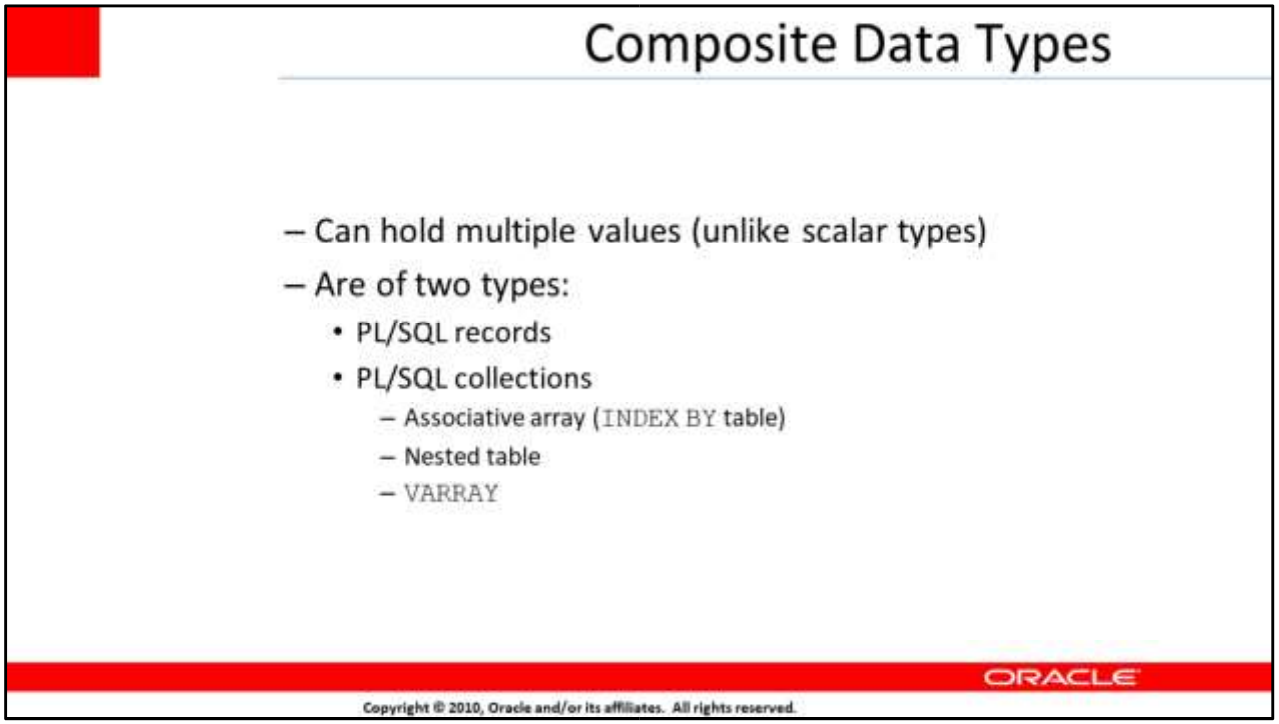


Composite Data Types

You learned that variables of the scalar data type can hold only one value, whereas a variable of the composite data type can hold multiple values of the scalar data type or the composite data type. There are two types of composite data types:

PL/SQL records: Records are used to treat related but dissimilar data as a logical unit. A PL/SQL record can have variables of different types. For example, you can define a record to hold employee details. This involves storing an employee number as NUMBER, a first name and last name as VARCHAR2, and so on. By creating a record to store employee details, you create a logical collective unit. This makes data access and manipulation easier.

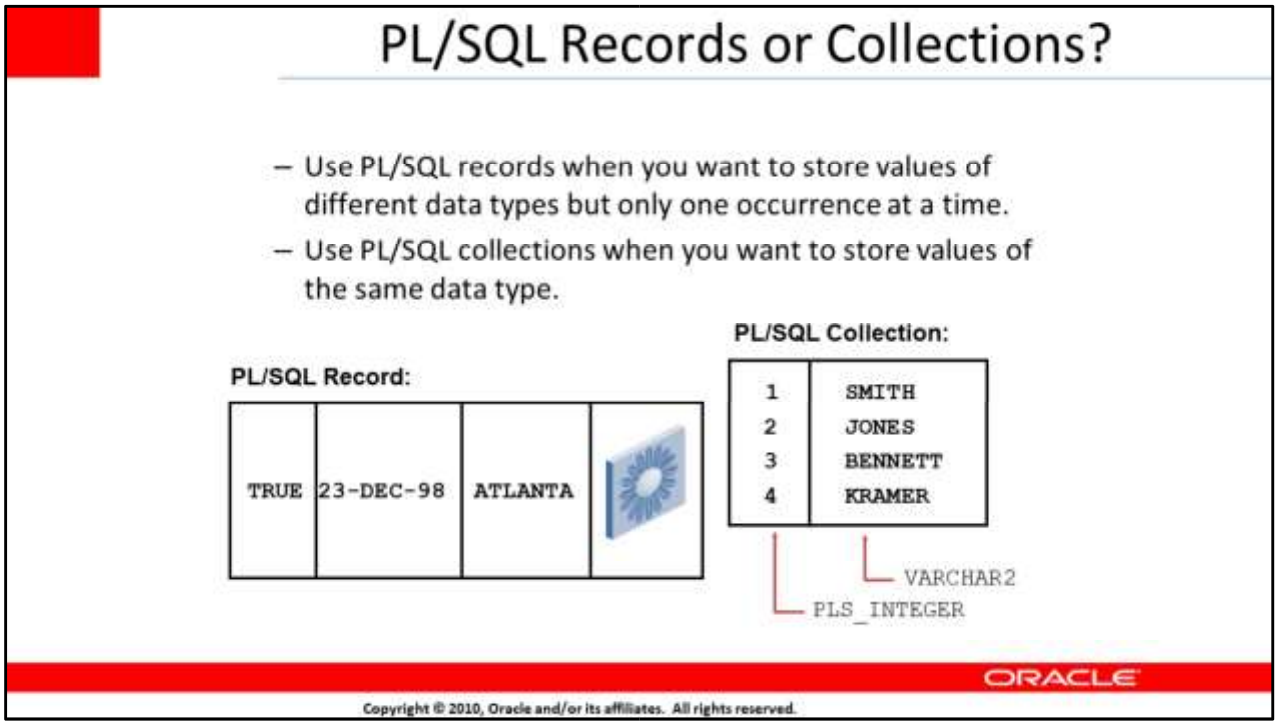
PL/SQL collections: Collections are used to treat data as a single unit. Collections are of three types:

Associative array
Nested table
VARRAY

Why Use Composite Data Types?

You have all the related data as a single unit. You can easily access and modify

data. Data is easier to manage, relate, and transport if it is composite. An analogy is having a single bag for all your laptop components rather than a separate bag for each component.



PL/SQL Records or Collections?

If both PL/SQL records and PL/SQL collections are composite types, how do you choose which one to use?

Use PL/SQL records when you want to store values of different data types that are logically related. For example, you can create a PL/SQL record to hold employee details and indicate that all the values stored are related because they provide information about a particular employee. Use PL/SQL collections when you want to store values of the same data type. Note that this data type can also be of the composite type (such as records). You can define a collection to hold the first names of all employees. You may have stored n names in the collection; however, name 1 is not related to name 2. The relation between these names is only that they are employee names. These collections are similar to arrays in programming languages such as C, C++, and Java.