

Database Programming

Using Data Types

Objectives

In this lesson, you will learn to:

- Create a table using `TIMESTAMP` and `TIMESTAMP WITH TIME ZONE` column data types
- Create a table using `INTERVAL YEAR TO MONTH` and `INTERVAL DAY TO SECOND` column data types
- Give examples of organizations and personal situations where it is important to know to which time zone a date-time value refers

Purpose

If you ever travel to another country, you'll quickly find out that the money in your pocket may not be that of the local currency. If you want to buy something, it will be necessary to convert your money into the currency of the local country.

Purpose (cont.)

This conversion process is a lot like dealing with data types in SQL. Different types of data have different types of characteristics, the purpose of which is to efficiently store data. In this lesson, you will learn more about data types and their uses.

Data Type Overview

Each value manipulated by Oracle has a data type. A value's data type associates a fixed set of properties with the value. These properties cause the database to treat values of one data type differently from values of another data type.

Data Type Overview (cont.)

Different data types offer several advantages:

- Columns of a single type produce consistent results. For example, DATE data type columns always produce date values.
- You cannot insert the wrong type of data into a column. For example, columns of data type DATE will prevent NUMBER type data from being inserted.

For these reasons, each column in a relational database can hold only one type of data. You cannot mix data types within a column.

Common Data Types

The most commonly used column data types are:

- For character values: CHAR (fixed size, maximum 2000 characters); VARCHAR2 (variable size, maximum 4000 characters); CLOB (variable size, maximum 128 terabytes)
- For number values: NUMBER (variable size, maximum precision 38 digits)

Common Data Types (cont.)

The most commonly used column data types are:

- For date and time values: DATE, TIMESTAMP, INTERVAL
- For binary values (eg. multimedia: JPG, WAV, MP3, and so on): RAW (variable size, maximum 2000 bytes); BLOB (variable size, maximum 128 terabytes).

Common Data Types (cont.)

The most commonly used column data types are:

- For character values, it is usually better to use VARCHAR2 or CLOB than CHAR, because it saves space.
- For example, an employee's last name is 'Chang'. In a VARCHAR2(30) column, only the 5 significant characters are stored: C h a n g. But in a CHAR(30) column, 25 trailing spaces would be stored as well, to make a fixed size of 30 characters.

Common Data Types (cont.)

The most commonly used column data types are:

- Number values can be negative as well as positive. For example, `NUMBER(6,2)` can store any value from +9999.99 down to -9999.99.

DATE-TIME Data Types

The DATE data type stores a value of centuries down to whole seconds, but cannot store fractions of a second. '21/Aug/2003 17:25:30' is a valid value, but '21/Aug/2003 17:25:30.255' is not.

The TIMESTAMP data type is an extension of the DATE data type which allows fractions of a second.

For example, TIMESTAMP(3) allows 3 digits after the whole seconds, allowing values down to milliseconds to be stored.

TIMESTAMP...With [LOCAL] Time Zone

Think about the time value '17:30'. Of course it means "half past five in the afternoon". But in which time zone? Is it half past five New York City time or Beijing time or Istanbul time or ?

In today's globalized organizations which operate in many different countries, it is important to know to which time zone a date-time value refers.

TIMESTAMP...With [LOCAL] Time Zone (cont.)

TIMESTAMP WITH TIME ZONE stores a time zone value as a displacement from Universal Coordinated Time or UCT (previously known as Greenwich Mean Time or GMT).

A value of '21/Aug/2003 08:00:00 -5:00' means 8:00 am 5 hours behind UTC.

This is US Eastern Standard Time (EST).

TIMESTAMP...With [LOCAL] Time Zone (cont.)

TIMESTAMP WITH LOCAL TIME ZONE is the same, but with one difference: when this column is SELECTed in a SQL statement, the time is automatically converted to the selecting user's time zone.

TIMESTAMP With...Time Zone Example

```
CREATE TABLE time_example  
(first_column TIMESTAMP WITH TIME ZONE,  
second_column TIMESTAMP WITH LOCAL TIME ZONE);
```

```
INSERT INTO time_example (first_column, second_column)  
VALUES  
( '15/Nov/2007 08:00:00 AM -05:00', '15/Nov/2007 08:00:00');
```

TIMESTAMP...With Time Zone Example (cont.)

Both values are stored with a time displacement of –05:00 hours (EST).

But now a user in Istanbul executes:

```
SELECT *  
FROM time_example;
```

FIRST_COLUMN

SECOND_COLUMN

-----	-----
15/Nov/2007 08:00:00 AM -05:00	15/Nov/2007 15:00:00

Istanbul time is 7 hours ahead of EST; when it's 8am in New York City, it's 3pm in Istanbul.

INTERVAL Data Types

These store the elapsed time, or interval of time, between two date-time values.

- **INTERVAL YEAR TO MONTH** stores a period of time measured in years and months.
- **INTERVAL DAY TO SECOND** stores a period of time measured in days, hours, minutes, and seconds.

```
CREATE TABLE time_example1  
(loan_duration    INTERVAL YEAR(3) TO MONTH,  
 day_duration     INTERVAL DAY(3) TO SECOND);
```

INTERVAL YEAR...TO MONTH

Syntax:

```
INTERVAL YEAR [(year_precision)] TO MONTH
```

year_precision is the maximum number of digits in the YEAR element. The default value of year_precision is 2.

INTERVAL YEAR...TO MONTH (cont.)

This example shows an interval of 120 months:

```
CREATE TABLE time_example2  
(loan_duration INTERVAL YEAR(3) TO MONTH);
```

```
INSERT INTO time_example2 (loan_duration)  
VALUES (INTERVAL '120' MONTH(3));
```

```
SELECT TO_CHAR (sysdate+loan_duration, 'dd-mon-yyyy')  
FROM time_example2;  
-- assume today's date is 26-Sep-2005
```

TO_CHAR(SYSDATE+LOAN_DURATION, 'DD-MON-YYYY')
26-SEP-2015

INTERVAL DAY...TO SECOND

Use this when you need a more precise difference between two date-time values.

Syntax:

```
INTERVAL DAY [day_precision)] TO SECOND [(fractional_seconds_precision)]
```

day_precision is the maximum number of digits in the DAY element. The default value of day_precision is 2.

fractional_seconds_precision is the number of digits in the fractional part of the SECOND date-time field. The default is 6.

INTERVAL DAY...TO SECOND (cont.)

This example shows an interval of 25 days:

```
CREATE TABLE time_example3  
(day_duration INTERVAL DAY(3) TO SECOND);
```

```
INSERT INTO time_example3 (day_duration)  
VALUES (INTERVAL '25' DAY(2));
```

```
SELECT TO_CHAR(sysdate + day_duration, 'DD-MON-YY') "25 Days"  
FROM time_example3;
```

25 Days
31-OCT-07

Terminology

Key terms used in this lesson included:

- CLOB
- BLOB
- TIMESTAMP
- TIMESTAMP WITH TIMEZONE
- TIMESTAMP WITH LOCAL TIMEZONE
- INTERVAL DAY TO SECOND
- INTERVAL DAY TO MONTH

Summary

In this lesson you have learned to:

- Create a table using `TIMESTAMP` and `TIMESTAMP WITH TIME ZONE` column data types
- Create a table using `INTERVAL YEAR TO MONTH` and `INTERVAL DAY TO SECOND` column data types
- Give examples of organizations and personal situations where it is important to know to which time zone a date-time value refers