

SQL 주요 함수

DUAL 테이블

❖ 24*60의 산술 연산

```
SELECT 24*60  
FROM employees;
```

- DUAL 테이블
 - 산술 연산을 위한 더미 테이블

DESC DUAL

```
SELECT 24*60  
FROM dual;
```

```
SELECT SYSDATE  
FROM dual;
```

숫자 함수

구분	설명
ABS	절대값을 구한다
COS	COSINE 값을 구한다
EXP	$e(2.71828183\dots)$ 의 n 승을 반환한다
FLOOR	소수점 아래를 잘라낸다
POWER	POWER(m,n) m 의 n 승을 구한다.
SIGN	SING(n) n 이 음수이면 -1, 0이면 0, 양수이면 1을 반환한다
SIN	SINE 값을 반환한다.
TAN	TANGENT 값을 반환한다.
ROUND	특정 자리수에서 반올림한다.
TRUNC	특정 자리수에서 잘라낸다
MOD	입력 받은 수를 나눈 나머지 값을 반환한다.

숫자 함수

❖ ABS 함수(절대값)

```
SELECT -10, ABS(-10)      → 10
FROM dual;
```

❖ FLOOR 함수(소수점이하 버림)

```
SELECT 34.5678, FLOOR(34.5678) → 34
FROM dual;
```

❖ ROUND 함수(반올림)

```
SELECT 34.5678, ROUND(34.5678) → 35
FROM dual;
```

```
SELECT 34.5678, ROUND(34.5678, 2) → 34.57
FROM dual;
```

```
SELECT 34.5678, ROUND(34.5678, -1) → 30
FROM dual;
```

숫자 함수

❖ TRUNC 함수(버림)

```
SELECT  TRUNC(34.5678, 2),      → 34.56
        TRUNC(34.5678, -1),    → 30
        TRUNC(34.5678)         → 34
FROM dual;
```

❖ MOD 함수(나머지)

```
SELECT  MOD(27, 2),      → 1
        MOD(27, 5),      → 2
        MOD(27, 7)       → 6
FROM dual;
```

❖ 사원 번호가 홀수인 사원 조회

```
SELECT *
FROM employees
WHERE MOD(EMPLOYEE_ID, 2) = 1;
```

문자 처리 함수

구분	설명
LOWER	소문자로 변환한다
UPPER	대문자로 변환한다
INITCAP	첫글자만 대문자로 나머지는 소문자로 변환한다
CONCAT	문자의 값을 연결한다
SUBSTR	문자를 잘라 추출(한글 1byte)한다.
SUBSTRB	문자를 잘라 추출(한글 2byte)한다.
LENGTH	문자의 길이를 반환(한글 1byte)한다.
LENGTHB	문자의 길이를 반환(한글 2byte)한다
INSTR	특정 문자 위치값을 반환(한글 1byte)한다
INSTRB	특정 문자 위치값을 반환(한글 2byte)한다
LPAD, RPAD	입력 받은 문자열과 기호를 정렬하여 특정 길이의 문자열로 반환한다.
TRIM	잘라내고 남은 문자를 표시한다
CONVERT	CHAR SET을 반환한다
CHR	ASCII 코드값으로 변환한다.
ASCII	ASCII 코드값을 문자로 변환한다
REPLACE	문자열에서 특정 문자를 변경한다

문자 처리 함수

❖ UPPER 함수

```
SELECT 'Welcome to Oracle', UPPER('Welcome to Oracle')  
FROM dual;
```

❖ LOWER 함수

```
SELECT 'Welcome to Oracle', LOWER('Welcome to Oracle')  
FROM dual;
```

❖ INITCAP 함수

```
SELECT 'WELCOME TO ORACLE', INITCAP('WELCOME TO ORACLE')  
FROM dual;
```

문자 처리 함수

❖ LENGTH 함수(문자의 길이)

```
SELECT LENGTH('Oracle'), LENGTH('오라클')  
FROM dual;
```

→ 6, 3

❖ LENGTHB 함수(바이트 수를 리턴)

- 메모리를 차지하는 바이트 수를 알려주는 함수

```
SELECT LENGTHB('Oracle'), LENGTHB('오라클')  
FROM dual;
```

→ 6, 9

문자 처리 함수

❖ SUBSTR

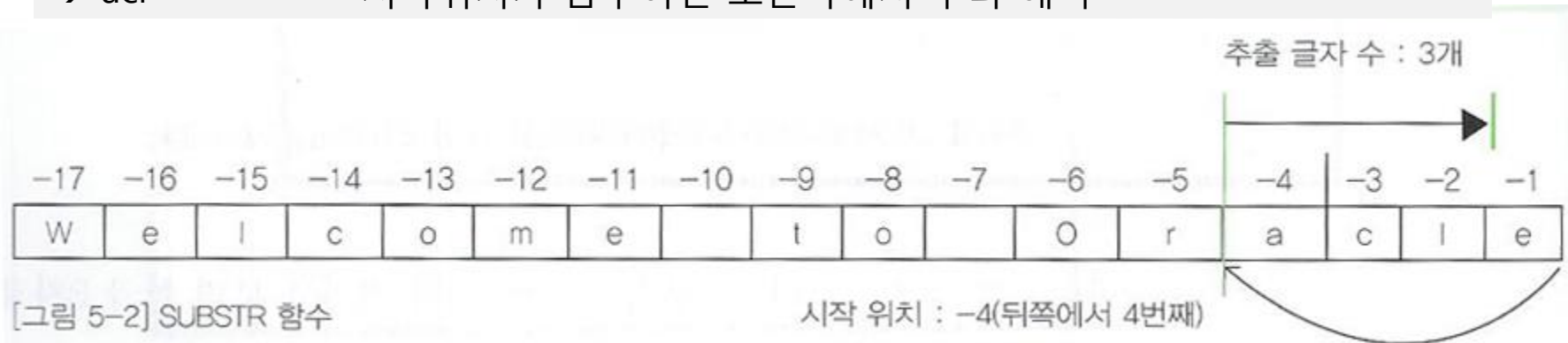
- SUBSTR(대상, 시작위치, 추출할 개수)

```
SELECT SUBSTR('Welcome to Oracle', 4, 3)  
FROM dual;
```

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
W	e	l	c	o	m	e		t	o		O	r	a	c	l	e

```
SELECT SUBSTR('Welcome to Oracle', -4, 3)  
FROM dual;
```

→ acl 시작위치가 음수이면 오른쪽에서 부터 해석



문자 처리 함수

❖ SUBSTR

- 사원의 입사 년도와 입사 월 조회

```
SELECT SUBSTR(hire_date, 1, 2)입사년도, SUBSTR(hire_date, 4, 2)월  
FROM employees;
```

- 9월에 입사한 사원 조회

```
SELECT *  
FROM employees  
WHERE SUBSTR(hire_date, 4, 2)='09'
```

문자 처리 함수

❖ 실습

- 03년에 입사한 사원을 조회하시오.
 - SUBSTR 함수를 이용한 경우
 - BETWEEN 연산자를 이용한 경우
- last_name이 'e'로 끝나는 사원을 조회하시오
 - LIKE 연산자를 이용한 경우
 - SUBSTR 함수를 이용한 경우

문자 처리 함수

❖ 한글에 대한 SUBSTR

```
SELECT SUBSTR('Welcome To Oracle', 3, 4),  
       SUBSTRB('Welcome To Oracle', 3, 4)  
FROM dual;  
→  
lcom      lcom
```

```
SELECT SUBSTR('웰컴투오라클', 3, 4),  
       SUBSTRB('웰컴투오라클', 3, 4)  
FROM dual;  
→  
투오라클   컴
```



문자 처리 함수

❖ INSTR

- 대상 문자열이나 컬럼에서 특정 문자가 나타나는 위치를 찾아냄

```
SELECT INSTR('WELCOME TO ORACLE', 'O')  
FROM dual;
```

→ 5

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
W	E	L	C	O	M	E		T	O		O	R	A	C	L	E

[형식]

INSTR(대상, 찾을 글자, 시작위치, 몇 번째 발견)

```
SELECT INSTR('WELCOME TO ORACLE', 'O', 6, 2)  
FROM dual;
```

→ 12

시작위치(6)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
W	E	L	C	O	M	E		T	O		O	R	A	C	L	E

↑ 두번째 O의 위치

문자 처리 함수

❖ INSTRB 함수

- 바이트 수를 기준으로 문자의 위치를 구하는 함수

```
SELECT INSTR('데이터베이스', '이', 3, 1),  
       INSTRB('데이터베이스', '이', 3, 1)  
FROM dual;
```

→ 5, 4

❖ 실습

- first_name의 세번째 자리가 r인 사원을 검색하는 쿼리를 작성한다
 - INSTR 함수를 이용하는 경우
 - SUBSTR함수를 이용하는 경우

문자 처리 함수

❖ LPAD/RPAD 함수

- 특정 기호로 채우는 함수

```
SELECT LPAD('Oracle', 20, '#')  
FROM dual;
```

→ 20자리 마련후 오른쪽에 대상 출력 후 빈 공간을 #으로 채움

```
SELECT RPAD('Oracle', 20, '#')  
FROM dual;
```

→ 20자리 마련후 왼쪽에 대상 출력 후 빈 공간을 #으로 채움

문자 처리 함수

❖ LTRIM/RTRIM 함수

- 문자열의 왼쪽/오른쪽 공백을 삭제

```
SELECT LTRIM(' Oracle ')
```

```
FROM dual;
```

```
SELECT RTRIM(' Oracle ')
```

```
FROM dual;
```

❖ TRIM 함수

- 컬럼이나 대상 문자열에서 특정 문자가 첫번째 글자이거나 마지막 글자이면 잘라내고 남은 문자열만 반환한다

```
SELECT TRIM('a' FROM 'aaaaOracleaaaa')
```

```
FROM dual;
```

```
SELECT TRIM(' Oracle ')
```

```
FROM dual;
```

```
→ Oracle
```


날짜 함수

구분	설명
SYSDATE	시스템의 현재 날짜를 반환한다.
MONTHS_BETWEEN	두 날짜 사이가 몇 개월인지 반환한다.
ADD_MONTHS	특정 날짜에 개월 수를 더한다.
NEXT_DAY	특정 날짜에서 최초로 도래하는 인자로 받은 요일의 날짜를 반환
LAST_DAY	해당 달의 마지막 날짜를 특정 기준으로 반올림한다.
ROUND	인자로 받은 날짜를 특정 기준으로 반올림한다
TRUNC	인자로 받은 날짜를 특정 기준으로 버린다.

날짜 함수

❖ SYSDATE 함수

```
SELECT SYSDATE  
FROM dual;
```

→ 14/07/12

❖ 날짜연산

```
SELECT SYSDATE-1 어제, SYSDATE+1 내일  
FROM dual;
```

→ 14/07/11, 14/07/13

❖ 실습

- 각 사원의 근무일 수를 계산하시오

날짜 함수

❖ ROUND 함수

- 형식

ROUND (date, format)

구분	설명
CC, SCC	4자리 연도 끝의 두 글자를 기준으로 반올림
SYYY, YYYY, YEAR SYEAR, YYY, YY, Y	년(7월1일부터 반올림)
DDD, D, J	일을 기준
HH, HH12, HH24	시를 기준
Q	한 분기의 두 번째 달의 16일 기준으로 반올림
MONTH, MON, MM, RM	월(16일을 기준으로 반올림)
DAY, DY, D	한 주가 시작되는 날짜
M	분을 기준

```
SELECT hire_date, ROUND(hire_date, 'MONTH')  
FROM employees;
```

날짜 함수

❖ TRUNC 함수

- 포맷 형식에 맞춰 날짜를 잘라낸다.
- 형식

TRUNC(date, format)

```
SELECT hire_date, TRUNC(hire_date, 'MONTH')  
FROM employees;
```

❖ MONTH_BETWEEN 함수

- 날짜와 날짜 사이의 날수를 구하는 함수
- 형식
 - MONTH_BETWEEN(date1, date2)

```
SELECT last_name, SYSDATE, hire_date,  
       MONTHS_BETWEEN(SYSDATE, hire_date) → 소수점 자리까지 표시  
FROM employees;
```

```
SELECT last_name, SYSDATE, hire_date,  
       TRUNC(MONTHS_BETWEEN(SYSDATE, hire_date))  
FROM employees;
```

날짜 함수

❖ ADD_MONTHS 함수

- 특정 개월수를 더한 날짜를 구하는 함수
- 형식

ADD_MONTHS(date, number)

```
SELECT last_name, hire_date, ADD_MONTHS(hire_date, 6)
FROM employees;
```

❖ NEXT_DAY 함수

- 해당 용일을 기준으로 가장 가까운 다음 요일에 해당되는 날짜를 반환
- 형식

NEXT_DAY(date, 요일)

```
SELECT SYSDATE, NEXT_DAY(SYSDATE, '수요일')
FROM dual;
```

날짜 함수

❖ LAST_DAY

- 해당 월의 마지막 날짜를 반환

```
SELECT hire_date, LAST_DAY(hire_date)
FROM employees;
```

CF] NLS_LANG의 변경

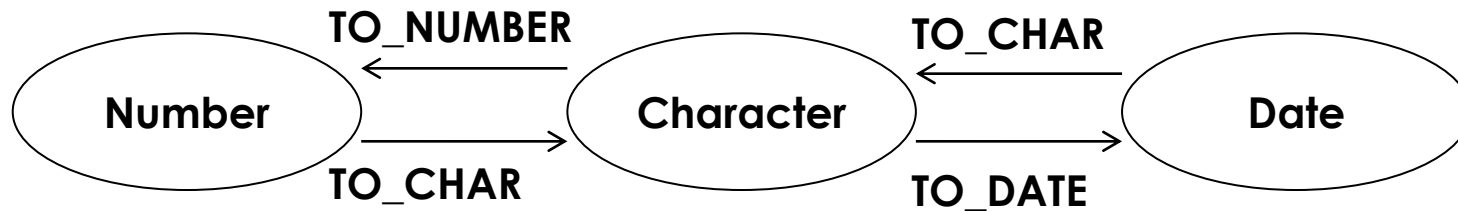
```
ALTER SESSION SET NLS_LANGUAGE=AMERICAN;
```

```
ALTER SESSION SET NLS_LANGUAGE=KOREAN;
```

형 변환 함수

❖ 형 변환 함수

구분	설명
TO_CHAR	날짜 혹은 숫자 형을 문자형으로 변환한다
TO_DATE	문자형을 날짜형으로 변환한다
TO_NUMBER	문자형을 숫자형으로 변환한다.



형 변환 함수

❖ TO_CHAR 함수

- 날짜형을 문자형으로 변환

- 형식

TO_CHAR(날짜 데이터, 출력형식)

구분	설명
YYYY	년도 표현(4자리)
YY	년도 표현(2자리)
MM	월을 숫자로 표현
MON	월을 알파벳으로 표현
DAY	요일 표현
DY	요일을 약어로 표현

형 변환 함수

- 현재 날짜를 다른 형식으로 출력

```
SELECT SYSDATE, TO_CHAR(SYSDATE, 'YYYY-MM-DD')  
FROM dual;
```

→ 14/07/13 2014-07-13

```
SELECT hire_date, TO_CHAR(hire_date, 'YYYY/MM/DD DAY')  
FROM employees;
```

→ 03/06/17 2003/06/17 화요일
 05/09/21 2005/09/21 수요일

```
SELECT hire_date, TO_CHAR(hire_date, 'YY/MON/DD DY')  
FROM employees;
```

→ 03/06/17 03/6월 /17 화
 05/09/21 05/9월 /21 수

형 변환 함수

- 시간의 출력

구분	설명
AM또는 PM	오전, 오후 시각 표시
A.M또는 P.M	오전, 오후 시각 표시
HH 또는 HH12	12시간으로 표시(1~12)
HH24	24시간으로 표시(0~23)
MI	분 표현
SS	초 표현

```
SELECT TO_CHAR(SYSDATE, 'YYYY/MM/DD, HH24:MI:SS')  
FROM dual;
```

→ 2014/07/13, 05:55:43

형 변환 함수

❖ 숫자형을 문자형으로 변환하기

구분	설명
0	자릿수를 나타내며 자릿수가 맞지 않을 경우 0으로 채움
9	자릿수를 나타내며 자릿수가 맞지 않아도 채우지 않음
L	각 지역별 통화 기호를 앞에 표시
. ,	소수점과 천 단위 자리 구분

```
SELECT TO_CHAR(1230000)
FROM dual;
```

```
SELECT last_name, salary, TO_CHAR(salary, 'L999,999')
FROM employees;
```

```
→ King          24000          ₩24,000
   Kochhar       17000          ₩17,000
```

```
SELECT TO_CHAR(123456, '0000000000'), TO_CHAR(123456, '999,999,999')
FROM dual;
```

```
→ 000123456      123,456
```

형 변환 함수

❖ TO_DATE 함수

- 형식

TO_DATE('문자', 'format')

```
SELECT last_name, hire_date
FROM employees
WHERE hire_date=20021020;
→ 에러 발생
```

```
SELECT last_name, hire_date
FROM employees
WHERE hire_date=TO_DATE(20020816, 'YYYYMMDD');
```

→ Faviet02/08/16

```
SELECT TRUNC(SYSDATE-TO_DATE('2008/01/01', 'YYYY/MM/DD'))
FROM dual;
```

형 변환 함수

❖ TO_NUMBER 함수

```
SELECT TO_NUMBER('20,000', '99,999') - TO_NUMBER('10,000', '99,999')  
FROM dual;
```

NULL을 다른 값으로 변환하는 NVL 함수

❖ 사원의 커미션 비율 조회

```
SELECT last_name, salary, commission_pct, job_id
FROM employees
ORDER BY job_id;
```

Colmenares	2500	PU_CLERK	
Raphaely	11000	PU_MAN	
Cambrault	11000	0.3	SA_MAN
Errazuriz	12000	0.3	SA_MAN

❖ NULL을 0으로 처리하여 연봉 계산

```
SELECT last_name, salary, commission_pct,
       salary*12 + (salary*12)*nvl(commission_pct, 0)
FROM employees
ORDER BY job_id;
```

Colmenares	2500	30000	
Raphaely	11000	132000	
Cambrault	11000	0.3	171600
Errazuriz	12000	0.3	187200

선택을 위한 DECODE 함수

❖ switch case문과 같은 기능 제공

- 여러가지 경우에서 선택할 수 있는 기능
- 형식

```
DECODE(표현식, 조건1, 결과 1  
        조건2, 결과2,  
        조건3, 결과3,  
        기본결과 n  
      )
```

```
SELECT department_id,  
       DECODE(department_id,  
              10, 'A',  
              20, 'B',  
              'DEFAULT')  
FROM employees  
ORDER BY department_id;
```

10	A
20	B
20	B
30	DEFAULT
30	DEFAULT

선택을 위한 DECODE 함수

```
SELECT last_name, department_id,  
       DECODE(department_id,  
              10, 'Administration',  
              20, 'Marketing',  
              30, 'Purchasing',  
              40, 'Human Resources') AS DNAME  
FROM employees;
```

Whalen	10	Administration
Hartstein	20	Marketing
Fay	20	Marketing
Mavris	40	Human Resources

조건에 따라 서로 다른 처리가 가능한 CASE 함수

❖ CASE 함수

○ 형식

```
CASE      WHEN 조건1 THEN 결과1
           WHEN 조건2 THEN 결과2
           WHEN 조건3 THEN 결과3
           ELSE 결과N
END
```

```
SELECT last_name, department_id,
       CASE      WHEN department_id=10 THEN 'Administration'
                  WHEN department_id=20 THEN 'Marketing'
                  WHEN department_id=30 THEN 'Purchasing'
                  WHEN department_id=40 THEN 'Human Resources'
       END AS DNAME
FROM employees;
```