

Java Web Programming 입문 11

(Oracle PL/SQL #03)

오늘의 키워드

▶ 1. SELECT문

- WHERE Condition

- AND, OR
- 수치 비교
- IS NULL, IS NOT NULL
- IN, NOT IN
- BETWEEN A AND B, NOT BETWEEN A AND B
- LIKE, NOT LIKE

▶ 2. 함수 (Function)

- 분류
- 대소문자 치환 함수
- 문자열 함수
- 소수점 제어 함수
- 나머지 연산 함수
- 날짜 함수

SELECT 문

▶ WHERE 절

```
SELECT [COLUMNS]
FROM [TABLES]
WHERE [CONDITIONS]
```

```
SQL> SELECT ENAME, SAL
      FROM EMP
      WHERE SAL >= 3000;
```

```
SQL> SELECT EMPNO, ENAME, SAL, DEPTNO
      FROM EMP
      WHERE ENAME = 'SCOTT'
```

```
SQL> SELECT EMPNO, ENAME, SAL, DEPTNO
      FROM EMP
      WHERE DEPTNO = 10;
```

ENAME	SAL		
SCOTT	3000	<-	[TRUE]
SMITH	800	<-	[FALSE]
KING	5000	<-	[TRUE]
.			
.			

SELECT 문

▶ AND, OR

- 소개팅 했다!

- 남자

- (남자1) 예쁘냐?
 - (남자2) 오 이쁘냐?
 - (남자3) 이쁘디?
 - (남자4) 그래서 예뻐?
 - (남자5) 아니 그니까, 이쁘냐고?!

- 여자

- (여자1) 뭐하는 사람인데?
 - (여자2) 키는?
 - (여자3) 연봉은?
 - (여자4) 나이는?
 - (여자5) 얼굴은?

- WHERE 절에서는 한번에 여러 조건부여가 가능

▶ EMP 테이블에서 SAL이 3000 이상이며, DEPTNO 는 10인 EMPNO, ENAME, SAL, DEPTNO 조회

```
SQL> SELECT EMPNO, ENAME, SAL, DEPTNO  
        FROM EMP  
        WHERE SAL >= 3000 AND DEPTNO = 10;
```

SELECT 문

▶ AND, OR

- A **AND** B : A와 B 모두 만족해야 TRUE

[AND]	TRUE	FALSE	NULL
TRUE	TRUE	FALSE	NULL
FALSE	FALSE	FALSE	FALSE
NULL	NULL	FALSE	NULL

- A **OR** B : A와 B중 어느 하나만 만족하여도 TRUE

[OR]	TRUE	FALSE	NULL
TRUE	TRUE	TRUE	TRUE
FALSE	TRUE	FALSE	NULL
NULL	TRUE	NULL	NULL

SELECT 문

▶ AND, OR

- AND 가 OR 보다 우선순위가 높다.

```
WHERE A AND B OR C AND D AND E OR F
```

1 4 2 3 5

- 그러나 괄호는 모든 연산보다 우선이다.

```
WHERE (A AND B) OR ((C AND D) AND E) OR F
```

- NOT 은 AND 연산보다 우선이다.

SELECT 문

▶ 수치와 관련된 녀석들

- =, >, <, >=, <=, !=, <>, ^=

연산	의미
A = B	A와 B가 같음
A > B	A보다 B가 큼
A < B	A보다 B가 작음
A >= B	A가 B보다 크거나 같음
A <= B	A가 B보다 작거나 같음
A != B	A와 B가 같지않음
A <> B	A와 B가 같지않음
A ^= B	A와 B가 같지않음

- 날짜의 크기도 비교가 가능

```
WHERE hiredate > '81/01/01';
```

BC

9999/12/31/23/59

2012/06/09

2011/08/12

SELECT 문

▶ IS NULL, IS NOT NULL

○ (X)

```
SQL> SELECT ENAME, SAL  
      FROM EMP  
      WHERE MGR = NULL;
```

○ (O)

```
SQL> SELECT ENAME, SAL  
      FROM EMP  
      WHERE MGR IS NULL;
```



```
SQL> SELECT ENAME, SAL  
      FROM EMP  
      WHERE MGR IS NOT NULL;
```


SELECT 문

▶ IN, NOT IN

- 조회 시, 특정 Column이 가지고 있는 값을 여러 개 지정해주고 싶다면?

```
SQL> SELECT ENAME, COMM  
      FROM EMP  
      WHERE COMM = 300  
            OR COMM = 500  
            OR COMM = 1400;
```



```
SQL> SELECT ENAME, COMM  
      FROM EMP  
      WHERE COMM IN (300, 500, 1400);
```



```
SQL> SELECT ENAME, COMM  
      FROM EMP  
      WHERE comm NOT IN (300, 500, 1400);
```

SELECT 문

- ▶ BETWEEN A AND B, NOT BETWEEN A AND B
 - 사이 값?

```
SQL> SELECT ENAME, SAL  
      FROM EMP  
      WHERE SAL >= 1500  
            AND SAL <= 3000;
```



```
SQL> SELECT ENAME, SAL  
      FROM EMP  
      WHERE SAL BETWEEN 1500 AND 3000;
```



```
SQL> SELECT ENAME, SAL  
      FROM EMP  
      WHERE SAL NOT BETWEEN 1500 AND 3000;
```

SELECT 문

▶ LIKE, NOT LIKE

- 문자열의 일부?

```
SQL> SELECT ENAME  
      FROM EMP  
      WHERE ENAME LIKE '%A%';
```

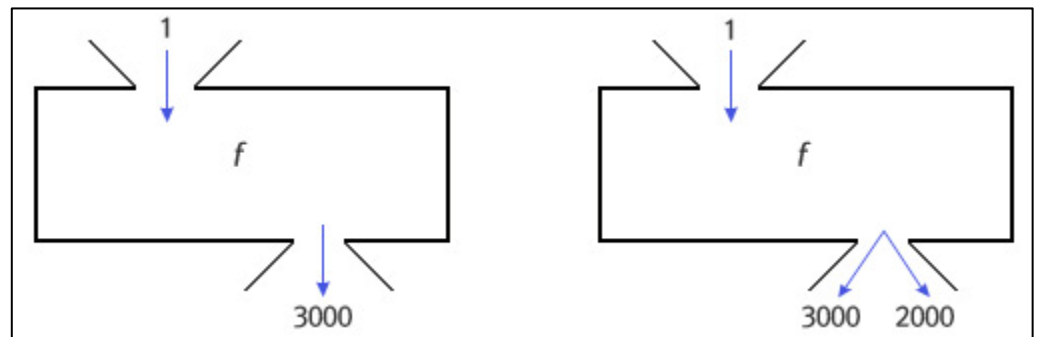
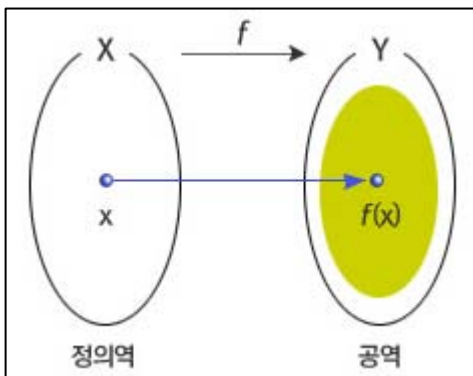
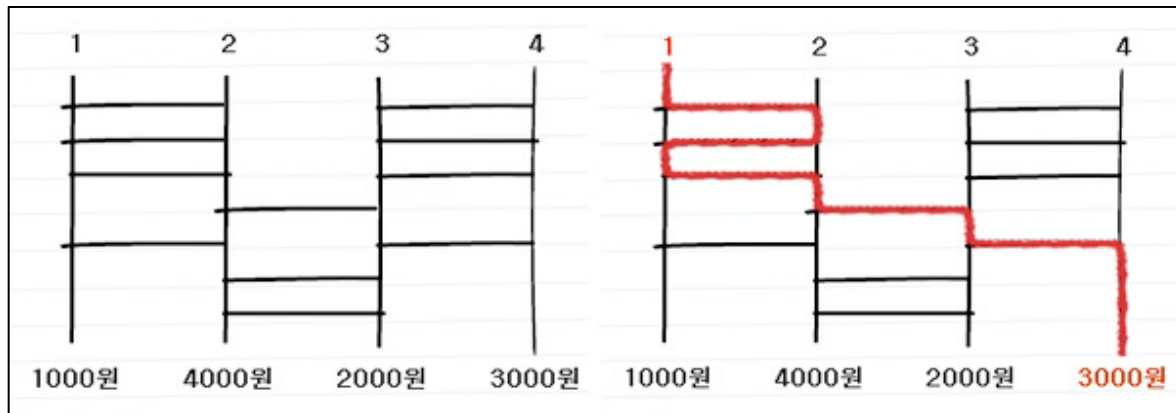
```
SQL> SELECT ENAME  
      FROM EMP  
      WHERE ENAME LIKE '_A%';
```

```
SQL> SELECT ENAME  
      FROM EMP  
      WHERE ENAME LIKE '__R%';
```

- _ : 한 개!
- % : 몇 개든!

함수 (Function)

▶ 리하~



함수 (Function)

- ▶ 누가 만들었느냐에 따라...
 - Built-In Function
 - 내장함수
 - Oracle 내에 이미 작성되어 있는 Function
 - User-Define Function
 - 사용자정의 함수
 - 사용자가 필요에 의해 직접 만든 Function
- ▶ Input(입력값)의 개수에 따라...
 - Single Row Function
 - Input 하나를 넣고 Output 이 하나 나오는 Function
 - Multiple Row Function
 - Input 여러 개를 넣고 Output 이 하나 나오는 Function

함수 (Function)

▶ 대소문자 치환 함수

```
SQL> SELECT UPPER(ENAME) , LOWER(ENAME) , INITCAP(ENAME)  
FROM EMP;
```

- UPPER

- 해당 Column에 있는 Row의 값을 모두 대문자로 치환하여 return

- LOWER

- 해당 Column에 있는 Row의 값을 소문자로 치환하여 return

- INITCAP

- 해당 Column에 있는 Row의 값의 첫부분만 대문자로 처리하여 return

함수 (Function)

▶ 문자열 함수

```
SQL> SELECT SUBSTR(ENAME,1,3)      ,  
            SUBSTR(ENAME,4)        ,  
            LENGTH(ENAME)          ,  
            INSTR(ENAME,'A')       ,  
            CONCAT(ENAME,JOB)      ,  
            LPAD(SAL,10,'*')       ,  
            RPAD(SAL,10,'*')       ,  
FROM EMP;
```

- SUBSTR(ENAME,1,3) : ENAME Column의 값을 첫글자부터 세번째자리 문자까지 반환
- SUBSTR(ENAME,4) : ENAME Column의 값을 이름 넷째자리부터 문자열끝까지 반환
- LENGTH(ENAME) : ENAME Column의 값 길이
- INSTR(ENAME,'A') : ENAME Column에서 A가 몇번째에 위치하는지를 반환
- CONCAT(ENAME, JOB) : ENAME Column 값과 JOB Column의 값을 연결 (|| 는 연산자, CONCAT은 함수)
- LPAD(SAL,10,'*') : Left Padding, SAL Column의 값을 10자리수로 만들고 남은 자리는 왼쪽부터 '*'로 채움
- RPAD(SAL,10,'*') : Right Padding, SAL Column의 값을 10자리수로 만들고 남은 자리는 오른쪽부터 '*'로 채움

함수 (Function)

▶ 소수점 제어 함수

```
SQL> SELECT CEIL (45.129)      ,  
           ROUND (45.129,2)   ,  
           ROUND (45.129,1)   ,  
           ROUND (45.129,0)   ,  
           ROUND (45.129,-1)  ,  
           TRUNC (45.129,2)   ,  
FROM EMP;
```

```
SQL> SELECT CEIL (45.129)      ,  
           ROUND (45.129,2)   ,  
           ROUND (45.129,1)   ,  
           ROUND (45.129,0)   ,  
           ROUND (45.129,-1)  ,  
           TRUNC (45.129,2)   ,  
FROM DUAL;
```

- **CEIL** : 올림
-> **CEIL**(45.129) - 소수점 이후 값 올림
- **ROUND** : 반올림
-> **ROUND**(45.129,x) - 소수점을 기준으로 x 번째에서 반올림 (0은 첫째자리)
- **TRUNC** : 버림
-> **TRUNC**(45.129,x) - 버림

함수 (Function)

▶ 나머지 연산 함수

```
SQL> SELECT 32-32+254/2+25  
FROM DUAL;
```

```
SQL> SELECT MOD(101,2)  
FROM DUAL;
```

```
SQL> SELECT MOD(SAL, COMM)  
FROM EMP;
```

함수 (Function)

▶ 날짜 함수

```
SQL> SELECT SYSDATE  
      FROM Dual;
```

- Oracle에서의 Date 타입
 - 세기, 년, 월, 일, 시간, 분, 초
 - SYSDATE : 현재 시스템의 Date 정보를 Return
- Oracle에서의 날짜연산
 - 날짜 + 숫자 : 특정 날짜에 숫자 만큼의 일수 후 날짜
 - 날짜 - 숫자 : 특정 날짜에 숫자 만큼의 일수 전 날짜
 - 날짜 - 날짜 : 두 날짜 사이의 일수 계산
 - 날짜 + 날짜 : 불가. SYSDATE + HIREDATE ? SF 영화?

함수 (Function)

▶ 날짜 함수

```
SQL> SELECT HIREDATE  
        MONTHS_BETWEEN(SYSDATE, HIREDATE)  
        ADD_MONTHS(HIREDATE, 6)  
        NEXT_DAY(HIREDATE, '금')  
        LAST_DAY(HIREDATE)  
FROM EMP;
```

- MONTHS_BETWEEN(SYSDATE, HIREDATE)
날짜간의 차이를 월수로 환산해주는 함수
- ADD_MONTHS(HIREDATE, 6)
첫번째 날짜에 뒤의 숫자만큼의 달(MONTH)을 더해줌
- NEXT_DAY(HIREDATE, '금')
HIREDATE 를 기준으로 돌아오는 금요일 날짜
- LAST_DAY(HIREDATE)
해당날짜 월에 가장 마지막 날짜를 표시

함수 (Function)

- ▶ 함수의 조합?!
 - 함수를 품은 함수

```
SQL> SELECT HIREDATE, TRUNC(MONTHS_BETWEEN(SYSDATE, HIREDATE), 0)  
      FROM EMP;
```

```
SQL> SELECT ENAME, NEXT_DAY(ADD_MONTHS(HIREDATE, 6), '금')  
      FROM EMP;
```