

Chapter 16. Oracle 9i 날짜 함수

이번 장에서는 Oracle 9i에서 새롭게 지원하는 날짜 함수에 대하여 자세히 알아본다.

표준시간대(Time Zone)

Oracle 9i부터 날짜 데이터 타입에 표준시간대를 입력 할 수 있는 데이터 타입과 이러한 데이터 타입을 조작 할 수 있는 함수들이 추가되었다. 추가된 데이터 타입과 함수를 이해하기 위해서는 표준시간대와 그리니치 표준시(GMT : Greenwich Mean Time)에 대한 개념을 알아야 한다. 우리가 살고 있는 지구는 전 세계를 24개의 시간대로 나누고 영국의 그리니치 천문대를 기준 표준시간대로 지정하여 다른 표준시간대들에 의해 참조된다. 예를 들어, 한국은 그리니치 표준시를 기준으로 +9:00 시간 빠르다.

TIMESTAMP 데이터 타입

Oracle 9i 이전에서는 날짜를 저장하기 위한 데이터 타입으로 DATE 타입이 있지만 Oracle 9i 부터는 표준시간대와 정밀한 시간(10^{-9} 초)을 저장 할 수 있는 TIMESTAMP 데이터 타입이 추가되었다.

DATE 타입에 새롭게 추가된 데이터 타입은 다음과 같다.

표 16-1. 추가된 날짜 타입

데이터 타입	표준 시간대	정밀도(초)
DATE	저장되지 않음	초
TIMESTAMP (fractional_seconds_precision)	저장되지 않음	fractional_seconds_precision 는 초의 소수점 자리수를 지정 하며, 지정가능한 값은 0~9이 며 디폴트는 6
TIMESTAMP (fractional_seconds_precision) WITH TIME ZONE	TIMESTAMP에 지역시간과 그리니치 표준시와의 시간차 가 저장	"
TIMESTAMP (fractional_seconds_precision) WITH LOCAL TIME ZONE	TIMESTAMP WITH TIME ZONE에 다음과 같은 사항을 제외하고 저장 - 날짜를 저장 할 때 데이터 베이스 표준시간대를 기 준으로 변환 저장 - 날짜가 검색될 때 세션의 표준시간대로 변환	"

위에서 TIMESTAMP WITH LOCAL TIME ZONE 타입에 날짜가 저장되면 데이터베이스에 지정된 표준시간대로 변환되어 저장된다. 예를 들어, 서울(+9:00)에서 뉴욕(-5:00)에 있는 데이터베이스의 TIMESTAMP WITH LOCAL TIME ZONE으로 정의된 컬럼에 '2004-6-27 20:00'를 저장하면 뉴욕의 표준시간대를 기준으로 '2004-6-27 6:00'가 저장된다. 이 후, 서울에서 입력된 데이터를 검색하면 '2004-6-27 20:00'가 검색되며, 뉴욕에서

검색하면 '2004-6-27 6:00'가 검색된다. 또한, Oracle 9i는 날짜 데이터 타입에 섬머타임(Daylight Savings Time)을 지원해주므로 섬머타임제가 시작되면 시간이 1시간 앞당겨지고 섬머타임제가 종료되면 원래 시간으로 설정된다.

TZ_OFFSET

TZ_OFFSET 함수는 입력된 표준시간대에 대하여 그리니치 표준시와의 시차를 리턴하는 함수이다. 예를 들어, TZ_OFFSET 함수가 +9:00를 리턴하면 입력된 표준시간대가 그리니치 표준시를 기준으로 9시간 빠르다는 것을 의미한다.

```
SQL> SELECT TZ_OFFSET('Asia/Seoul') FROM DUAL;

TZ_OFFSET
-----
+09:00

SQL> SELECT TZ_OFFSET('Europe/London') FROM DUAL;

TZ_OFFSET
-----
+01:00
```

TZ_OFFSET의 입력 값으로 사용 할 수 있는 표준시간대 이름을 확인하려면 V\$TIMEZONE_NAMES를 검색하면 된다.

```
SQL> SELECT TZNAME FROM V$TIMEZONE_NAMES
2  WHERE UPPER(TZNAME) LIKE '%SEOUL%';

TZNAME
-----
Asia/Seoul
Asia/Seoul
Asia/Seoul
```

CURRENT_DATE

CURRENT_DATE 함수는 세션의 표준시간대에서 현재 날짜를 리턴한다. 그러므로, 세션의 표준시간대가 변경되면 CURRENT_DATE 함수의 결과는 달라지게 된다. 현재 세션에서의 표준시간대와 날짜를 확인해보자. 먼저, 정확한 시간을 표시하기 위해 ALTER SESSION 명령으로 날짜 표시 형식을 변경한다.

```
SQL> ALTER SESSION
2 SET NLS_DATE_FORMAT = 'YYYY-MM-DD HH24:MI:SS';
```

세션이 변경되었습니다.

```
SQL> SELECT SESSIONTIMEZONE, CURRENT_DATE FROM DUAL;
```

SESSIONTIMEZONE	CURRENT_DATE
+09:00	2004-06-27 22:03:51

세션의 표준시간대를 변경하여 현재 날짜를 확인해보자. 표준시간대가 변경되었으므로 CURRENT_DATE 함수의 결과값도 변경되었다.

```
SQL> ALTER SESSION
2 SET TIME_ZONE = '+5:0';
```

세션이 변경되었습니다.

```
SQL> SELECT SESSIONTIMEZONE, CURRENT_DATE FROM DUAL;
```

SESSIONTIMEZONE	CURRENT_DATE
+05:00	2004-06-27 18:05:26

참고로 SYSDATE 함수는 표준시간대 설정과는 무관한 결과를 리턴한다.

```
SQL> SELECT SYSDATE FROM DUAL;
```

SYSDATE
2004-06-27 22:06:42

CURRENT_TIMESTAMP

CURRENT_TIMESTAMP 함수도 CURRENT_DATE 함수와 마찬가지로 표준시간대의 설정에 따라 그 결과도 달라진다. 반면, CURRENT_TIMESTAMP 함수는 CURRENT_DATE 함수에 비해 좀 더 정밀한 시간을 표시한다. 즉, TIMESTAMP WITH TIME ZONE 데이터 타입으로 출력된다.

```
SQL> SELECT SESSIONTIMEZONE, CURRENT_TIMESTAMP FROM DUAL;
```

SESSIONTIMEZONE	CURRENT_TIMESTAMP
+05:00	04/06/27 18:11:45.218000 +05:00

마찬가지로 표준시간대를 변경하면 리턴 값도 달라진다.

```
SQL> ALTER SESSION
      2 SET TIME_ZONE = '+9:0';
```

세션이 변경되었습니다.

```
SQL> SELECT SESSIONTIMEZONE, CURRENT_TIMESTAMP FROM DUAL;
```

SESSIONTIMEZONE	CURRENT_TIMESTAMP
-----	-----
+09:00	04/06/27 22:13:07.747000 +09:00

LOCALTIMESTAMP

LOCALTIMESTAMP 함수는 CURRENT_TIMESTAMP와 마찬가지로 세션의 표준시간대에서 현재 날짜를 리턴한다. 그러나 LOCALTIMESTAMP 함수의 리턴값은 TIMESTAMP 타입으로 출력되며, CURRENT_TIMESTAMP 함수의 리턴값은 TIMESTAMP WITH TIME ZONE 타입으로 출력된다.

```
SQL> SELECT CURRENT_TIMESTAMP, LOCALTIMESTAMP FROM DUAL;
```

CURRENT_TIMESTAMP	LOCALTIMESTAMP
-----	-----
04/06/27 22:26:18.664000 +09:00	04/06/27 22:26:18.664000

```
SQL> ALTER SESSION
      2 SET TIME_ZONE = '+5:0';
```

세션이 변경되었습니다.

```
SQL> SELECT CURRENT_TIMESTAMP, LOCALTIMESTAMP FROM DUAL;
```

CURRENT_TIMESTAMP	LOCALTIMESTAMP
-----	-----
04/06/27 18:26:59.122000 +05:00	04/06/27 18:26:59.122000

DBTIMEZONE, SESSIONTIMEZONE

DBTIMEZONE 함수는 디폴트 데이터베이스 표준시간대로서 운영체제의 표준시간대와 동일하다. 디폴트 데이터베이스 표준시간대는 CREATE DATABASE 문장의 SET TIME_ZONE 구문으로 지정 가능하며, 세션에서는 ALTER SESSION 명령에 의해 변경이 가능하다.

```
SQL> SELECT DBTIMEZONE FROM DUAL;
```

DBTIME

-07:00

SESSIONTIMEZONE 함수는 세션의 표준시간대를 리턴한다.

```
SQL> SELECT SESSIONTIMEZONE FROM DUAL;
```

```
SESSIONTIMEZONE
```

```
-----
+09:00
```

EXTRACT

EXTRACT는 입력값인 날짜 데이터 타입에서 지정된 값을 추출하는 함수이다. EXTRACT의 문법은 다음과 같다.

```
SELECT EXTRACT ([YEAR] [MONTH] [DAY] [HOUR] [MINUTE] [SECOND]
                [TIMEZONE_HOUR] [TIMEZONE_MINUTE]
                [TIMEZONE_REGION] [TIMEZONE_ABBR]
FROM [datetime_value_expression]
     [interval-value_expression]);
```

SYSDATE 함수의 결과값으로부터 년도를 추출하는 방법은 다음과 같다.

```
SQL> SELECT EXTRACT (YEAR FROM SYSDATE) FROM DUAL;
```

```
EXTRACT(YEARFROMSYSDATE)
```

```
-----
2004
```

사원 테이블에서 10번 부서에 근무하는 직원의 입사월을 출력하면 다음과 같다.

```
SQL> SELECT ENAME, HIREDATE, EXTRACT (MONTH FROM HIREDATE)
2 FROM EMP
3 WHERE DEPTNO = 10;
```

ENAME	HIREDATE	EXTRACT(MONTHFROMHIREDATE)
CLARK	81/06/09	6
KING	81/11/17	11
MILLER	82/01/23	1

FROM_TZ

FROM_TZ 함수는 TIMESTAMP 타입의 데이터를 TIMESTAMP WITH TIME ZONE 타입의 데이터로 변환해준다. FROM_TZ 함수의 문법은 다음과 같다.

```
FROM_TZ(TIMESTAMP timestamp_value, time_zone_value)
```

TIMESTAMP 데이터인 '2004-06-27 20:00:00'를 TIMESTAMP WITH TIME ZONE 타입의 데이터로 변환하면 다음과 같다.

```
SQL> SELECT FROM_TZ(TIMESTAMP '2004-06-27 20:00:00', '+5:0') FROM DUAL;

FROM_TZ(TIMESTAMP'2004-06-2720:00:00','+5:0')
-----
04/06/27 20:00:00.000000000 +05:00

SQL> SELECT FROM_TZ(TIMESTAMP '2004-06-27 20:00:00', 'Asia/Seoul') FROM DUAL;

FROM_TZ(TIMESTAMP'2004-06-2720:00:00','ASIA/SEOUL')
-----
04/06/27 20:00:00.000000000 ASIA/SEOUL
```

TO_TIMESTAMP, TO_TIMESTAMP_TZ

TO_TIMESTAMP와 TO_TIMESTAMP_TZ 함수는 문자열을 각각 TIMESTAMP 타입 데이터와 TIMESTAMP WITH TIME ZONE 타입 데이터로 변환해준다.

문자열 '2004-06-27 20:00:00'를 TIMESTAMP 타입 데이터로 변환하면 다음과 같다.

```
SQL> SELECT TO_TIMESTAMP ('2004-06-27 20:00:00', 'YYYY-MM-DD HH24:MI:SS')
2 FROM DUAL;

TO_TIMESTAMP('2004-06-2720:00:00','YYYY-MM-DDHH24:MI:SS')
-----
04/06/27 20:00:00.000000000
```

문자열 '2004-06-27 20:00:00'를 TIMESTAMP WITH TIME ZONE 타입의 데이터로 변환하면 다음과 같다.

```
SQL> SELECT TO_TIMESTAMP_TZ ('2004-06-27 20:00:00 +9:00',
2 'YYYY-MM-DD HH24:MI:SS TZh:TzM')
3 FROM DUAL;

TO_TIMESTAMP_TZ('2004-06-2720:00:00+9:00','YYYY-MM-DDHH24:MI:SSTZh:TzM')
-----
04/06/27 20:00:00.000000000 +09:00
```

TO_YMINTERVAL

TO_YMINTERVAL 함수는 문자열을 INTERVAL YEAR TO MONTH 타입 데이터로 변환해준다.

사원 테이블에서 10번 부서에 근무하는 직원의 입사일과 입사일로부터 1년 2개월 후의 날짜를 출력하면 다음과 같다.

```
SQL> SELECT ENAME, HIREDATE, HIREDATE+TO_YMINTERVAL('01-02') HIREDATE2
2  FROM EMP
3  WHERE DEPTNO = 10;
```

ENAME	HIREDATE	HIREDATE
CLARK	81/06/09	82/08/09
KING	81/11/17	83/01/17
MILLER	82/01/23	83/03/23

복습

1. 'Canada/Yukon'의 표준시간대를 검색하시오.
2. 현재 세션의 표준시간대를 검색하시오.
3. 사원 테이블에서 SMITH의 입사년도, 입사월, 입사일을 다음과 같이 출력하는 SQL 문장을 작성하시오.

ENAME	년도	월	일
SMITH	1980	12	17

4. 세션의 날짜 표시 형식을 'YYYY-MM-DD HH24:MI:SS'로 변경하시오.
5. 세션의 현재 날짜를 출력하시오. 단, 표준시간대도 표시하시오.