

04강 다양한 함수

내장 함수

내장 함수

- 오라클은 많은 내장함수를 가진다. – 전체 함수는 공식문서를 통해 확인하는 것이 좋다.(영문)
- 크게 나누면 문자열 함수, 숫자 및 수학함수, 날짜/시간 함수, 형변환함수, 분석함수, 확장 함수, 기타함수등으로 나눌수 있고 이중 형변환 함수는 앞서 살펴 봤다.

내장 함수 - 문자열 함수

종류	설명
ASCII(영문자)	한 문자의 아스키값을 반환
CHR(숫자)	숫자 아스키값에 해당하는 문자반환
ASCIISTR(한글)	유니코드문자를 입력하면 16진수의 유니코드 반환
UNISTR('유니코드')	유니코드값을 입력하면 해당 문자를 반환

```
SELECT ASCII('T'), CHR(84), ASCIISTR('천'), UNISTR('\CC9C') FROM DUAL;
```

내장 함수 - 문자열 함수

종류	설명
LENGTH(문자열)	문자열의 길이 반환
LENGTHB(문자열)	문자열의 바이트 수 반환
LENGTHC(문자열)	유니코드 문자열 길이 반환

```
SELECT LENGTH('한글'), LENGTH('AB'), LENGTHB('한글'), LENGTHB('AB'), LENGTHC('한글') FROM DUAL;
```

내장 함수 - 문자열 함수

종류	설명
CONCAT(문자열1,문자열2)	두개의 문자열을 결합
문자열1 문자열2	두개 이상의 문자열 결합
INSTR(기준문자열, 부분문자열, 찾기 시작위치)	기준 문자열에서 부분 문자열을 찾아 문자열이 시작하는 위치를 반환
INSTRB(기준문자열, 부분문자열, 찾기 시작위치)	기준 문자열에서 부분 문자열을 찾아 문자열이 시작하는 위치를 반환, 단 Byte단위로 센다.

```
SELECT CONCAT('이것은', '오라클이다'), '이것이' || '오라클' || '이다' FROM DUAL;
```

```
SELECT INSTR('이것이 Oracle이다. 이것도 오라클이다.', '이것') FROM DUAL;
```

```
SELECT INSTR('이것이 Oracle이다. 이것도 오라클이다.', '이것', 2) FROM DUAL;
```

```
SELECT INSTRB('이것이 Oracle이다. 이것도 오라클이다.', '이것', 2) FROM DUAL;
```

내장 함수 - 문자열 함수

종류	설명
LOWER(문자열)	문자열을 소문자로
UPPER(문자열)	문자열을 대문자로
INITCAP(문자열)	단어의 첫글자만 대문자 나머지는 소문자로
REPLACE(문자열, 원래문자열, 바꿀 문자열)	문자열에서 원래 문자열을 찾아서 바꿀 문자열로 바꿔준다.
TRANSLATE(문자열, 원래 문자열, 바꿀 문자열)	REPLACE는 바꿀 문자열을 통째로 바꾸지만 TRANSLATE는 한 글자씩 찾아서 변환해준다.

내장 함수 - 문자열 함수

```
SELECT LOWER('abcdEFGH'),UPPER('abcdEFGH'),INITCAP('thIs is oraCLe') FROM DUAL;
```

```
SELECT REPLACE('이것이 Oracle이다','이것이','This is') FROM DUAL;
```

```
SELECT TRANSLATE('이것이 Oracle이다','이것','AB') FROM DUAL;
```

```
SELECT ename, LOWER(ename),job,INITCAP(job) FROM employee;
```

```
SELECT eno,ename,dno FROM employee  
WHERE ename='scott';
```

```
SELECT eno,ename,dno FROM employee  
WHERE ename=UPPER('scott');
```

```
SELECT eno,ename,dno FROM employee  
WHERE INITCAP(ename)='Scott';
```

내장 함수 – 문자열 함수

종류	설명
SUBSTR(문자열, 시작위치, 길이)	시작위치에서 길이만큼 문자열을 반환
REVERSE(문자열)	문자열 순서를 거꾸로 한다.
LPAD(문자열,길이,채울 문자열)	문자열을 길이만큼 늘린 다음 빈 곳을 채울 문자열로 채운다. 길이는 바이트 단위로 센다.
RPAD(문자열,길이,채울 문자열)	
LTRIM(문자열, 제거할 문자)	문자열의 왼쪽/오른쪽에서 제거할 문자를 제거한다. 생략시 공백을 제거
RTRIM(문자열, 제거할 문자)	
TRIM(제거할방향 제거할문자 FROM 문자열)	문자열의 앞뒤의 제거할 문자열을 제거한다. 방향 생략시 앞뒤 모두, 제거할 문자열이 없으면 공백을 제거 방향: LEADING(앞) TRAILING(뒤) BOTH(양쪽)
REGEXP_COUNT(문자열, 문자)	문자열에서 문자의 개수를 센다.

내장 함수 - 문자열 함수

```
SELECT SUBSTR('대한독립만세',3,2) FROM DUAL;
SELECT REVERSE('oracle') FROM DUAL;
SELECT LPAD('이것이',10,'##'), RPAD('이것이',10,'##') FROM DUAL;
SELECT LTRIM('        이번생은'), RTRIM('망했어요ㅋㅋㅋㅋㅋㅋ', 'ㅋ') FROM DUAL;
SELECT TRIM('        이번생은        '), TRIM(BOTH 'ㅋ' FROM 'ㅋㅋㅋㅋ망했어요ㅋㅋㅋㅋㅋㅋ') FROM DUAL;
SELECT REGEXP_COUNT('이것이 오라클이다.', '이') FROM DUAL;
```

```
SELECT * FROM employee
WHERE substr(ename,-1,1)='N';
```

```
SELECT * FROM employee
WHERE substr(hiredate,1,2)='87';
```

```
SELECT LPAD(salary,10,'*') FROM employee;
SELECT RPAD(salary,10,'*') FROM employee;
```

내장 함수 - 수학, 숫자 함수

종류	설명
ABS(숫자)	절대값
ACOS(숫자), ASIN(숫자), ATAN(숫자), ATAN2(숫자1, 숫자2), SIN(숫자), COS(숫자), TAN(숫자)	삼각함수
EXP(숫자), LN(숫자), LOG(숫자)	지수, 로그함수

내장 함수 - 수학, 숫자 함수

종류	설명
CEIL(숫자), FLOOR(숫자), ROUND(숫자)	올림, 내림, 반올림
MOD(숫자1, 숫자2)	숫자1을 숫자2로 나눈 나머지
POWER(숫자1, 숫자2)	거듭제곱
SQRT(숫자)	제곱근
SIGN(숫자)	음수인지 양수인지 판단해서 -1, 1을 반환(0은 0)
TRUNC(숫자, 정수)	숫자를 소수점을 기준으로 정수위치까지 구하고 나머지 버린다.

```
SELECT ABS(-100) FROM DUAL;
```

```
SELECT CEIL(4.7), FLOOR(4.7), ROUND(4.7) FROM DUAL;
```

```
SELECT MOD(256, 10) FROM DUAL;
```

```
SELECT POWER(2, 3) FROM DUAL;
```

```
SELECT SQRT(9) FROM DUAL;
```

```
SELECT SIGN(10), SIGN(0), SIGN(-10.23) FROM DUAL;
```

```
SELECT TRUNC(12345.12345, 2), TRUNC(12345.12345, -2) FROM DUAL;
```

내장 함수 - 날짜 시간 함수

종류	설명
ADD_MONTHS(숫자/-숫자)	개월을 더하거나 뺀 결과
+/-	날짜를 더하거나 뺀 결과
CURRENT_DATE	연/월/일 현재날짜를 구한다.
SYSDATE	연/월/일 현재 날짜는 구한다
CURRENT_TIMESTAMP	연/월/일 시:분:초 지역 형식으로 현재 날짜와 시간을 구한다.
EXTRACT(형식 FROM DATE '날짜')	날짜에서 연,월,일,시,분,초를 구한다 YEAR,MONTH,DAY.HOUR,MINUTE,SECOND
LAST_DAY(날짜)	주어진 날짜의 마지막 날짜를 구한다(해당 달이 며칠까지 있는지 확인)
NEXT_DAY(날짜, 요일)	주어진 날짜의 다음에 오는 요일의 날짜를 구한다.
MONTHS_BETWEEN(날짜1,날짜2)	두 날짜 사이에 오는 개월수를 계산해서 소수점 단위로 반환
ADD_MONTHS(날짜, 숫자)	날짜에 숫자만큼의 개월을 더한다.
NEXT_DAY(날짜,'요일')	지정된 날짜 이후에 돌아오는 요일에 해당하는 날을 반환 (요일은 숫자로 지정가능 일요일:7, 월요일:1, 화요일:2, 수요일:3, 목요일:4, 금요일:5, 토요일:6)

내장 함수 - 날짜 시간 함수

- 추가 날짜 함수

- ROUND(날짜, 포맷), TRUNC(날짜, 포맷)

포맷 모델	
CC, SCC	4자리 연도의 끝 두자리를 기준으로 반올림
SYYY,YYYY, YEAR, SYEAR, YYY, YY, Y	년(7월1일 부터 반올림)
DDD, D, J	일 기준
HH, HH12, HH24	시 기준
Q	한 분기의 두번째 달에 16일을 기준으로 반올림
MONTH, MON, MM, RM	월(16일을 기준으로 반올림)
DAY, DY, D	한주가 시작되는 날짜 (한 주의 절반이 지나면 다음주 시작일, 한 주의 절반이 안지나면 이번주 시작일)
MI	분을 기준

내장 함수 - 날짜 시간 함수

```
SELECT ADD_MONTHS('2020-01-01',5), ADD_MONTHS(SYSDATE,-5) FROM DUAL;
```

```
SELECT TO_DATE('2020-01-01')+5,SYSDATE-5 FROM DUAL;
```

```
SELECT CURRENT_DATE, SYSDATE, CURRENT_TIMESTAMP FROM DUAL;
```

```
SELECT EXTRACT(YEAR FROM DATE '2019-06-27'), EXTRACT(DAY FROM SYSDATE) FROM DUAL;
```

```
SELECT LAST_DAY('2019-05-05') FROM DUAL;
```

```
SELECT NEXT_DAY('2020-01-03','월요일'), NEXT_DAY(SYSDATE,'일요일') FROM DUAL;
```

```
SELECT MONTHS_BETWEEN(SYSDATE,'1985-02-17') FROM DUAL;
```

```
SELECT hiredate,TRUNC(hiredate,'MONTH') FROM employee;
```

```
SELECT ROUND(sysdate-hiredate) AS 근무일수 FROM employee;
```

```
SELECT sysdate, NEXT_DAY(sysdate,'토요일') FROM dual;
```

```
SELECT ename,hiredate, ADD_MONTHS(hiredate,6) FROM employee;
```

```
SELECT ename, sysdate,hiredate,  
       TRUNC(MONTHS_BETWEEN(sysdate,hiredate)) FROM employee;
```

```
SELECT ename,hiredate,  
       LAST_DAY(hiredate) FROM employee;
```

내장 함수 - 추가 형변환 함수

종류	설명
BIN_TO_NUM(2진수)	주어진 2진수를 10진수로 변경한다.(2진수는 콤마로 구분해서 사용)
NUMTODSINTERVAL(숫자,'표현식')	숫자가 며칠에 해당하는 지 반환 (표현식에는 DAY, HOUR, MINUTE, SECOND 등 사용)
NUMTOYMINTERVAL(숫자,'표현식')	숫자가 몇 년 몇 월에 해당하는 지 반환 (표현식에는 MONTH, YEAR 등 사용)

```
SELECT BIN_TO_NUM(1,0), BIN_TO_NUM(1,0,0,1,1,1) FROM DUAL;
```

```
SELECT NUMTODSINTERVAL(48,'HOUR'), NUMTODSINTERVAL(360000,'SECOND') FROM DUAL;
```

```
SELECT NUMTOYMINTERVAL(37,'MONTH'), NUMTOYMINTERVAL(1.5,'YEAR') FROM DUAL;
```

내장 함수 - 기타 일반 함수

종류	설명
NVL(컬럼명(표현식), 대체데이터)	해당 컬럼의 데이터가 NULL이면 대체 데이터로 변환해서 사용된다.
NVL2(컬럼명, 데이터1, 데이터2)	해당 컬럼의 데이터가 NULL이면 데이터2반환, 아니면 데이터1반환
NULLIF(표현식1,표현식2)	표현식1과 표현식2의 결과가 같으면 NULL반환 아니면 표현식1 반환
COALESCE(표현식1,표현식2...표현식N)	여러 인자들중 NULL이 아닌 첫번째 인자를 반환
DECODE(표현식, 조건1, 결과1, 조건2, 결과2, 조건3, 결과3, ..., 기본결과n)	switch ~case문과 비슷하다 표현식의 값이 조건1과 일치하면 결과1 반환 조건2과 일치하면 결과2 반환 조건3과 일치하면 결과3 반환 모든 조건이 일치하지 않으면 기본결과 반환
CASE 표현식 WHEN 조건1 THEN 결과1 WHEN 조건2 THEN 결과2 WHEN 조건3 THEN 결과3 ELSE 결과N END	if~else문과 비슷하다

내장 함수 - 기타 일반 함수

```
SELECT ename, salary, commission,  
       NVL(commission,0),  
       salary*12+NVL(commission,0)  
FROM employee;
```

```
SELECT ename, salary, commission,  
       NVL2(commission,salary*12+commission,salary*12)  
FROM employee;
```

```
SELECT NULLIF('A','A'), NULLIF('A','B')  
FROM dual;
```

```
SELECT ename,salary,commission,  
       COALESCE(commission,salary,0)  
FROM employee;
```

```
SELECT ename,dno,  
       DECODE(dno, 10, 'ACCOUNTING',  
              20, 'RESEARCH',  
              30, 'SALES',  
              40, 'OPERATIONS',  
              'DEFAULT') AS DNAME  
FROM employee;
```

```
SELECT ename,dno,  
       CASE WHEN dno=10 THEN 'ACCOUNTING'  
            WHEN dno=20 THEN 'RESEARCH'  
            WHEN dno=30 THEN 'SALES'  
            WHEN dno=40 THEN 'OPERATIONS'  
            ELSE 'DEFAULT'  
       END AS DNAME  
FROM employee;
```

내장 함수 - 순위 함수

- 순위 함수는 RANK(), NTILE(), DENSE_RANK(), ROW_NUMBER()등이 있다.
- 순위함수는 데이터들의 비교가 필수이므로 앞서 나온 예제처럼 예제를 작성할 수 없고 저장된 DB 스키마를 사용해야 한다.

-순위 함수 사용법

```
순위함수() OVER(  
    PARTITION BY <컬럼>  
    ORDER BY <컬럼>  
)
```

내장 함수 - 순위 함수

- ROW_NUMBER() : 기본 순위 함수
- 키를 기준으로 순위를 정할 때

```
SELECT ROW_NUMBER()OVER(ORDER BY height) ,userName, height,addr FROM userTBL;
```

- 전체 순위가 아닌 지역별 순위를 정할 때

```
SELECT ROW_NUMBER()OVER(PARTITION BY addr ORDER BY height) ,userName, height,addr FROM userTBL;
```

내장 함수 - 순위 함수

- DENSE_RANK() : 동일한 두 데이터를 같은 순위로 매기는 함수

```
SELECT DENSE_RANK() OVER(ORDER BY height DESC),userName, height, addr FROM userTBL;
```

- RANK() : 동일한 두 데이터를 같은 순위로 매기고 그 다음 순위는 상위 개수에 맞는 순위로 다시 매기는 함수

```
SELECT RANK() OVER(ORDER BY height DESC),userName, height, addr FROM userTBL;
```

- NTILE(그룹 수) : 전체 순위를 매긴 후 그룹을 지어줄 때 사용

```
SELECT NTILE(3) OVER(ORDER BY height DESC),userName, height, addr FROM userTBL;
```

내장 함수 - 분석 함수

- 분석함수는 순위함수와 마찬가지로 여러 데이터를 기반으로 연산한다.
- 평균, 백분율, 누계등의 계산을 진행하게 된다.
- LEAD(), FIRST_VALUE(), CUME_DIST(), PERCENTILE_COUNT()등 이 있다.

내장 함수 - 분석 함수

- LEAD() : 비교대상의 데이터를 가져오기 위한 함수

사용될 인자 : LEAD(컬럼이름, 비교대상 레코드, 다음행이 없을 시 반환 값)

```
SELECT userName, addr, height, height - LEAD(height,1,0) OVER(ORDER BY height DESC) FROM userTbl;
```

LEAD(height,1,0) => 사용할 컬럼은 height이고 1은 비교대상이 다음 1번째 행을 의미한다. 0은 비교대상이 없는 즉 다음 1번째 행이 없는 경우 0을 반환한다는 의미

- LAG() : LEAD()와 같다 다만 비교대상이 이전행이다.

```
SELECT userName, addr, height, height - LAG(height,1,0) OVER(ORDER BY height ASC) FROM userTbl;
```

내장 함수 - 분석 함수

- FIRST_VALUE() : 가장 큰 데이터와의 비교를 할 때 사용하는 함수

사용될 인자 : FIRST_VALUE(컬럼명) 해당 컬럼에서 가장 큰 값(ASC정렬일 경우 가장 작은 값)을 반환

```
SELECT userName, addr, height, FIRST_VALUE(height) OVER(ORDER BY height DESC) FROM userTbl;
```

지역별로 그룹화 해서 해당 지역의 첫번째 값을 추출할 수도 있다

```
SELECT userName, addr, height, FIRST_VALUE(height) OVER(PARTITION BY addr ORDER BY height DESC) FROM userTbl;
```

내장 함수 - 분석 함수

- PERCENTILE_CONT() : 해당 컬럼의 중앙값을 계산한다.

사용될 인자 : PERCENTILE_CONT(백분율값) 0.0~1.0사이의 백분율 값을 넣을 수 있다. 0.5를 입력시
중앙값

PERCENTILE_CONT(백분율값) WITHIN GROUP(정렬할 컬럼)

```
SELECT DISTINCT addr, PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY height) OVER(PARTITION BY addr) FROM userTBL;
```