



ORACLE

Academy



Database Programming with SQL

13-2

Uso de Tipos de Datos

ORACLE
Academy



Objetivos

- En esta lección, aprenderá a:
 - Crear una tabla utilizando los tipos de dato de columna `TIMESTAMP` y `TIMESTAMP WITH TIME ZONE`
 - Crear una tabla utilizando los tipos de dato de columna `INTERVAL YEAR TO MONTH` e `INTERVAL DAY TO SECOND`
 - Dar ejemplos de organizaciones y situaciones personales en las que es importante conocer a qué zona horaria se refiere un valor de fecha y hora
 - Enumerar y dar un ejemplo de cada uno de los tipos de dato de número, fecha y carácter



Objetivo

- Si alguna vez viaja a otro país, descubrirá rápidamente que el dinero en su bolsillo puede no ser el de la moneda local
- Si desea comprar algo, será necesario convertir el dinero en la moneda del país local
- Este proceso de conversión se parece mucho a tratar con tipos de dato en SQL
- Los distintos tipos de dato tienen tipos diferentes de características, cuyo objetivo es almacenar de manera eficaz los datos
- En esta lección, aprenderá más sobre los tipos de dato y sus usos

Visión General de Tipo de Dato

- Cada valor manipulado por Oracle tiene un tipo de dato
- El tipo de dato de un valor asocia un juego fijo de propiedades al valor
- Estas propiedades hacen que la base de datos trate los valores de un tipo de dato de forma diferente a valores de otro tipo de dato

Visión General de Tipo de Dato

- Los distintos tipos de dato ofrecen varias ventajas:
 - Las columnas de un solo tipo producen resultados consistentes
 - Por ejemplo, las columnas del tipo de dato DATE siempre producen valores de fecha
 - No se puede insertar el tipo de dato incorrecto en una columna. Por ejemplo, las columnas del tipo de dato DATE impedirán que se inserten datos de tipo NUMBER
- Por este motivo, cada columna de una base de datos relacional solo puede contener un tipo de dato
- No puede combinar los tipos de dato en una columna

Tipos de Dato Comunes

- A continuación se muestran los tipos de dato de columna más utilizados para los valores de carácter y número
- Para valores de carácter:
 - CHAR (tamaño fijo, máximo de 2000 caracteres)
 - VARCHAR2 (tamaño variable, máximo de 4000 caracteres)
 - CLOB (tamaño variable, máximo de 128 terabytes)
- Para valores de número:
 - NUMBER (tamaño variable, precisión máxima de 38 dígitos)

Tipos de Dato Comunes

- A continuación se muestran los tipos de dato de columna más utilizados para los valores de fecha, hora y binario
- Para valores de fecha y hora:
 - DATE
 - TIMESTAMP
 - INTERVAL
- Para valores binarios (p. ej., multimedia: JPG, WAV, MP3, etc.):
 - RAW (tamaño variable, máximo de 2000 bytes)
 - BLOB (tamaño variable, máximo de 128 terabytes)

Tipos de Dato Comunes

- Para valores de caracteres, normalmente es mejor utilizar VARCHAR2 o CLOB que CHAR, porque se ahorra espacio
- Por ejemplo, el apellido de un empleado es "Chang"
- En una columna VARCHAR2(30), solo se almacenan los 5 caracteres importantes: C h a n g
- Pero en una columna CHAR(30), se almacenarán también 25 espacios finales, para formar un tamaño fijo de 30 caracteres
- Los valores de número pueden ser negativos, así como positivos. Por ejemplo, NUMBER(6,2) puede almacenar cualquier valor de +9999,99 hasta -9999,99

Tipos de Dato DATE-TIME

- El tipo de dato DATE almacena un valor de siglos hasta segundos enteros, pero no se pueden almacenar las fracciones de un segundo
- "21-Aug-2003 17:25:30" es un valor válido, pero "21-Aug-2003 17:25:30.255" no lo es
- El tipo de dato TIMESTAMP es una extensión del tipo de dato DATE que permite fracciones de un segundo
- Por ejemplo, TIMESTAMP(3) permite 3 dígitos después de los segundos completos, lo que permite almacenar valores de hasta milisegundos

Tipos de Dato DATE-TIME

- Ejemplo de TIMESTAMP:

```
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-15 10.52.29.123456 AM
16-JUL-15 08.17.08.000000 AM
16-JUL-15 08.17.16.610293 AM

TIMESTAMP...With [LOCAL] Time Zone

- Piense en el valor de hora "17:30". Por supuesto, significa "cinco y media de la tarde"
- ¿Pero en qué zona horaria?
- ¿Es cinco y media de la hora de Nueva York, de Pekín, de Estambul tiempo... ?
- En las organizaciones globalizadas actuales que operan en muchos países distintos, es importante saber a qué zona horaria hacer referencia un valor de fecha-hora

TIMESTAMP...With [LOCAL] Time Zone

- `TIMESTAMP WITH TIME ZONE` almacena un valor de zona horaria como un desplazamiento de la Hora Universal Coordinada o UCT (también conocida como Hora Media de Greenwich o GMT)
- Un valor "21-Aug-2003 08:00:00 -5:00" significa 8:00 a.m. 5 horas por debajo de UTC
- Esta es Hora Oficial Oriental de EE. UU. (EST)

TIMESTAMP...With [LOCAL] Time Zone

- Ejemplo de TIMESTAMP WITH TIME ZONE:

```
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
16-JUL-15 08.49.47.126056 AM -07:00
10-JUN-15 10.52.29.123456 AM +02:00

TIMESTAMP...With [LOCAL] Time Zone

- **TIMESTAMP WITH LOCAL TIME ZONE** es similar, pero con una diferencia: si esta columna se selecciona en una sentencia SQL, la hora se convierte automáticamente a la zona horaria del usuario que realiza la selección
- Ejemplo de **TIMESTAMP With...Time Zone**:

```
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');
```


Ejemplo de TIMESTAMP...With Time Zone

- Ambos valores se almacenan con un desplazamiento de hora de -07:00 horas (PST)
- Pero ahora un usuario en Estambul ejecuta:

```
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

- La hora de Estambul es de 9 horas por delante de PST; cuando son las 8:00 a.m. en Los Ángeles, son las 5:00 p.m. en Estambul

Tipos de Dato INTERVAL

- Estos almacenan el tiempo o intervalo de tiempo transcurrido entre dos valores de fecha y hora
- INTERVAL YEAR TO MONTH almacena un período de tiempo medido en años y meses
- INTERVAL DAY TO SECOND almacena un período de tiempo medido en días, horas, minutos y segundos



INTERVAL YEAR...TO MONTH

- Sintaxis:

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

- year_precision es el número máximo de dígitos del elemento YEAR
- El valor por defecto de year_precision es 2



INTERVAL YEAR...TO MONTH

- En este ejemplo se muestra 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);
```

Suponga que la fecha de hoy es: 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 (120 meses a partir de ahora)	3 years 6 months from now (3 años y 6 meses a partir de ahora)
17-Jul-2027	17-Jan-2021

INTERVAL DAY...TO SECOND

- Utilice esta opción cuando necesite una diferencia más precisa entre dos valores de fecha-hora
- Sintaxis:

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

- Day_precision es el número máximo de dígitos del elemento DAY
- El valor por defecto de day_precision es 2
- fractional_seconds_precision es el número de dígitos de la parte fraccionaria del campo de fecha/hora SECOND
- El valor por defecto es 6

INTERVAL DAY...TO SECOND

- En este ejemplo se muestra INTERVAL DAY TO SECOND:

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

```
CREATE TABLE time_ex5  
(day_duration1 INTERVAL DAY(3) TO SECOND,  
 day_duration2 INTERVAL DAY(3) 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 (25 días a partir de ahora)	precise days and time from now (Días y hora precisos a partir de ahora)
11-Aug-2017	21-Jul-2017 01:13:17

Terminología

- Entre los términos clave utilizados en esta lección se incluyen:
 - CLOB
 - BLOB
 - TIMESTAMP
 - TIMESTAMP WITH TIMEZONE
 - TIMESTAMP WITH LOCAL TIMEZONE
 - INTERVAL DAY TO SECOND
 - INTERVAL DAY TO MONTH

Resumen

- En esta lección, ha aprendido lo siguiente:
 - Crear una tabla utilizando los tipos de dato de columna `TIMESTAMP` y `TIMESTAMP WITH TIME ZONE`
 - Crear una tabla utilizando los tipos de dato de columna `INTERVAL YEAR TO MONTH` e `INTERVAL DAY TO SECOND`
 - Dar ejemplos de organizaciones y situaciones personales en las que es importante conocer a qué zona horaria se refiere un valor de fecha y hora
 - Enumerar y dar un ejemplo de cada uno de los tipos de dato de número, fecha y carácter





ORACLE

Academy

