

The logo for Oracle Academy. The word "ORACLE" is in a bold, orange, sans-serif font. Below it, the word "Academy" is in a smaller, dark gray, sans-serif font. The entire logo is centered on a light gray background, which is framed by dark gray horizontal bars at the top and bottom.

ORACLE

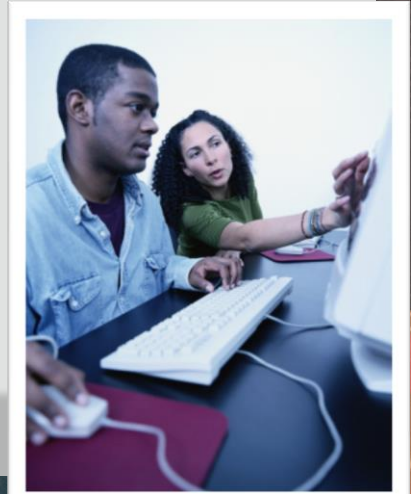
Academy

Database Programming with PL/SQL

2-3

Recognizing Data Types

ORACLE
Academy



Copyright © 2020, Oracle and/or its affiliates. All rights reserved.

Objectives

- This lesson covers the following objectives:
 - Define data type and explain why it is needed
 - List and describe categories of data types
 - Give examples of scalar and composite data types

Purpose

- As a result of the Database Programming with SQL course, you should be familiar with several data types that were used when defining the type of data stored in the different columns of a table (NUMBER, VARCHAR2, DATE, etc.)
- PL/SQL includes a variety of data types for use when defining variables, constants, and parameters
- As with table columns, these data types specify what type and size of data can be stored in a particular location

PL/SQL Data Types

- PL/SQL supports five categories of data type
- A data type specifies a storage format, constraints, and a valid range of values

Data Type	Description
Scalar	Holds a single value with no internal elements.
Composite	Contains multiple internal elements that can be manipulated individually.
Large Object (LOB)	Holds values called locators that specify the location of large objects (such as graphic images) that are stored out of line.
Reference	Holds values called pointers that point to a storage location.
Object	Is a schema object with a name, attributes, and methods. An object data type is similar to the class mechanism supported by C++ and Java.

Scalar, Composite, and LOB data types are covered in this course.

Before a variable, constant, or parameter can be referenced in a PL/SQL program, it must be defined using a data type in the declarative section of a PL/SQL program.

Scalar Data Types

- Scalar data types:
 - Hold a single value
 - Have no internal components
- Can be classified into four categories:
 - Character
 - Number
 - Date
 - Boolean
- Note : The BOOLEAN data type may be new to you, as it is not a column data type used in table definitions



Scalar Data Types

- The different data types specify:
 - What type and size of data can be stored in a particular location
 - The range of values the variable can have
 - The set of operations that can be applied to values of that type

Scalar Data Types

- The Character, Number, and Date categories include several data types
- SCALAR CHARACTER data types:
 - CHAR
 - VARCHAR2
 - LONG
- SCALAR NUMERIC data types:
 - NUMBER
 - PLS_INTEGER
- SCALAR DATE data types:
 - DATE
 - TIMESTAMP
 - TIMESTAMP WITH TIME ZONE

Note : For more information and the complete list of scalar data types, refer to the PL/SQL User's Guide and Reference.

Scalar Data Types: Character (or String)

- Character data types also are known as strings and allow storage of alphanumeric data (letters, numbers, and symbols)

Data Type	Description
CHAR [(maximum_length)]	Base type for fixed-length character data up to 32,767 characters. If you do not specify a maximum_length, the default length is set to 1.
VARCHAR2 (maximum_length)	Base type for variable-length character data up to 32,767 characters. VARCHAR2 is optimized for performance or efficiency, depending on the size.
LONG	Character data of variable length up to 2 gigabytes size.

- Ex. `v_first_name VARCHAR2(20) := 'Neena';`

Scalar Data Types: Number

- Number data types allow storage of integers, decimals, and a positive or negative indicator

Data Type	Description
NUMBER	Floating-point number from 1E-130 to 10E125.
NUMBER(p,s)	Fixed-point number with precision p. Precision includes scale s and can range from 1 to 38. Scale can range from -84 to 127 and determines where rounding occurs as well as the fixed number of decimal places to store.
NUMBER(p)	Integers with maximum number of digits p (range 1-38).
PLS_INTEGER	Requires less storage and is faster than NUMBER.

- EX. `v_salary NUMBER(8,2) := 9999.99;`

Scalar Data Types: Date

- Date data types provide storage of dates and times

Data Type	Description
DATE	Base type for dates and times. DATE values include the time of day in seconds since midnight. The range for dates is between 1-Jan-4712 BCE and 31-Dec-9999 CE.
TIMESTAMP	TIMESTAMP extends the DATE data type to store the year, month, day, hour, minute, second, and fraction of seconds.
TIMESTAMP WITH TIME ZONE	TIMESTAMP WITH TIME ZONE extends the TIMESTAMP data type to include a time-zone displacement—that is, the difference (in hours and minutes) between local time and Coordinated Universal Time (UTC).

- Ex. `v_hire_date DATE := '15-Apr-2015';`

TIMESTAMP WITH LOCAL TIME ZONE - This data type differs from **TIMESTAMP WITH TIME ZONE** in that when you insert a value into a database column, the value is normalized to the database time zone, and the time-zone displacement is not stored in the column. When you retrieve the value, the Oracle server returns the value in your local session time zone.

INTERVAL YEAR TO MONTH - Use the **INTERVAL YEAR TO MONTH** data type to store and manipulate intervals of years and months.

INTERVAL DAY TO SECOND - Use the **INTERVAL DAY TO SECOND** data type to store and manipulate intervals of days, hours, minutes, and seconds.

Scalar Data Types: Boolean

- Use the BOOLEAN data type to store the logical values TRUE, FALSE, and NULL
- Only logic operations are allowed on BOOLEAN variables
- Column values cannot be fetched into a BOOLEAN variable and a table column cannot be defined with a BOOLEAN data type
- EX. `v_control BOOLEAN := TRUE;`
- The BOOLEAN data type is available in PL/SQL, but is not valid in SQL



Composite Data Types

- Composite data types have internal components, sometimes called elements, that can be manipulated individually
- It may be helpful to think of a scalar type as being like a single column value in a table, while a record data type is like a whole row of a table
- Composite data types include the following:
 - RECORD
 - TABLE
 - VARRAY

RECORD and TABLE data types are covered later in this course.

Record Composite Data Type

- A composite variable that contains internal components that match the data structure of the EMPLOYEES table can be created using:

```
v_emp_record employees%ROWTYPE;
```

- The internal elements can be referenced by prefixing the column-name with the record-name:

```
v_emp_record.first_name
```

LOB (Large Object) Data Type

- LOB data types allow you to store blocks of unstructured data (such as text, graphic images, video, or audio) up to 4 gigabytes in size
- A database column can be a LOB data type
- There are four LOB data types:
 - Character large object (CLOB)
 - Binary large object (BLOB)
 - Binary file (BFILE)
 - National language character large object (NCLOB)

LOB Data Type

- LOB data types store locators, which point to large objects stored in an external file
- LOB data types allow efficient, random, piece-wise access to the data
- CLOB, BLOB, and NCLOB data is stored in the database, either inside or outside of the row
- BFILE data is stored in operating system files outside the database.

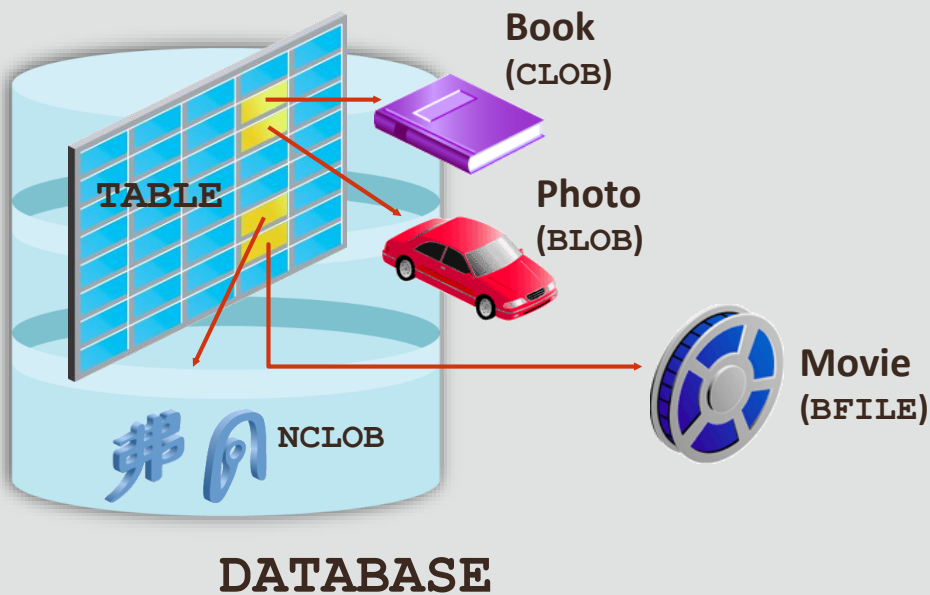
The character large object (CLOB) data type is used to store large blocks of text data in the database.

The binary large object (BLOB) data type is used to store large binary objects (ex. graphic image) in the database. When you insert or retrieve such data to and from the database, the database does not interpret the data. External applications that use this data must interpret the data.

The national language character large object (NCLOB) data type is used to store large blocks of NCHAR data in the database.

The binary file (BFILE) data type is used to store large binary objects (ex. a video or audio file) in operating system files outside the database.

LOB Data Type



ORACLE
Academy

PLSQL 2-3
Recognizing Data Types

Copyright © 2020, Oracle and/or its affiliates. All rights reserved.

17

BLOB, CLOB, and NCLOB objects can be stored in-line (inside the row) or out-of-line (outside the row, but still in the database, as indicated in the image).

BFILE objects are stored outside the database in operating system files.

Terminology

- Key terms used in this lesson included:
 - BFILE
 - BLOB
 - CLOB
 - Composite
 - LOB

- BFILE – Store large binary objects outside of the database.
- BLOB – Store large binary objects in the database.
- CLOB – Store large blocks of character data in the database.
- Composite – Contain internal elements that that can be manipulated individually.
- LOB – Data types that hold values called locators, which specify the location of large objects (such as graphic images).

Terminology

- Key terms used in this lesson included:
 - NCLOB
 - Object
 - Reference
 - Scalar

- NCLOB (National Language Character Large Object) – Store large blocks of NCHAR data in the database.
- Object – A schema object with a name, attributes, and methods.
- Reference – Hold values, called pointers, which point to a storage location.
- Scalar – Hold a single value with no internal components.

Summary

- In this lesson, you should have learned how to:
 - Define data type and explain why it is needed
 - List and describe categories of data types
 - Give examples of scalar and composite data types

The Oracle Academy logo is centered on a light gray background. It features the word "ORACLE" in a bold, orange, sans-serif font, with the letters "R" and "A" having a unique, slanted design. Below "ORACLE" is the word "Academy" in a dark gray, sans-serif font. The entire logo is framed by two horizontal dark gray bars, one at the top and one at the bottom.

ORACLE

Academy