

## 3-3. SQL 함수 – 사용자 정의 함수 및 기타

홍형경

[chariehong@gmail.com](mailto:chariehong@gmail.com)

2020.11

# 1. 사용자 정의 함수

- 사용자 정의 함수

- 직접 함수 만들어 사용
- PL/SQL을 사용해 개발

- PL/SQL

- 프로그래밍 언어의 특성을 수용한 SQL 확장판
- SQL은 물론 프로그래밍에 필요한 조건문(IF), 반복문(LOOP, FOR, WHILE) 추가
- PL/SQL은 본 강의에서 다루지 않음

## 2. 사용자 정의 함수를 만들어 볼까?

. 어떤 함수를 만들어야 할까?

- 반복적으로 사용되는 로직

- 부서 번호로 부서명을 가져오는 함수를 만들어 보자

  - ➔ 부서명을 가져오기 위해서는 EMPLOYEES와 DEPARTMENTS 테이블 조인해야 함

- 함수는 매개변수를 받아 결과를 반환

  - ➔ 부서번호(DEPARTMENT\_ID)를 매개변수로 받아 부서명(DEPARTMENT\_NAME)을 반환

## 2. 사용자 정의 함수를 만들어 볼까?

### (1) get\_dept\_name 함수 생성

```
CREATE OR REPLACE FUNCTION get_dept_name (p_dept_id NUMBER)
RETURN VARCHAR2
IS
    v_return VARCHAR2(80);
BEGIN
    SELECT department_name
    INTO v_return
    FROM departments
    WHERE department_id = p_dept_id;

    RETURN v_return;
END;
```

## 2. 사용자 정의 함수를 만들어 볼까?

### (1) get\_dept\_name 함수

```
SELECT get_dept_name(10),  
       get_dept_name(100)  
FROM DUAL;
```

GET_DEPT_NAME(10)	GET_DEPT_NAME(100)
Administration	Finance

DEPARTMENT_ID	DEPARTMENT_NAME
1	10 Administration
2	20 Marketing
3	30 Purchasing
4	40 Human Resources
5	50 Shipping
6	60 IT
7	70 Public Relations
8	80 Sales
9	90 Executive
10	100 Finance
11	110 Accounting
12	120 Treasury
13	130 Corporate Tax
14	140 Control And Credit
15	150 Shareholder Services
16	160 Benefits
17	170 Manufacturing
18	180 Construction
19	190 Contracting
20	200 Operations
21	210 IT Support
22	220 NOC
23	230 IT Helpdesk
24	240 Government Sales
25	250 Retail Sales
26	260 Recruiting
27	270 Payroll

## 2. 사용자 정의 함수를 만들어 볼까?

### (1) get\_dept\_name 함수

```
SELECT employee_id,  
       first_name || ' ' || last_name emp_name,  
       department_id,  
       get_dept_name(department_id) dept_name  
FROM employees;
```

EMPLOYEE_ID	EMP_NAME	DEPARTMENT_ID	DEPT_NAME
100	Steven King	90	Executive
101	Neena Kochhar	90	Executive
102	Lex De Haan	90	Executive
103	Alexander Hunold	60	IT
104	Bruce Ernst	60	IT
105	David Austin	60	IT
106	Valli Pataballa	60	IT
107	Diana Lorentz	60	IT
108	Nancy Greenberg	100	Finance
109	Daniel Faviet	100	Finance
110	John Chen	100	Finance
111	Ismael Sciarra	100	Finance
112	Jose Manuel Urman	100	Finance
113	Luis Popp	100	Finance
114	Den Raphaely	30	Purchasing
115	Alexander Khoo	30	Purchasing
116	Shelli Baida	30	Purchasing
117	Sigal Tobias	30	Purchasing
118	Guy Himuro	30	Purchasing
119	Karen Colmenares	30	Purchasing
120	Matthew Weiss	50	Shipping
121	Adam Fripp	50	Shipping

## 2. 사용자 정의 함수를 만들어 볼까?

### (2) IsNumber 함수

- 문자형 매개변수 값을 받아 이 값이 숫자인지 판별하는 함수
- MSSQL에는 빌트인 함수로 제공됨 (IsDate 함수도 제공)
- 오라클에서는 제공되지 않으므로, 직접 만들어 보자
- 숫자가 맞으면 0을 반환, 틀리면 1을 반환

## 2. 사용자 정의 함수를 만들어 볼까?

### (2) IsNumber 함수 생성

```
CREATE OR REPLACE FUNCTION IsNumber (p_number VARCHAR2)
RETURN NUMBER
IS
    v_return NUMBER;
BEGIN
    SELECT TO_NUMBER(p_number)
    INTO v_return
    FROM DUAL;

    RETURN 0;
EXCEPTION WHEN OTHERS THEN
    RETURN 1;
END;
```



## 2. 사용자 정의 함수를 만들어 볼까?

### (2) IsNumber 함수 사용

```
SELECT IsNumber('123')  
       ,IsNumber('ABc')  
FROM DUAL;
```

ISNUMBER('123')	ISNUMBER('ABC')
0	1

## 2. 사용자 정의 함수를 만들어 볼까?

### (2) IsNumber 함수 사용

```
CREATE TABLE TO_NUMBER_TEST (  
    NUMBER_CONF VARCHAR2(100)  
);
```

```
INSERT INTO TO_NUMBER_TEST VALUES ('1');  
INSERT INTO TO_NUMBER_TEST VALUES ('2');  
INSERT INTO TO_NUMBER_TEST VALUES ('3');  
INSERT INTO TO_NUMBER_TEST VALUES ('4');  
INSERT INTO TO_NUMBER_TEST VALUES ('5');  
INSERT INTO TO_NUMBER_TEST VALUES ('6');  
INSERT INTO TO_NUMBER_TEST VALUES ('7');  
INSERT INTO TO_NUMBER_TEST VALUES ('8');  
INSERT INTO TO_NUMBER_TEST VALUES ('9');  
INSERT INTO TO_NUMBER_TEST VALUES ('10');
```

```
INSERT INTO TO_NUMBER_TEST VALUES ('11');  
INSERT INTO TO_NUMBER_TEST VALUES ('1a');  
INSERT INTO TO_NUMBER_TEST VALUES ('13');  
INSERT INTO TO_NUMBER_TEST VALUES ('14');  
COMMIT;
```

## 2. 사용자 정의 함수를 만들어 볼까?

### (2) IsNumber 함수 사용

```
SELECT *  
FROM TO_NUMBER_TEST;
```

```
SELECT TO_NUMBER(NUMBER_CONF)  
FROM TO_NUMBER_TEST;
```

	NUMBER_CONF
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	1a
13	13
14	14

```
ORA-01722: 수치가 부적합합니다  
01722, 00000 - "invalid number"  
*Cause:   The specified number was invalid.  
*Action:  Specify a valid number.
```

## 2. 사용자 정의 함수를 만들어 볼까?

### (2) IsNumber 함수 사용

```
SELECT CASE WHEN IsNumber(number_conf) = 0 THEN TO_NUMBER(number_conf)
        ELSE 0
        END TONUMBERS
FROM TO_NUMBER_TEST;
```

	NUMBER_CONF
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	1a
13	13
14	14

	TONUMBERS
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	0
13	13
14	14

## 2. 사용자 정의 함수를 만들어 볼까?

- 사용자 정의 함수는 가급적 사용 자제
- 기능은 편리하나 성능 측면에서는 좋지 않음
- 특히 함수 내에서 대량 데이터를 조회하는 경우, 성능에 악영향  
➔ 이런 경우는 쿼리를 직접 작성해 처리해야 함
- 꼭 사용해야 한다면 RESULT\_CACHE 기능 활성화

### 3. SQL 함수를 사용해 재미있는 문제 풀이

1. 서기가 시작된 1월 1일부터 오늘까지 1조원을 쓰려면 매일 얼마를 써야 하는지 구하시오.

(최종 결과는 소수점 첫 째 자리에서 반올림 )

- 필요한 내용 : 1조 (10000000000000), 서기시작 ~ 오늘까지 일 수
- 오늘까지 일 수 → 1년은 365일, 년도를 구하면 년도까지 일 수는 구할 수 있음  
→ 작년 12월 31일까지 일 수 + 금년 오늘까지 일 수
- 1조 / (작년 12월 31일까지 일 수 + 금년 오늘까지 일 수 )

### 3. SQL 함수를 사용해 재미있는 문제 풀이

1. 서기가 시작된 1월 1일부터 오늘까지 1조원을 쓰려면 매일 얼마를 써야 하는지 구하시오.

(최종 결과는 소수점 첫 째 자리에서 반올림 )

- 작년 12월 31일까지 일 수

➔  $(\text{현재년도} - 1) * 365$

```
select (TO_NUMBER(TO_CHAR(SYSDATE, 'YYYY')) - 1) * 365
```

```
from dual;
```

- 금년 오늘까지 일 수

➔ `SELECT TO_NUMBER(TO_CHAR(sysdate, 'ddd'))`

```
FROM DUAL;
```

### 3. SQL 함수를 사용해 재미있는 문제 풀이

1. 서기가 시작된 1월 1일부터 오늘까지 1조원을 쓰려면 매일 얼마를 써야 하는지 구하시오.  
(최종 결과는 소수점 첫 째 자리에서 반올림 )

```
SELECT TO_NUMBER(TO_CHAR(SYSDATE, 'YYYY')) - 1 LAST_YEAR
      ,TO_CHAR(SYSDATE, 'DDD') DAYS
      ,(( TO_NUMBER(TO_CHAR(SYSDATE, 'YYYY')) - 1) * 365 ) + TO_NUMBER(TO_CHAR(SYSDATE, 'DDD')) DAYS_ALL
      , ROUND(10000000000000 / ( (( TO_NUMBER(TO_CHAR(SYSDATE, 'YYYY')) - 1) * 365 ) + TO_NUMBER(TO_CHAR(SYSDATE, 'DDD'))),0) TRILLIONS
FROM DUAL;
```

LAST_YEAR	DAYS	DAYS_ALL	TRILLIONS
2019	155	737090	1356686



### 3. SQL 함수를 사용해 재미있는 문제 풀이

2. 2019년 9월 10일 애인과 처음 만났다. 100일, 500일, 1000일 기념 파티를 하고 싶은데, 언제인지 계산하기가 힘드네요 ㅠㅠ... 계산해 주세요.

```
SELECT TO_DATE('2019-09-10', 'YYYY-MM-DD') + 100 AS "100일"  
      ,TO_DATE('2019-09-10', 'YYYY-MM-DD') + 500 AS "500일"  
      ,TO_DATE('2019-09-10', 'YYYY-MM-DD') + 1000 AS "1000일"  
FROM DUAL ;
```

100일	500일	1000일
2019-12-19 00:00:00	2021-01-22 00:00:00	2022-06-06 00:00:00

## ● SQL 함수를 사용해 재미있는 문제 풀이

### 3. 524288 이란 숫자가 있다. 이 수는 2의 몇 승일까?

```
SELECT log(2, 524288)
FROM DUAL;
```

[illegible]

- SQL 함수를 사용해 재미있는 문제 풀이

#### 4. 2050년 2월의 마지막 날은 무슨 요일일까?

```
SELECT TO_CHAR(LAST_DAY(TO_DATE('20500201', 'YYYYMMDD')), 'DAY')  
FROM DUAL;
```

```
TO_CHAR(LAST_DAY(TO_DATE('20500201', 'YYYYMMDD')), 'DAY')  
월요일
```

- SQL 함수를 사용해 재미있는 문제 풀이

5. 현재일자(2020-11-23) 기준 ROUND(SYSDATE, 'YYYY') 를 실행하면 2021-01-01이 반환된다.  
그럼 언제 시점부터 2021-01-01이 반환될까?

```
SELECT ROUND(TO_DATE('2020-06-30 23:59:59', 'YYYY-MM-DD HH24:MI:SS'), 'YYYY')  
        AS THIS_YEAR  
      ,ROUND(TO_DATE('2020-07-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'YYYY')  
        AS NEXT_YEAR  
FROM DUAL;
```

THIS_YEAR	NEXT_YEAR
2020-01-01 00:00:00	2021-01-01 00:00:00