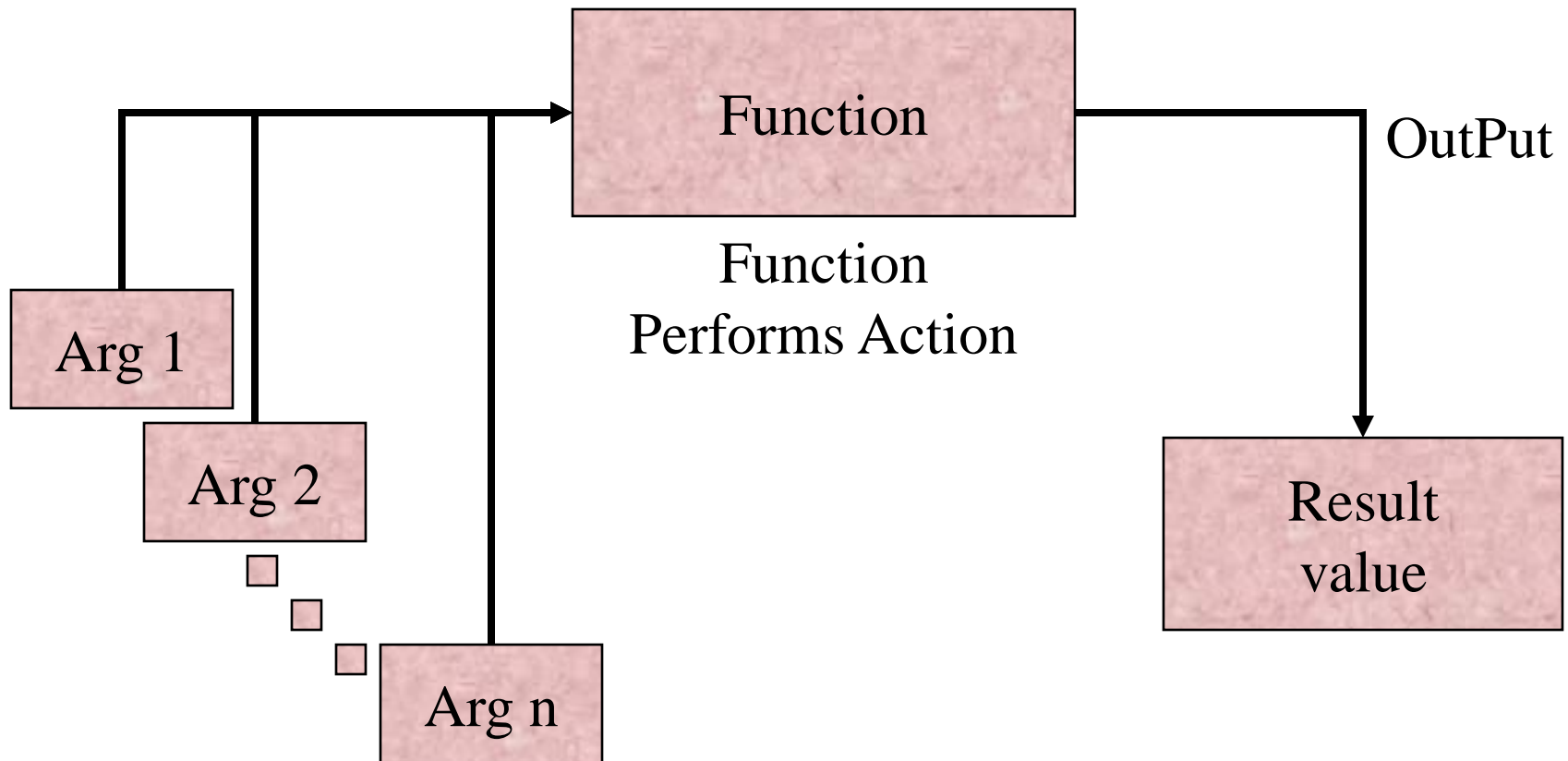


## 제 3 장 단일 행 함수

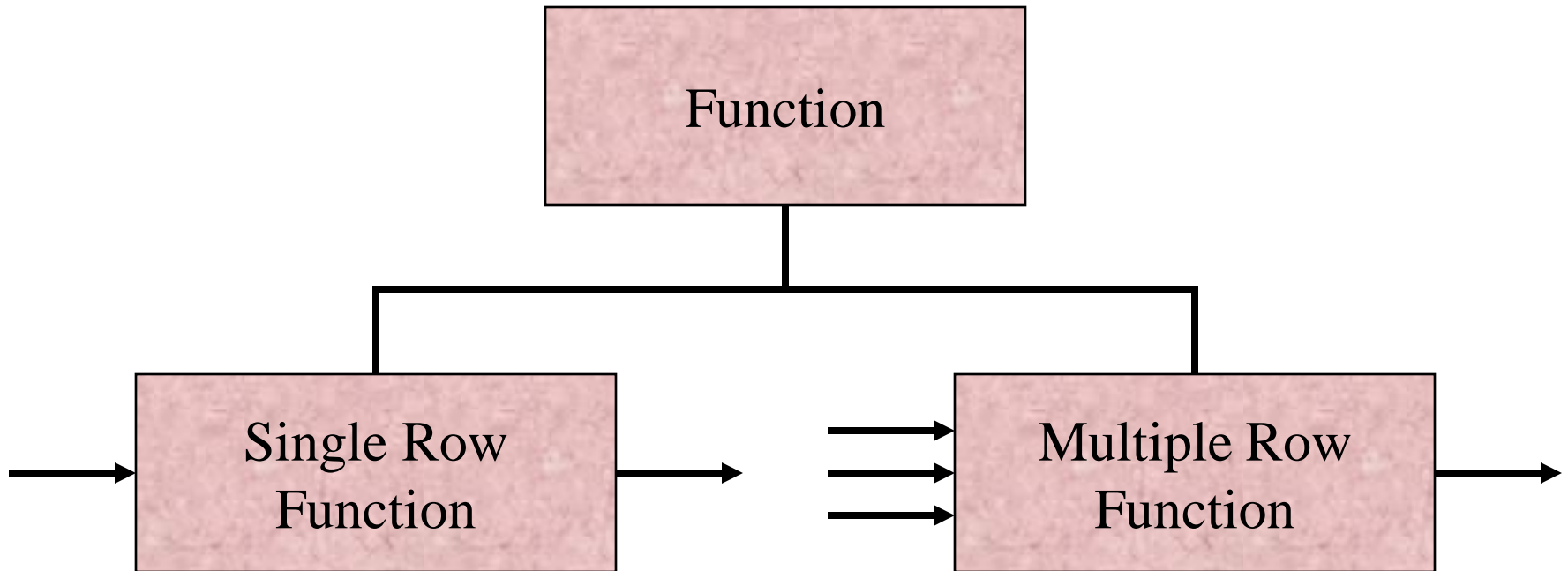
# 목적

- **SQL** 에서 이용 가능한 다양한 종류의 함수를 기술.
- **Select** 문장에서 문자, 숫자 그리고 날짜 함수를 사용
- 변환 함수 사용 기술

# SQL 함수



# SQL 함수의 두 가지 유형

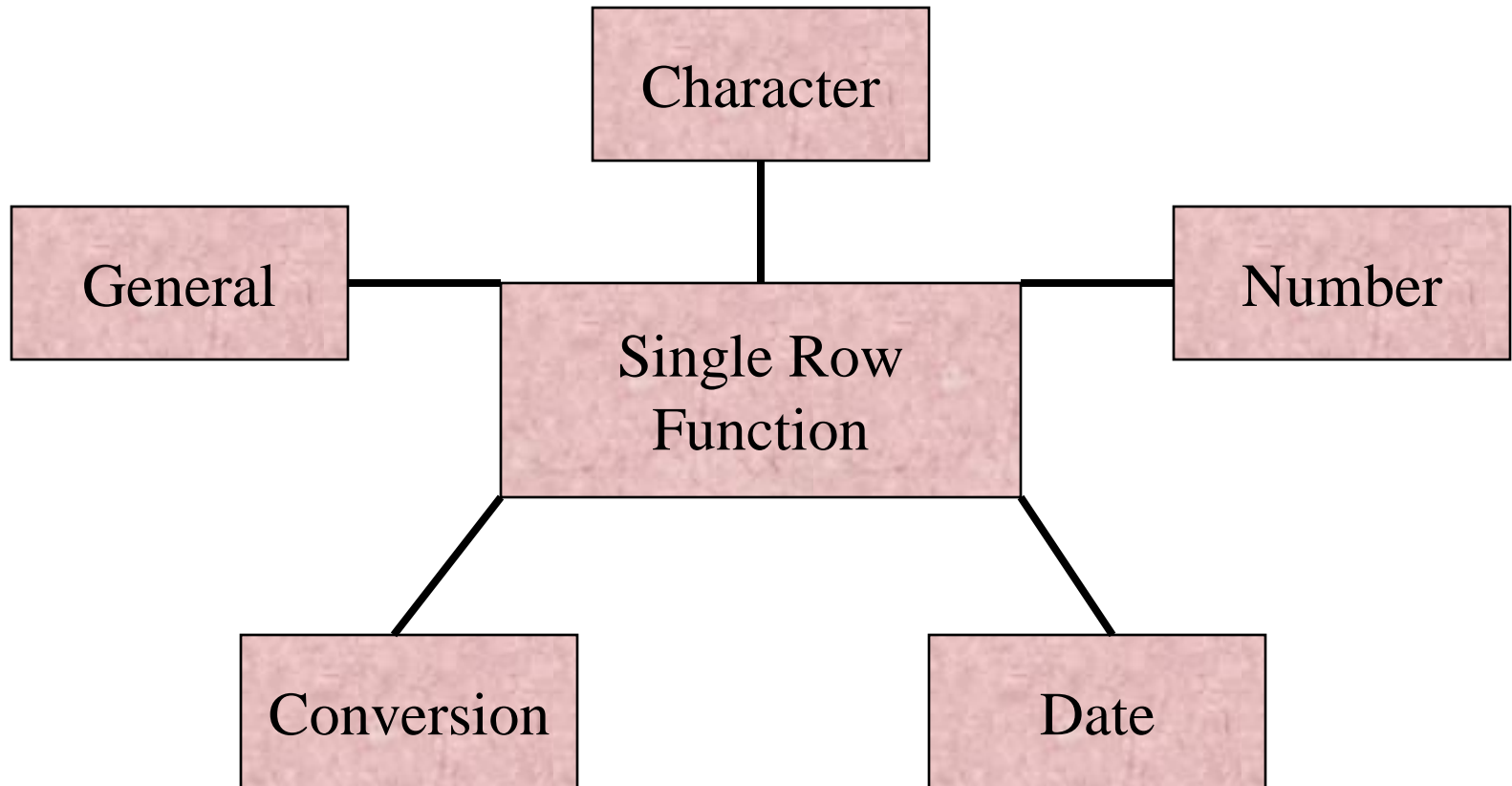


# 단일 행 함수

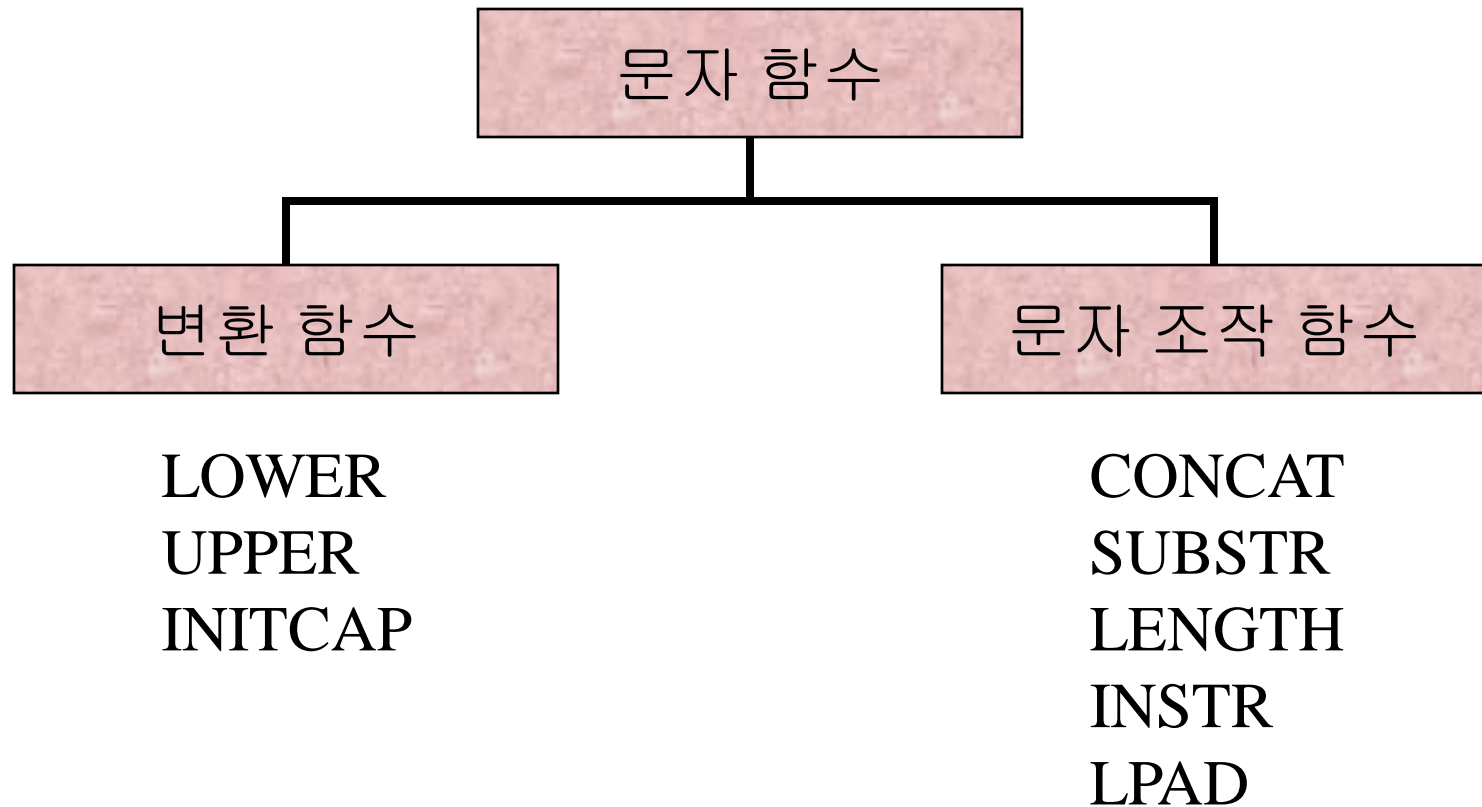
- 데이터 값을 조작
- 인수들을 받고 하나의 결과를 리턴
- 리턴될 각각의 행에 적용
- 행별로 하나의 결과 리턴
- 데이터형 수정 가능
- 중첩 가능

Function\_name (column | expression, [arg1, arg2,...])

# 단일 행 함수



# 문자 함수



# 대소문자 변환 함수

- 문자 스트링에 대한 대소문자 변환

함수	결과
LOWER('SQL Course')	sql course
UPPER('SQL Course')	SQL COURSE
INITCAP('SQL Course')	Sql Course



# 문자 조작 함수

## # 문자 스트링 조작

함수	결과
CONCAT('Good', 'String')	GoodString
SUBSTR('String',1,3)	Str
LENGTH('String')	6
INSTR('String','r')	3
LPAD(sal,10,'*')	*****5000
RPAD(sal,10,'*')	5000*****

# 숫자 함수

- **ROUND** – 명시된 소수점으로 반올림  
**ROUND(45.926, 2) = 45.93**
- **TRUNC** – 명시된 소수점으로 절하  
**TRUNC(45.926,2) = 45.92**
- **MOD** – 나머지 값  
**MOD(1600,300) = 100**

# 날짜 작업

- 오라클은 세기, 년, 월, 일, 시간, 분, 초의 내부 숫자 형식으로 날짜를 저장합니다.
- 디폴트 날짜 형식 **DD-MON-YY**
- **DUAL** 은 **SYSDATE**를 보기 위한 더미테이블 이다.  
**DUAL** – **sys** 에 의해 소유되며 모든 사용자가 액세스 할 수 있다.  
**DUMMY**라는 하나의 열과 **X**값을 가지는 하나의 행을 포함.
- **DUAL** 테이블은 오직 한번만 값을 리턴 하고자 할 때 유용.
- 가령 사용자 데이터를 가진 테이블에서 파생되지 않는 상수 또는 표현식의 값의 경우.  
**Ex) select sysdate from dual;**

# 날짜 연산

- 날짜에서 숫자를 더하거나 빼어 날짜 결과를 리턴
- 날짜 사이의 일 수를 알기 위해서 두 개의 날짜를 뺄
- 시간을 **24**로 나누어서 시간을 날짜에 더함.

작업	결과	설명
날짜 + 숫자	날짜	일 수를 날짜에 더함
날짜 - 숫자	날짜	날짜에서 일 수를 뺄
날짜 - 날짜	일 수	어떤 날짜에서 다른 날짜를 뺄
날짜 + 숫자/24	날짜	시간을 날짜에 더함

# 날짜를 가진 산술 연산자 사용

```
SQL> select ename, (SYSDATE – hiredate)/7 weeks  
2   From emp  
3   Where deptno = 10
```

ENAME	WEEKS
KING	830.93709
CLARK	853.93709
MILLER	821.36566

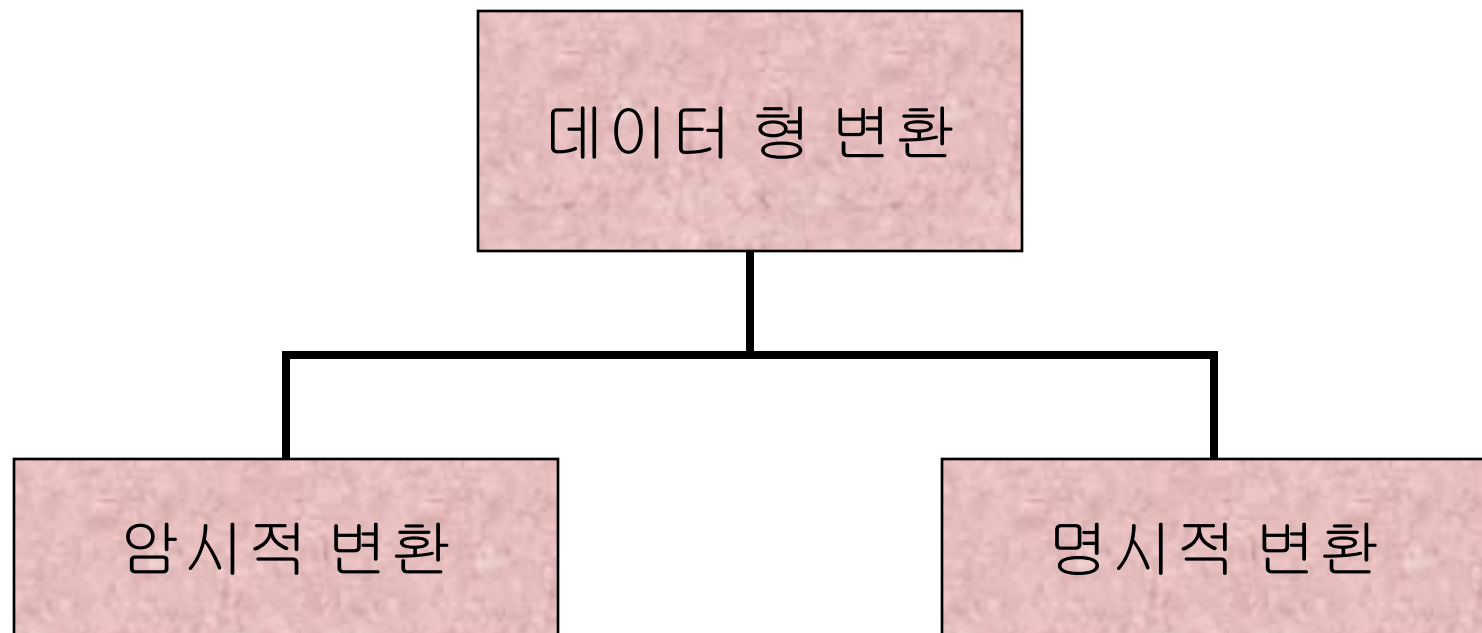
## 날짜 함수

함수	설명
MONTHS_BETWEEN	두 날짜 사이의 월 수
ADD_MONTHS	월을 날짜에 더함
NEXT_DAY	다음 요일에 대한 날짜
LAST_DAY	월의 마지막 날
ROUND	날짜를 반올림
TRUNC	날짜를 절삭

## 날짜 함수의 사용

- **MONTH\_BETWEEN('01-SEP-95','11-JAN-94')**  
**= 19.6774149**
- **ADD\_MONTHS('11-JAN-94',6) = 11-JUL-94**
- **NEXT\_DAY('01-SEP-95','FRIDAY') = '08-SEP-95'**
- **LAST\_DAY('01-SEP-95') = '30-SEP-95'**
- **ROUND('25-JUL-95','MONTH') = 01-AUG-95**
- **ROUND('25-JUL-95','YEAR') = 01-JAN-96**
- **TRUNC('25-JUL-95','MONTH') = 01-JUL-95**
- **TRUNC('25-JUL-95','YEAR') = 01-JAN-95**

# 변환 함수





## 암시적 데이터 형 변환

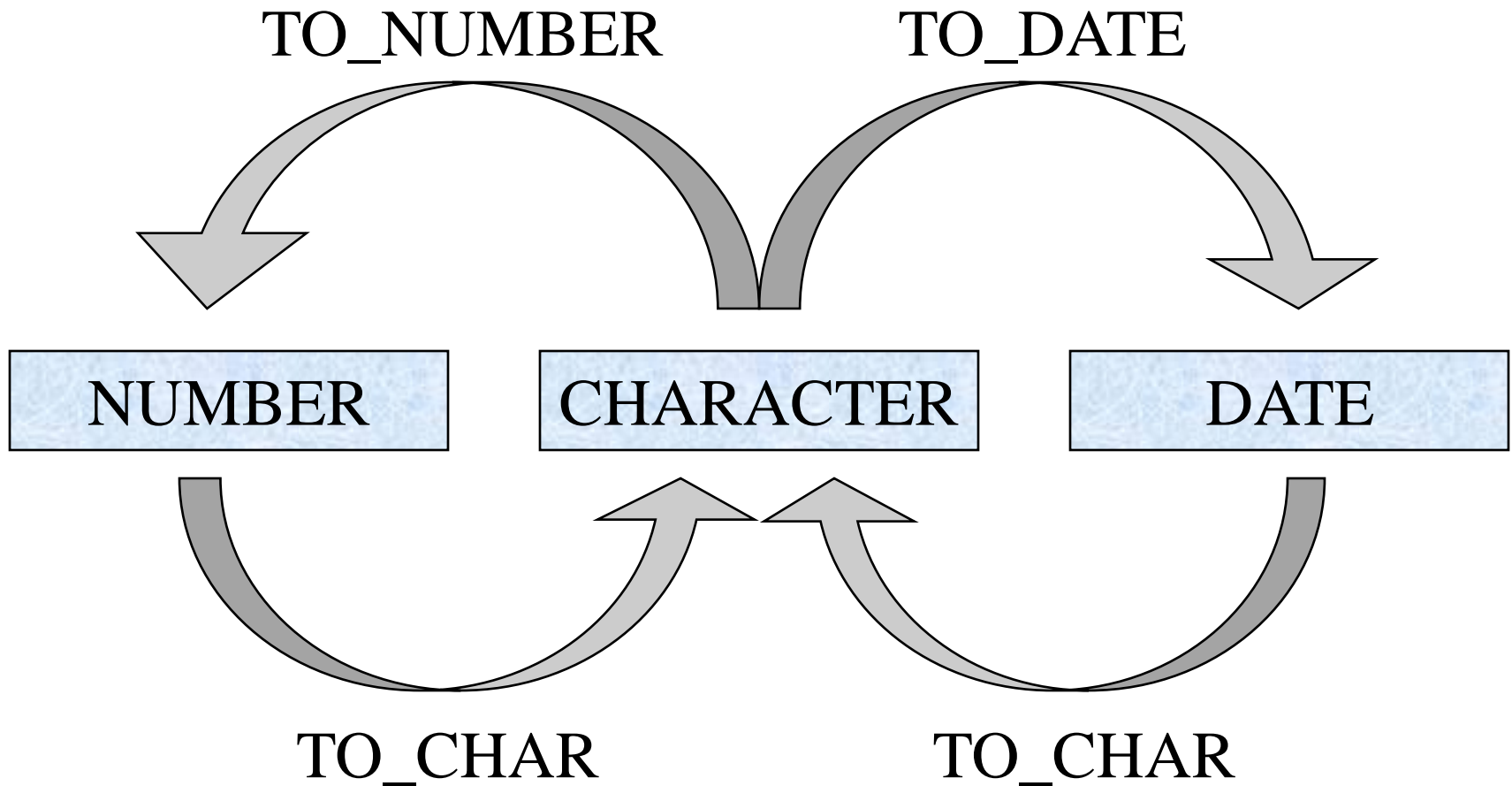
- 값 할당시, 오라클은 자동으로 다음을 변환할 수 있다.

From	To
VARCHAR2 or CHAR	NUMBER
VARCHAR2 or CHAR	DATE
NUMBER	VARCHAR2
DATE	VARCHAR2

- 표현식 계산의 경우 오라클은 자동으로 다음을 변환할 수 있다.

From	To
VARCHAR2 or CHAR	NUMBER
VARCHAR2 or CHAR	DATE

# 명시적 데이터 형 변환



# 날짜를 가진 TO\_CHAR 함수

```
TO_CHAR(date, 'fmt')
```

포맷 모델:

- 단일 인용부호로 둘러 싸여 있어야 하고 대소문자를 구분
- 어떤 타당한 날짜 형식 요소도 포함할 수 있다
- 데이터 값을 콤마로 구분한다

## 날짜 형식 모델 요소

SCC or CC	세기	MM	두자리 값의 월
YYYY or SYYYY	년도	MONTH	9자리 월
YYY or YY or Y	년도, 자리수	MON	세 자리의 월 이름
Y,YYY	콤마가 있는 년도	RM	로마 숫자 월
IYYY,IYY,IY,I	ISO 표준 년도	WW or W	년이나 월의 주
SYEAR or YEAR	문자로 표현된 년	DDD or DD or D	년, 월 또는 주의 일
BC or AD	BC/AD 지시자	DAY	9자리 요일
B.C. or A.D.	.이 있는 BC/AD 지시자	DY	세자리 요일 이름
Q	년의 4분의 1	J	Julian Day

\* Julian day – BC 4713년 12월 31일 이후의 요일수

## 날짜 형식 모델 요소

- 시간 요소는 날짜의 시간 부분을 형식화  
**HH24:MI:SS AM – 15:45:32 PM**
- 문자 스트링에 이중 인용부호를 둘러싸서 문자 스트링 추가  
**DD “of” MONTH – 12 of OCTOBER**
- 숫자 접미사는 숫자를 문자로 명시  
**ddspth – fourteenth**

# 숫자를 가진 TO\_CHAR 함수

TO\_CHAR(number, 'fmt')

- **TO\_CHAR** 함수를 사용하여 숫자 값을 문자로 디스플레이 하기 위해서 위의 형식을 사용

**9** – 숫자

**0** – 0이 표시되도록 한다

**\$** – 달러 기호

**L** – 지역 화폐 기호

**.** – 명시된 위치에 소수점

**,** – 명시된 위치에 콤마

**MI** – 우측에 마이너스 기호

**PR** – 음수를 괄호로 묶음

**EEEE** – 과학적인 부호표기

**V** – 10을 n 번 곱함

**B** – 0값을 0이 아닌 공백으로 표시

# TO\_NUMBER 와 TO\_DATE 함수

- **TO\_NUMBER** 함수를 사용하여 문자 스트링을 숫자 형식으로 변환  
**TO\_NUMBER(char)**
- **TO\_DATE** 함수를 사용하여 문자 스트링을 날짜 형식으로 변환  
**TO\_DATE(char[, 'fmt'])**

# RR 날짜 형식

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



# NVL 함수

- **Null 값을 실제 값으로 변환**
- **사용될 수 있는 데이터형은 날짜, 문자, 숫자.**
- **데이터형은 일치해야 함.**
  - **NVL(comm, 0)**
  - **NVL(hiredate, '01-JAN-97')**
  - **NVL(job, 'No Job Yet')**

# DECODE 함수

- **CASE 나 IF-THEN-ELSE** 문장의 조건적 조회를 가능하게 함.

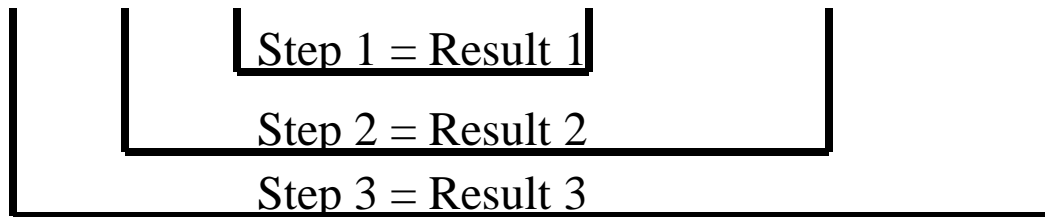
```
DECODE(col/expression, search1, result1 [, search2, result2,...]  
      [, default])
```

```
SQL> Select job, sal, DECODE(job, 'ANALYST', sal * 1.1,  
                                'CLERK',    sal * 1.15,  
                                'MANAGER',  sal * 1.20,  
                                sal) Revised Salary from EMP
```

# 중첩 함수

- 단일 행 함수는 여러 레벨에 걸쳐 중첩될 수 있다.
- 중첩 함수는 가장 하위 레벨에서 가상 상위 레벨 순으로 계산된다.

F3(F2(F1(col, arg1), arg2), arg3)



# 요약

다음을 위해서 함수를 사용

- 데이터 계산 수행
- 개별적인 데이터 값 수정
- 행의 그룹에 대한 결과 조작
- 디스플레이를 위한 날짜 형식 변경
- 열 데이터형 변환