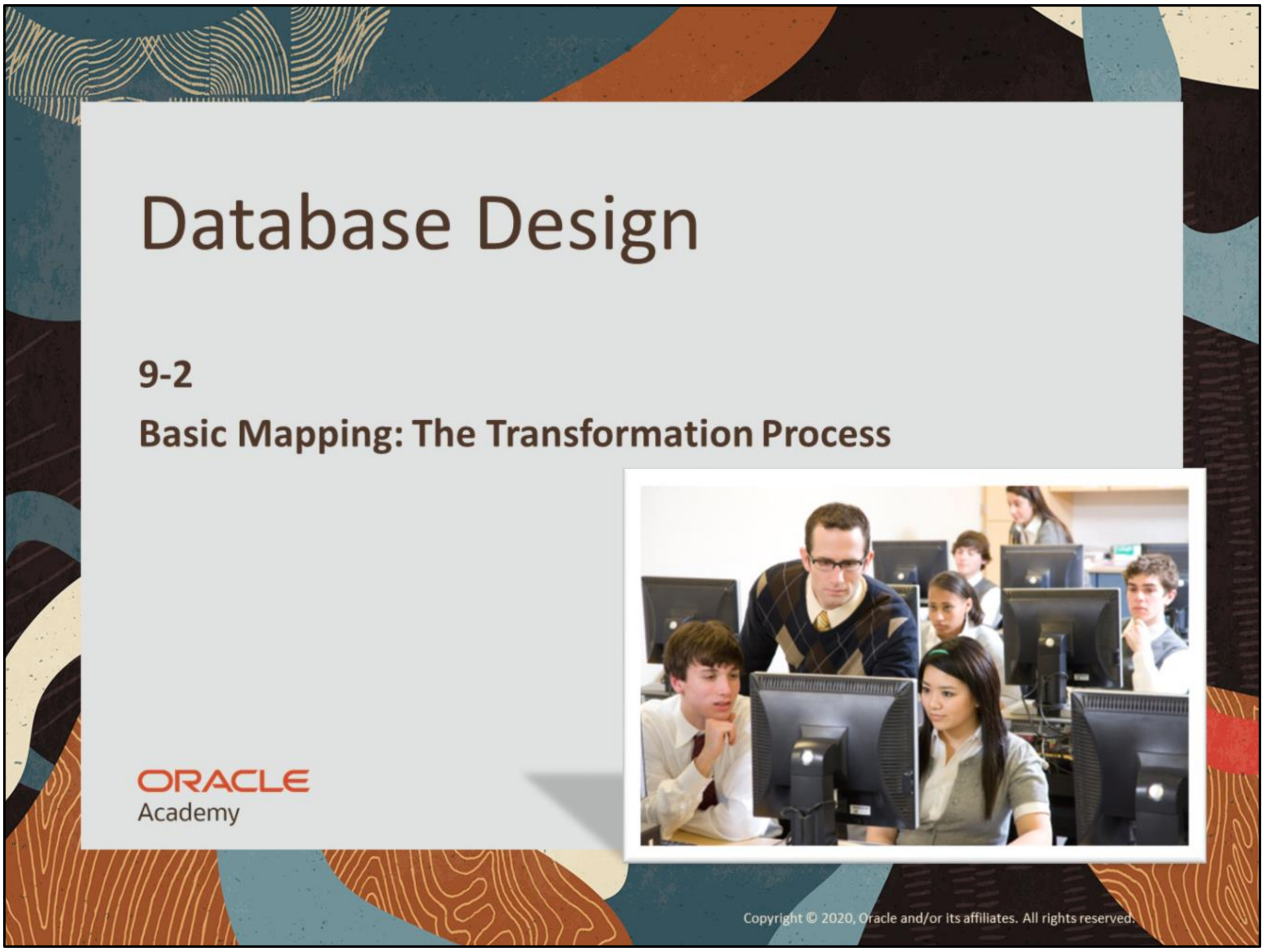
ORACLE

Academy

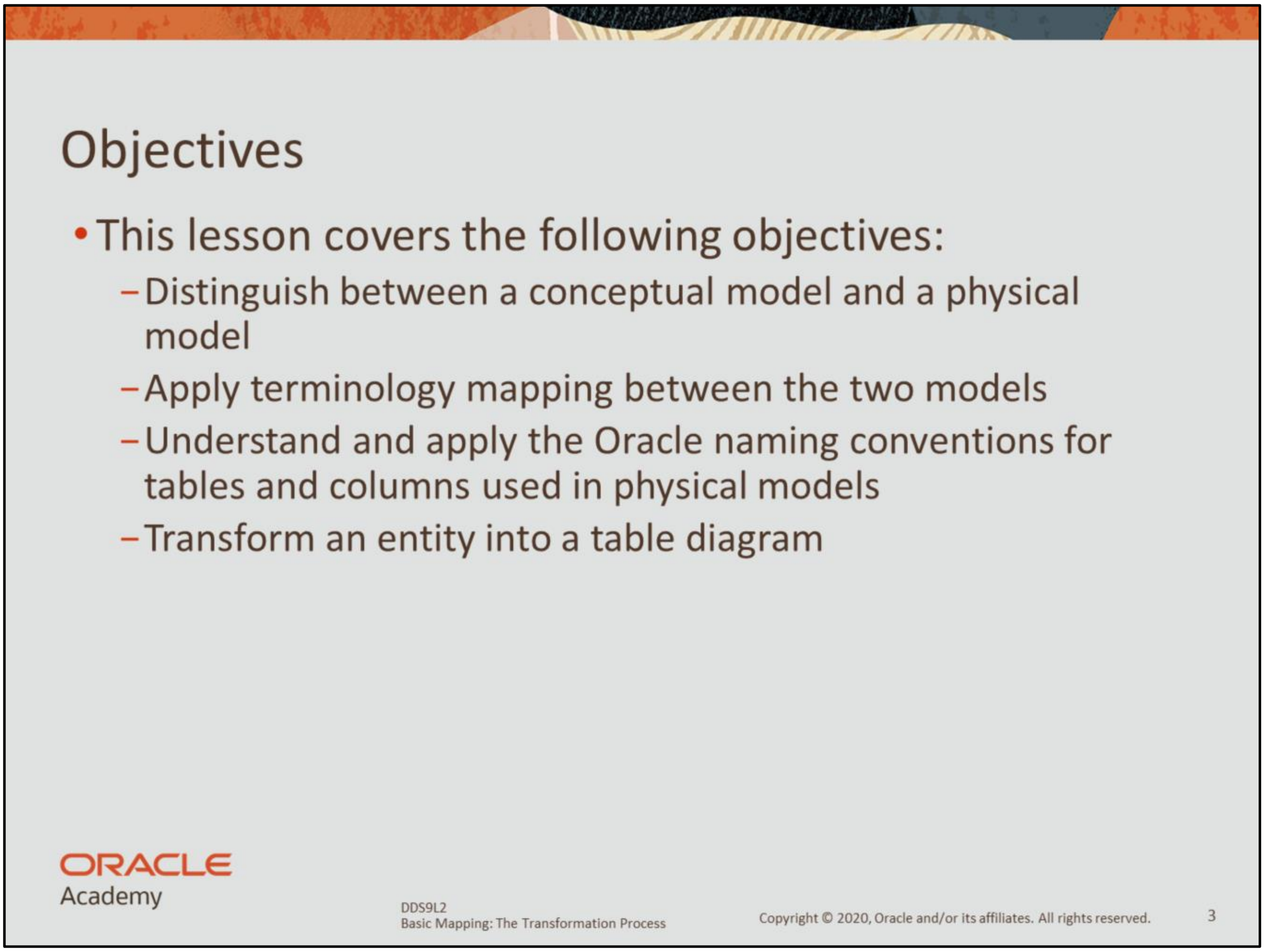


Database Design

9-2

Basic Mapping: The Transformation Process

ORACLE
Academy



Objectives

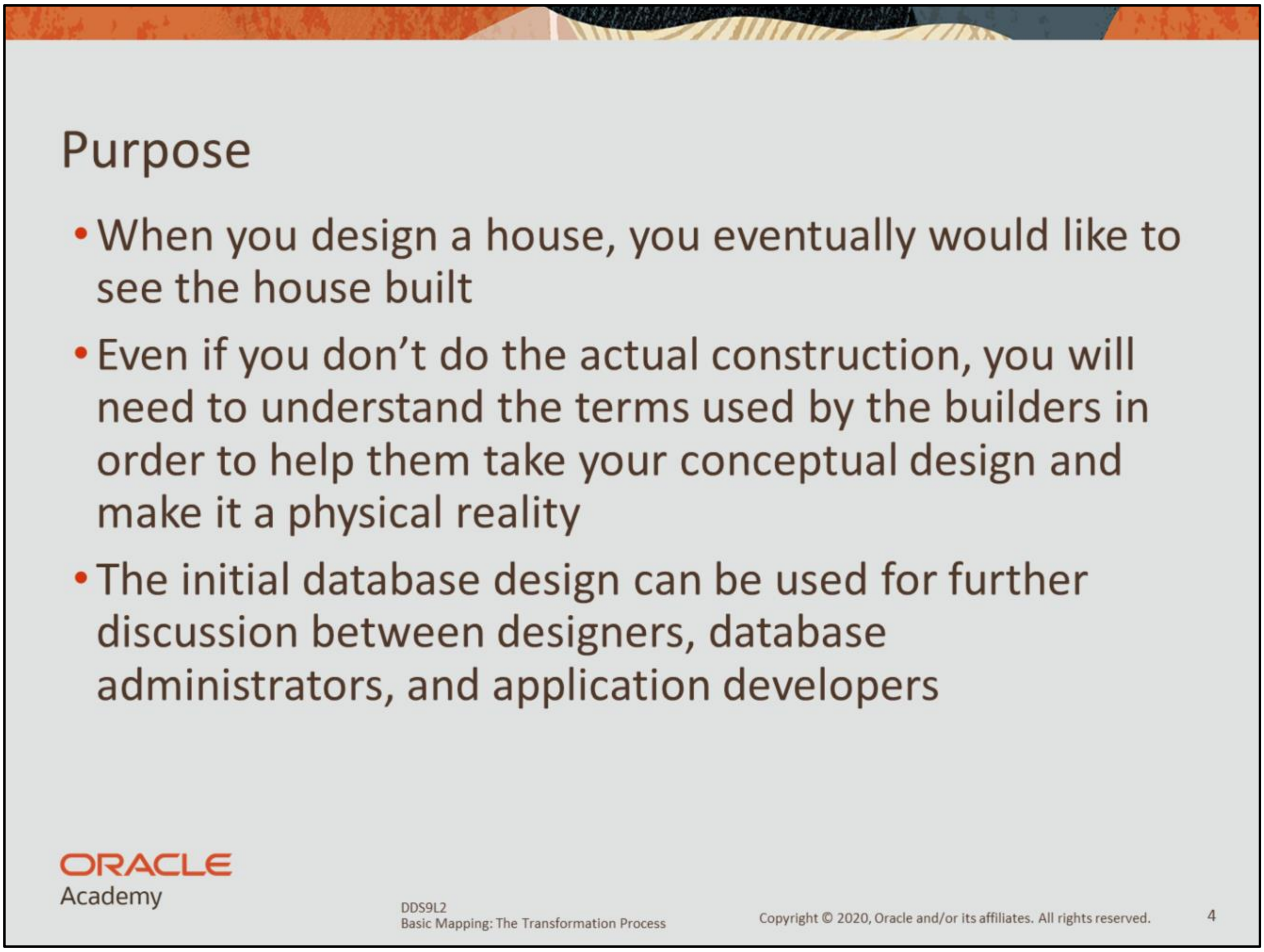
- This lesson covers the following objectives:
  - Distinguish between a conceptual model and a physical model
  - Apply terminology mapping between the two models
  - Understand and apply the Oracle naming conventions for tables and columns used in physical models
  - Transform an entity into a table diagram

ORACLE
Academy

You may not see a great difference between the conceptual model and the relational design at this point.
This stage of the design process transforms an ERD into table definitions. Table definitions are then used to create the physical database.

We are transforming the terminology we have used to build our conceptual model into the equivalent relational database terminology, following naming conventions and restrictions.

Simple entities (like the ones they will see in this lesson) are very similar to relational tables. However, once we get to foreign keys, arcs, and subtypes, there will be differences.

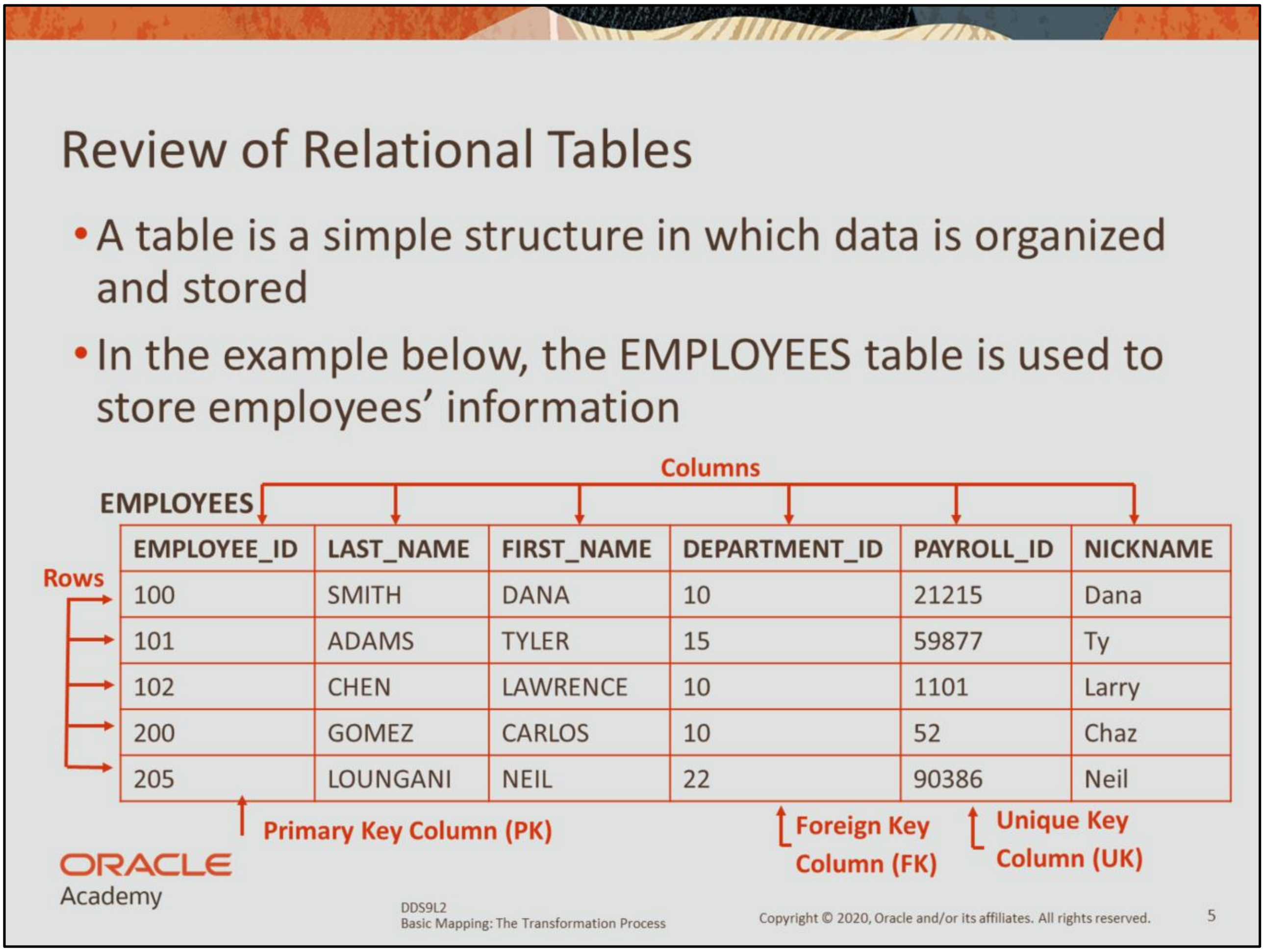


Purpose

- When you design a house, you eventually would like to see the house built
- Even if you don’t do the actual construction, you will need to understand the terms used by the builders in order to help them take your conceptual design and make it a physical reality
- The initial database design can be used for further discussion between designers, database administrators, and application developers

ORACLE
Academy

When we create a conceptual model, we are focused on the business and its rules. When we create a database design, the focus will be on database issues of storage, speed of transactions, security, etc. For example, in a Data Warehouse, the physical model is often deliberately de-normalized to give faster performance.
Although these are important issues, they should not be considered before or above the business requirements. Data modeling pays attention to the business requirements, regardless of implementation. You may have the fastest and most secure database in the world, but if it doesn’t meet your business requirements, it’s not going to be of much use.



Review of Relational Tables

- A table is a simple structure in which data is organized and stored
- In the example below, the EMPLOYEES table is used to store employees' information

Columns

EMPLOYEES

Rows

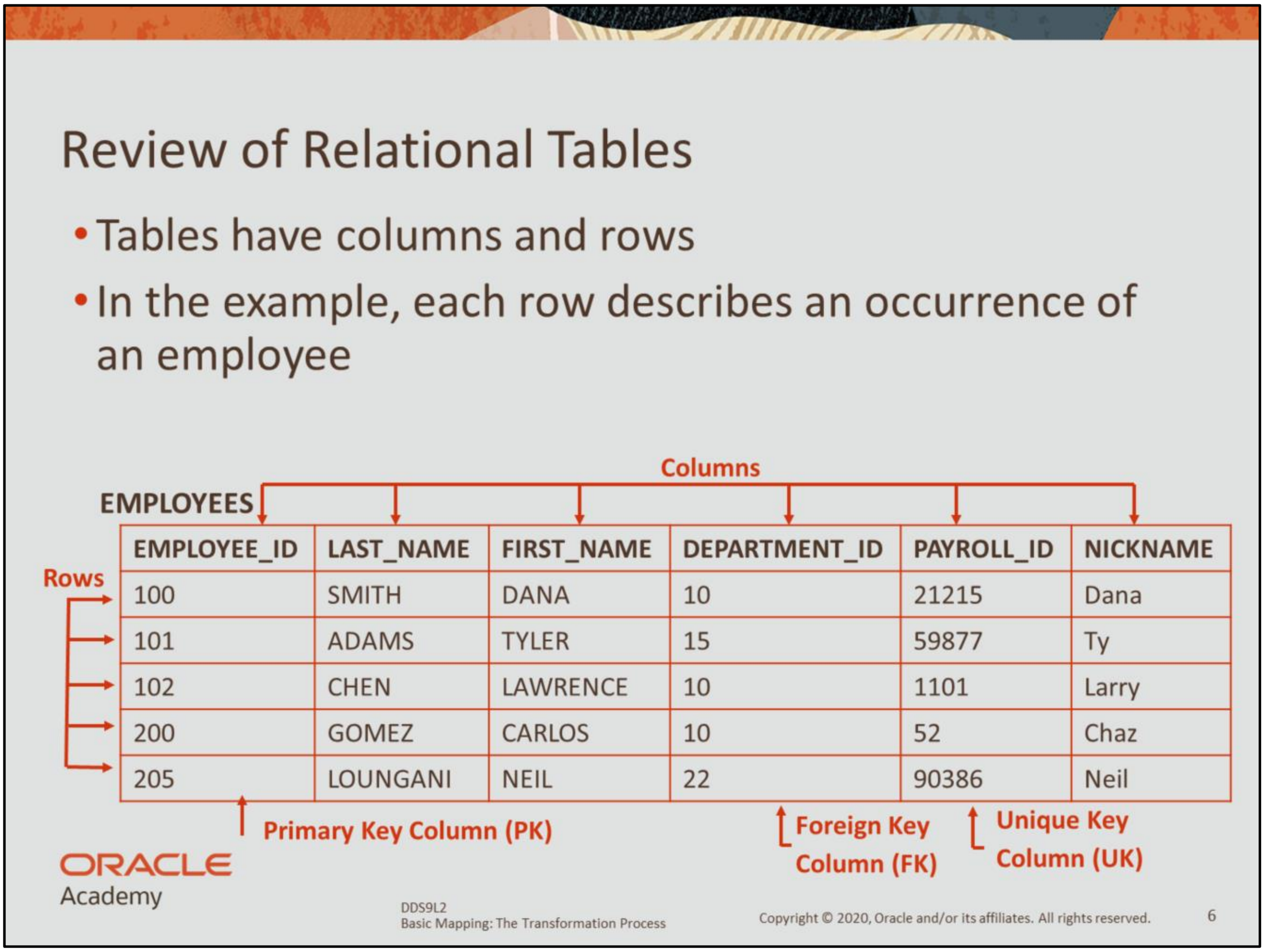
EMPLOYEE_ID	LAST_NAME	FIRST_NAME	DEPARTMENT_ID	PAYROLL_ID	NICKNAME
100	SMITH	DANA	10	21215	Dana
101	ADAMS	TYLER	15	59877	Ty
102	CHEN	LAWRENCE	10	1101	Larry
200	GOMEZ	CARLOS	10	52	Chaz
205	LOUNGANI	NEIL	22	90386	Neil

Primary Key Column (PK)

Foreign Key Column (FK)

Unique Key Column (UK)

We will discuss primary, foreign and unique keys later in this lesson.



Review of Relational Tables

- Tables have columns and rows
- In the example, each row describes an occurrence of an employee

Columns

EMPLOYEES

Rows

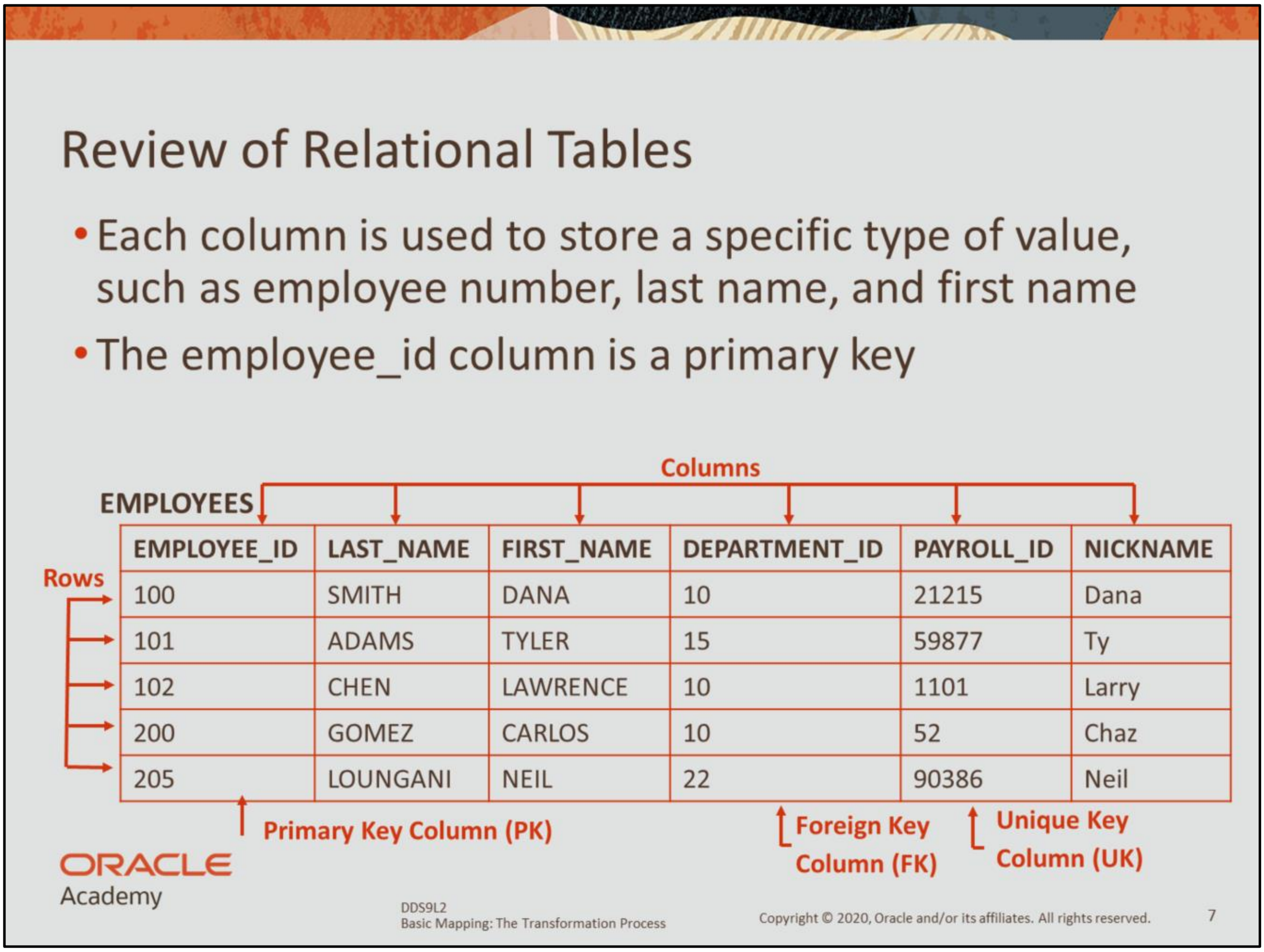
EMPLOYEE_ID	LAST_NAME	FIRST_NAME	DEPARTMENT_ID	PAYROLL_ID	NICKNAME
100	SMITH	DANA	10	21215	Dana
101	ADAMS	TYLER	15	59877	Ty
102	CHEN	LAWRENCE	10	1101	Larry
200	GOMEZ	CARLOS	10	52	Chaz
205	LOUNGANI	NEIL	22	90386	Neil

Primary Key Column (PK)

Foreign Key Column (FK)

Unique Key Column (UK)

ORACLE Academy



Review of Relational Tables

- Each column is used to store a specific type of value, such as employee number, last name, and first name
- The employee_id column is a primary key

Columns

EMPLOYEES

Rows

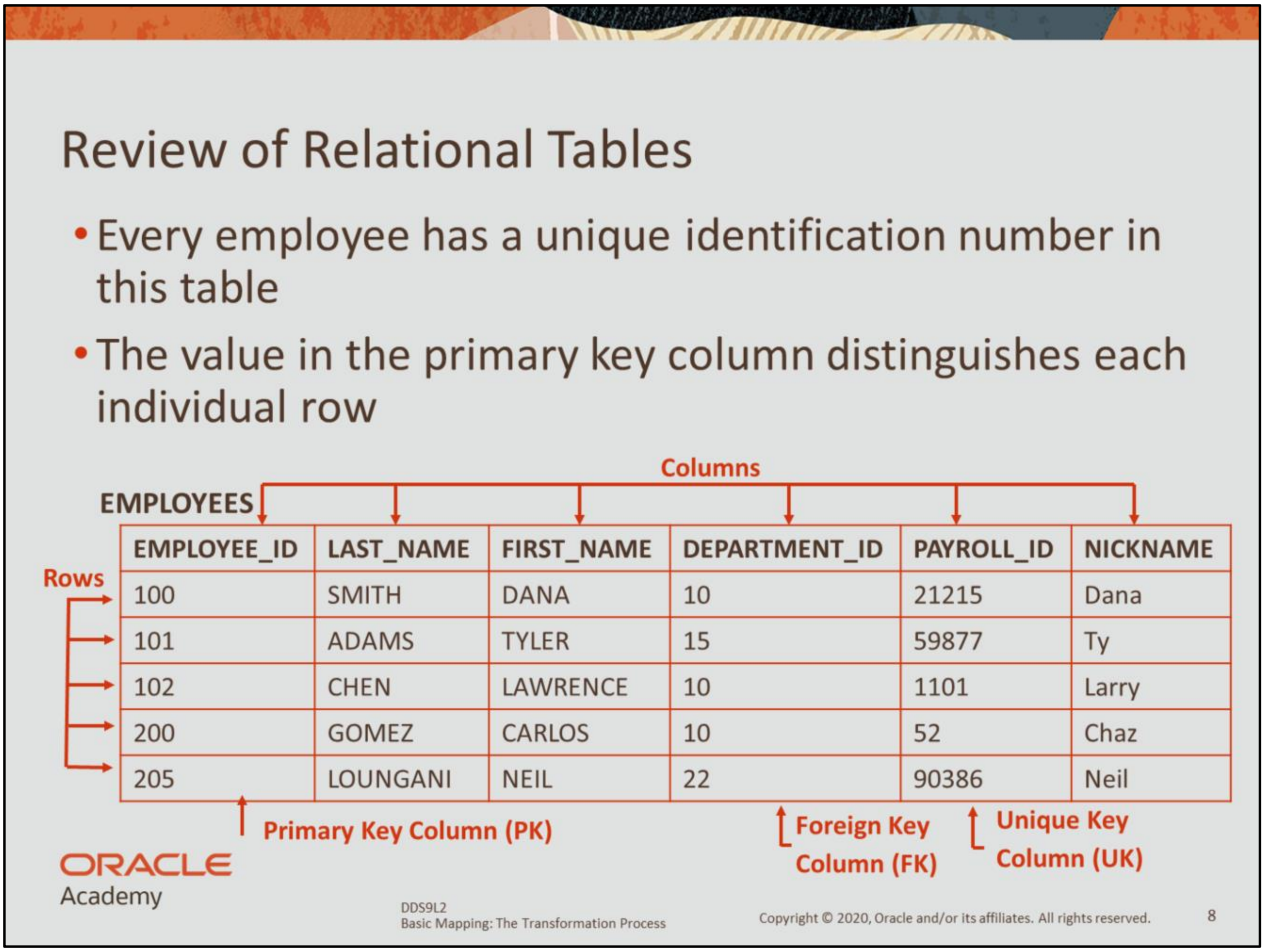
EMPLOYEE_ID	LAST_NAME	FIRST_NAME	DEPARTMENT_ID	PAYROLL_ID	NICKNAME
100	SMITH	DANA	10	21215	Dana
101	ADAMS	TYLER	15	59877	Ty
102	CHEN	LAWRENCE	10	1101	Larry
200	GOMEZ	CARLOS	10	52	Chaz
205	LOUNGANI	NEIL	22	90386	Neil

Primary Key Column (PK)

Foreign Key Column (FK)

Unique Key Column (UK)

ORACLE Academy



Review of Relational Tables

- Every employee has a unique identification number in this table
- The value in the primary key column distinguishes each individual row

Columns

EMPLOYEES

Rows

EMPLOYEE_ID	LAST_NAME	FIRST_NAME	DEPARTMENT_ID	PAYROLL_ID	NICKNAME
100	SMITH	DANA	10	21215	Dana
101	ADAMS	TYLER	15	59877	Ty
102	CHEN	LAWRENCE	10	1101	Larry
200	GOMEZ	CARLOS	10	52	Chaz
205	LOUNGANI	NEIL	22	90386	Neil

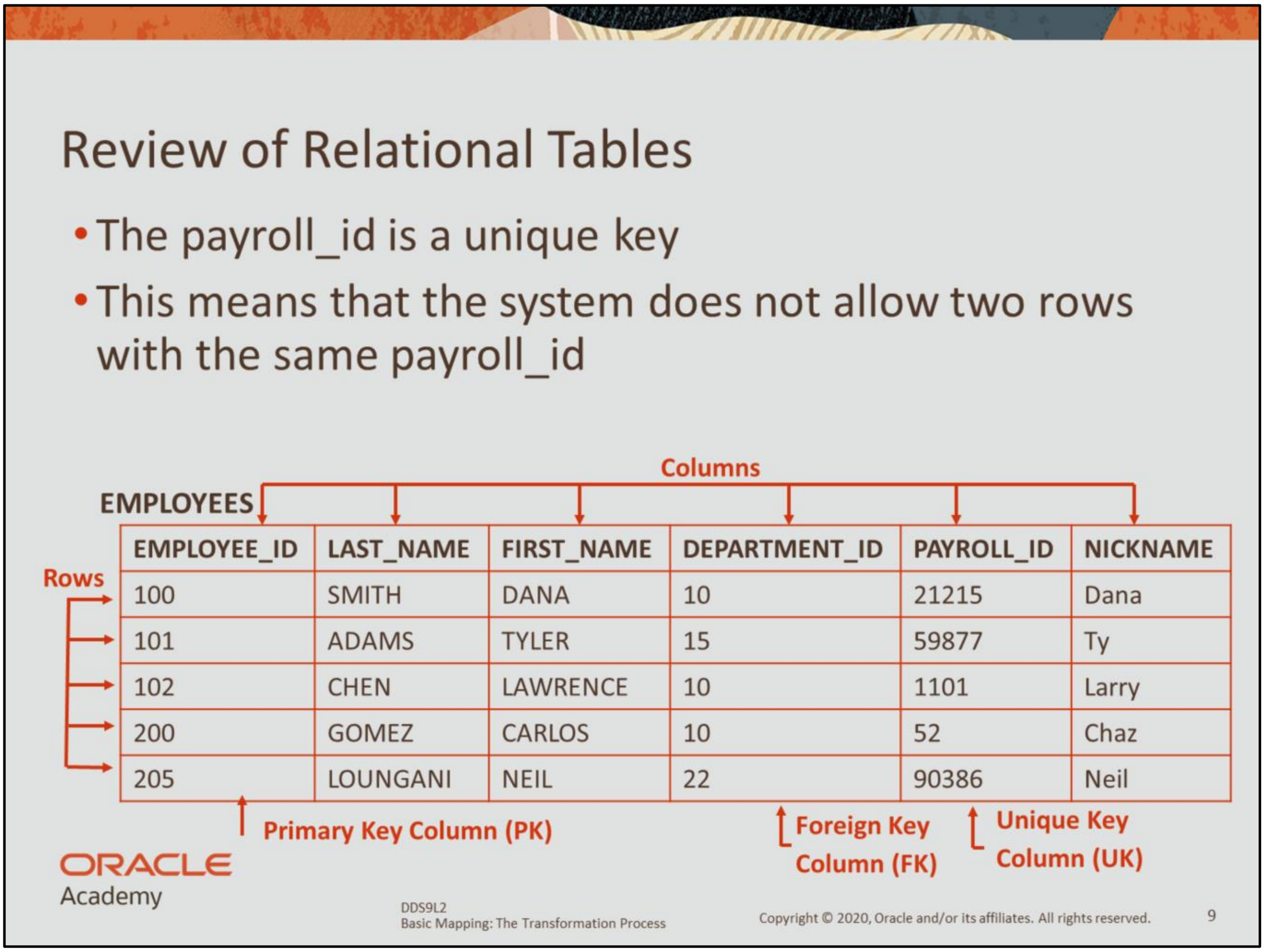
Primary Key Column (PK)

Foreign Key Column (FK)

Unique Key Column (UK)

ORACLE Academy

DDS9L2
Basic Mapping: The Transformation Process



Review of Relational Tables

- The payroll_id is a unique key
- This means that the system does not allow two rows with the same payroll_id

EMPLOYEES

Columns

Rows

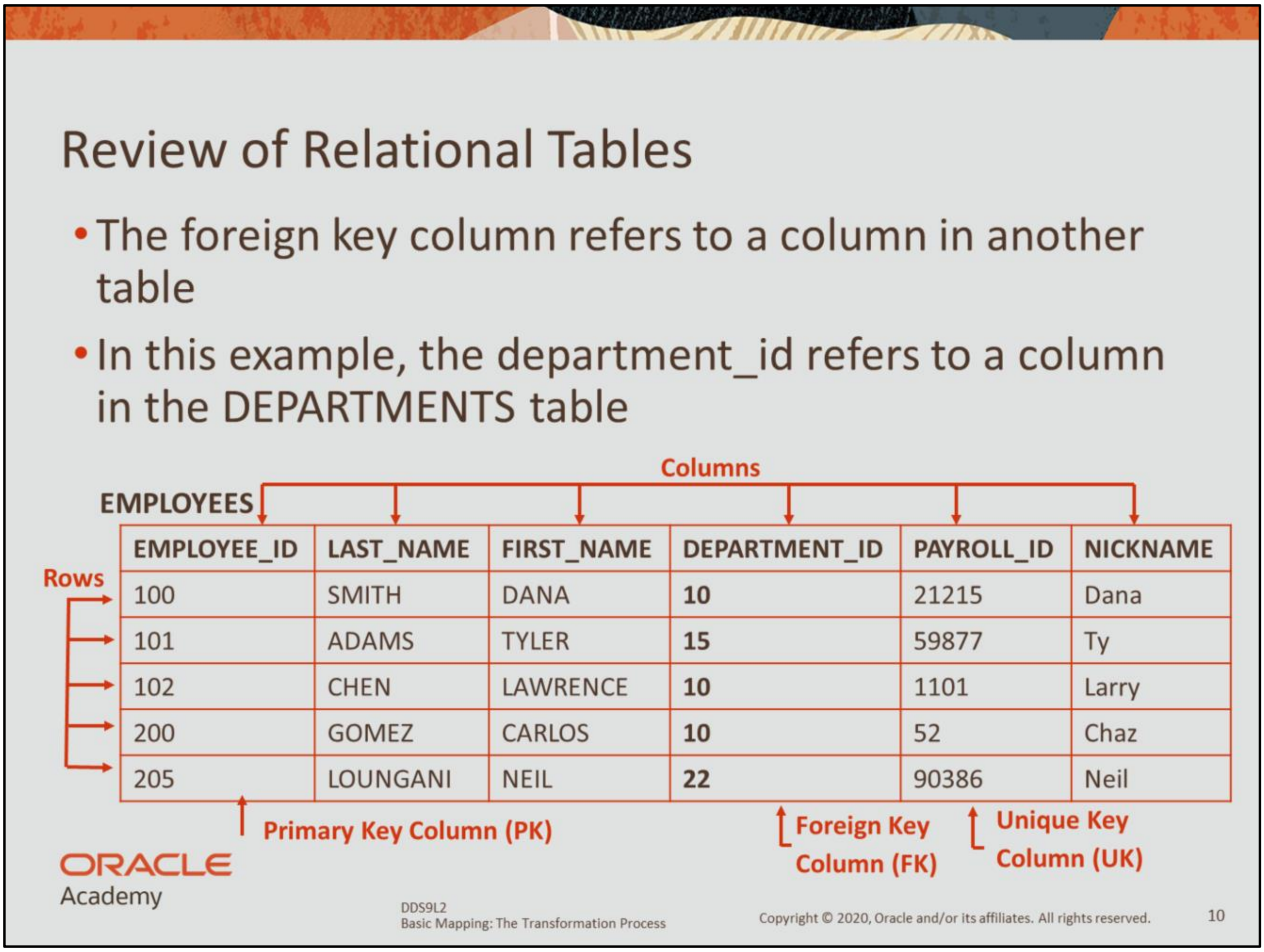
EMPLOYEE_ID	LAST_NAME	FIRST_NAME	DEPARTMENT_ID	PAYROLL_ID	NICKNAME
100	SMITH	DANA	10	21215	Dana
101	ADAMS	TYLER	15	59877	Ty
102	CHEN	LAWRENCE	10	1101	Larry
200	GOMEZ	CARLOS	10	52	Chaz
205	LOUNGANI	NEIL	22	90386	Neil

Primary Key Column (PK)

Foreign Key Column (FK)

Unique Key Column (UK)

ORACLE Academy



Review of Relational Tables

- The foreign key column refers to a column in another table
- In this example, the department_id refers to a column in the DEPARTMENTS table

Columns

EMPLOYEES

Rows

EMPLOYEE_ID	LAST_NAME	FIRST_NAME	DEPARTMENT_ID	PAYROLL_ID	NICKNAME
100	SMITH	DANA	**10**	21215	Dana
101	ADAMS	TYLER	**15**	59877	Ty
102	CHEN	LAWRENCE	**10**	1101	Larry
200	GOMEZ	CARLOS	**10**	52	Chaz
205	LOUNGANI	NEIL	**22**	90386	Neil

Primary Key Column (PK)

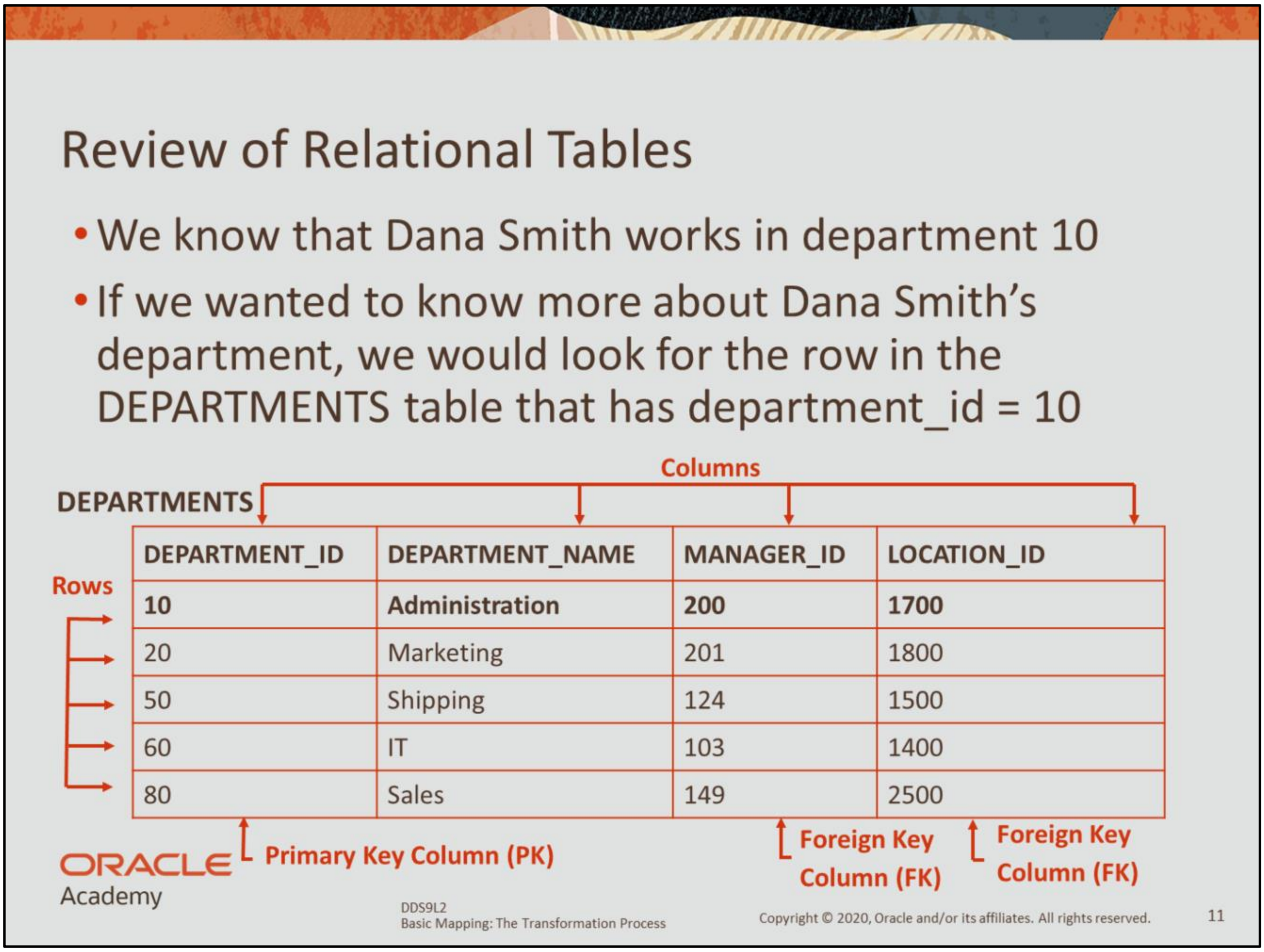
Foreign Key Column (FK)

Unique Key Column (UK)

ORACLE Academy

DDS9L2
Basic Mapping: The Transformation Process

10



Review of Relational Tables

- We know that Dana Smith works in department 10
- If we wanted to know more about Dana Smith's department, we would look for the row in the DEPARTMENTS table that has department_id = 10

Columns

DEPARTMENTS

Rows

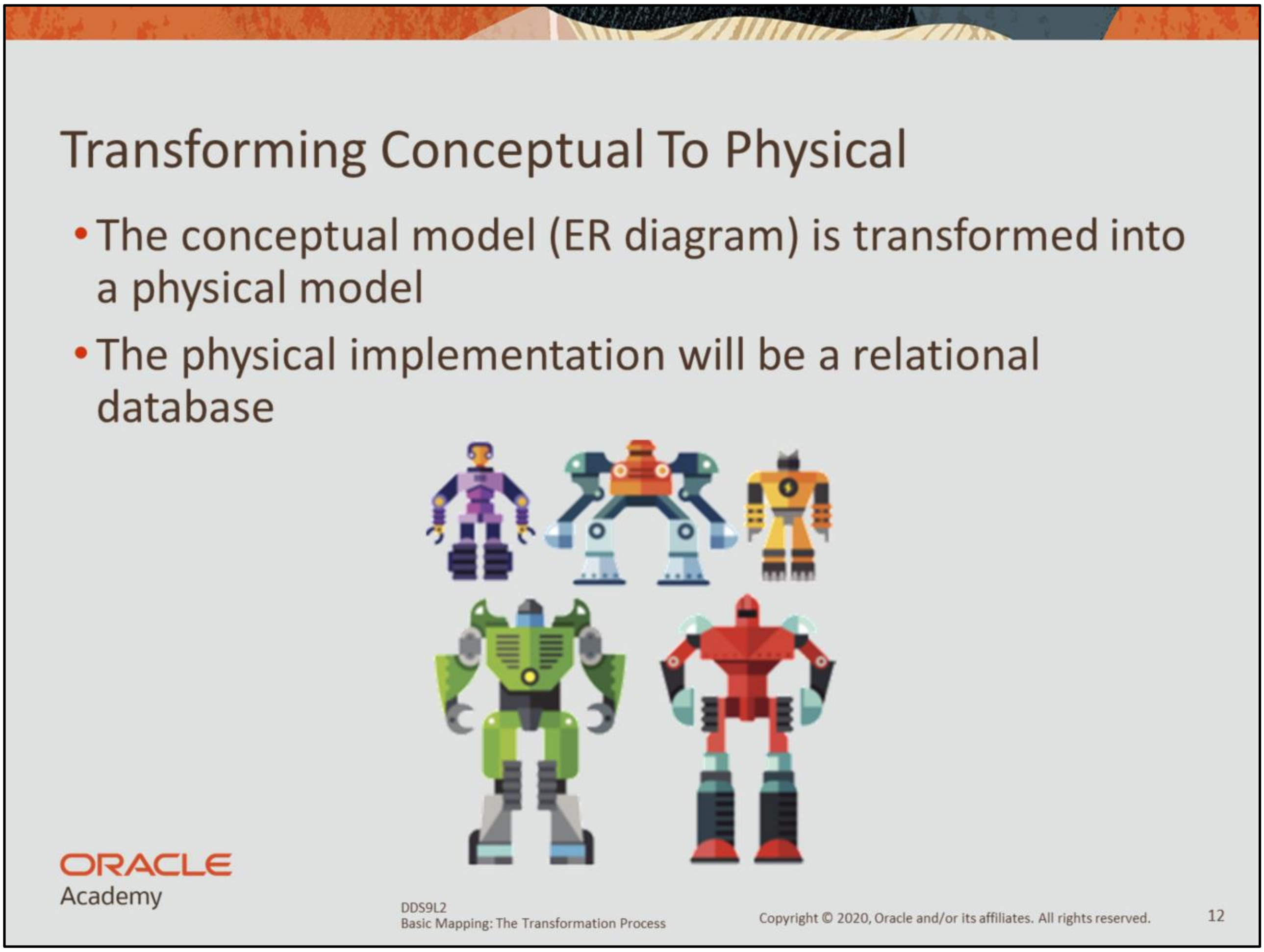
DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	**Administration**	**200**	**1700**
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500

Primary Key Column (PK)

Foreign Key Column (FK)

Foreign Key Column (FK)

ORACLE Academy

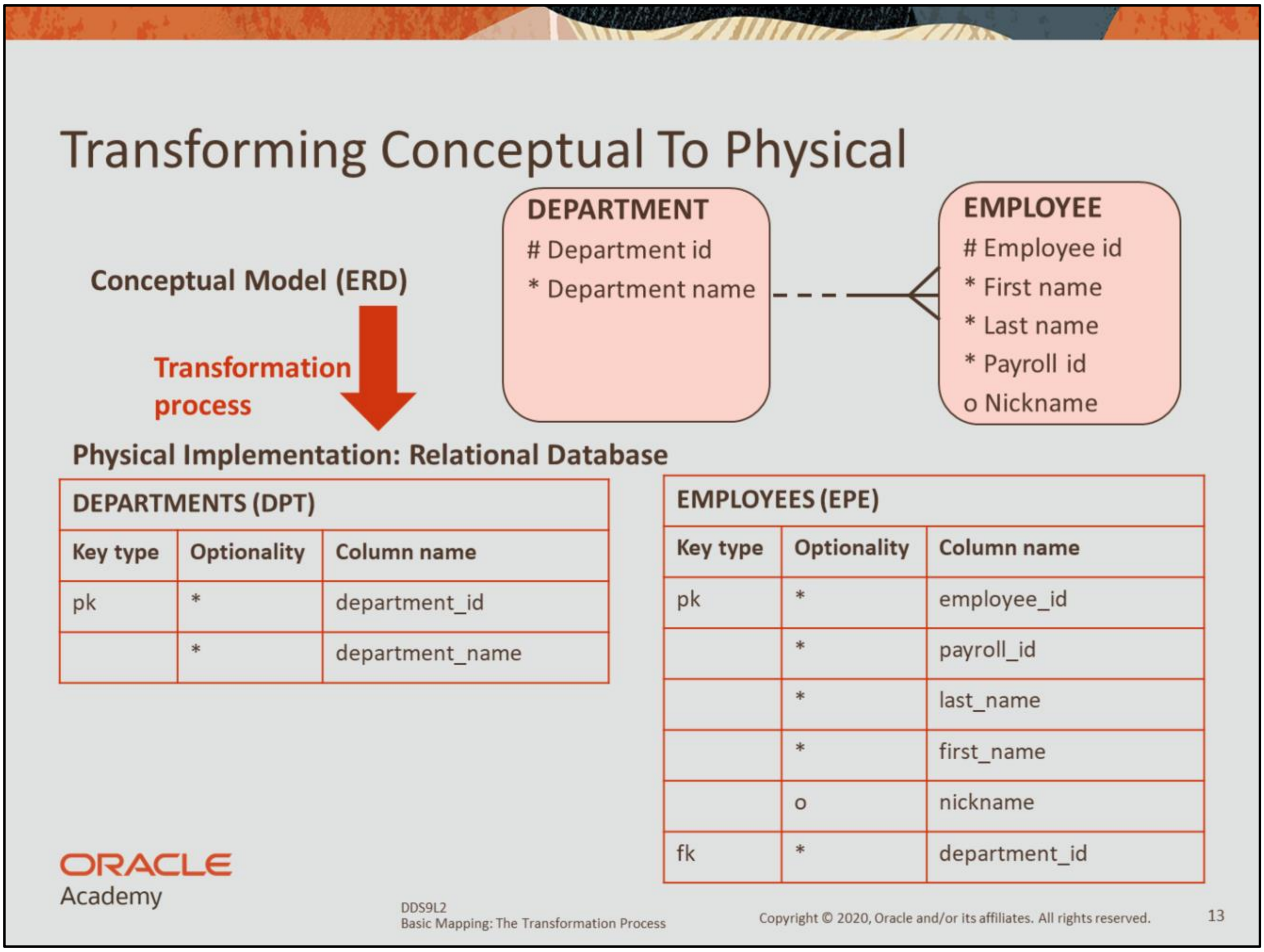


Transforming Conceptual To Physical

- The conceptual model (ER diagram) is transformed into a physical model
- The physical implementation will be a relational database

ORACLE
Academy

Transform: To change the elements of an ERD (entities, attributes, relationships) into database elements (tables, attributes, foreign keys).



Transforming Conceptual To Physical

Conceptual Model (ERD)

Transformation process

Physical Implementation: Relational Database

DEPARTMENTS (DPT)

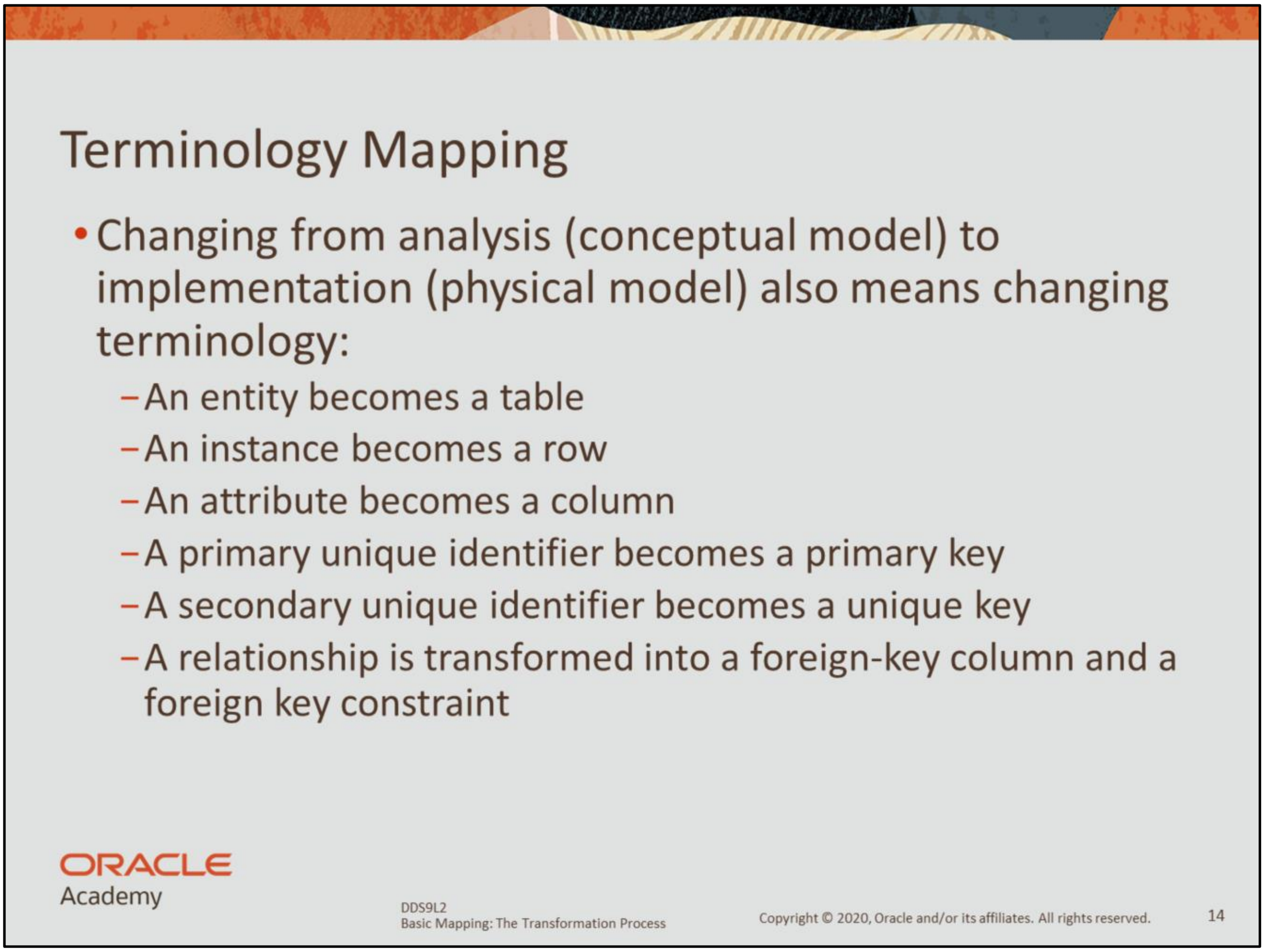
Key type	Optionality	Column name
pk	*	department_id
	*	department_name

EMPLOYEES (EPE)

Key type	Optionality	Column name
pk	*	employee_id
	*	payroll_id
	*	last_name
	*	first_name
	o	nickname
fk	*	department_id

ORACLE Academy

The EMPLOYEE entity in the ERD (conceptual model) transforms into the diagram of the EMPLOYEES table, which represents the definition of the table in the relational model (physical implementation). The notations in the table diagram will be explained later in this lesson.

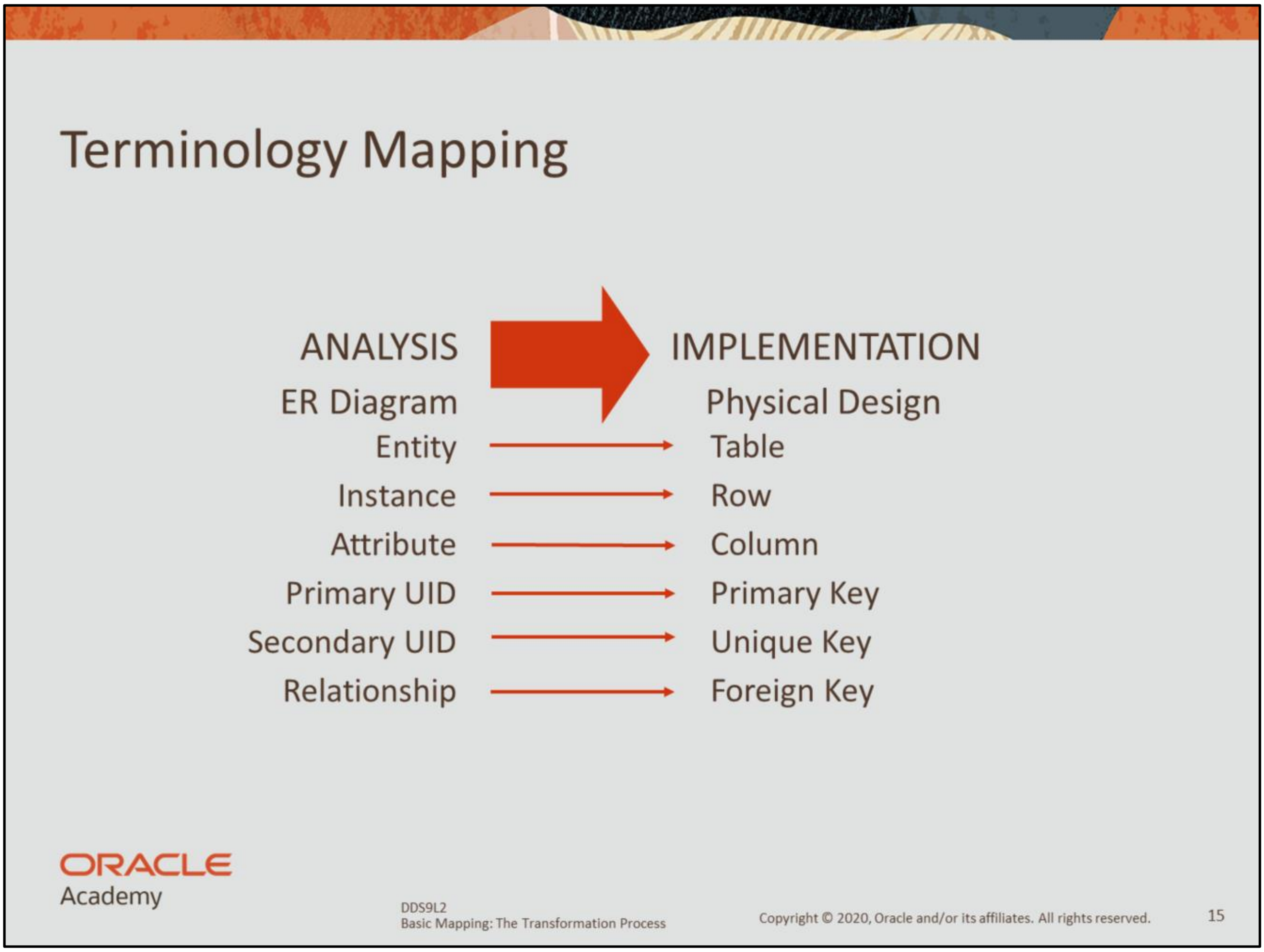


Terminology Mapping

- Changing from analysis (conceptual model) to implementation (physical model) also means changing terminology:
  - An entity becomes a table
  - An instance becomes a row
  - An attribute becomes a column
  - A primary unique identifier becomes a primary key
  - A secondary unique identifier becomes a unique key
  - A relationship is transformed into a foreign-key column and a foreign key constraint

ORACLE
Academy

Map: To associate the elements of an ERD (entities, attributes, relationships) with database elements (tables, attributes, foreign keys).



Terminology Mapping

ANALYSIS	IMPLEMENTATION
ER Diagram	Physical Design
Entity	Table
Instance	Row
Attribute	Column
Primary UID	Primary Key
Secondary UID	Unique Key
Relationship	Foreign Key

ORACLE
Academy

Analysis and design are phases of the system development life cycle (to be discussed more later). When designing a system, analysis precedes design. Data modeling is done in the analysis phase. When you are satisfied that you have captured the business requirements in the data model, you move on to the design phase, where the ERD is mapped to a physical implementation.

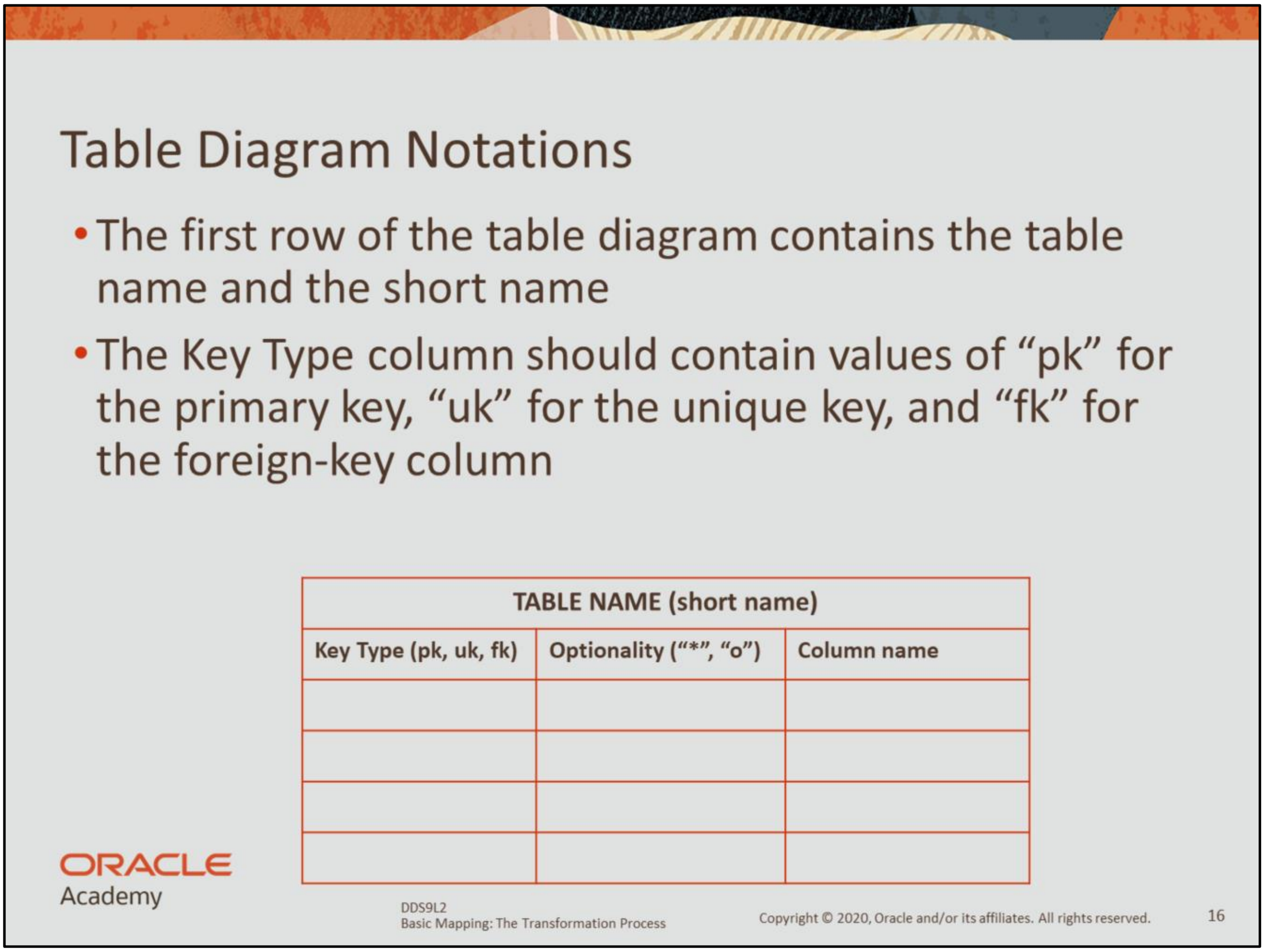


Table Diagram Notations

- The first row of the table diagram contains the table name and the short name
- The Key Type column should contain values of “pk” for the primary key, “uk” for the unique key, and “fk” for the foreign-key column

TABLE NAME (short name)		
Key Type (pk, uk, fk)	Optionality (“*”, “o”)	Column name

ORACLE Academy

In these simple examples there is a one-to-one mapping between conceptual and physical terminology (for example one entity becomes one table) but that this will not always be true in more complex models.

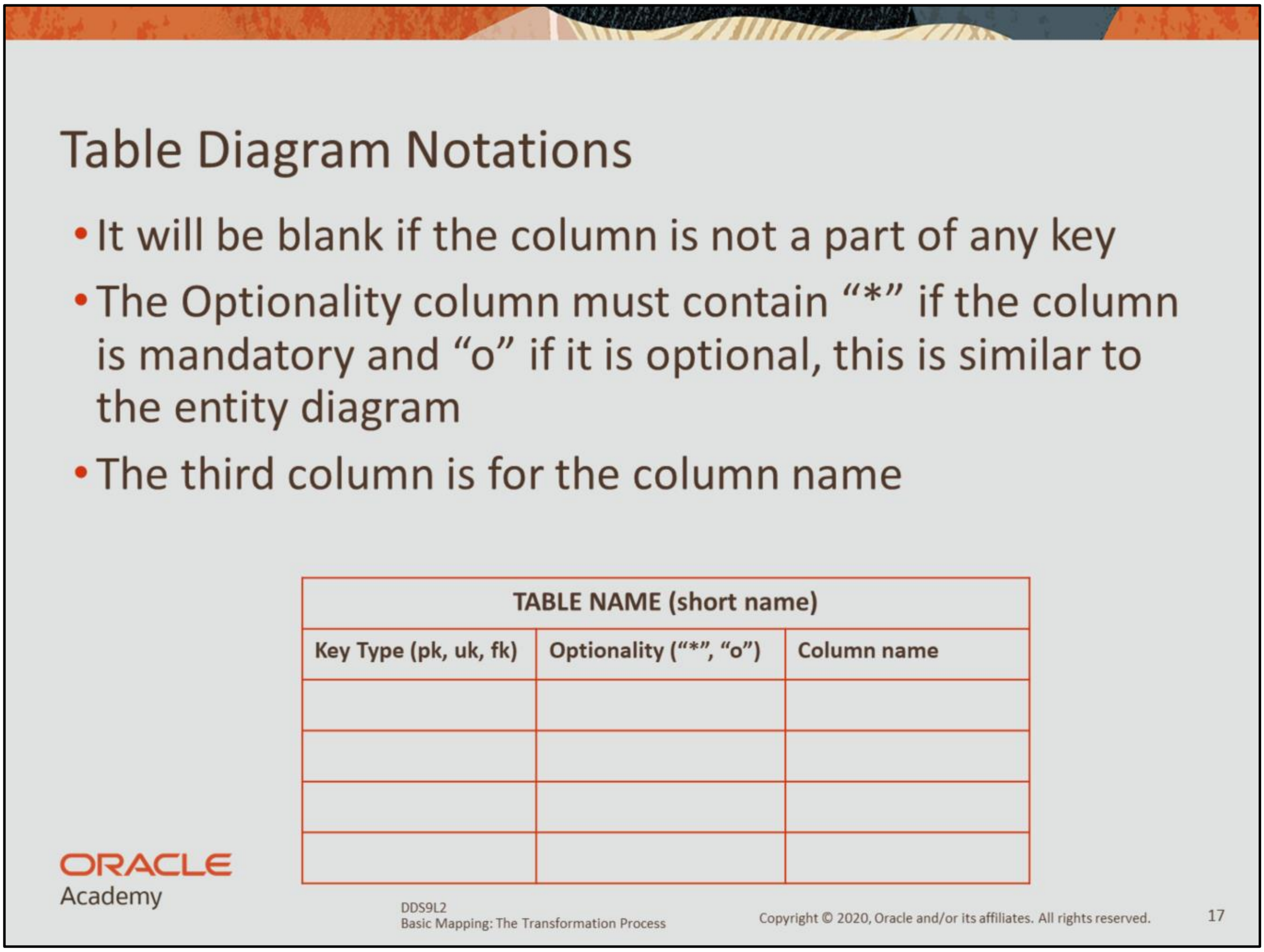
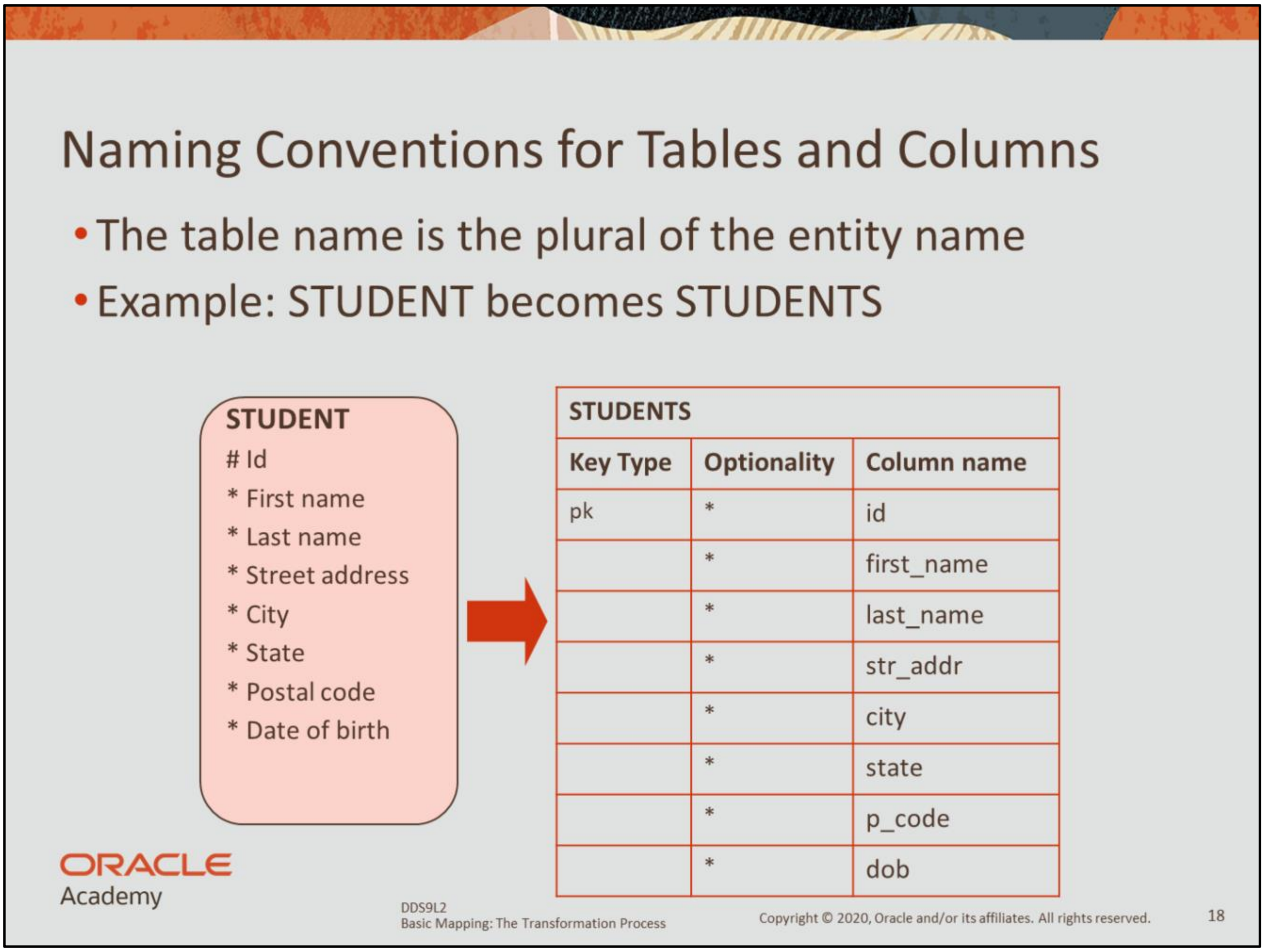


Table Diagram Notations

- It will be blank if the column is not a part of any key
- The Optionality column must contain “*” if the column is mandatory and “o” if it is optional, this is similar to the entity diagram
- The third column is for the column name

TABLE NAME (short name)		
Key Type (pk, uk, fk)	Optionality (“*”, “o”)	Column name

ORACLE
Academy



Naming Conventions for Tables and Columns

- The table name is the plural of the entity name
- Example: STUDENT becomes STUDENTS

STUDENT

Id

* First name

* Last name

* Street address

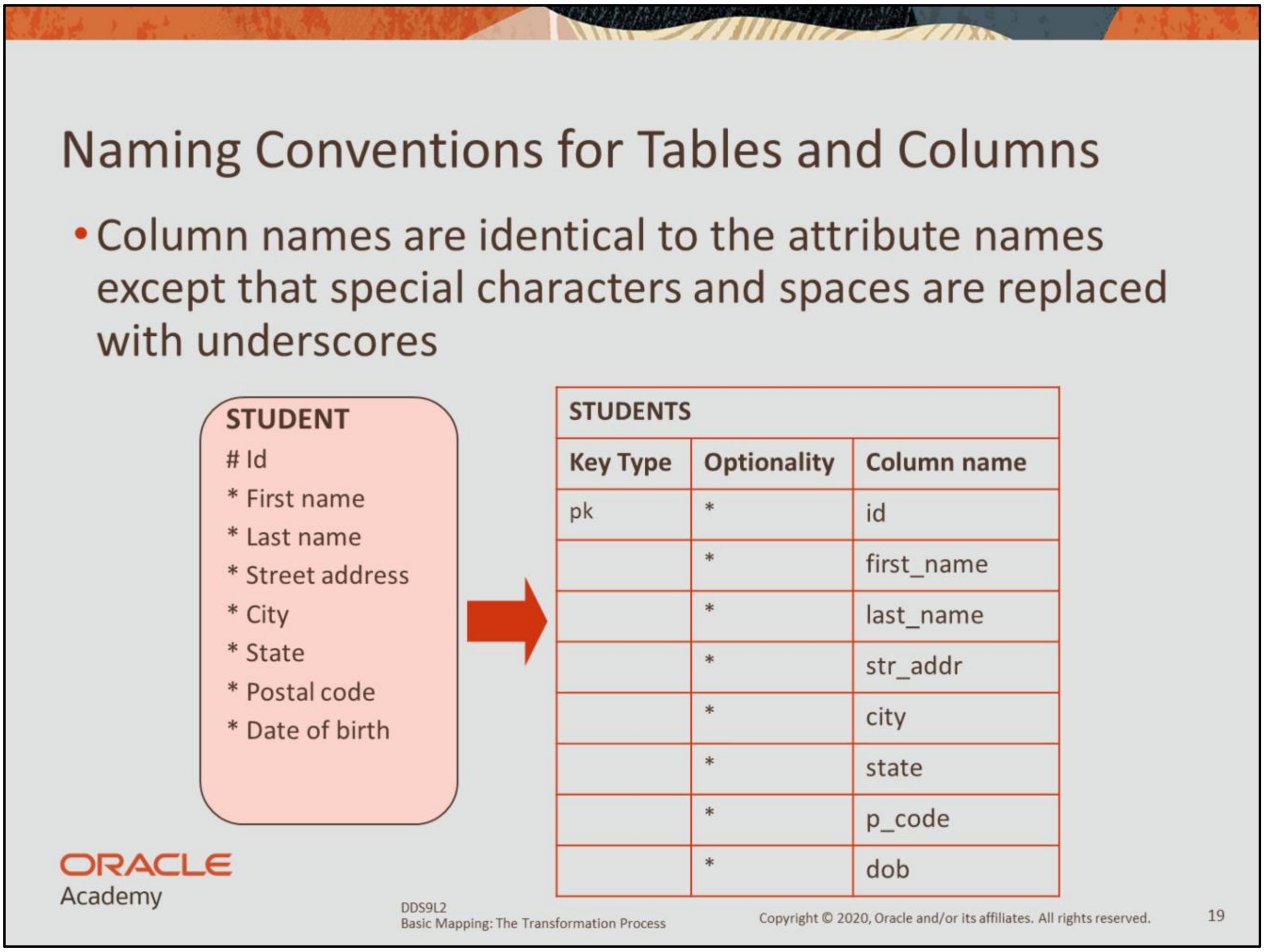
* City

* State

* Postal code

* Date of birth

STUDENTS		
Key Type	**Optionality**	**Column name**
pk	*	id
	*	first_name
	*	last_name
	*	str_addr
	*	city
	*	state
	*	p_code
	*	dob



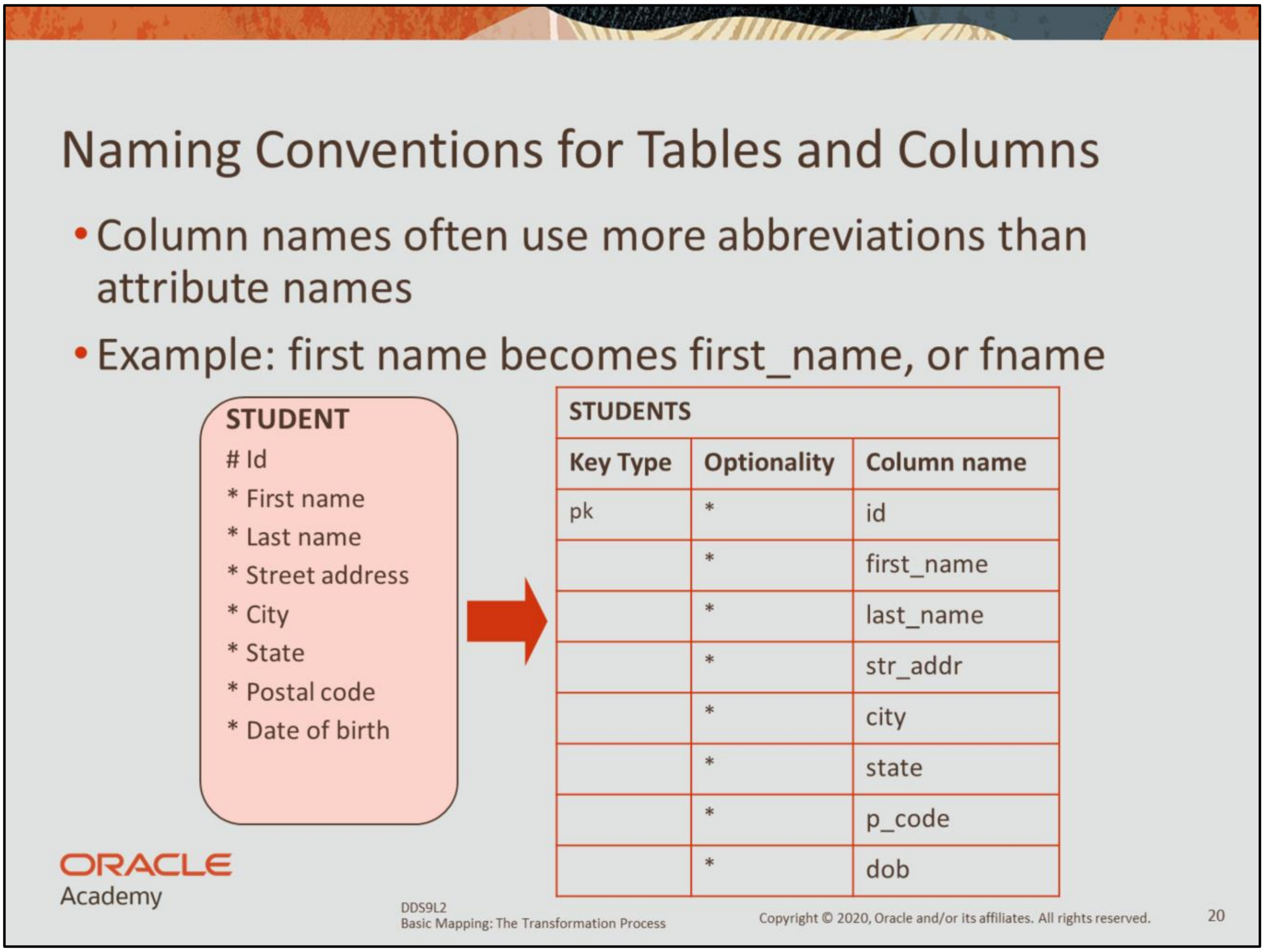
Naming Conventions for Tables and Columns

- Column names are identical to the attribute names except that special characters and spaces are replaced with underscores

STUDENT
Id
* First name
* Last name
* Street address
* City
* State
* Postal code
* Date of birth

STUDENTS		
Key Type	**Optionality**	**Column name**
pk	*	id
	*	first_name
	*	last_name
	*	str_addr
	*	city
	*	state
	*	p_code
	*	dob

ORACLE
Academy



Naming Conventions for Tables and Columns

- Column names often use more abbreviations than attribute names
- Example: first name becomes first_name, or fname

STUDENT
Id
* First name
* Last name
* Street address
* City
* State
* Postal code
* Date of birth

STUDENTS

Key Type	Optionality	Column name
pk	*	id
	*	first_name
	*	last_name
	*	str_addr
	*	city
	*	state
	*	p_code
	*	dob

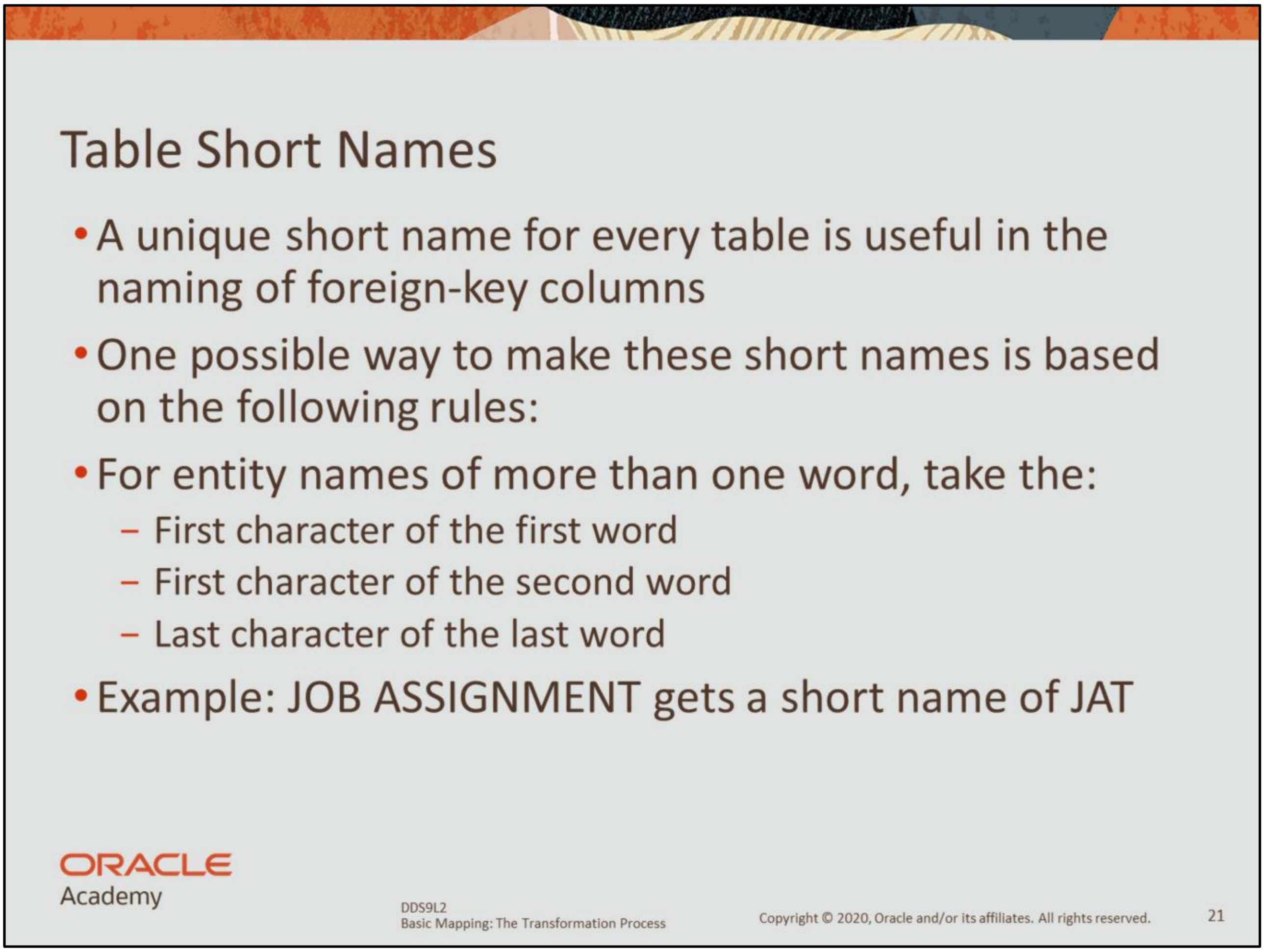


Table Short Names

- A unique short name for every table is useful in the naming of foreign-key columns
- One possible way to make these short names is based on the following rules:
- For entity names of more than one word, take the:
  - First character of the first word
  - First character of the second word
  - Last character of the last word
- Example: JOB ASSIGNMENT gets a short name of JAT

ORACLE Academy

DDS9L2
Basic Mapping: The Transformation Process

Short names are NOT mandatory, simply useful. The suggested "rules" are one of several possible conventions for determining short names.

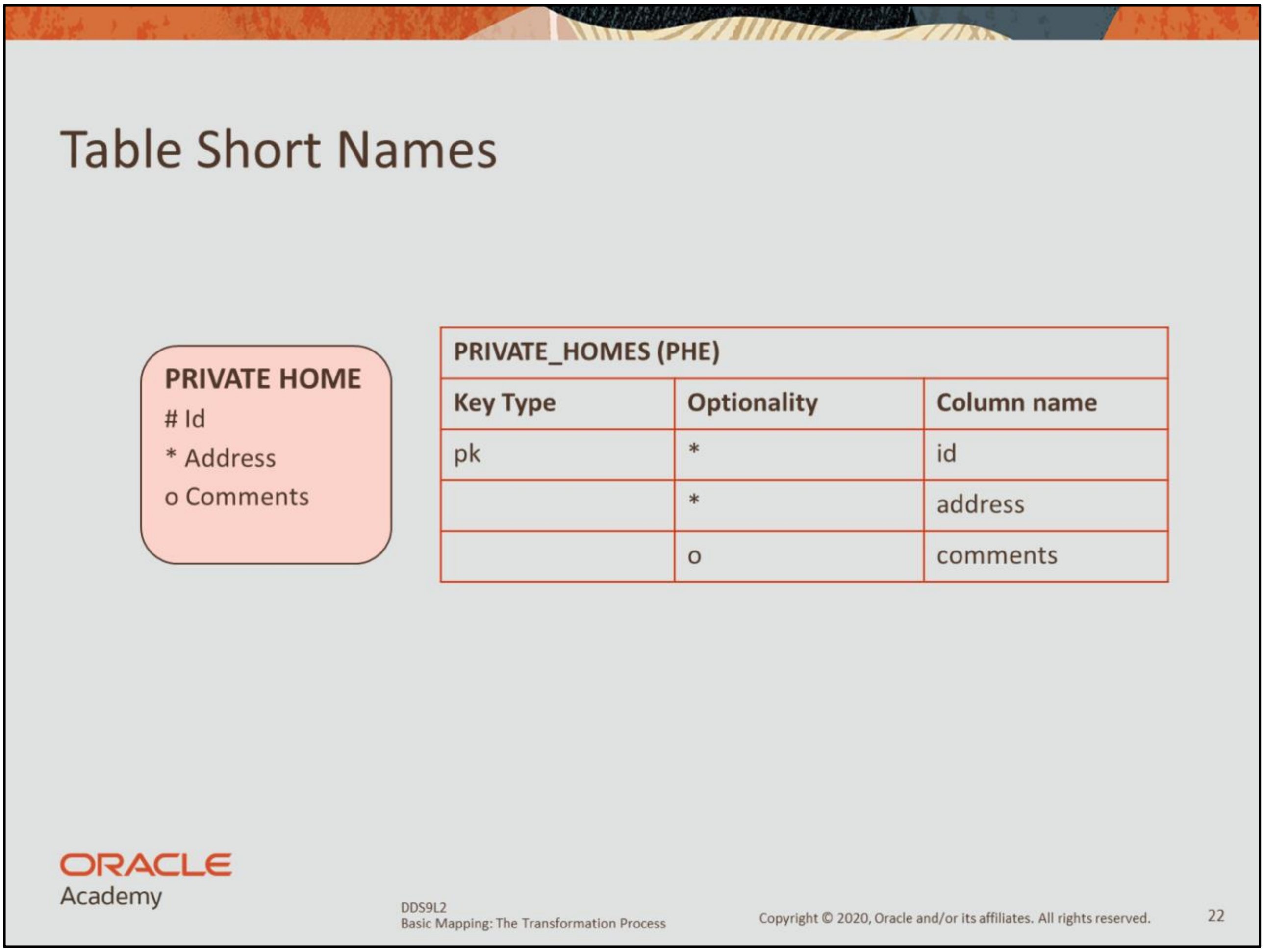


Table Short Names

PRIVATE HOME
Id
* Address
o Comments

PRIVATE_HOMES (PHE)		
Key Type	**Optionality**	**Column name**
pk	*	id
	*	address
	o	comments

ORACLE
Academy

DDS9L2
Basic Mapping: The Transformation Process

These rules do not guarantee uniqueness, but experience has proved that duplicated names are relatively rare. In the case of identical short names, just add a number to the one that is used less. Example: CTR and CTR1.

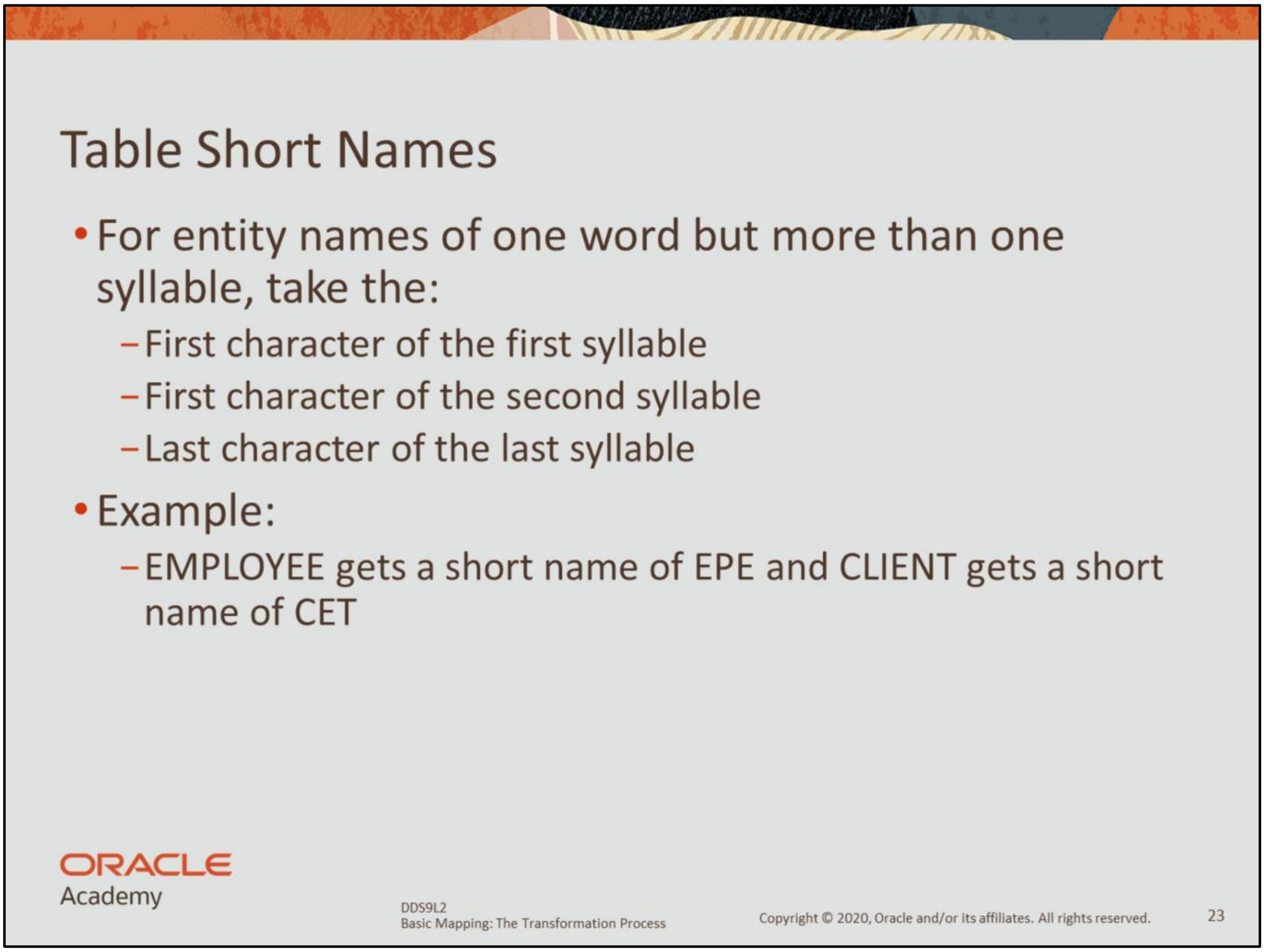
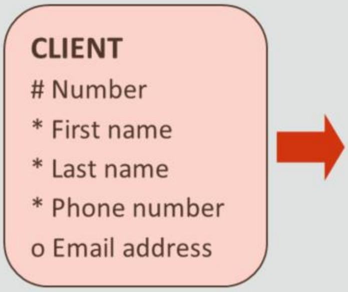


Table Short Names

- For entity names of one word but more than one syllable, take the:
  - First character of the first syllable
  - First character of the second syllable
  - Last character of the last syllable
- Example:
  - EMPLOYEE gets a short name of EPE and CLIENT gets a short name of CET

Table Short Names



CLIENT
Number
* First name
* Last name
* Phone number
o Email address

CLIENTS (CET)		
Key Type	**Optionality**	**Column name**
pk	*	client_num
	*	first_name
	*	last_name
	*	phone_num
	*	email_addr

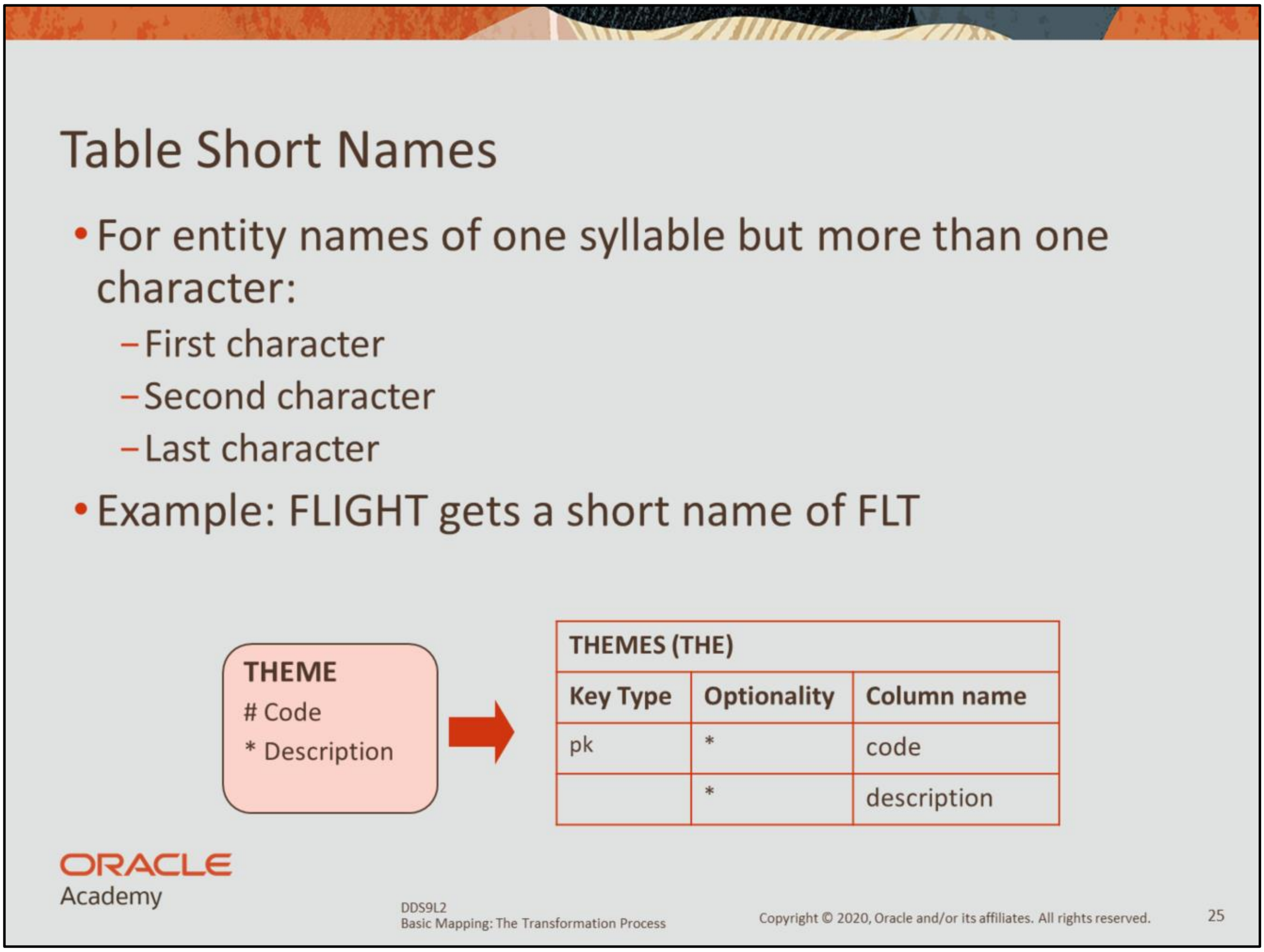
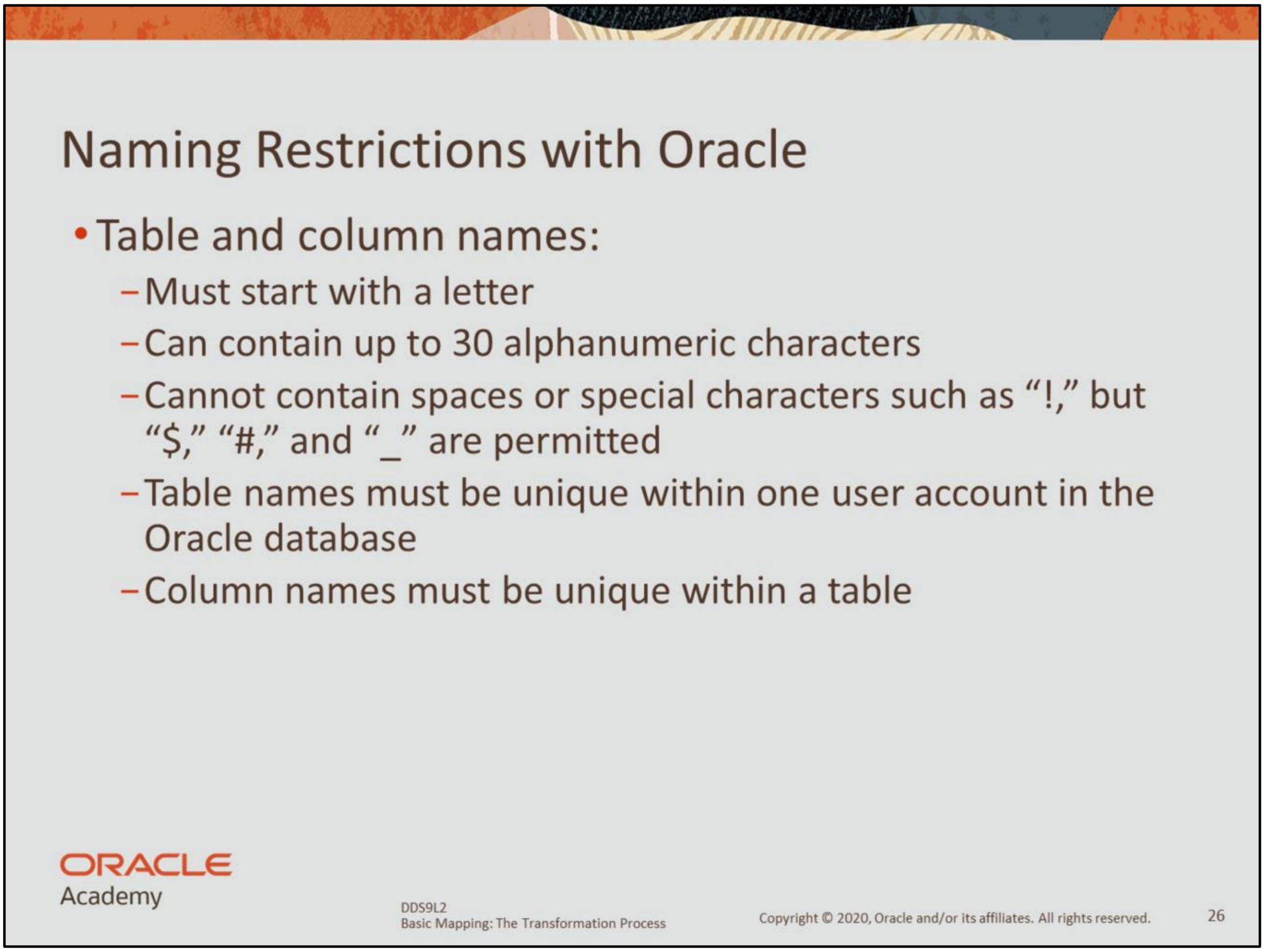


Table Short Names

- For entity names of one syllable but more than one character:
  - First character
  - Second character
  - Last character
- Example: FLIGHT gets a short name of FLT

THEMES (THE)		
Key Type	Optionality	Column name
pk	*	code
	*	description

ORACLE Academy

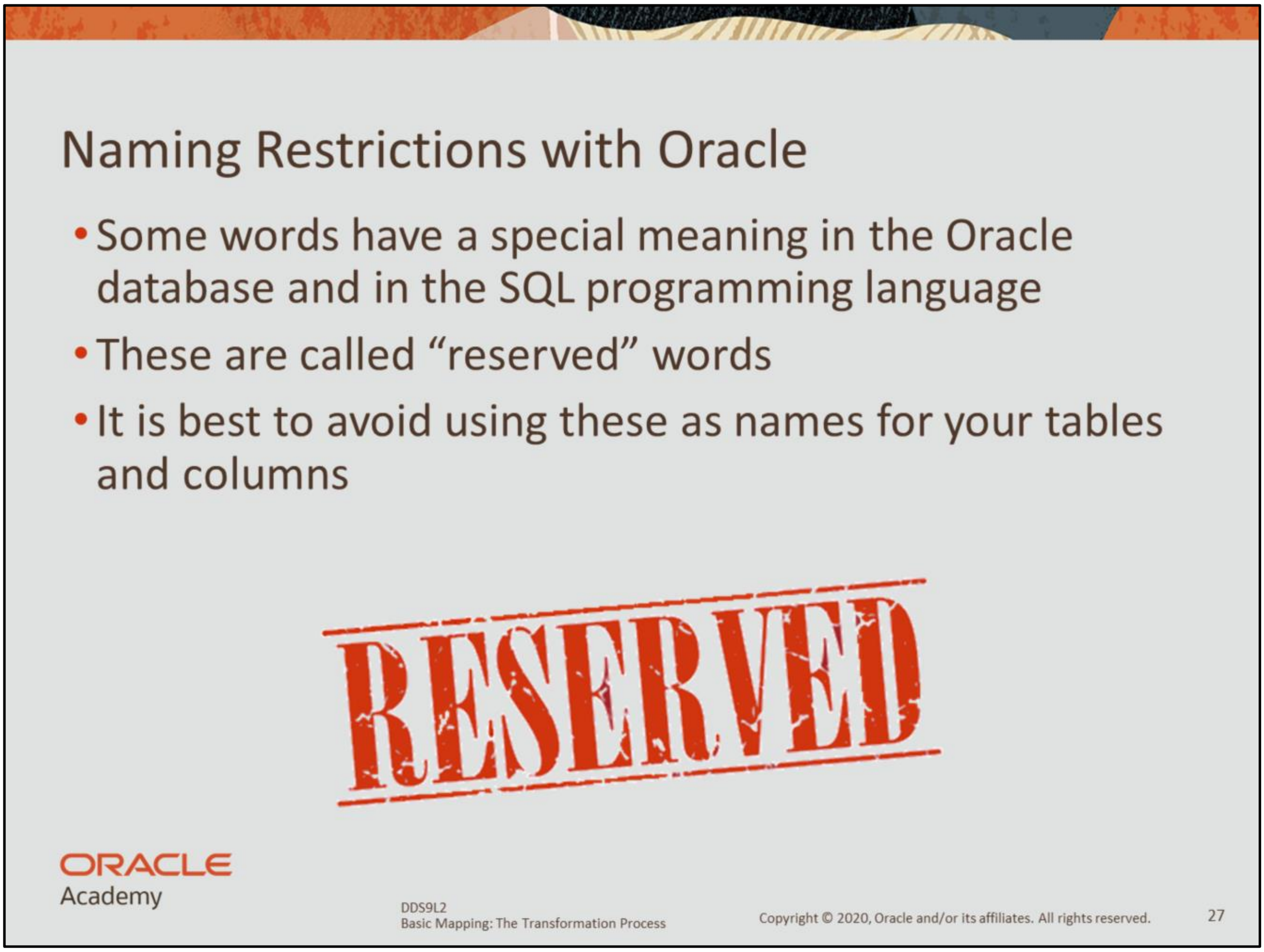


Naming Restrictions with Oracle

- Table and column names:
  - Must start with a letter
  - Can contain up to 30 alphanumeric characters
  - Cannot contain spaces or special characters such as “!,” but “$,” “#,” and “_” are permitted
  - Table names must be unique within one user account in the Oracle database
  - Column names must be unique within a table

ORACLE
Academy

Note that Oracle table and column names can contain underscores but not hyphens. For example, SALES_ORDERS is a valid table name but SALES-ORDERS is not.
All database systems make recommendations on naming objects (such as tables). If you do not use an Oracle database, you should still decide on a naming convention and make sure it is compatible with the database system that you have chosen.



Naming Restrictions with Oracle

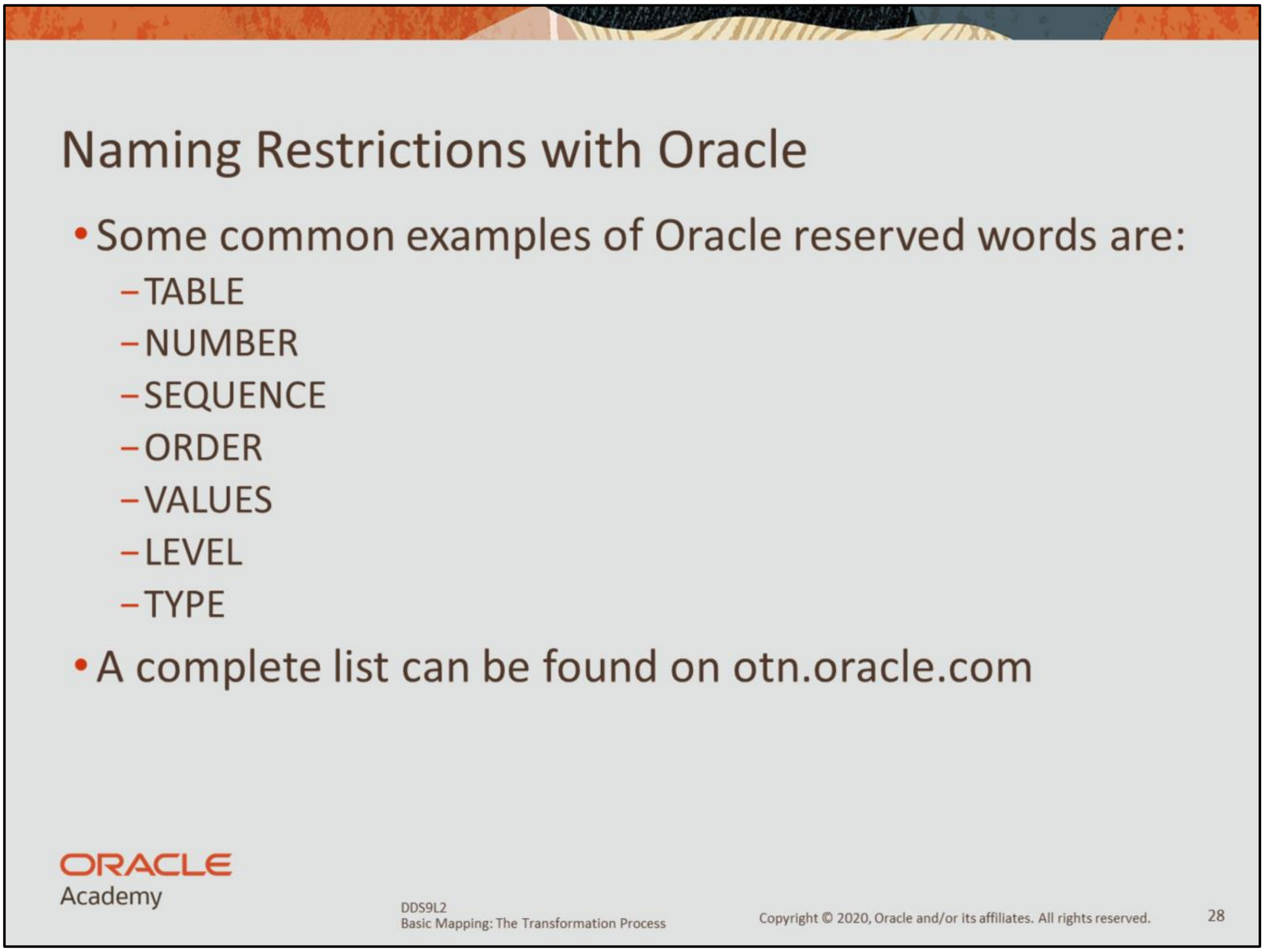
- Some words have a special meaning in the Oracle database and in the SQL programming language
- These are called “reserved” words
- It is best to avoid using these as names for your tables and columns

RESERVED

ORACLE
Academy

DDS9L2
Basic Mapping: The Transformation Process

The next slide gives some examples of reserved words.

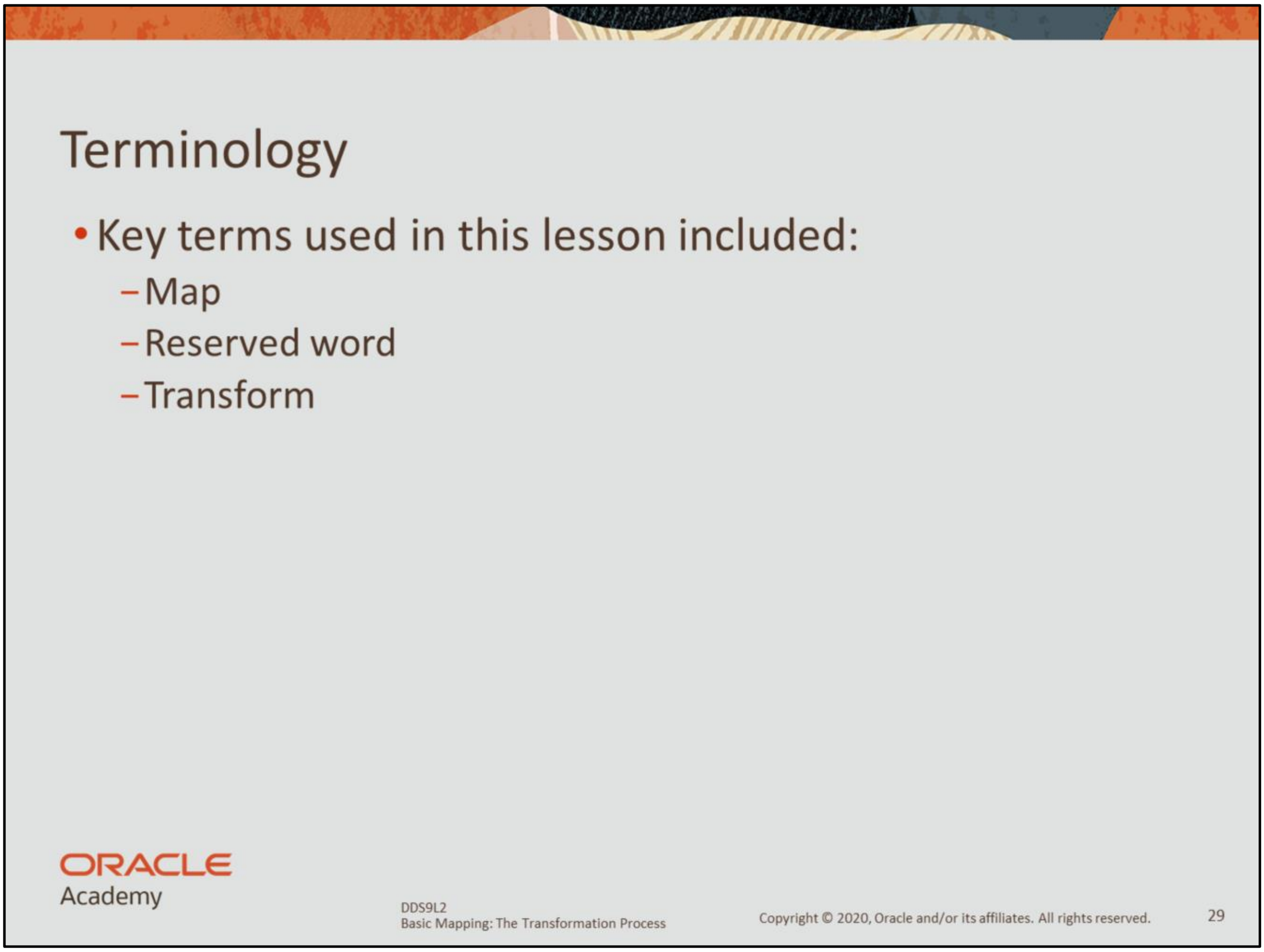


Naming Restrictions with Oracle

- Some common examples of Oracle reserved words are:
  - TABLE
  - NUMBER
  - SEQUENCE
  - ORDER
  - VALUES
  - LEVEL
  - TYPE
- A complete list can be found on otn.oracle.com

ORACLE
Academy

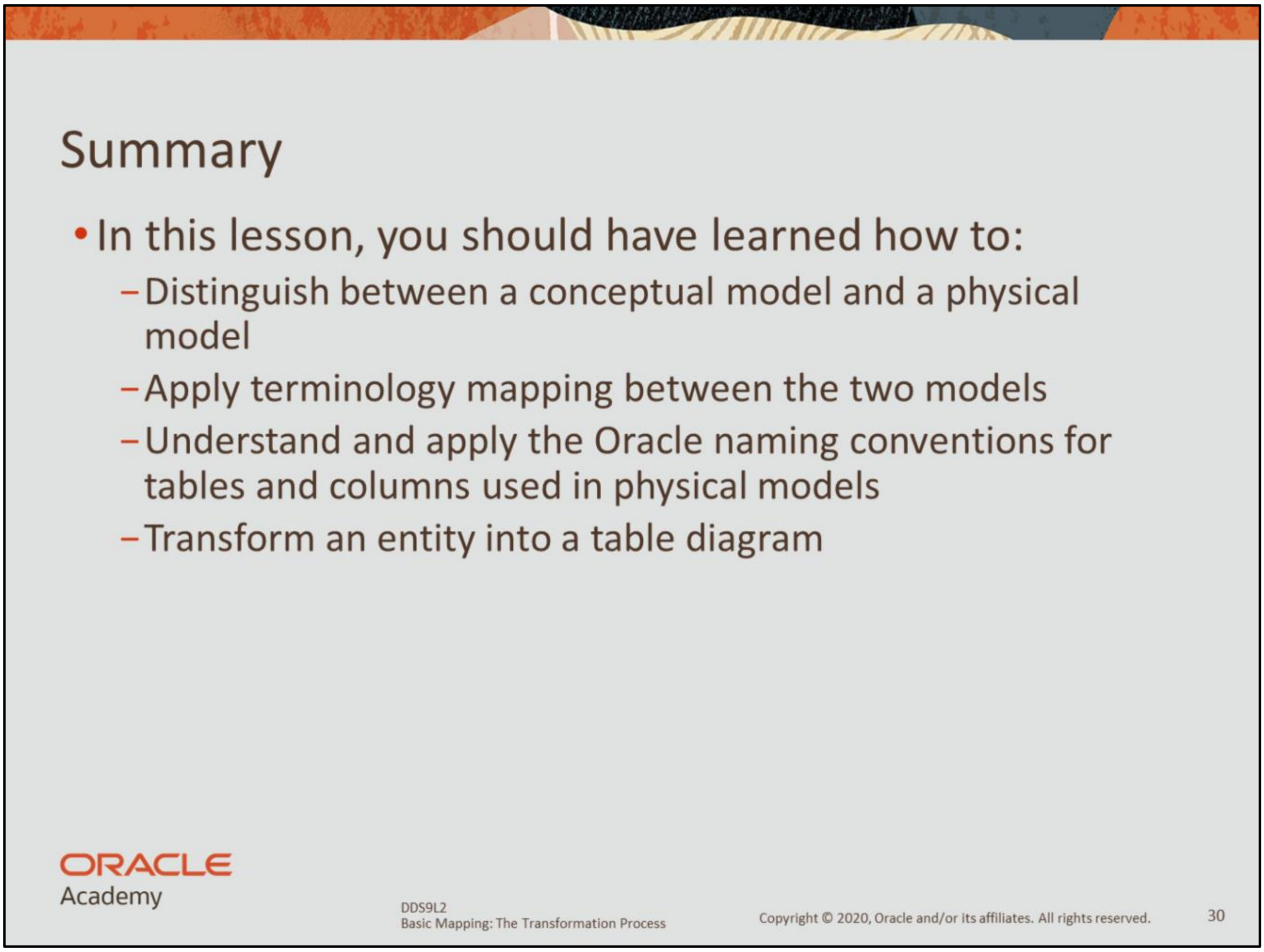
This site requires you to sign up, but it is free. It's a valuable source of technical information on all Oracle products.



Terminology

- Key terms used in this lesson included:
  - Map
  - Reserved word
  - Transform

ORACLE
Academy



Summary

- In this lesson, you should have learned how to:
  - Distinguish between a conceptual model and a physical model
  - Apply terminology mapping between the two models
  - Understand and apply the Oracle naming conventions for tables and columns used in physical models
  - Transform an entity into a table diagram

ORACLE Academy

ORACLE

Academy