Database Programming with SQL

13-2
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

• List and provide an example of each of the number, date, and character data types
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.

• 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

• 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 for character and number values are below.

• 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

- The most commonly used column data types for date, time, and binary values are below.

- 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

• 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.

• 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.
DATE-TIME Data Types

• TIMESTAMP example:

CREATE TABLE time_ex1
  ( exact_time TIMESTAMP);

INSERT INTO time_ex1
VALUES ('10/Jun/2015 10:52:29.123456');

INSERT INTO time_ex1
VALUES (SYSDATE);

INSERT INTO time_ex1
VALUES (SYSTIMESTAMP);

SELECT *
FROM time_ex1;

<table>
<thead>
<tr>
<th>EXACT_TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>10-JUN-15 10.52.29.123456 AM</td>
</tr>
<tr>
<td>16-JUL-15 08.17.08.000000 AM</td>
</tr>
<tr>
<td>16-JUL-15 08.17.16.610293 AM</td>
</tr>
</tbody>
</table>
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

• 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

• TIMESTAMP WITH TIME ZONE example:

```sql
CREATE TABLE time_ex2
( time_with_offset TIMESTAMP WITH TIME ZONE);

INSERT INTO time_ex2
VALUES (SYSTIMESTAMP);

INSERT INTO time_ex2
VALUES ('10/Jun/2015 10:52:29.123456 AM +2:00');

SELECT *
FROM time_ex2;
```

<table>
<thead>
<tr>
<th>TIME_WITH_OFFSET</th>
</tr>
</thead>
<tbody>
<tr>
<td>16-JUL-15 08.49.47.126056 AM -07:00</td>
</tr>
<tr>
<td>10-JUN-15 10.52.29.123456 AM +02:00</td>
</tr>
</tbody>
</table>
TIMESTAMP...With [LOCAL] Time Zone

• TIMESTAMP WITH LOCAL TIME ZONE is similar, 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:

```sql
CREATE TABLE time_ex3
( first_column TIMESTAMP WITH TIME ZONE,
  second_column TIMESTAMP WITH LOCAL TIME ZONE);

INSERT INTO time_ex3
  (first_column, second_column)
VALUES
  ('15/Jul/2015 08:00:00 AM -07:00', '15/Nov/2007 08:00:00');
```
TIMESTAMP...With Time Zone Example

• Both values are stored with a time displacement of –07:00 hours (PST).

• But now a user in Istanbul executes:

```
SELECT *
FROM time_ex3;
```

<table>
<thead>
<tr>
<th>FIRST_COLUMN</th>
<th>SECOND_COLUMN</th>
</tr>
</thead>
<tbody>
<tr>
<td>15-JUL-15 08.00.00.000000 AM -07:00</td>
<td>15-NOV-07 05.00.00.000000 PM</td>
</tr>
</tbody>
</table>

• Istanbul time is 9 hours ahead of PST; when it's 8am in Los Angeles, it's 5pm 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.
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

• This example shows INTERVAL YEAR TO MONTH:

```
CREATE TABLE time_ex4
(loan_duration1 INTERVAL YEAR(3) TO MONTH,
 loan_duration2 INTERVAL YEAR(2) TO MONTH);

INSERT INTO time_ex4 (loan_duration1, loan_duration2)
VALUES (INTERVAL '120' MONTH(3),
        INTERVAL '3-6' YEAR TO MONTH);
```

Assume today's date is: 17/Jul/2015

```
SELECT SYSDATE + loan_duration1 AS "120 months from now",
       SYSDATE + loan_duration2 AS "3 years 6 months from now"
FROM time_ex4;
```

<table>
<thead>
<tr>
<th>120 months from now</th>
<th>3 years 6 months from now</th>
</tr>
</thead>
<tbody>
<tr>
<td>17/Jul/2025</td>
<td>17/Jan/2019</td>
</tr>
</tbody>
</table>
INTERVAL DAY...TO SECOND

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

• Syntax:

\[
\text{INTERVAL \ DAY \ [day\_precision]) \ TO \ SECOND \ [(fractional\_seconds\_precision)]}
\]

• 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

- This example shows interval DAY TO SECOND:

```sql
CREATE TABLE time_ex5
(day_duration1 INTERVAL DAY(3) TO SECOND,
 day_duration2 INTERVAL DAY(3) TO SECOND);

INSERT INTO time_ex5 (day_duration1, day_duration2)
VALUES (INTERVAL '25' DAY(2), INTERVAL '4 10:30:10' DAY TO SECOND);

SELECT SYSDATE + day_duration1 AS "25 Days from now",
 TO_CHAR(SYSDATE + day_duration2, 'dd/Mon/yyyy hh:mi:ss')
AS "precise days and time from now"
FROM time_ex5;
```

<table>
<thead>
<tr>
<th>25 Days from now</th>
<th>precise days and time from now</th>
</tr>
</thead>
<tbody>
<tr>
<td>11/Aug/2015</td>
<td>21/Jul/2015 01:13:17</td>
</tr>
</tbody>
</table>
Terminology

Key terms used in this lesson included:

• CLOB
• BLOB
• TIMESTAMP
• TIMESTAMP WITH TIME_ZONE
• TIMESTAMP WITH LOCAL TIME_ZONE
• 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

• List and provide an example of each of the number, date, and character data types