

Tipps & Tricks: Timestamps, Intervalle und dazugehörige Funktionen

Bereich:	SQL	Erstellung:	01/2005 MP
Versionsinfo:	9.2, 10.2, 11.1	Letzte Überarbeitung:	06/2009 EF

Timestamps, Zeitzonen und Intervalle

Timestamp-Datentypen

Mit Oracle 9i wurden 3 Timestamp-Datentypen eingeführt, die Bruchteile von Sekunden anzeigen können.

- **TIMESTAMP[(n)]**: ermöglicht die Angabe von Sekundenbruchteilen; n liegt dabei zwischen 0...9, Default-Wert ist 6; der entsprechende Initialisierungs-Parameter lautet NLS_TIMESTAMP_FORMAT.
- **TIMESTAMP[(n)] WITH TIME ZONE**: beinhaltet zusätzlich die Abweichung von UTC (Coordinated Universal Time); zwei Werte werden dabei als gleich erachtet, wenn sie der gleichen Zeitzone angehören; der entsprechende Initialisierungs-Parameter lautet NLS_TIMESTAMP_TZ_FORMAT.
- **TIMESTAMP[(n)] WITH LOCAL TIME ZONE**: beinhaltet ebenfalls die Abweichung von UTC, jedoch wird diese nicht explizit in der Datenbank gespeichert; stattdessen wird der Wert auf die Zeitzone der Datenbank umgerechnet und als solche eingetragen; ein SELECT liefert die lokale Session-Zeit.

Beispiel:

```
ALTER SESSION SET time_zone = '-5:00';
ALTER SESSION SET nls_date_format = 'dd.mm.rr hh24:mi:ss';

CREATE TABLE new_times (
  zeit1  TIMESTAMP(3),
  zeit2  TIMESTAMP(3) WITH TIME ZONE,
  zeit3  TIMESTAMP(3) WITH LOCAL TIME ZONE);

INSERT INTO new_times VALUES (sysdate, sysdate, sysdate);
INSERT INTO new_times VALUES (TIMESTAMP'2009-06-20 23:20:20',
                                TIMESTAMP'2009-06-20 23:20:20 -7:00',
                                TIMESTAMP'2009-06-20 23:20:20');

SELECT * FROM new_times;
```

ZEIT1	ZEIT2	ZEIT3
30.06.09 09:50:52,000	30.06.09 09:50:52,000 -05:00	30.06.09 09:50:52,000
20.06.09 23:20:20,000	20.06.09 23:20:20,000 -07:00	20.06.09 23:20:20,000

Wenn man jetzt versucht, die Zeitzone der Datenbank zu ändern, erhält man eine Fehlermeldung:

```
ALTER DATABASE SET time_zone = '+5,00';

ORA-30079: Datenbank-Zeitzone kann nicht geändert werden, wenn Datenbank über Spalten
TIMESTAMP WITH LOCAL TIME ZONE verfügt
```

Interval-Datentypen

- **INTERVAL YEAR[(n)] TO MONTH:** gibt Zeitintervalle in Jahren und Monaten an. Optional kann man die Stellenanzahl der Jahreszahl angeben, der Default-Wert ist 2. Ein Intervall von einem Jahr und 13 Monaten muss man als 2 Jahre und 1 Monat ausdrücken.
- **INTERVAL DAY[(n)] TO SECOND:** gibt Zeitintervalle in Tagen, Stunden, Minuten, Sekunden und Sekundenbruchteilen an. Per Default wird der Tag 2-stellig angegeben. Sekundenbruchteile werden bis zur 6. Nachkommastelle ausgegeben (0-9 Nachkommastellen sind möglich).

Intervalle können zu DateTime-Werten addiert oder von ihnen abgezogen werden und es ergibt sich wieder ein DateTime-Wert. Sie können aber auch miteinander addiert, voneinander subtrahiert, mit einer Zahl multipliziert oder durch eine dividiert werden, es ergibt sich immer wieder ein Intervall.

Beispiel:

```
CREATE TABLE halbwertszeiten (
    element_name VARCHAR2(30),
    isotop        VARCHAR2(30),
    hwz_lang      INTERVAL YEAR TO MONTH,
    hwz_kurz      INTERVAL DAY TO SECOND);

INSERT INTO halbwertszeiten VALUES ('Americium', '241Am', '432-2', null);
INSERT INTO halbwertszeiten VALUES ('Plutonium', '238Pu', '87-9', null);
INSERT INTO halbwertszeiten VALUES ('Caesium', '137Cs', '30-2', null);
INSERT INTO halbwertszeiten VALUES ('Strontium', '90Sr', '28-8', null);
INSERT INTO halbwertszeiten VALUES ('Tritium', '3H', '12-4', null);
INSERT INTO halbwertszeiten VALUES ('Cobalt', '60Co', '5-4', null);
INSERT INTO halbwertszeiten VALUES ('Schwefel', '35S', null, '87 12:00:00.0');
INSERT INTO halbwertszeiten VALUES ('Iod', '131I', null, '8 4:48:00.0');
INSERT INTO halbwertszeiten VALUES ('Radon', '222Rn', null, '3 19:12:00.0');
INSERT INTO halbwertszeiten VALUES ('Francium', '223Fr', null, '0 0:22:00.0');
INSERT INTO halbwertszeiten VALUES ('Thorium', '223Th', null, '0 0:0:0.6');
```

```
SELECT * FROM halbwertszeiten;
```

ELEMENT_NAME	ISOTOP	HWZ_LANG	HWZ_KURZ
Plutonium	238Pu	+87-09	
Caesium	137Cs	+30-02	
Strontium	90Sr	+28-08	
Tritium	3H	+12-04	
Cobalt	60Co	+05-04	
Schwefel	35S		+87 12:00:00.000000
Iod	131I		+08 04:48:00.000000
Radon	222Rn		+03 19:12:00.000000
Francium	223Fr		+00 00:22:00.000000
Thorium	223Th		+00 00:00:00.600000

EXTRACT

Die Funktion EXTRACT ermöglicht es, einzelne Datums- oder Uhrzeitelemente aus einem Datums-/Uhrzeitwert oder einem INTERVAL-Wert herauszuholen. Sinn macht die Funktion eigentlich nur bei Intervallen, weil man bei den normalen Datumstypen mit TO_CHAR besser bedient ist.

Beispiele:

```
SELECT ename ,
EXTRACT(YEAR FROM (SYSDATE - hiredate) YEAR TO MONTH ) Jahre ,
EXTRACT(MONTH FROM (SYSDATE - hiredate) YEAR TO MONTH ) Monate
FROM emp ;
```

ENAME	JAHRE	MONATE
SMITH	28	6
ALLEN	28	4
WARD	28	4
JONES	28	3
MARTIN	27	9
....		

```
SELECT element_name ,
EXTRACT(YEAR FROM hwz_lang) Jahre ,
EXTRACT(MONTH FROM hwz_lang) Monate ,
EXTRACT(DAY FROM hwz_kurz) Tage ,
EXTRACT(HOUR FROM hwz_kurz) Stunden ,
EXTRACT(MINUTE FROM hwz_kurz) Minuten
FROM halbwertszeiten ;
```

ELEMENT_NAME	JAHRE	MONATE	TAGE	STUNDEN	MINUTEN
Plutonium	87	9			
Caesium	30	2			
Strontium	28	8			
Tritium	12	4			
Cobalt	5	4			
Schwefel			87	12	0
Iod			8	4	48
Radon			3	19	12
Francium			0	0	22
Thorium			0	0	0

Neue Datumsfunktionen

Gleichzeitig mit den neuen Datentypen wurden auch neue Datums- und Konvertierungsfunktionen (s.u.) eingeführt. Die nachfolgende Tabelle listet die Funktionen auf, die gemeinsam mit den neuen Datentypen eingeführt wurden:

- **CURRENT_DATE** gibt das aktuelle Datum und die Uhrzeit der Session als Datentyp DATE zurück (SYSDATE bezieht sich dagegen auf die Datenbank)
- **CURRENT_TIMESTAMP** gibt das aktuelle Datum und die Uhrzeit der Session als Datentyp TIMESTAMP WITH TIME ZONE zurück
- **LOCALTIMESTAMP** gibt das aktuelle Datum und die Uhrzeit der Session als Datentyp TIMESTAMP zurück
- **SYSTIMESTAMP** gibt das aktuelle Datum und die Uhrzeit der Datenbank als Datentyp TIMESTAMP WITH TIME ZONE zurück.
- **DBTIMEZONE** gibt die Zeitzone der Datenbank zurück (UTC)
- **SESSIONTIMEZONE** gibt die Zeitzone der Session zurück
- **TZ_OFFSET (SESSIONTIMEZONE | DBTIMEZONE | Zonenname | <offset>)**

gibt die Differenz der Zeitzone (angegeben als Name, Abkürzung oder Offset) im Vergleich zur aktuellen Zeitzone der Datenbank zurück

Beispiele:

```
ALTER SESSION SET time_zone = '-7:00';
ALTER SESSION SET nls_date_format = 'dd.mm.rr hh24:mi:ss';
SELECT systimestamp FROM dual;

SYSTIMESTAMP
-----
30.06.09 10:37:49,421000 +02:00

SELECT current_timestamp FROM dual;

CURRENT_TIMESTAMP
-----
30.06.09 01:38:03,250000 -07:00

SELECT localtimestamp FROM dual;

LOCALTIMESTAMP
-----
30.06.09 01:38:18,812000

SELECT current_date FROM dual;

CURRENT_DATE
-----
30.06.09 01:38:36

SELECT sysdate FROM dual;

SYSDATE
-----
30.06.09 10:38:50

SELECT dbtimezone FROM dual;

DBTIME
-----
+00:00

SELECT sessiontimezone FROM dual;

SESSIONTIMEZONE
-----
-07:00
```

Mittels der Funktion `timezone_offset` kann man auch die Differenz zwischen der Zeitzone des Datenbankservers und einer gegebenen Zeitzone, z.B. der amerikanischen Ostküste, ermitteln. Weil sich der Offset des Servers immer auf UTC (die Abkürzung ist eine "sprachneutrale" Abkürzung der Ausdrücke Coordinated Universal Time bzw. Temps Universel Coordonné) bezieht, ändert er sich nicht, wenn die Zeitzone der Session umgestellt wird.

```
SELECT TZ_OFFSET( 'US/Eastern' ) FROM dual;
```

```
TZ_OFFSETS
-----
-04:00

ALTER SESSION SET time_zone = '+2:00';
SELECT TZ_OFFSET('US/Eastern') FROM dual;

TZ_OFFSETS
-----
-04:00
```

Konvertierungsfunktionen

- FROM_TZ(timestamp_wert, timezone_wert) wandelt einen TIMESTAMP-Wert in einen TIMESTAMP WITH TIME ZONE-Wert um.
- TO_TIMESTAMP(char[, format[, nlsparam]]) wandelt einen String in einen TIMESTAMP um, optional kann ein Formatstring angegeben werden.
- TO_TIMESTAMP_TZ(char[, format[, nlsparam]]) wandelt einen String in einen TIMESTAMP WITH TIME ZONE um.
- TO_DSINTERVAL(char[, nlsparam]) wandelt einen String in ein INTERVAL DAY TO SECOND um.
- TO_YMINTERVAL(char) wandelt einen String in ein INTERVAL YEAR TO MONTH um.

Beispiele:

```
SELECT FROM_TZ(TIMESTAMP '2009-06-20 08:00:00', '3:00') zeitstempel FROM dual;

ZEITSTEMPEL
-----
20.06.09 08:00:00,000000000 +03:00

SELECT FROM_TZ(localtimestamp, '-7:00') zeitstempel FROM dual;

ZEITSTEMPEL
-----
30.06.09 04:18:58,125000 -07:00

ALTER SESSION SET time_zone = '-4:00';
SELECT TO_TIMESTAMP_TZ(sysdate, 'dd.mm.yyyy hh24:mi:ss') zeitstempel FROM dual;

ZEITSTEMPEL
-----
30.06.09 13:19:16 -04:00

SELECT TO_DSINTERVAL('150 12:06:45') Intervall FROM dual;

INTERVALL
-----
+000000150 12:06:45.000000000

SELECT ename, hiredate+TO_YMINTERVAL('01-06') einst_datum_plus_eineinhalb FROM emp;

ENAME      EINST_DATUM_PLUS_EINEINHALB
-----
SMITH      17.06.82
ALLEN      20.08.82
WARD       22.08.82
```

```
JONES    02.10.82
MARTIN   28.03.83
...
```

Die Namen der Zeitzonen und ihre Abkürzungen erhält man übrigens über die View V\$TIMEZONE_NAMES.

```
SELECT tzname, tzabbrev FROM v$timezone_names
WHERE SUBSTR(TZ_OFFSET(tzname),1,6) = '+02:00'
AND tzname LIKE '%Berlin%';
```

TZNAME	TZABBREV
-----	-----
Europe/Berlin	LMT
Europe/Berlin	CET
Europe/Berlin	CEST
Europe/Berlin	CEMT