

6.1 함수(Function)

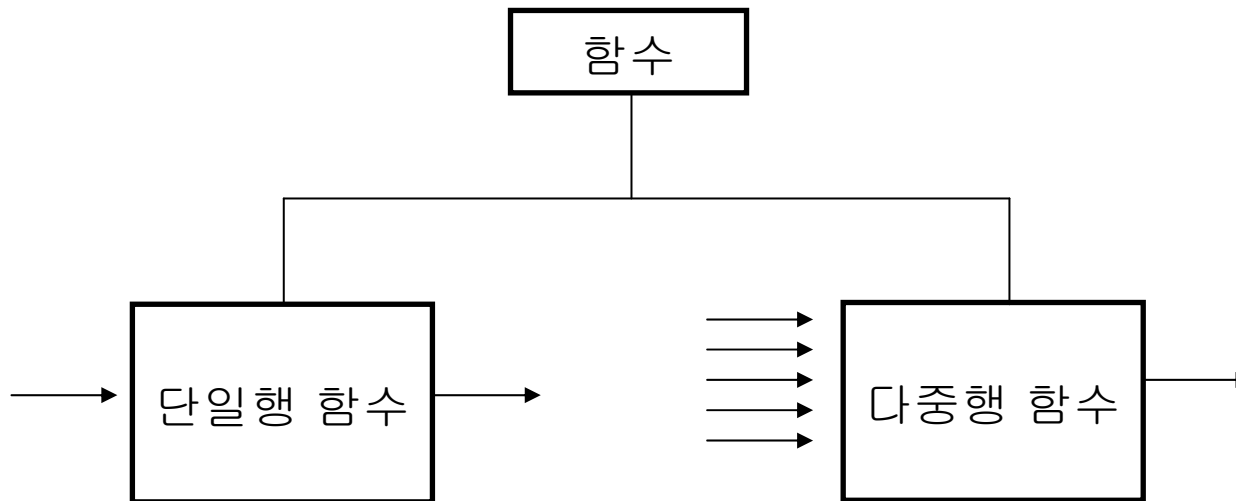
- 함수의 종류

- 단일 행 함수

- : 모든 행에 대해서 각각 적용되어 행의 개수와 동일한 개수의 결과를 리턴.

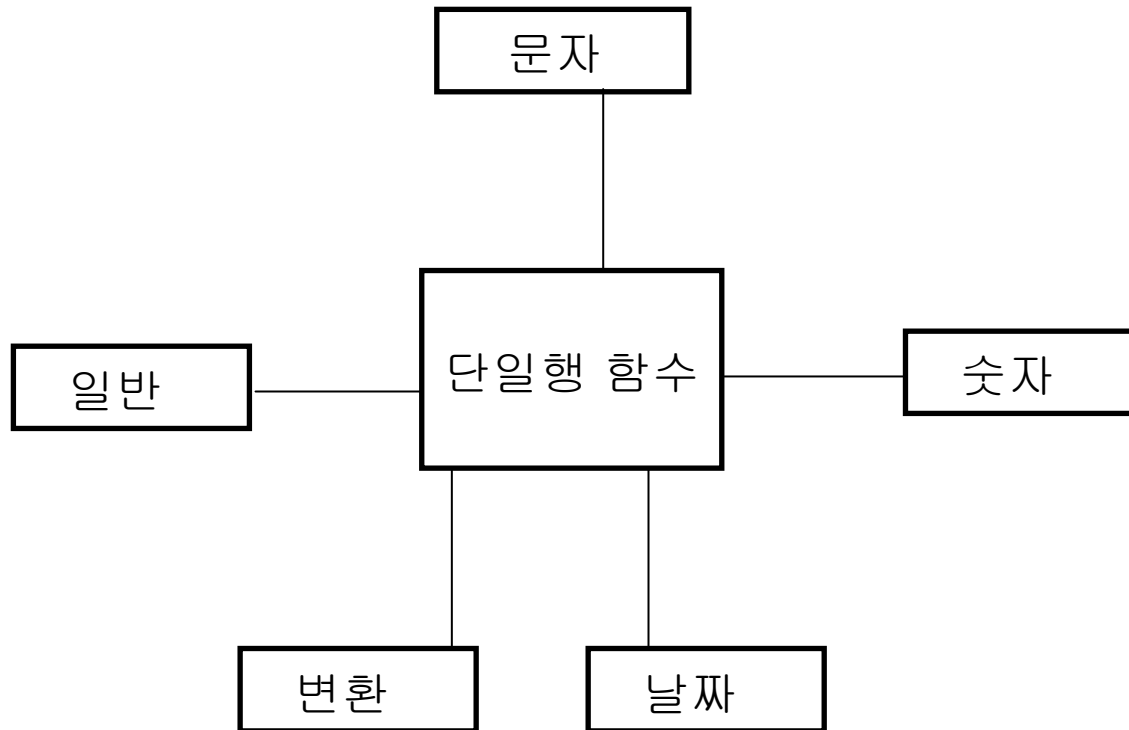
- 다중 행 함수 (그룹 함수)

- : 검색되는 모든 행에 대해서 한번만 적용되고 한 건의 결과만을 리턴.



6.2 함수(Function)

- 단일 행 함수의 종류



6.3 함수(Function)

■ 문자 함수 – 대소문자 조작함수

함수	설명
LOWER(<i>column</i> <i>expression</i>)	영문자를 소문자로 변환
UPPER(<i>column</i> <i>expression</i>)	영문자를 대문자로 변환
INITCAP(<i>column</i> <i>expression</i>)	영문자열의 첫 번째 문자를 대문자로 변환
CONCAT(<i>column1</i> <i>expression1</i> , <i>column2</i> <i>expression2</i>)	문자열을 결합, 의 결과와 동등
SUBSTR(<i>column</i> <i>expression</i> , <i>m</i> [, <i>n</i>])	입력된 문자열의 <i>m</i> 번째 문자부터 <i>n</i> 개의 문자열을 추출

```
SQL> SELECT EMPNO, ENAME, LOWER(ENAME), UPPER(ENAME)
2 FROM EMP
3 WHERE EMPNO = 7369;
```

EMPNO	ENAME	LOWER(ENAM	UPPER(ENAM
7369	SMITH	smith	SMITH

```
SQL> SELECT EMPNO, ENAME
2 FROM EMP
3 WHERE ENAME = UPPER('smith');
```

EMPNO	ENAME
7369	SMITH

```
SQL> SELECT EMPNO, ENAME, INITCAP(ENAME)
2 FROM EMP
3 WHERE EMPNO = 7369;
```

EMPNO	ENAME	INITCAP(EN
7369	SMITH	Smith

```
SQL> SELECT EMPNO, ENAME, JOB, CONCAT(ENAME, JOB)
2 FROM EMP
3 WHERE EMPNO = 7369;
```

EMPNO	ENAME	JOB	CONCAT(ENAME, JOB)
7369	SMITH	CLERK	SMITHCLERK

```
SQL> SELECT EMPNO, JOB, SUBSTR(JOB, 6, 3)
2 FROM EMP
3 WHERE EMPNO = 7499;
```

EMPNO	JOB	SUBSTR
7499	SALESMAN	MAN

6.4 함수(Function)

■ 문자 함수 – 문자열 조작함수

함수	설명
LENGTH(<i>column</i> <i>expression</i>)	입력된 문자열의 전체 문자 개수를 리턴
INSTR(<i>column</i> <i>expression</i> , ' <i>string</i> ', [, <i>m</i>], [, <i>n</i>])	입력된 문자열의 <i>m</i> 번째 문자부터 ' <i>string</i> '이 <i>n</i> 번째 나오는 위치를 리턴, <i>m</i> 의 디폴트 값은 1
LPAD(<i>column</i> <i>expression</i> , <i>n</i> , ' <i>string</i> ') RPAD(<i>column</i> <i>expression</i> , <i>n</i> , ' <i>string</i> ') TRIM(LEADING TRAILING BOTH <i>trim_character</i> FROM <i>trim_source</i>)	전체 문자의 개수가 <i>n</i> 개가 되도록 입력된 문자열의 왼쪽에 ' <i>string</i> '을 추가 전체 문자의 개수가 <i>n</i> 개가 되도록 입력된 문자열의 오른쪽에 ' <i>string</i> '을 추가 <i>trim_source</i> 에서 <i>trim_character</i> 를 제거
REPLACE(<i>text</i> , <i>search_string</i> , <i>replacement_string</i>)	<i>text</i> 에서 <i>search_string</i> 을 <i>replacement_string</i> 으로 교체

```
SQL> SELECT EMPNO, ENAME, LENGTH(ENAME)
2  FROM EMP
3  WHERE EMPNO = 7499;

SQL> SELECT EMPNO, JOB, INSTR(JOB, 'A', 1, 2)
2  FROM EMP
3  WHERE EMPNO = 7844;
```

EMPNO	ENAME	LENGTH(ENAME)	EMPNO	JOB	INSTR(JOB, 'A', 1, 2)
7499	ALLEN	5	7844	SALESMAN	7

6.5 함수(Function)

- 문자 함수 – 문자열 조작함수

```
SQL> SELECT EMPNO, SAL, LPAD(SAL, 6, '*'), RPAD(SAL, 6, '*')  
2 FROM EMP;
```

EMPNO	SAL	LPAD(SAL,6,'*')	RPAD(SAL,6,'*')
7369	800	***800	800***
7499	1600	**1600	1600**
7521	1250	**1250	1250**
7566	2975	**2975	2975**

```
SQL> SELECT EMPNO, JOB,  
2 TRIM(LEADING 'S' FROM JOB) AS LEADING,  
3 TRIM(TRAILING 'N' FROM JOB) AS TRAILING  
4 FROM EMP  
5 WHERE EMPNO = 7844
```

EMPNO	JOB	LEADING	TRAILING
7844	SALESMAN	ALESMAN	SALESMA

```
SQL> SELECT EMPNO, JOB, REPLACE(JOB, 'MAN', 'PERSON')  
2 FROM EMP  
3 WHERE EMPNO = 7844;
```

EMPNO	JOB	REPLACE(JOB, 'MAN', 'PERSON')
7844	SALESMAN	SALESPERSON

6.6 함수(Function)

■ 숫자 함수

함수	설명
ROUND(column expression, n)	입력된 숫자를 반올림하여 소수점 n자리로 리턴, n이 음수인 경우 정수 자리에서 반올림
TRUNC(column expression, n)	입력된 숫자를 내림하여 소수점 n자리로 리턴
MOD(m, n)	m을 n으로 나눈 나머지 리턴

```
SQL> SELECT ROUND(45.923, 2), ROUND(45.923, 0),  
2  ROUND(45.923, -1)  
3  FROM DUAL;
```

```
ROUND(45.923,2)  ROUND(45.923,0)  ROUND(45.923,-1)  
-----  
45.92          46          50
```

```
SQL> SELECT TRUNC(45.923, 2), TRUNC(45.923),  
2  TRUNC(45.923, -2)  
3  FROM DUAL;
```

```
TRUNC(45.923,2)  TRUNC(45.923)  TRUNC(45.923,-2)  
-----  
45.92          45          0
```

```
SQL> SELECT MOD(100, 30) FROM DUAL;
```

```
MOD(100,30)  
-----  
10
```

```
SQL> SELECT CEIL(15.3) FROM DUAL;
```

```
CEIL(15.3)  
-----  
16
```

값 이상의 가장 작은 정수

```
SQL> SELECT FLOOR( 15.3 ) FROM DUAL;
```

```
FLOOR(15.3)  
-----  
15
```

값 이하의 가장 큰 정수

6.7 함수(Function)

■ 날짜 타입

: 오라클은 날짜타입을 저장할 때, 내부적으로 세기,년도,월,일,시간,분,초 모두를 저장한다.
그러나 화면에 표시되는 형식은 기본적으로 RR/MM/DD (DD-MON-RR) 이다.

■ SYSDATE 함수

: 데이터베이스 서버에 설정되어 있는 날짜를 리턴.

```
SQL> SELECT SYSDATE FROM DUAL;
```

```
SYSDATE  
-----  
09/09/03
```

```
SQL> SELECT SYSTIMESTAMP FROM DUAL;
```

```
SYSTIMESTAMP  
-----  
09/09/03 15:10:00.765000 +09:00
```

■ 날짜 연산

연산	결과	설명
날짜 + 숫자	날짜	날짜에 일수를 더한다
날짜 - 숫자	날짜	날짜에서 일수를 뺀다
날짜 - 날짜	숫자(일수)	두 날짜의 차이(일수)를 계산한다
날짜 + 숫자/24	날짜	날짜에 시간을 더한다

6.8 함수(Function)

사원테이블에서 사원들이 입사 후 몇 주가 경과되었는지 조회.

```
SQL> SELECT EMPNO, ENAME, (SYSDATE-HIREDATE)/7 "주"  
2 FROM EMP;
```

EMPNO	ENAME	주
7369	SMITH	1226.08576
7499	ALLEN	1216.80004
7521	WARD	1216.51433
...		
7900	JAMES	1175.9429
7902	FORD	1175.9429
7934	MILLER	1168.65719

6.9 함수(Function)

■ 날짜 함수

함수	설명
MONTHS_BETWEEN(<i>date1</i> , <i>date2</i>)	<i>date1</i> 과 <i>date2</i> 의 차이를 월 단위로 계산한다
ADD_MONTHS(<i>date</i> , <i>n</i>)	<i>date</i> 에 <i>n</i> 개월을 더한다
NEXT_DAY(<i>date</i> , ' <i>char</i> ')	<i>date</i> 이후 ' <i>char</i> '로 지정된 요일의 날짜를 계산한다
LAST_DAY(<i>date</i>)	<i>date</i> 의 해당 월에서 마지막 일을 계산한다
ROUND(<i>date</i> [, ' <i>fmt</i> '])	<i>date</i> 에서 ' <i>fmt</i> '로 지정된 자리수로 반올림한다
TRUNC(<i>date</i> [, ' <i>fmt</i> '])	<i>date</i> 에서 ' <i>fmt</i> '로 지정된 자리수로 내림한다

```
SQL> SELECT EMPNO, ENAME, MONTHS_BETWEEN(SYSDATE, HIREDATE)
2 FROM EMP
3 WHERE EMPNO = 7839;
```

EMPNO	ENAME	MONTHS_BETWEEN(SYSDATE, HIREDATE)
7839	KING	270.987419

```
SQL> SELECT EMPNO, ENAME, HIREDATE, ADD_MONTHS(HIREDATE, 6)
2 FROM EMP
3 WHERE EMPNO = 7839;
```

EMPNO	ENAME	HIREDATE	ADD_MONT
7839	KING	81/11/17	82/05/17

```
SQL> SELECT EMPNO, ENAME, HIREDATE, NEXT_DAY(HIREDATE, '수')
2 FROM EMP
3 WHERE EMPNO = 7839;
```

EMPNO	ENAME	HIREDATE	NEXT_DAY
7839	KING	81/11/17	81/11/18

```
SQL> SELECT EMPNO, ENAME, HIREDATE, LAST_DAY(HIREDATE)
2 FROM EMP
3 WHERE EMPNO = 7839;
```

EMPNO	ENAME	HIREDATE	LAST_DAY
7839	KING	81/11/17	81/11/30

```
SQL> SELECT EMPNO, ENAME, HIREDATE,
2 ROUND(HIREDATE, 'MONTH'),
3 ROUND(HIREDATE, 'YEAR')
4 FROM EMP
5 WHERE EMPNO = 7839;
```

EMPNO	ENAME	HIREDATE	ROUND(HI	ROUND(HI
7839	KING	81/11/17	81/12/01	82/01/01

```
SQL> SELECT EMPNO, ENAME, HIREDATE,
2 TRUNC(HIREDATE, 'MONTH'),
3 TRUNC(HIREDATE, 'YEAR')
4 FROM EMP
5 WHERE EMPNO = 7839;
```

EMPNO	ENAME	HIREDATE	TRUNC(HI	TRUNC(HI
7839	KING	81/11/17	81/11/01	81/01/01

6.10 함수(Function)

- 변환 함수

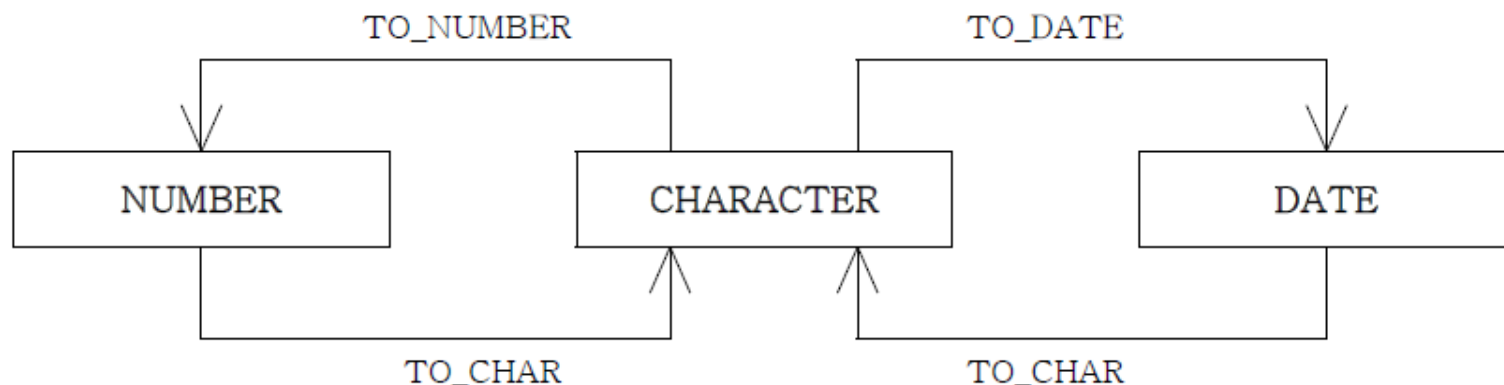
: 데이터 타입을 다른 데이터 타입으로 변환하는 함수.

- 데이터 타입의 변환 방법

1. 묵시적 형 변환

변환 전	변환 후
VARCHAR2 또는 CHAR	NUMBER
VARCHAR2 또는 CHAR	DATE
NUMBER	VARCHAR2
DATE	VARCHAR2

2. 명시적 형 변환



6.11 함수(Function)

```
SQL> SELECT EMPNO , ENAME , HIREDATE
2 FROM EMP
3 WHERE HIREDATE = '82/01/23';
```

EMPNO	ENAME	HIREDATE
7934	MILLER	82/01/23

```
SQL> SELECT EMPNO , ENAME , HIREDATE
2 FROM EMP
3 WHERE HIREDATE = TO_DATE( '82/01/23' );
```

EMPNO	ENAME	HIREDATE
7934	MILLER	82/01/23

```
SQL> SELECT EMPNO ,ENAME
2 FROM EMP
3 WHERE EMPNO = '7900';
```

EMPNO	ENAME
7900	JAMES

```
SQL> SELECT EMPNO , ENAME
2 FROM EMP
3 WHERE EMPNO = TO_NUMBER('7900');
```

EMPNO	ENAME
7900	JAMES

- TO_CHAR(type [, 'fmt'])
: NUMBER 또는 DATE 타입을 'fmt' 형식에 맞게 CHAR 타입으로 변환.
- TO_NUMBER(char [, 'fmt'])
: char 타입을 'fmt' 형식에 맞게 NUMBER 타입으로 변환.
- TO_DATE(char [, 'fmt'])
: char 타입을 'fmt' 형식에 맞게 DATE 타입으로 변환.

6.12 함수(Function)

형식	설명
YYYY	년도를 숫자로 표시
YEAR	년도를 영문으로 표시
MM	월을 숫자로 표시
MONTH	월을 영문으로 표시
MON	월을 영문 3자리로 축약해서 표시
DY	요일을 영문 3자리로 축약해서 표시, 한글 환경에서는 한글 1자로 표시
DAY	요일을 영문으로 표시, 한글 환경에서는 한글 3자로 표시
DD	일을 숫자로 표시

형식	설명
AM 또는 PM	오전 또는 오후를 표시
A.M. 또는 P.M.	오전 또는 오후를 표시
HH 또는 HH12 또는 HH24	시간 또는 1~12시 또는 1~24시
MI	분(0~59)
SS	초(0~59)
SSSSS	자정 이후 초(0~86399)

형식	설명	SQL> SELECT EMPNO,ENAME,HIREDATE
9	숫자로 표시	2 FROM EMP
0	숫자의 앞부분을 0으로 표시	3 WHERE HIREDATE=TO_DATE('DECEMBER 17, 1980','MONTH DD,YYYY');
\$	달러 표시	
L	지역 화폐 단위 표시	EMPNO ENAME HIREDATE
.	소수점 표시	-----
,	1000 단위 구분자 표시	7369 SMITH 17-DEC-80

```
SQL> SELECT empno , ename ,
2 to_char ( hiredate , 'fmDD MONTH YYYY')
3 FROM emp;
```

EMPNO	ENAME	TO_CHAR(HIREDATE, 'FMD
7369	SMITH	17 DECEMBER 1980
7499	ALLEN	20 FEBRUARY 1981
7521	WARD	22 FEBRUARY 1981
7566	JONES	2 APRIL 1981
7654	MARTIN	28 SEPTEMBER 1981

```
SQL> SELECT EMPNO,ENAME,
2 TO_CHAR(SAL, 'L99,999.00') "급여"
3 FROM EMP;
```

EMPNO	ENAME	급여
7369	SMITH	₩800.00
7499	ALLEN	₩1,600.00

6.13 함수(Function)

■ RR 날짜 형식

: 년도를 2자리로 입력 시 적절하게 1900년대 또는 2000년대의 4자리 연도로 변환.

현재 년도	명시된 날짜	RR 형식	YY 형식
1995	27-OCT-95	1995	1995
1995	27-OCT-17	2017	1917
2001	27-OCT-17	2017	2017
2001	27-OCT-95	1995	2095

		명시된 두 자리 년도	
		0-49	50-99
현재 년도의 두 자리값	0-49	리턴되는 날짜는 현재의 세기	리턴되는 날짜는 현재 세기 이 전의 세기
	50-99	리턴 날짜는 현재 세기 이후의 세기	리턴 날짜는 현재 세기

6.14 함수(Function)

■ 일반 함수

함수	설명
<code>NVL(expr1, expr2)</code>	<code>expr1</code> 이 NULL이면 <code>expr2</code> 를 리턴
<code>NVL2(expr1, expr2, expr3)</code>	<code>expr1</code> 이 NULL이 아니면 <code>expr2</code> , NULL이면 <code>expr3</code> 를 리턴
<code>NULLIF(expr1, expr2)</code>	<code>expr1</code> 과 <code>expr2</code> 가 같으면 NULL, 다르면 <code>expr1</code> 을 리턴
<code>COALESCE(expr1, expr2, ... , exprn)</code>	인자들 중에서 NULL이 아닌 첫 번째 인자를 리턴

```
SQL> SELECT EMPNO, ENAME, SAL, COMM, SAL*12+COMM, SAL*12+NVL(COMM, 0)
2 FROM EMP;
```

EMPNO	ENAME	SAL	COMM	SAL*12+COMM	SAL*12+NVL(COMM,0)
7369	SMITH	800			9600
7499	ALLEN	1600	300	19500	19500
7521	WARD	1250	500	15500	15500
7566	JONES	2975			35700
7934	MILLER	1300			15600

```
SQL> SELECT EMPNO, ENAME, SAL, COMM, SAL+COMM, NVL2(COMM, SAL+COMM, SAL)
2 FROM EMP;
```

EMPNO	ENAME	SAL	COMM	SAL+COMM	NVL2(COMM, SAL+COMM, SAL)
7369	SMITH	800			800
7499	ALLEN	1600	300	1900	1900
7521	WARD	1250	500	1750	1750
7566	JONES	2975			2975
7934	MILLER	1300			1300

6.15 함수(Function)

■ CASE 함수

```
CASE expr WHEN comparison_expr1 THEN return_expr1
      [WHEN comparison_expr2 THEN return_expr2
      WHEN comparison_exprn THEN return_exprn
      ELSE else_expr]
END
```

```
SQL> SELECT EMPNO, ENAME, SAL, JOB,
2  CASE JOB WHEN 'ANALYST' THEN SAL*1.1
3           WHEN 'CLERK' THEN SAL*1.2
4           WHEN 'MANAGER' THEN SAL*1.3
5           WHEN 'PRESIDENT' THEN SAL*1.4
6           WHEN 'SALESMAN' THEN SAL*1.5
7           ELSE SAL
8  END "급여"
9  FROM EMP;
```

EMPNO	ENAME	SAL	JOB	급여
7369	SMITH	800	CLERK	960
7499	ALLEN	1600	SALESMAN	2400
7521	WARD	1250	SALESMAN	1875
7566	JONES	2975	MANAGER	3867.5
7654	MARTIN	1250	SALESMAN	1875

6.16 함수(Function)

■ DECODE 함수

```
DECODE(col/expression, search1, result1  
      [, search2, result2, ... ,]  
      [, default])
```

```
SQL> SELECT EMPNO, ENAME, SAL, JOB,  
2      DECODE(JOB, 'ANALYST' , SAL*1.1,  
3              'CLERK'   , SAL*1.2,  
4              'MANAGER' , SAL*1.3,  
5              'PRESIDENT', SAL*1.4,  
6              'SALESMAN' , SAL*1.5, SAL) "급여"  
7 FROM EMP;
```

EMPNO	ENAME	SAL	JOB	급여
7369	SMITH	800	CLERK	960
7499	ALLEN	1600	SALESMAN	2400
7521	WARD	1250	SALESMAN	1875
7566	JONES	2975	MANAGER	3867.5
7654	MARTIN	1250	SALESMAN	1875

6.17 함수(Function)

■ 실습 예제

1. 사원테이블에서 입사일이 12월인 사원의 사번, 사원명, 입사일 검색하시오
2. 다음과 같은 결과를 검색할 수 있는 SQL 문장을 작성하시오.

EMPNO	ENAME	급여
7369	SMITH	*****800
7499	ALLEN	*****1600
...		
7934	MILLER	*****1300

- 3 다음과 같은 결과를 검색할 수 있는 SQL 문장을 작성하시오.

EMPNO	ENAME	입사일
7369	SMITH	1980-12-17
7499	ALLEN	1981-02-20
...		
7934	MILLER	1982-01-23

6.18 함수(Function)

- 실습 예제

4 . 사원 테이블에서 급여에 따라 사번, 이름, 급여, 등급을 검색하는 SQL 문장을 작성하시오 (CASE 함수 이용)

급여	등급
0~1000	E
1001~2000	D
2001~3000	C
3001~4000	B
4001~5000	A