

The logo for Oracle Academy. It features the word "ORACLE" in a bold, orange, sans-serif font. Below it, the word "Academy" is written 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 SQL

13-2

Using Data Types

ORACLE
Academy



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

Objectives

- This lesson covers the following objectives:
 - 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

Fractions of seconds can be important to some businesses. Currency and stock exchanges need to know the precise time that transactions occur, as prices can change hundreds of times in a second, so they need to be able to track the precise time of transactions / price fluctuations to fractions of a second.

DATE-TIME Data Types

- TIMESTAMP example:

```
CREATE TABLE time_ex1  
(exact_time TIMESTAMP);
```

```
INSERT INTO time_ex1  
VALUES ('10-Jun-2017 10:52:29.123456');
```

```
INSERT INTO time_ex1  
VALUES (SYSDATE);
```

```
INSERT INTO time_ex1  
VALUES (SYSTIMESTAMP);
```

```
SELECT *  
FROM time_ex1;
```

EXACT_TIME
10-JUN-17 10.52.29.123456 AM
20-NOV-20 01.36.33.000000 PM
20-NOV-20 01.36.46.873234 PM

The time_ex1 table has one column of type TIMESTAMP.

Enters a TIMESTAMP value as a literal .

Uses SYSDATE function, but SYSDATE only returns the time to the nearest second, so no microseconds are recorded.

Uses the SYSTIMESTAMP function. This is similar to SYSDATE, but also returns fractions of seconds, so the time is inserted to the nearest microsecond.

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 UTC (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:

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

```
INSERT INTO time_ex2  
VALUES (SYSTIMESTAMP);
```

```
INSERT INTO time_ex2  
VALUES ('10-Jun-2017 10:52:29.123456 AM +2:00');
```

```
SELECT *  
FROM time_ex2;
```

TIME_WITH_OFFSET
20-NOV-20 01.30.55.904850 PM +00:00
10-JUN-17 10.52.29.123456 AM +02:00

ORACLE
Academy

DP 13-2
Using Data Types

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

14

The time_ex2 table has one column of type TIMESTAMP WITH TIME ZONE.

Uses the SYSTIMESTAMP function to add time with offset from UTC. As the APEX server is located in California USA, the time zone is UTC – 7:00 (Pacific Standard Time, or PST)

Enters a literal value TIMESTAMP WITH LOCAL TIME with local time of UTC+2:00 .

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:**

```
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-2017 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;
```

FIRST_COLUMN	SECOND_COLUMN
15-JUL-15 08.00.00.000000 AM -07:00	15-NOV-07 05.00.00.000000 PM

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

The `TIMESTAMP WITH LOCAL TIME ZONE` data type stores the timestamp without time zone information. It converts the time to the database time zone every time the data is sent to and from a client.

Due to the limitations of the hosted online version of APEX, the above example will always show the local time as UTC -07:00 (PST).

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-2017

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

120 months from now	3 years 6 months from now
17-Jul-2027	17-Jan-2021

ORACLE
Academy

DP 13-2
Using Data Types

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

19

The table is created with 2 columns of type INTERVAL YEAR TO MONTHS.

The INSERT statement adds an INTERVAL of 120 months (10 years) to Loan_duration1 and an interval of 3 years and 6 months to loan_duration2.

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

- This example shows interval DAY TO SECOND:

```
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;
```

25 Days from now	precise days and time from now
11-Aug-2017	21-Jul-2017 01:13:17

ORACLE
Academy

DP 13-2
Using Data Types

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

21

This table is created with 2 INTERVAL DAY TO SECOND columns.

The INSERT statement adds an INTERVAL of 25 days to day_duration1, and 4 days, 10 hours, 30 minutes and 10 seconds to day_duration2.

The SELECT statement takes the current date and adds the durations stored in the columns.

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 should 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

The logo for Oracle Academy is centered on a light gray background. It features the word "ORACLE" in a bold, orange, sans-serif font. Below it, the word "Academy" is written in a smaller, 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