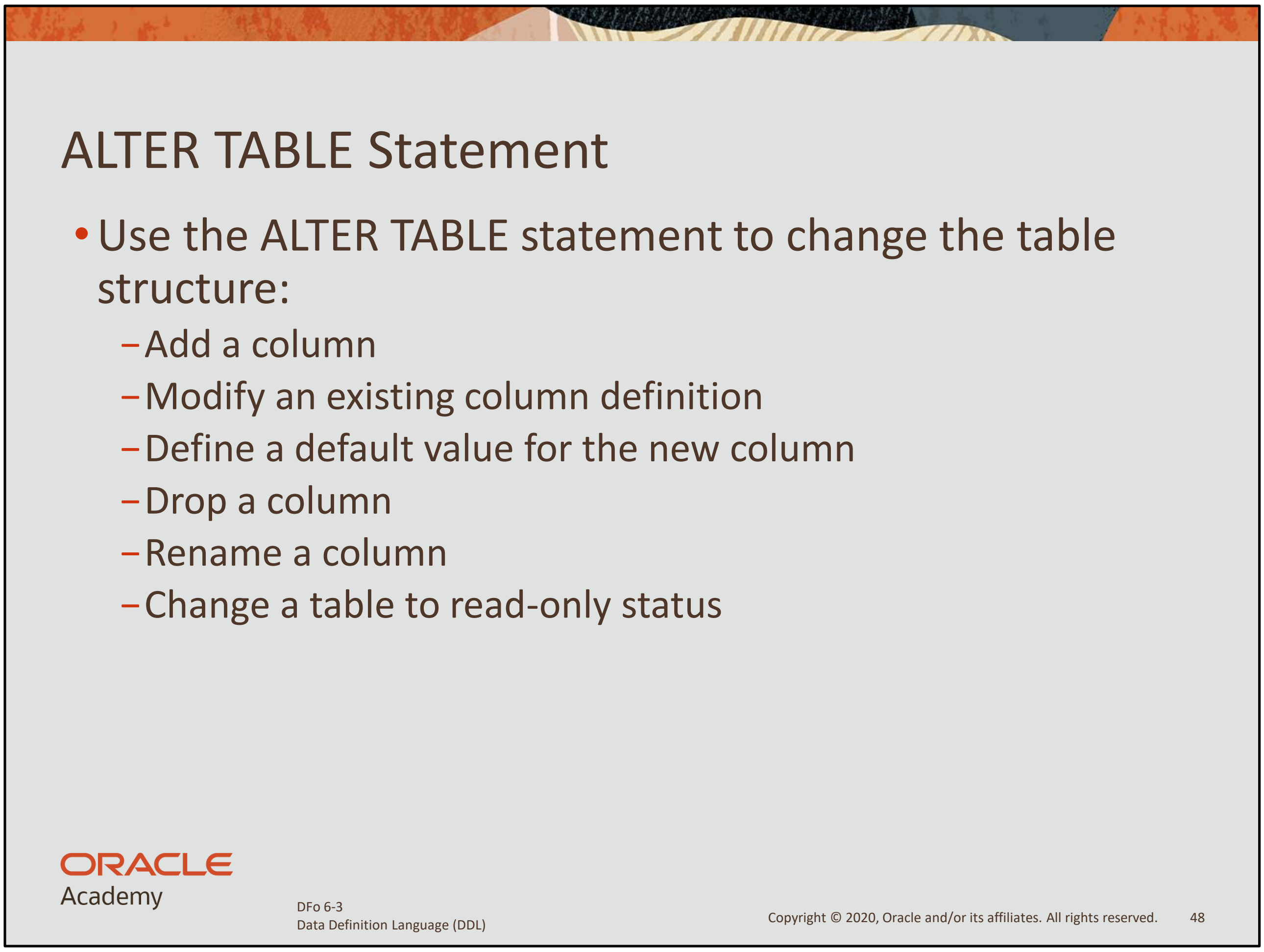
```
CREATE TABLE book_transactions(
   id                VARCHAR2(6),
   tran_date         DATE DEFAULT SYSDATE NOT NULL,
   type              VARCHAR2(10) ,
   book_id           VARCHAR2(6) ,
   member_id         NUMBER(4),
   CONSTRAINT btn_id_pk PRIMARY KEY (ID),
   CONSTRAINT bok_btn_fk FOREIGN KEY (book_id)
                               REFERENCES books(id),
   CONSTRAINT bok_mbr_fk FOREIGN KEY (member_id)
                               REFERENCES members(id)
);
```

ORACLE
Academy

Data Definition Language

- Creating tables is part of SQL's Data Definition Language
- Other **DDL** statements include :
  - **ALTER** : to modify an object's structure
  - **DROP** : to remove an object from the database
  - **RENAME** : to rename a database object

ORACLE
Academy



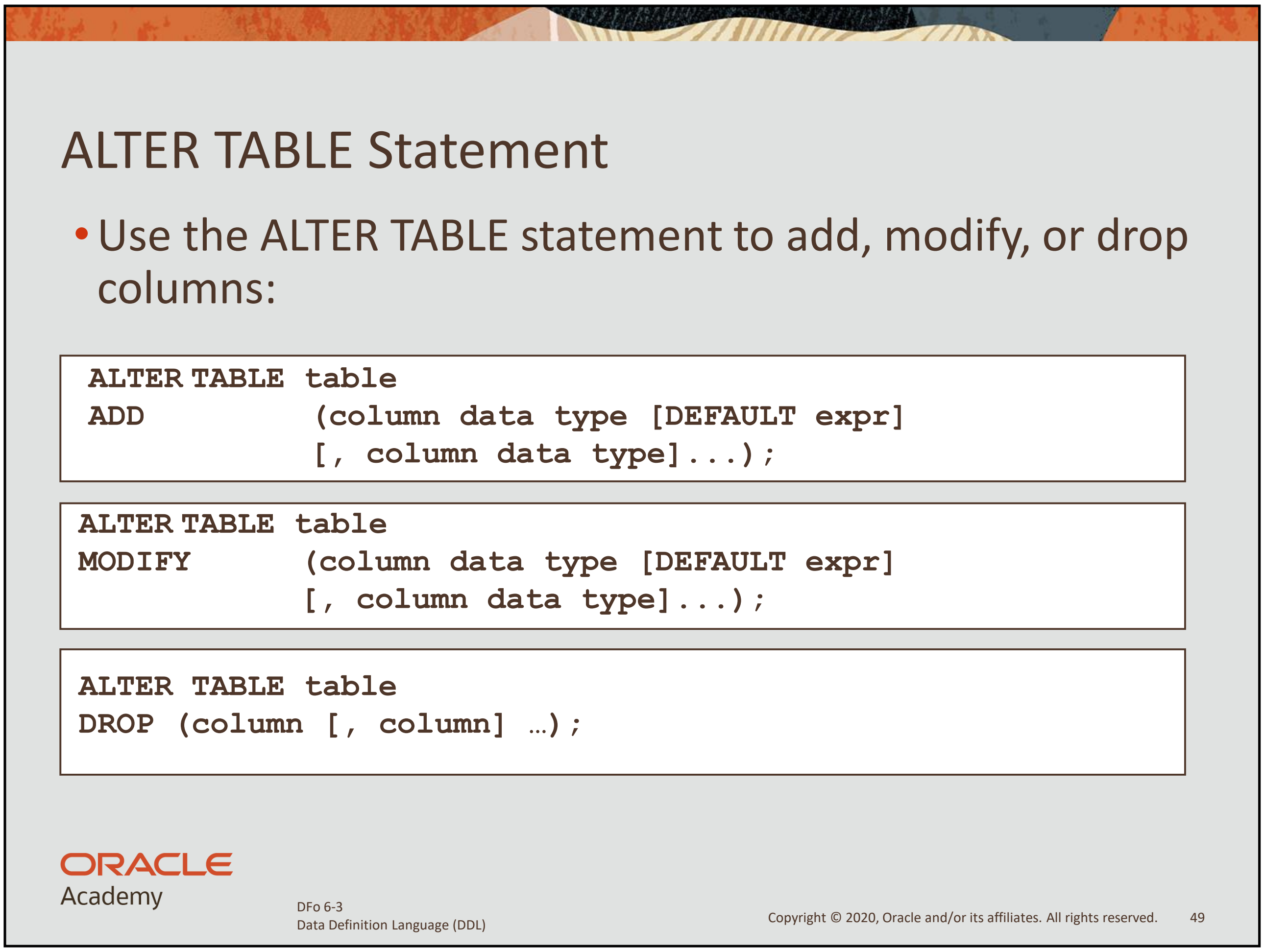
ALTER TABLE Statement

- Use the ALTER TABLE statement to change the table structure:
  - Add a column
  - Modify an existing column definition
  - Define a default value for the new column
  - Drop a column
  - Rename a column
  - Change a table to read-only status

ORACLE Academy

After you create a table, you may need to change the table structure for any of the following reasons:

- You omitted a column.
- Your column definition or its name needs to be changed.
- You need to remove columns.
- You want to put the table in read-only mode



ALTER TABLE Statement

- Use the ALTER TABLE statement to add, modify, or drop columns:

```
ALTER TABLE table
ADD         (column data type [DEFAULT expr]
            [, column data type]...);
```

```
ALTER TABLE table
MODIFY      (column data type [DEFAULT expr]
            [, column data type]...);
```

```
ALTER TABLE table
DROP (column [, column] …);
```

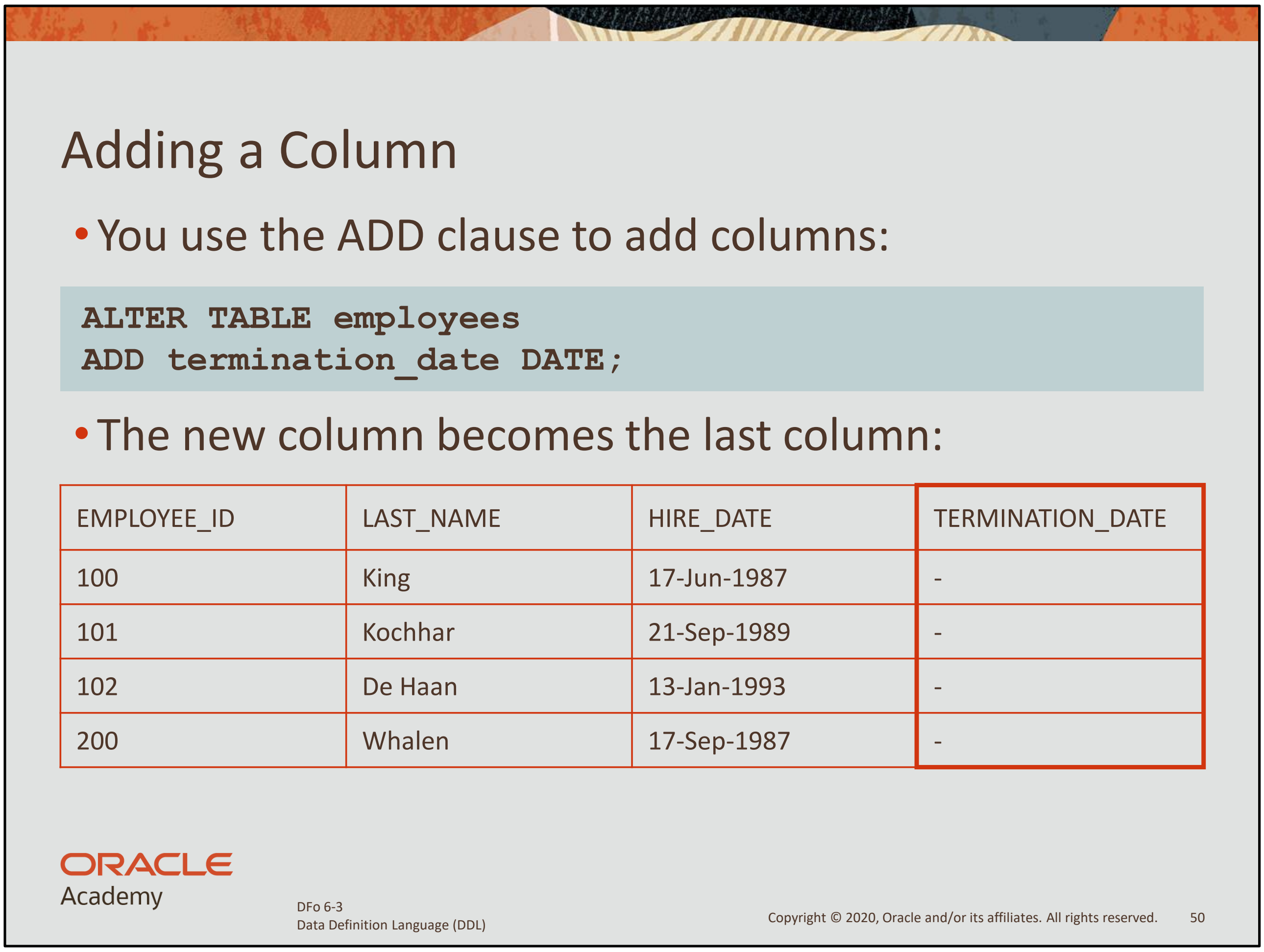
ORACLE
Academy

DFo 6-3
Data Definition Language (DDL)

49

In the syntax:

- *table* is the name of the table.
- ADD|MODIFY|DROP is the type of modification.
- *column* is the name of the column.
- *data type* is the data type and length of the column.
- DEFAULT *expr* specifies the default value for a column.



Adding a Column

- You use the ADD clause to add columns:

```
ALTER TABLE employees
ADD termination_date DATE;
```

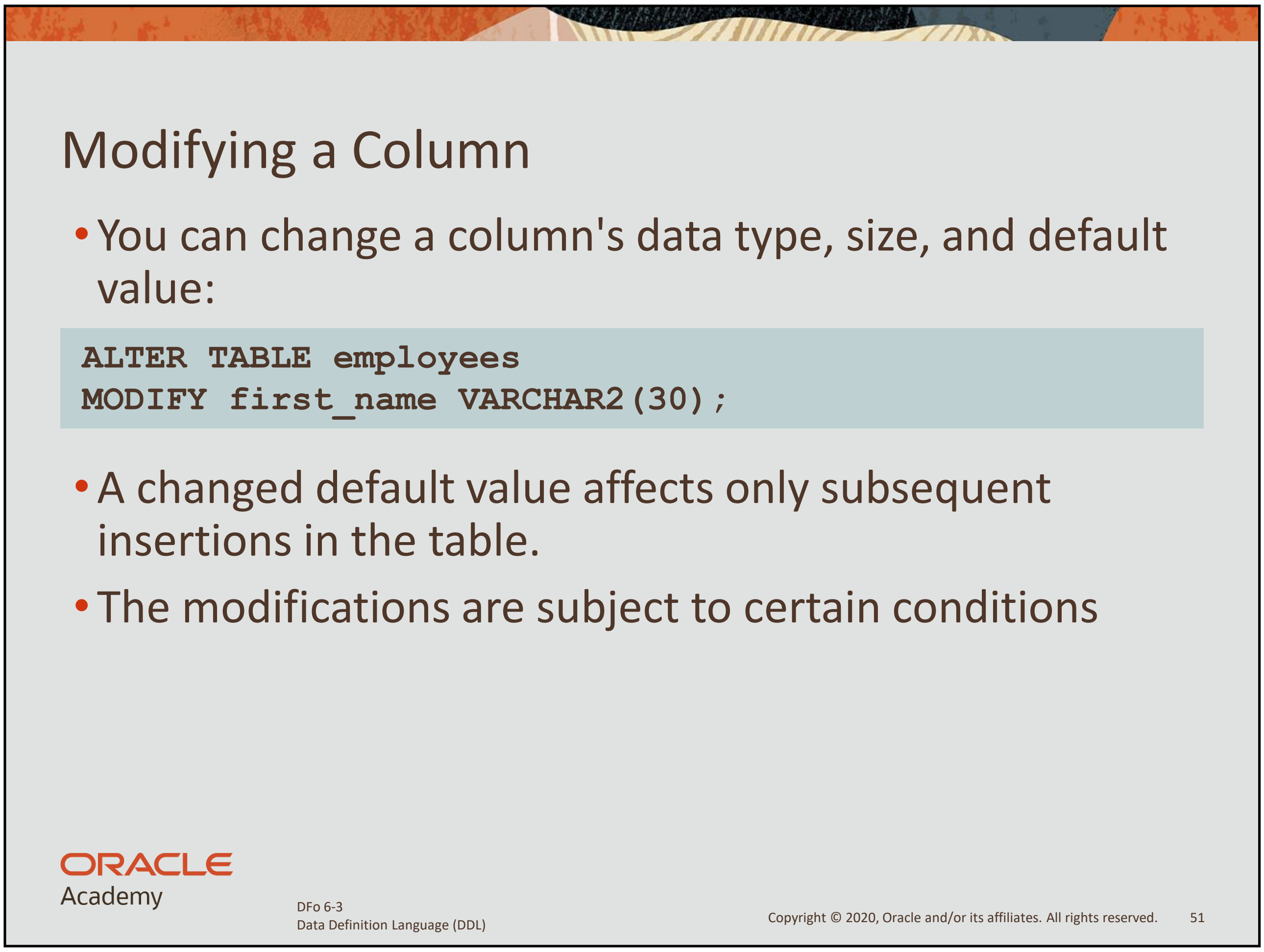
- The new column becomes the last column:

EMPLOYEE_ID	LAST_NAME	HIRE_DATE	TERMINATION_DATE
100	King	17-Jun-1987	-
101	Kochhar	21-Sep-1989	-
102	De Haan	13-Jan-1993	-
200	Whalen	17-Sep-1987	-

ORACLE Academy

Note: If a table already contains rows when a column is added, the new column is initially null or takes the default value for all rows. You can add a mandatory NOT NULL column to a table that contains data in the other columns only if you specify a default value. You can add a NOT NULL column to an empty table without the default value.

See slide 9 for code to create this table.



Modifying a Column

- You can change a column's data type, size, and default value:

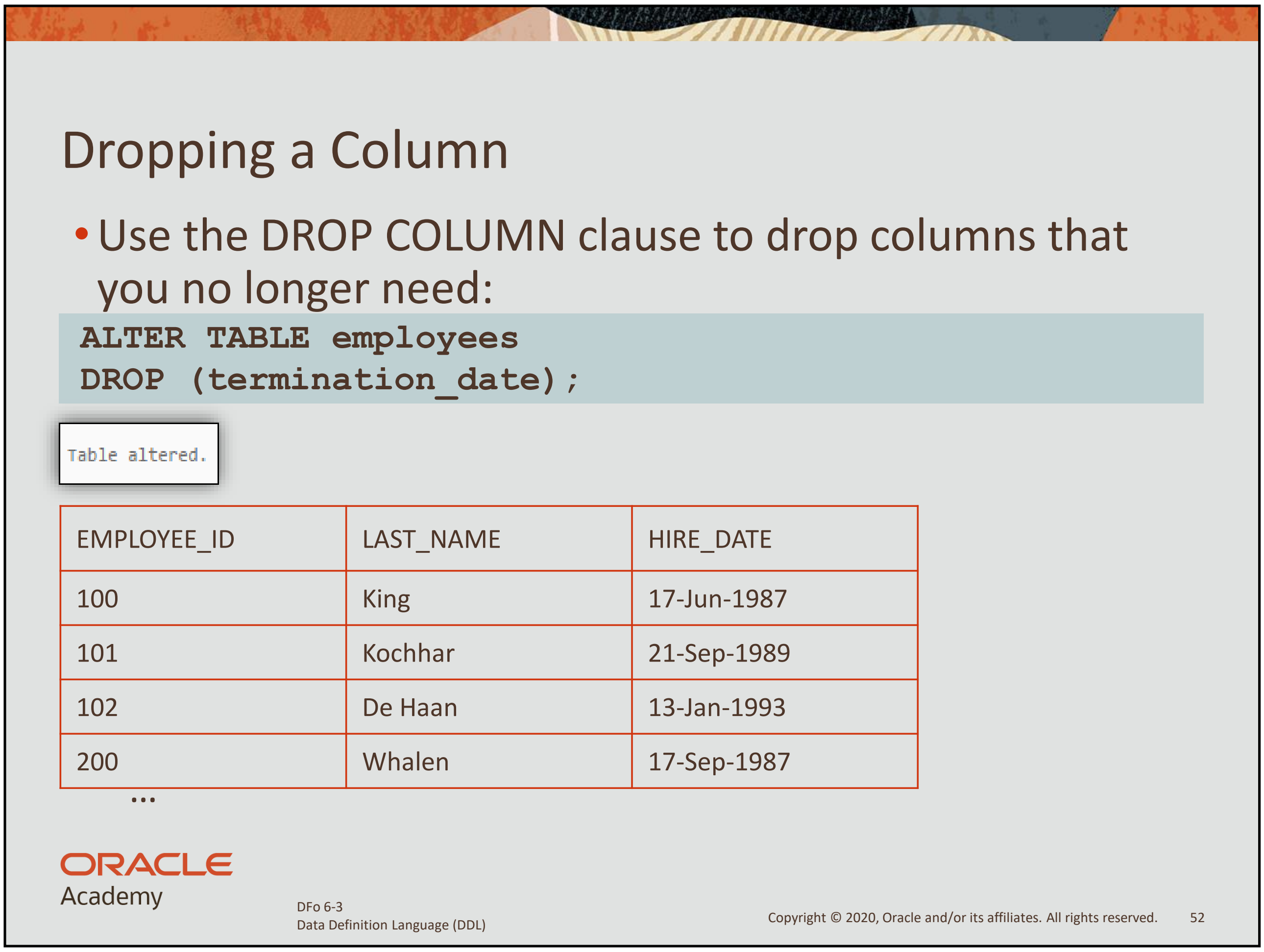
```
ALTER TABLE employees
MODIFY first_name VARCHAR2(30);
```

- A changed default value affects only subsequent insertions in the table.
- The modifications are subject to certain conditions

ORACLE
Academy

Here are the guidelines for modifying a column:

- You can increase the width or precision of a numeric column.
- You can increase the width of character columns.
- You can decrease the width of a column if:
  - The column contains only null values .
  - The table has no rows.
  - The decrease in column width is not less than the existing values in that column.
- You can change the data type if the column contains only null values. The exception to this is CHAR-to-VARCHAR2 conversions, which can be done with data in the columns.
- You can convert a CHAR column to the VARCHAR2 data type or convert a VARCHAR2 column to the CHAR data type only if the column contains null values or if you do not change the size.
- A change to the default value of a column affects only subsequent insertions in the table.
- You can add a NOT NULL constraint by using the MODIFY clauses.



Dropping a Column

- Use the DROP COLUMN clause to drop columns that you no longer need:

```
ALTER TABLE employees
DROP (termination_date);
```

```
Table altered.
```

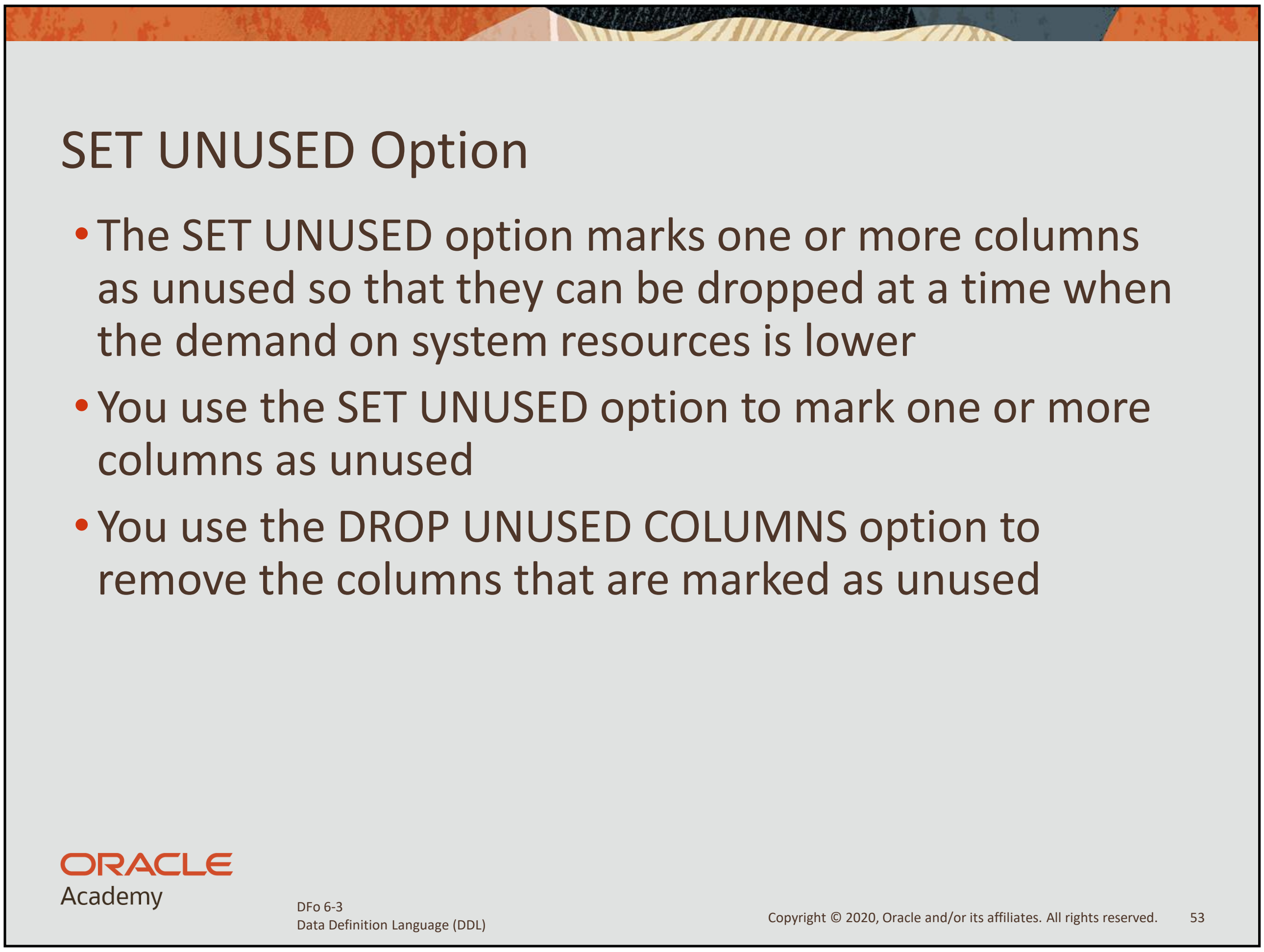
EMPLOYEE_ID	LAST_NAME	HIRE_DATE
100	King	17-Jun-1987
101	Kochhar	21-Sep-1989
102	De Haan	13-Jan-1993
200	Whalen	17-Sep-1987

…

ORACLE Academy

Here are the guidelines for dropping a column:

- The column may or may not contain data.
- With the ALTER TABLE DROP COLUMN statement, only one column can be dropped at a time.
- The table must have at least one column after it is altered.
- After a column is dropped, it cannot be recovered.
- A primary key that is referenced by another column cannot be dropped, unless the cascade option is added.
- Dropping a column can take a while if the column has a large number of values. In this case, it may be better to set it to be unused and drop it when there are fewer users on the system. That way, you avoid extended locks.



SET UNUSED Option

- The SET UNUSED option marks one or more columns as unused so that they can be dropped at a time when the demand on system resources is lower
- You use the SET UNUSED option to mark one or more columns as unused
- You use the DROP UNUSED COLUMNS option to remove the columns that are marked as unused

ORACLE Academy

DFo 6-3
Data Definition Language (DDL)

53

Unused columns are treated as if they were dropped, even though their column data remains in the table's rows.

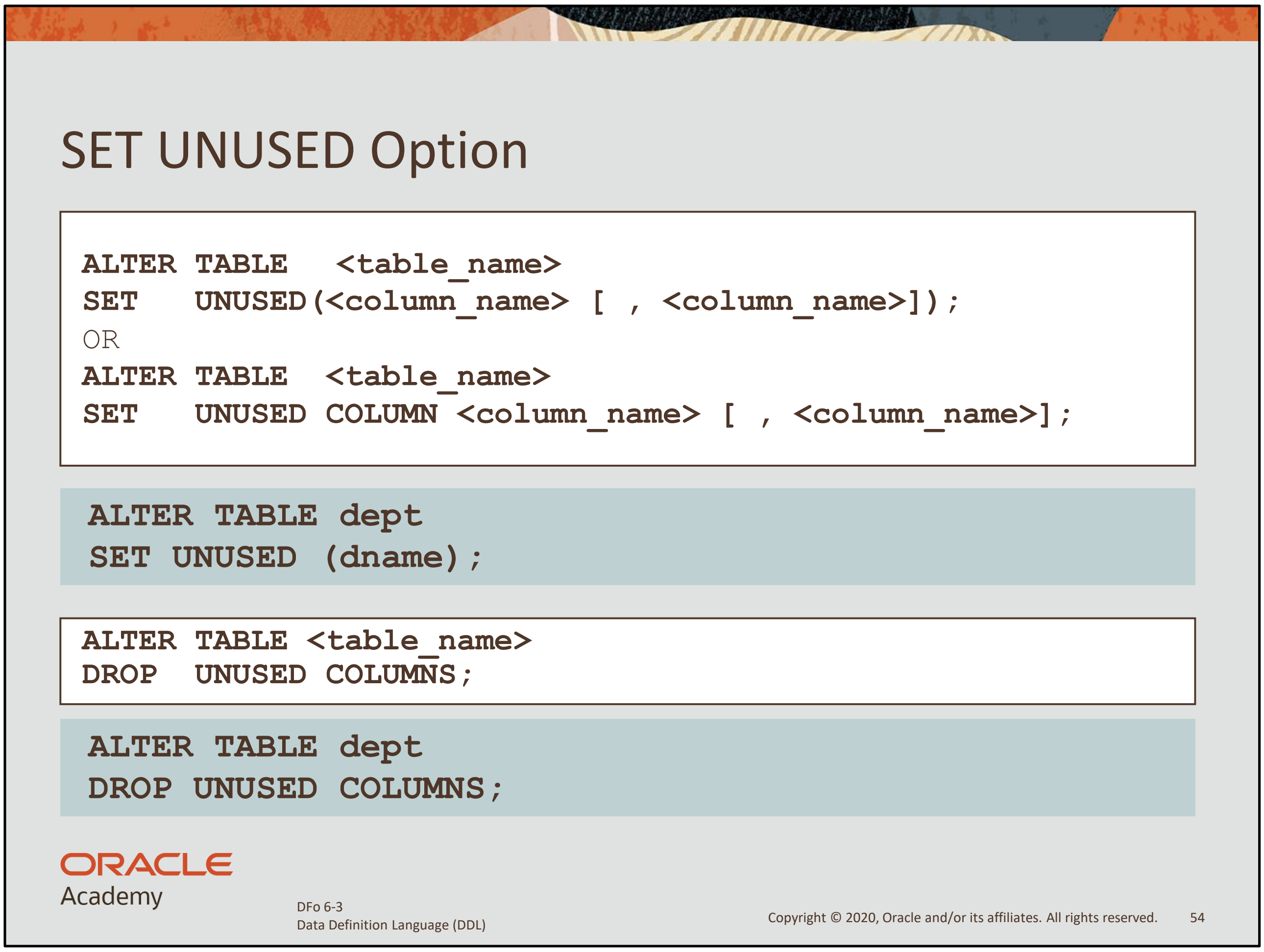
After a column is marked as unused, you have no access to that column. A SELECT * query does not retrieve data from unused columns. In addition, the names and types of columns marked as unused are not displayed during a DESCRIBE statement, and you can add a new column with the same name as an unused column.

You can specify the ONLINE keyword to indicate that data manipulation language (DML) operations on the table are allowed while marking the column or columns UNUSED. The following code example shows the use of SET UNUSED COLUMN, which sets a column unused forever with the addition of the ONLINE keyword:

```
ALTER TABLE dept80 SET UNUSED(hire_date)ONLINE;
```

The SET UNUSED information is stored in the USER_UNUSED_COL_TABS dictionary view.

Note: The guidelines for setting a column to be UNUSED are similar to those for dropping a column.



SET UNUSED Option

```
ALTER TABLE  <table_name>
SET   UNUSED(<column_name> [ , <column_name>]);
OR
ALTER TABLE  <table_name>
SET   UNUSED COLUMN <column_name> [ , <column_name>];
```

```
ALTER TABLE dept
SET UNUSED (dname);
```

```
ALTER TABLE <table_name>
DROP   UNUSED COLUMNS;
```

```
ALTER TABLE dept
DROP UNUSED COLUMNS;
```

ORACLE
Academy

On setting a column as UNUSED, you have the option of dropping that column.

You can use the DROP UNUSED COLUMNS to remove all columns that are currently marked as unused from the table. You can use this statement when you want to reclaim the extra disk space from the unused columns in the table. If the table contains no unused columns, the statement returns with no errors.

Note: A subsequent DROP UNUSED COLUMNS physically removes all unused columns from a table, similar to a DROP COLUMN.

Case Scenario: Altering Tables

Faculty

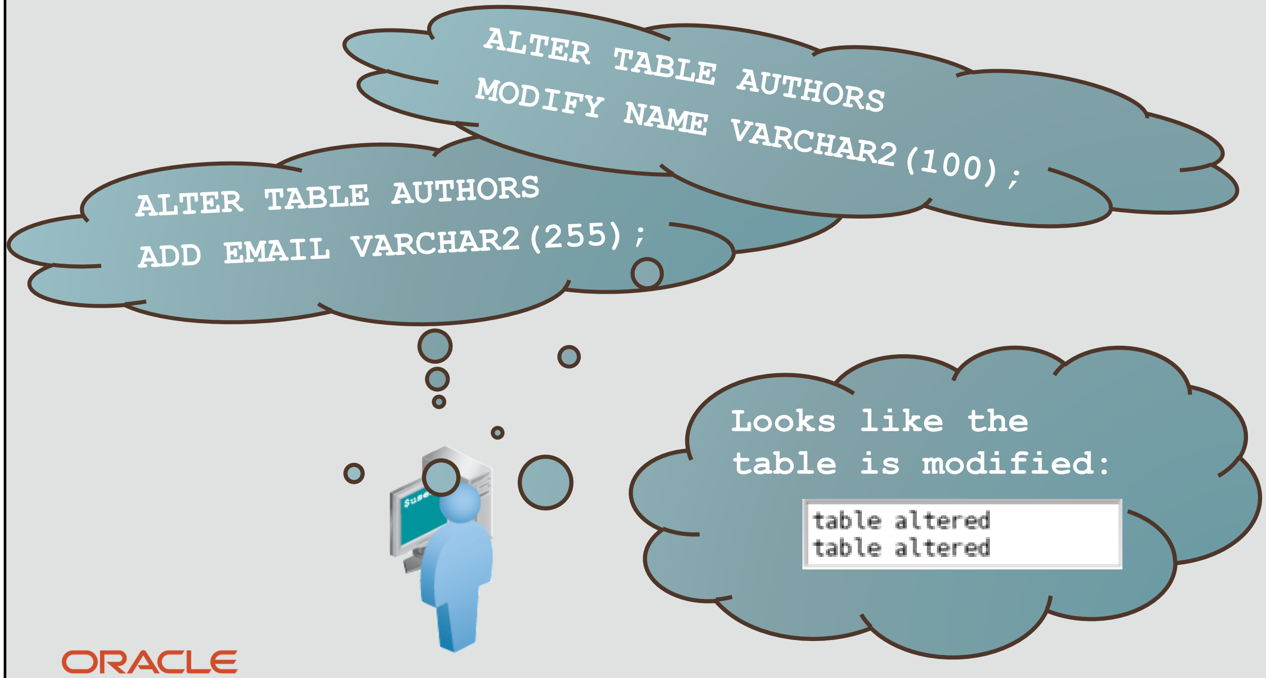
Sean, I was reviewing the AUTHORS table and realized that.
The author's email address field is missing.
The author's name column length needs to be increased.
Can you make these changes?

Sure, I can do it. Because the modification is adding a new column and is increasing the column length, this should not be an issue.

Student

ORACLE
Academy

Case Scenario: Altering Tables

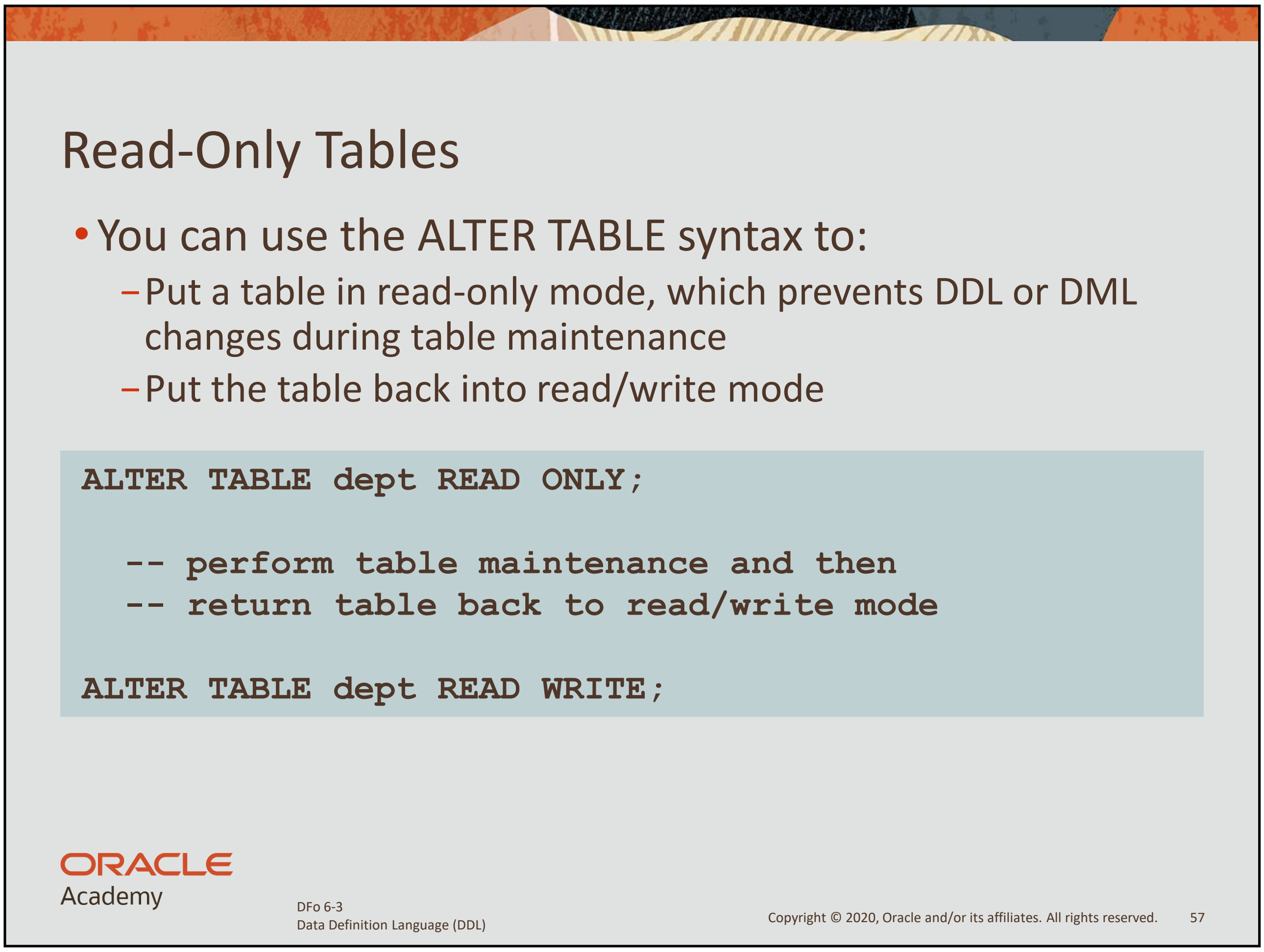


```
ALTER TABLE AUTHORS
MODIFY NAME VARCHAR2(100);
```

```
ALTER TABLE AUTHORS
ADD EMAIL VARCHAR2(255);
```

Looks like the table is modified:

```
table altered
table altered
```



Read-Only Tables

- You can use the ALTER TABLE syntax to:
  - Put a table in read-only mode, which prevents DDL or DML changes during table maintenance
  - Put the table back into read/write mode

```
ALTER TABLE dept READ ONLY;

  -- perform table maintenance and then
  -- return table back to read/write mode

ALTER TABLE dept READ WRITE;
```

ORACLE Academy

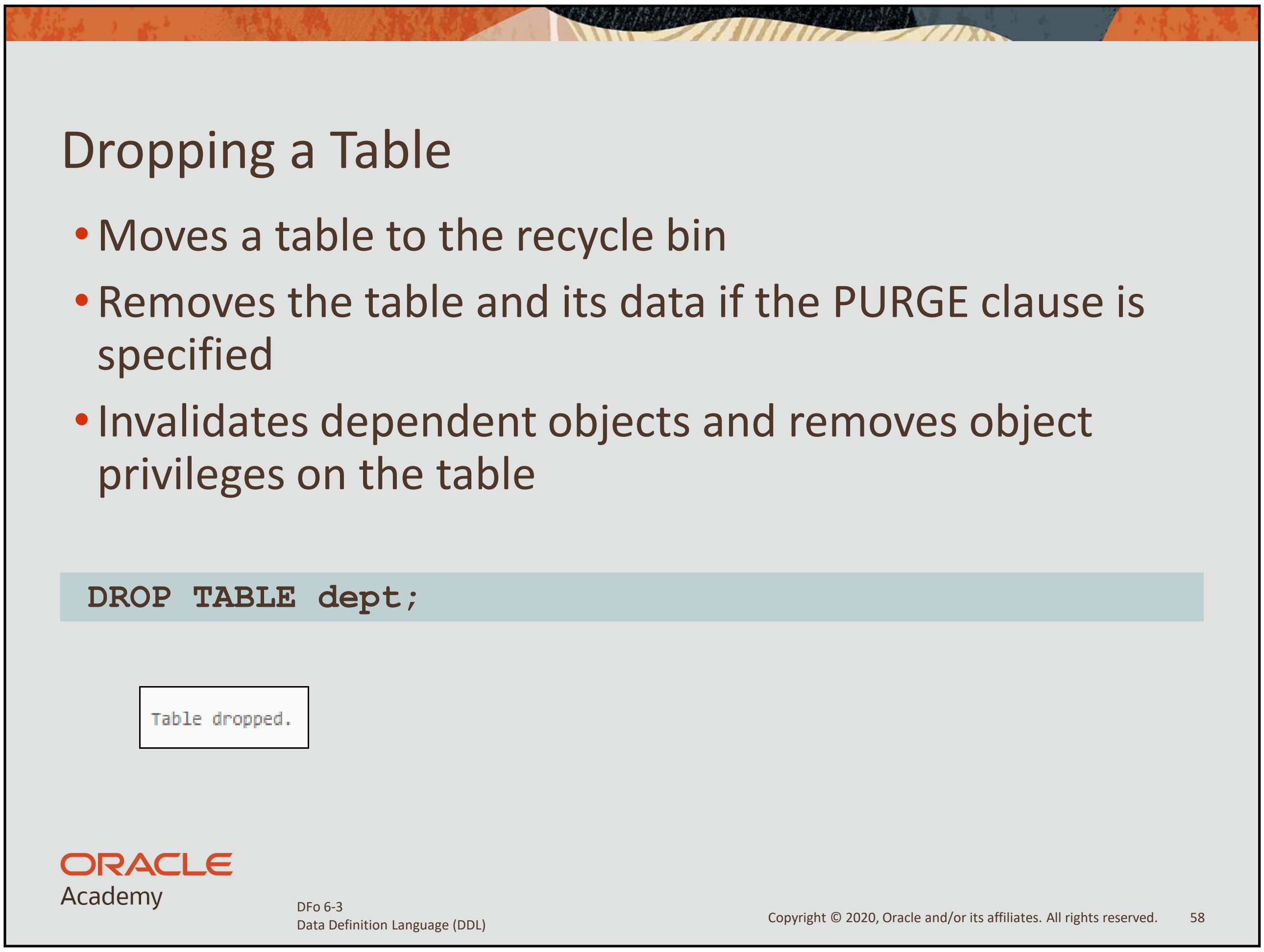
DFo 6-3
Data Definition Language (DDL)

57

Here are the guidelines for putting a table in read-only mode:

- You can specify READ ONLY to place a table in read-only mode.
- When a table is in read-only mode, you cannot issue any DML statements that affect the table or any SELECT ... FOR UPDATE statements.
- You can issue DDL statements as long as they do not modify any data in the table.
- Operations on indexes associated with the table are allowed when the table is in read-only mode.
- Specify READ/WRITE to return a read-only table to read/write mode.

Note: You can drop a table that is in READ ONLY mode. The DROP command is executed only in the data dictionary, so access to the table contents is not required. The space used by the table is not reclaimed until the tablespace is made read/write again, and then the required changes can be made to the block segment headers, and so on.



Dropping a Table

- Moves a table to the recycle bin
- Removes the table and its data if the PURGE clause is specified
- Invalidates dependent objects and removes object privileges on the table

```
DROP TABLE dept;
```

```
Table dropped.
```

ORACLE Academy

DFo 6-3
Data Definition Language (DDL)

58

Unless you specify the PURGE clause, the DROP TABLE statement does not result in space being released back to the tablespace for use by other objects, and the space continues to count toward the user's space quota. Dropping a table invalidates the dependent objects and removes object privileges on the table.

When you drop a table, the database loses all data in the table and all indexes associated with it.

Syntax

DROP TABLE *table* [PURGE]

In the syntax, *table* is the name of the table.

Here are the guidelines for dropping a table:

- All data is deleted from the table.
- Any views and synonyms remain, but they are invalid.
- Any pending transactions are committed.
- Only the creator of the table or a user with the DROP ANY TABLE privilege can remove a table.

Project Exercise

- DFo_6_3_Project
  - Oracle Baseball League Store Database
  - Using DDL to build and maintain database tables

ORACLE
Academy

Summary

- In this lesson, you should have learned how to:
  - Identify the steps needed to create database tables
  - Describe the purpose of the DDL
  - List the DDL operations needed to build and maintain a database's tables

ORACLE
Academy