3-3. SQL 함수 – 사용자 정의 함수 및 기타

홍형경 chariehong@gmail.com 2021.06

#### 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

- . 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

(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;
```

(1) get\_dept\_name 함수

SELECT get\_dept\_name(10), get\_dept\_name(100) FROM DUAL;

	GET_DEPT_NAME(100)
Administration	Finance

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

(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
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 Urma	
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

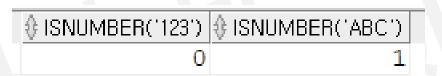
- (2) IsNumber 함수
- 문자형 매개변수 값을 받아 이 값이 숫자인지 판별하는 함수
- · MSSQL에는 빌트인 함수로 제공됨 (IsDate 함수도 제공)
- 오라클에서는 제공되지 않으므로, 직접 만들어 보자
- 숫자가 맞으면 0을 반환, 틀리면 1을 반환

(2) 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) IsNumber 함수 사용

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



#### (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) 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) 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

		-		
	\$ TON	IUME	BEF	RS
1				1
2				2
3				3
4				4
5				5
6				6
7				7
8				8
9				9
10			1	0
11			1	1
12				0
13			1	3
14			1	4

- · 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

- 1. 서기가 시작된 1월 1일부터 오늘까지 1조원을 쓰려면 매일 얼마를 써야 하는지 구하시오. (최종 결과는 소수점 첫 째 자리에서 반올림 )
- 필요한 내용 : 1조 (100000000000), 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 )

- 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
  - → (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;

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(100000000000 / ( (( TO_NUMBER(TO_CHAR(SYSDATE, 'YYYY')) - 1) * 365 ) + TO_NUMBER(TO_CHAR(SYSDATE, 'DDD'))),0) TRILLIONS
FROM DUAL;
```

	∯ DAYS	∯ DAYS_ALL	
2020	165	737465	1355997

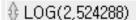
2. 2021년 6월 10일 애인과 처음 만났다. 100일, 500일, 1000일 기념 파티를 하고 싶은데, 언제인지 계산하기가 힘드네요 ㅠㅠ... 계산해 주세요.

```
SELECT TO_DATE('2021-06-10', 'YYYY-MM-DD') + 100 AS "100일"
       ,TO_DATE('2021-06-10', 'YYYY-MM-DD') + 500 AS "500일"
       ,TO_DATE('2021-06-10', 'YYYY-MM-DD') + 1000 AS "1000일"
FROM DUAL;
```

⊕ 500일 ᇴ 1000일 2021-09-18 00:00:00 2022-10-23 00:00:00 2024-03-06 00:00:00

3. 524288 이란 숫자가 있다. 이 수는 2의 몇 승일까?

SELECT log(2, 524288) FROM DUAL;





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

FROM DUAL;

5. 현재일자(2021-06-14) 기준 ROUND(SYSDATE, 'YYYY') 를 실행하면 2021-01-01이 반환된다. 그럼 언제 시점부터 2022-01-01이 반환될까?

SELECT ROUND(TO\_DATE('2021-06-30 23:59:59', 'YYYY-MM-DD HH24:MI:SS'), 'YYYY')

AS THIS\_YEAR

,ROUND(TO\_DATE('2021-07-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'YYYY')

AS NEXT\_YEAR