Database Design

Basic Mapping: The Transformation Process

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

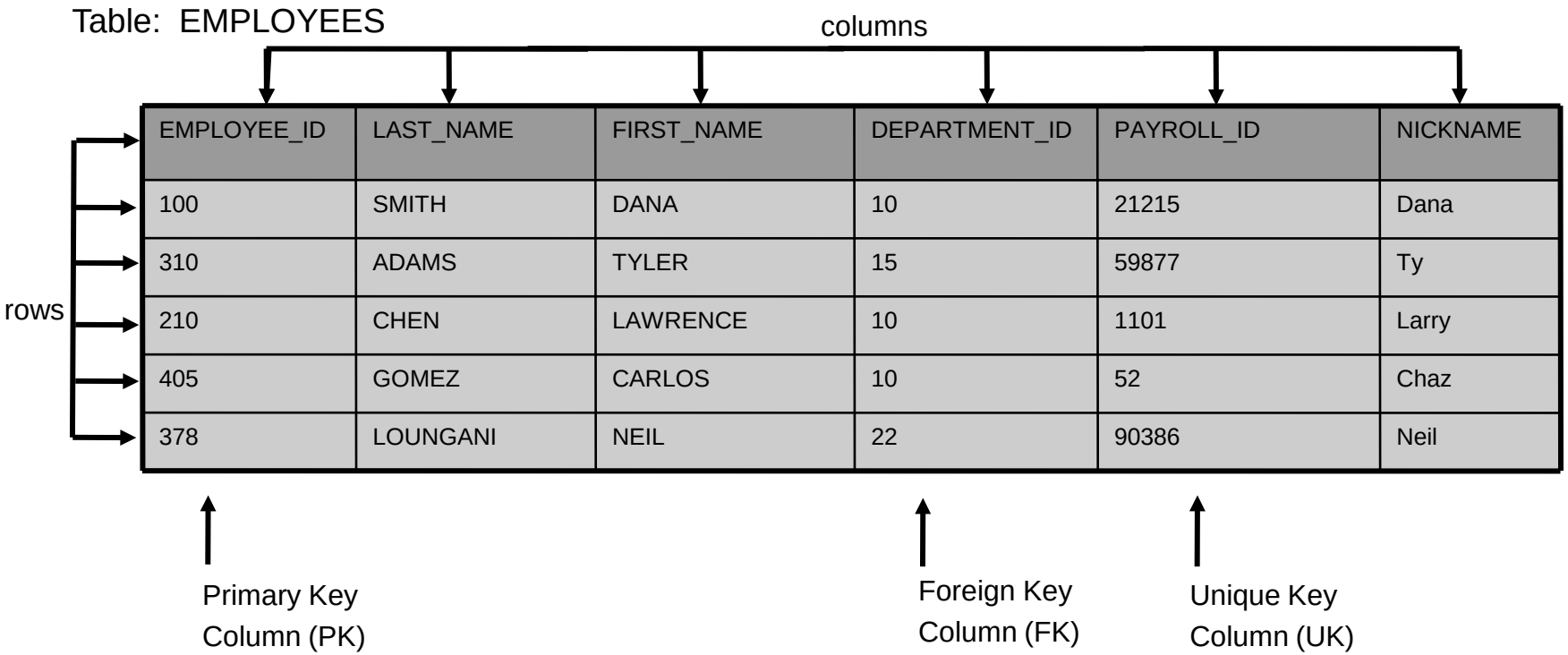
Purpose

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.

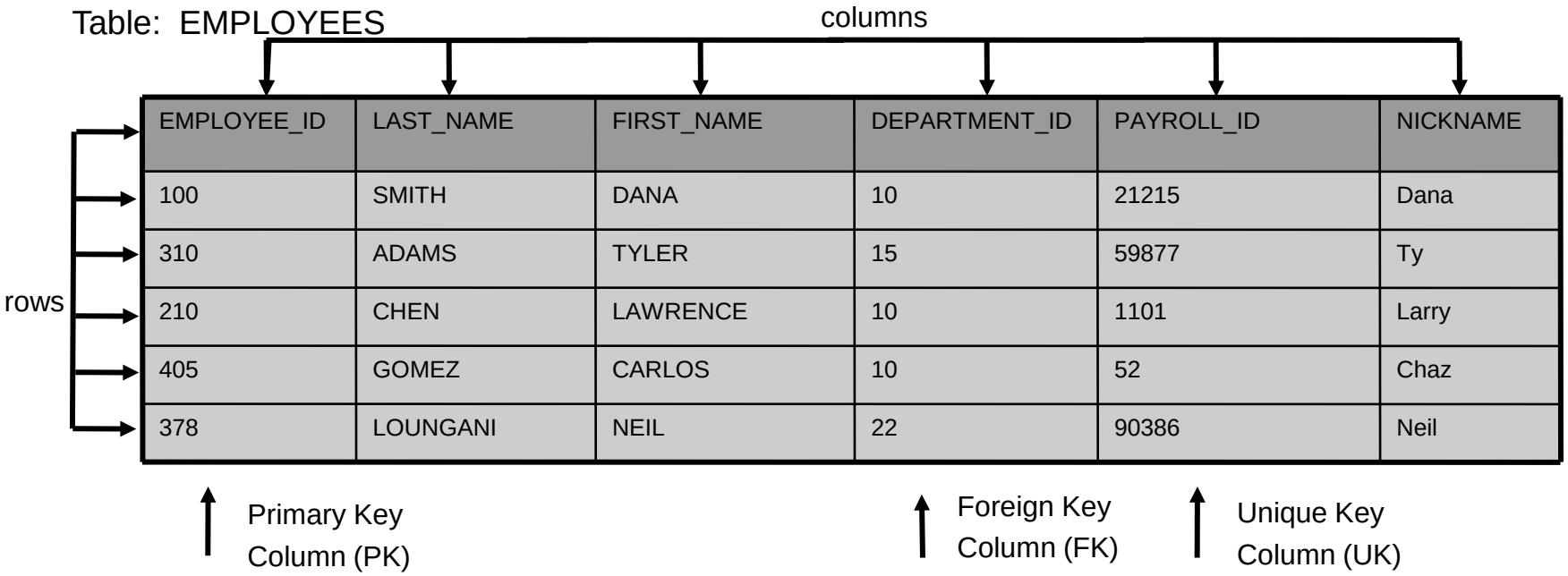
Review of Relational Tables

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.



Review of Relational Tables (cont.)

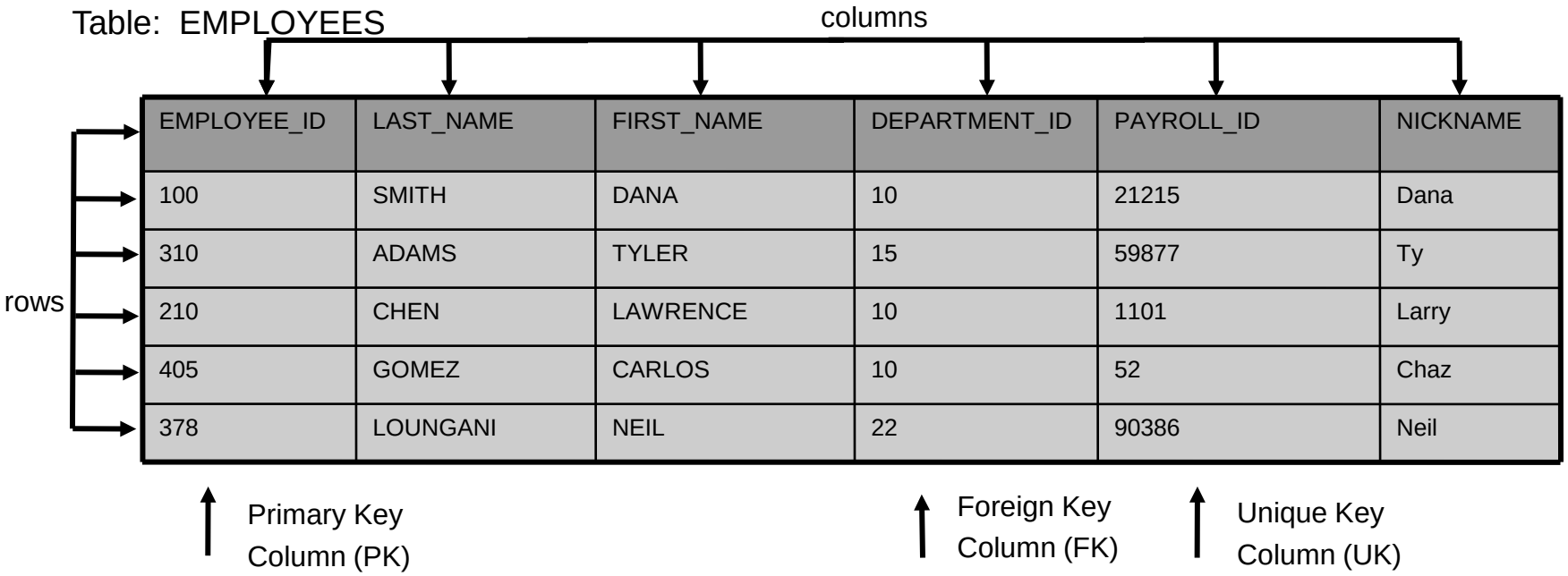
Tables have columns and rows. In the example, each row describes an occurrence of an employee. Each column is used to store a specific type of value, such as employee number, last name, and first name.



EMPLOYEE_ID	LAST_NAME	FIRST_NAME	DEPARTMENT_ID	PAYROLL_ID	NICKNAME
100	SMITH	DANA	10	21215	Dana
310	ADAMS	TYLER	15	59877	Ty
210	CHEN	LAWRENCE	10	1101	Larry
405	GOMEZ	CARLOS	10	52	Chaz
378	LOUNGANI	NEIL	22	90386	Neil

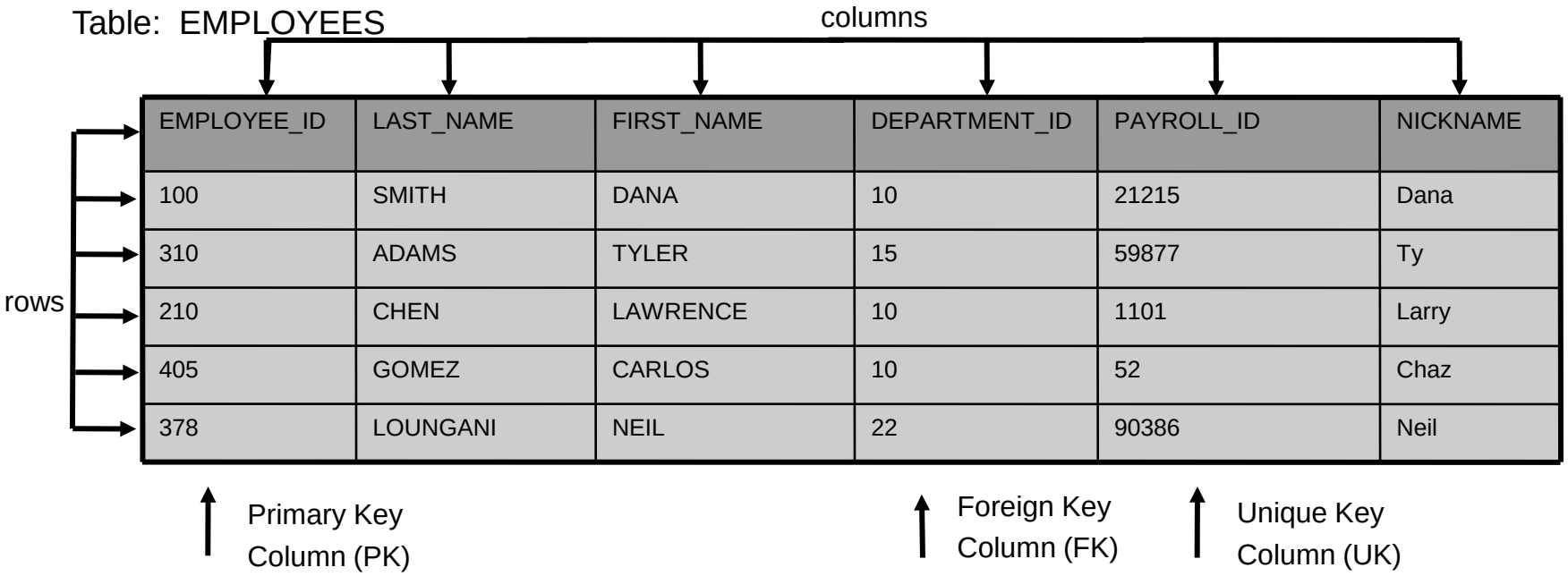
Review of Relational Tables (cont.)

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



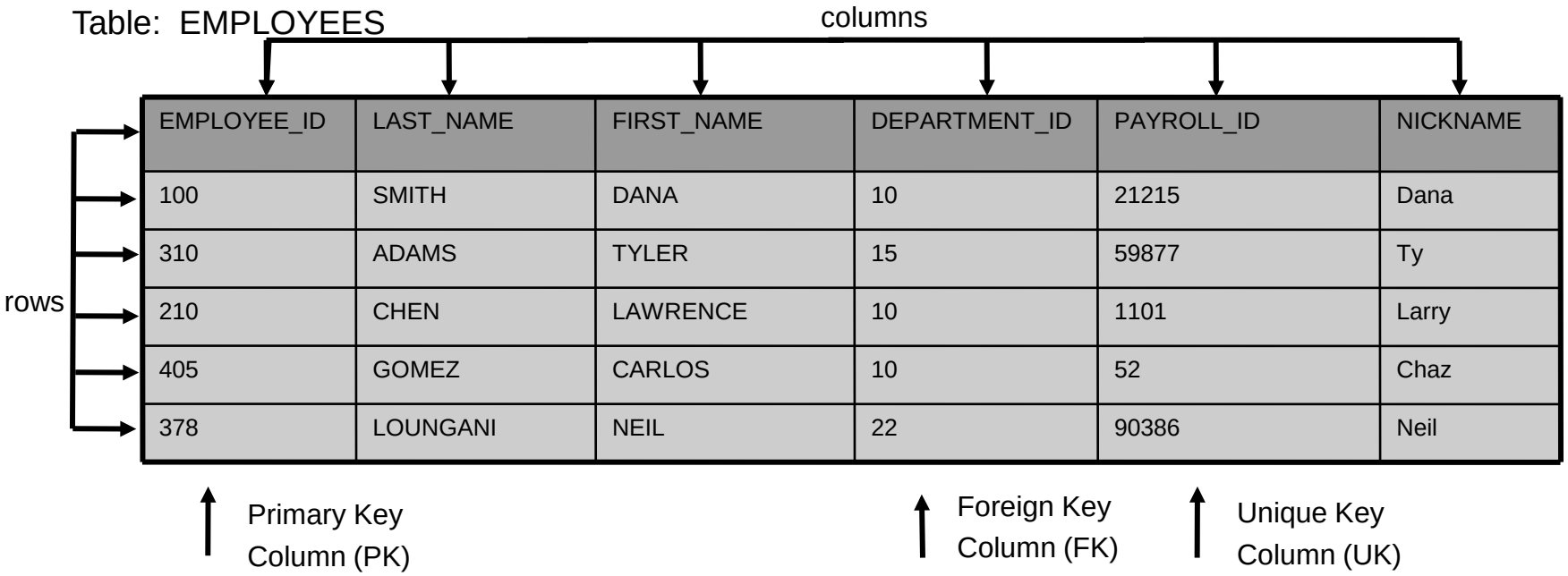
Review of Relational Tables (cont.)

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



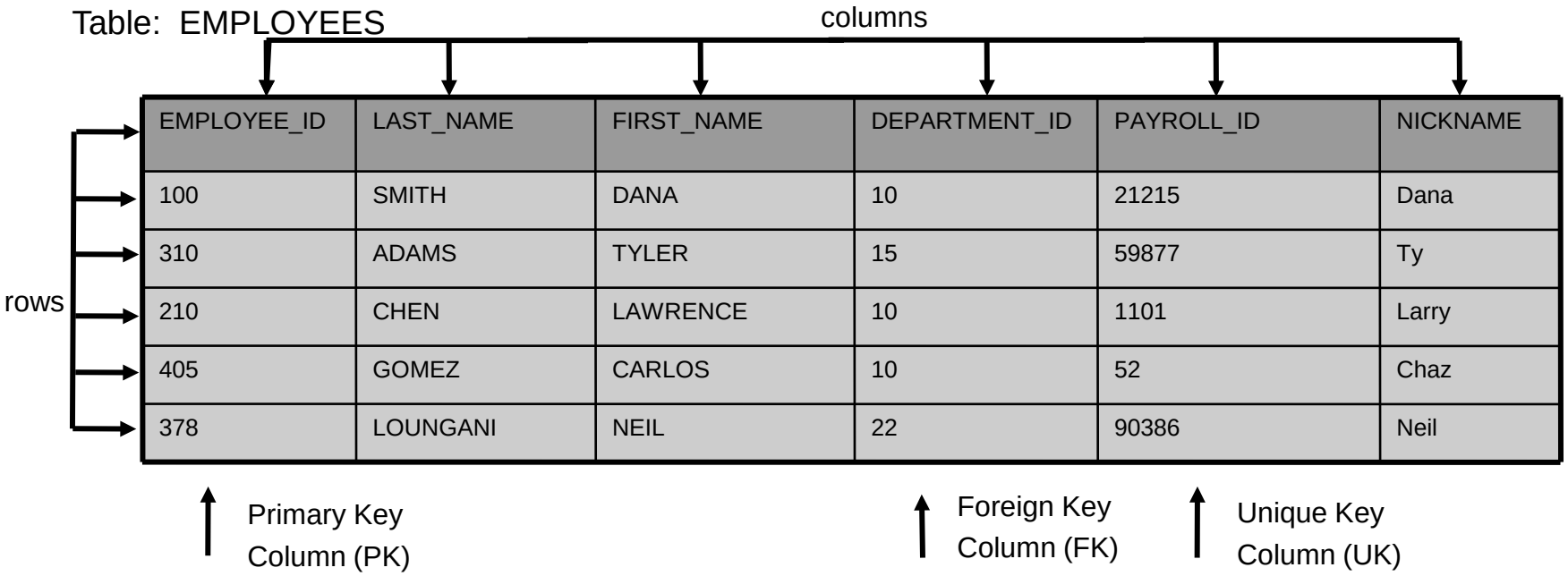
Review of Relational Tables (cont.)

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



Review of Relational Tables (cont.)

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.

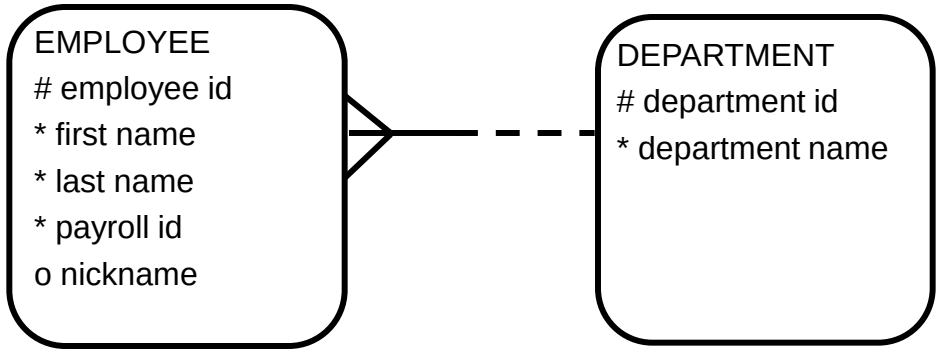


Transforming Conceptual To Physical

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

Transforming Conceptual To Physical (cont.)

Conceptual Model (ERD)



Transformation process

Physical Implementation: Relational Database

EMPLOYEES (EPE)

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

DEPARTMENTS (DPT)

Key type	Optionality	Column name
pk	*	department_id
	*	department_name

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.

Terminology Mapping (cont.)

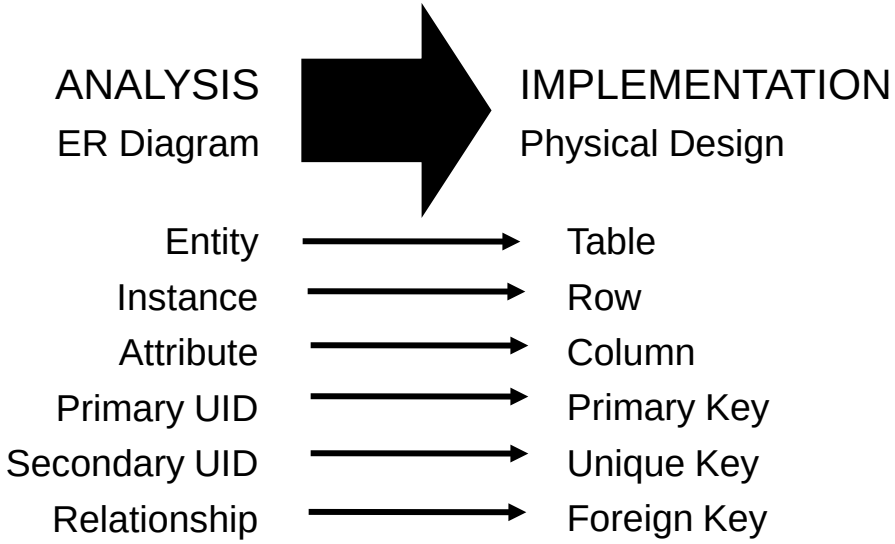


Table Diagram Notations

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. It will be blank if the column is not a part of any key.

TABLE NAME (short name)

Key Type (pk, uk, fk)	Optionality (“*”, “o”)	Column Name

Table Diagram Notations (cont.)

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.

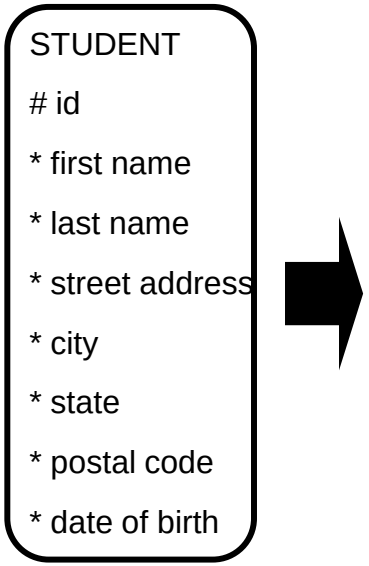
TABLE NAME (short name)

Key Type (pk, uk, fk)	Optionality (“*”, “o”)	Column Name

Naming Conventions for Tables and Columns

The table name is the plural of the entity name.

Example: STUDENT becomes STUDENTS

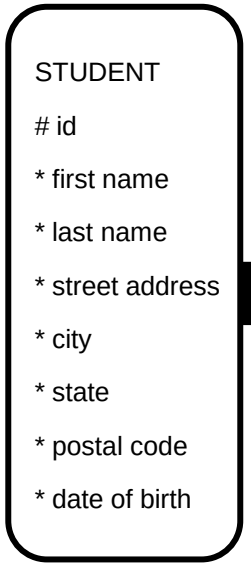


STUDENTS

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

Naming Conventions for Tables and Columns (cont.)

- Column names are identical to the attribute names except that special characters and spaces are replaced with underscores. Column names often use more abbreviations than attribute names.
- Example: first name becomes first_name, or fname



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

Table Short Names (cont.)

PRIVATE HOMES (PHE)

Key Type	Optionality	Column Name
pk	*	id
	*	address
	*	comments

Table Short Names (cont.)

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 (cont.)

CLIENT
number
* first name
* last name
* phone number
o email address

CLIENTS (CET)

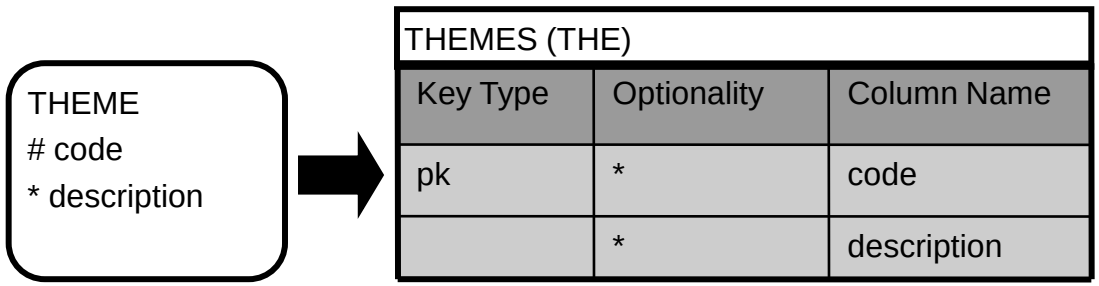
Key Type	Optionality	Column Name
pk	*	client_num
	*	first_name
	*	last_name
	*	phone_num
	o	email_addr

Table Short Names (cont.)

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



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.

Naming Restrictions with Oracle (cont.)

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.

Naming Restrictions with Oracle (cont.)

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.

Terminology

Key terms used in this lesson included:

- Map
- Reserved word
- Transform

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