Single-Row Functions

문자 함수

숫자 함수

날짜 함수

중첩 함수

일반 함수

실습

단일 행 함수

- 데이터 값을 조작한다.
- 인수를 받아들여 한 개의 값을 반환한다.
- 각각의 행에 적용 된다.
- 행 별로 하나의 값을 반환한다.
- 데이터 타입을 바꾸어 준다.
- 중첩이 가능하다.
- 인수는 하나의 컬럼이나 표현식이 될 수 있다.

```
function_name [(arg1, arg2, ...)]
```

문자 함수

Function	Purpose
LOWER(column|expression)	문자열을 소문자로 변환
UPPER(column|expression)	문자열을 대문자로 변환
INITCAP(column|expression)	단어의 첫글자는 대문자로, 나머지는 소문자로 변환
CONCAT(column1|expression1, column2|expression2)	첫 문자열과 두 번째 문자열을 연결, || 연산자와 같다.
SUBSTR(column|expression, m[,n]	m과 n 사이에 있는 문자열만 반환(m이 음수이면, 문자열의 끝에서 시작. n을 생략하면, 문자열 끝까지 모두 반환)
LENGTH(column|expression)	표현식에서 문자의 갯수를 반환
INSTR(column|expression, 'string', [,m], [n])	문자열에서 string의 위치를 반환(m부터 탐색을 시작, n번째 탐색을 반환)
LPAD(column|expression, n, 'string') RPAD(column|expression, n, 'string')	n의 자릿수를 가진 문자열 형태로 오른쪽 정렬을 한다. n의 자릿수를 가진 문자열 형태로 왼쪽 정렬을 한다.
TRIM(leading|trailing|both trim_character FROM trim_source)	문자열에서 첫 문자, 마지막 문자(또는 모두)를 정리할 수 있다.
REPLACE(text, search_string, replacement_string)	문자열에서 찾고자하는 텍스트를 탐색하여 있으면, replacement_string으로 바꾼다.

문자 함수

Function	Result
LOWER('SQL Course')	sql course
UPPER('SQL Course')	SQL COURSE
INITCAP('SQL Course')	Sql Course

```
SELECT employee_id, last_name, department_id
FROM employees
WHERE last_name = 'higgins';
```

no rows selected

```
SELECT employee_id, last_name, department_id
FROM employees
WHERE LOWER(last_name) = 'higgins';
```

문자 함수

Function	Result
CONCAT(‘Hello’, ‘World’)	HelloWorld
SUBSTR(‘HelloWorld’,1,5)	Hello
LENGTH(‘HelloWorld’)	10
INSTR(‘HelloWorld’, ’W’)	6
LPAD(salary,10,‘*’)	*****24000
RPAD(salary,10,‘*’)	24000*****
TRIM(‘H’ FROM ’HelloWorld’)	elloWorld

```
SELECT employee_id,  CONCAT(first_name,last_name) NAME,
LENGTH(last_name), INSTR(last_name, ‘a’) “Contains ‘a’?”
FROM employees
WHERE SUBSTR(last_name, -1, 1) = ‘n’;
```

숫자 함수

Function	Purpose
ROUND(column|expression, n)	컬럼, 표현식, 또는 값에 대해 소수점 n번째 자리에서 반올림 한다. n이 생략되면, 소수점은 없다.
TRUNC(column|expression, n)	컬럼, 표현식, 또는 값에 대해 소수점 n번째 자리의 나머지는 버린다. n이 생략되면 n의 기본값은 0이다.
MOD(m, n)	m을 n으로 나눈 나머지를 반환한다.

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

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

- **DUAL**은 함수 및 계산 결과를 보는데 사용할 수 있는 더미 테이블(dummy table) 이다.

숫자 함수

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

```
SELECT last_name, salary, MOD(salary, 5000)
FROM employees
WHERE job_id = 'SA_REP';
```

날짜 사용

DateFMT2To4Year = *cut-off-year*

15는 1915일까, 2015일까?

- 오라클 데이터베이스는 날짜를 내부적인 숫자 형식으로 저장한다.
  century, year, month, day, hours, minutes, seconds.
- 기본적인 표시 형식은 YY / MM / DD이다.
  - 연도의 마지막 두 자리만 기술함으로써 20세기에 21세기의 날짜를 저장할 수 있다.
  - 같은 방법으로 21세기에 20세기의 날짜를 저장할 수 있다.

```
SELECT last_name, hire_date
FROM employees
WHERE last_name LIKE 'G%';
```

- · SYSDATE 함수는 현재 데이터베이스 서버의 다음을 반환하는 함수이다.
  - Date
  - Time

날짜 계산

- 날짜에서 숫자를 더하거나 빼서 날짜 결과를 반환한다.
- 날짜 사이의 일수를 계산하기 위하여 두 개의 날짜를 뺀다.
- 시간을 24로 나눔으로써 시간을 날짜에 더한다.

Operation	Result	Description
date + number	Date	Adds a number of days to a date
date - number	Date	Substracts a number of days from a date
date - date	Number of days	Substracts one date from another
date + number/24	Date	Adds a number of hours to a date

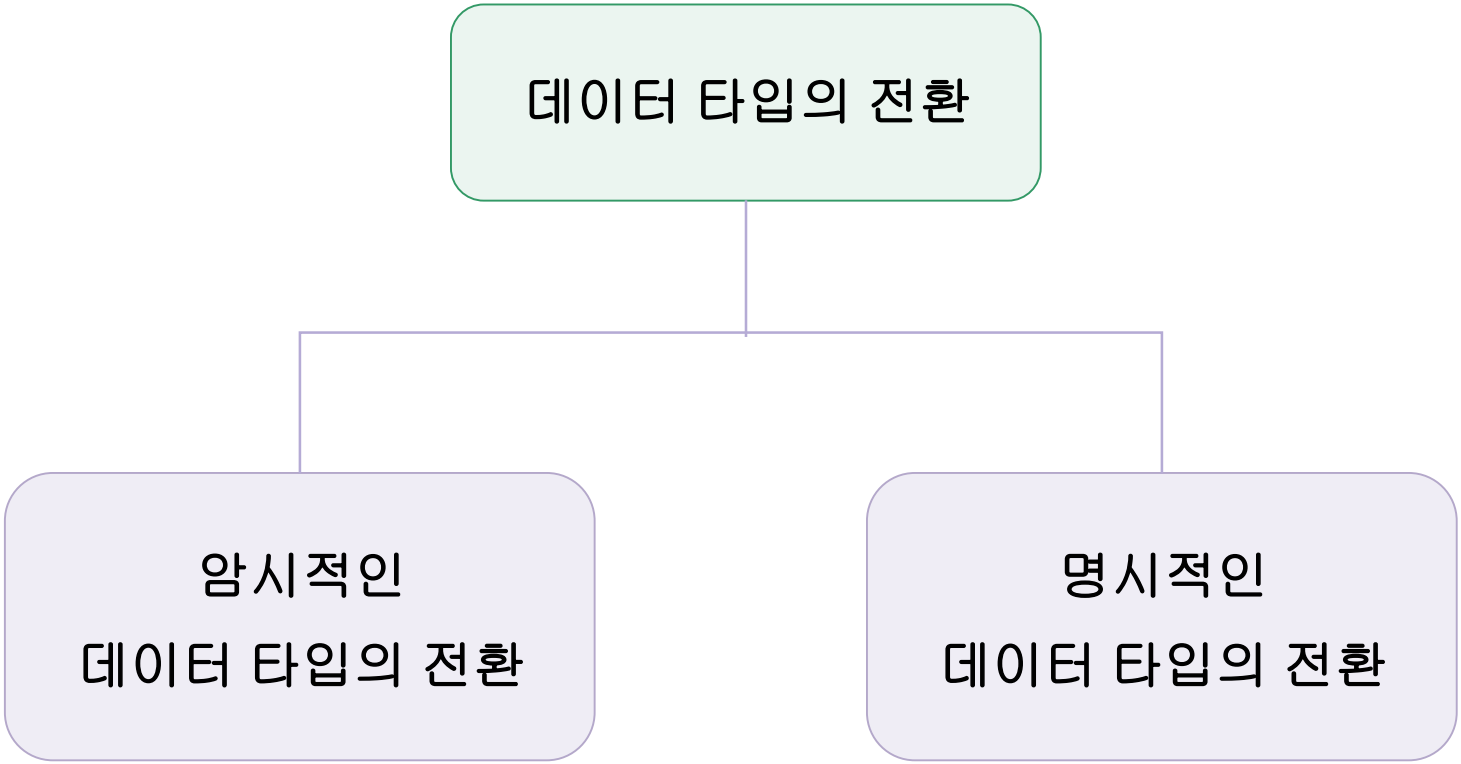
```
SELECT last_name, (SYSDATE-hire_date)/7 AS WEEKS
FROM employees
WHERE department_id = 90;
```

날짜 함수

Function	Description
MONTHS_BETWEEN	두 날짜 사이의 달 수
ADD_MONTHS	날짜에 달 수를 더한다.
NEXT_DAY	명시한 날짜 이후의 첫 번째 해당 요일 일짜
LAST_DAY	달의 마지막 날
ROUND	정오를 기준으로 날짜 반올림
TRUNC	날짜에서 시간 부분을 버림

- **Using Date Functions**
  - MONTHS_BETWEEN('95/09/01', '94/01/11')
  - ADD_MONTHS('94/01/11', 6)
  - NEXT_DAY('95/09/01', '일요일')
  - LAST_DAY('95/02/01')

변환 함수



Implicit Data Type Conversion

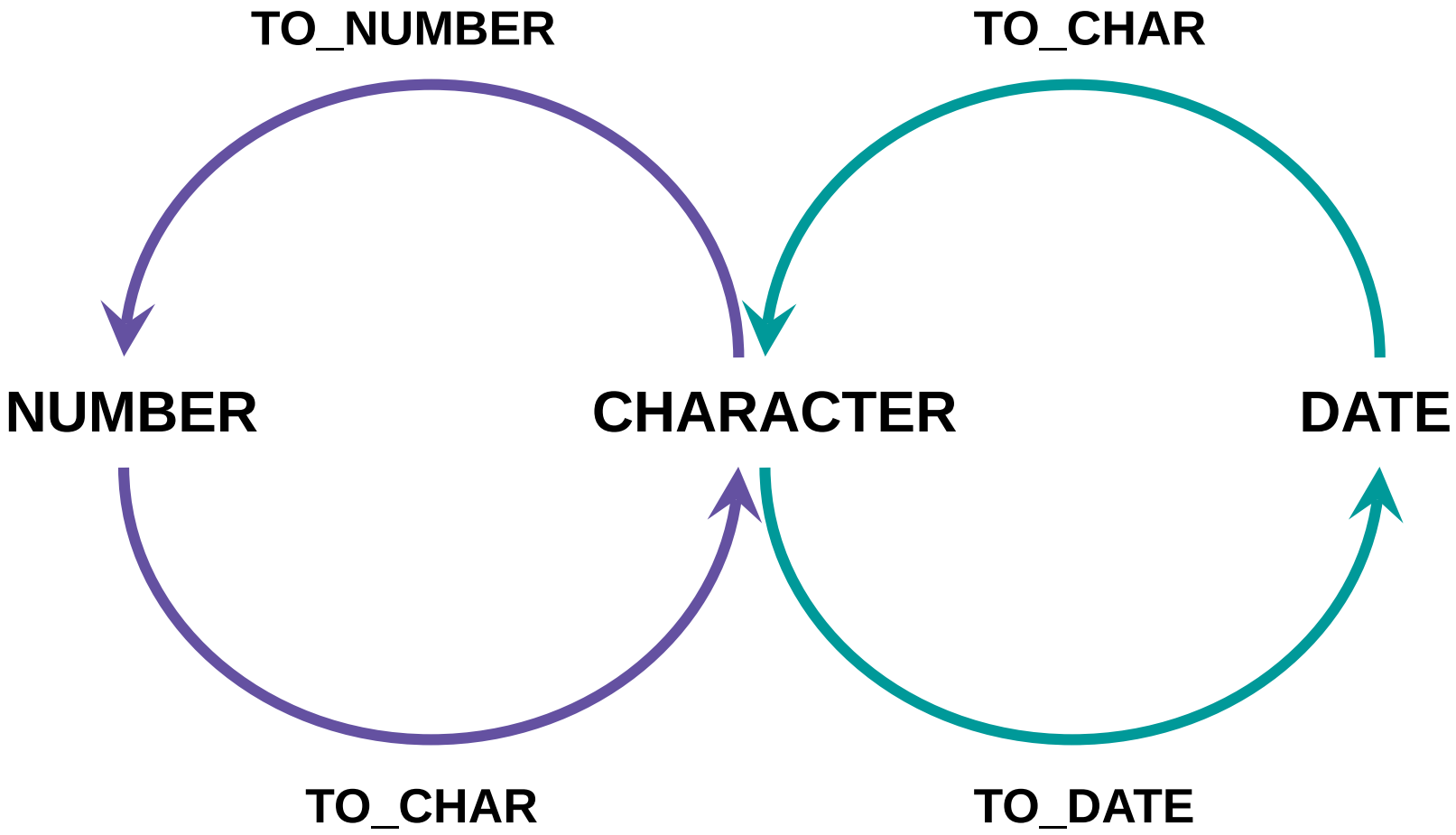
- 값 할당시, 오라클 서버는 자동으로 다음과 같이 변환한다.

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

- 표현식의 값을 구하기 위하여, 오라클 서버는 자동으로 다음과 같이 변환한다.

From	To
VARCHAR2 or CHAR	NUMBER
VARCHAR2 or CHAR	DATE

Explicit Data Type Conversion



Elements of the Date Format Model

요소	설명
YYYY	년도를 완전한 숫자로 표시
YEAR	명시한 년도를 문자로 표시
MM	월을 2자리 숫자로 표시
MONTH	월을 완전한 이름으로 표시
MON	월을 3자리 약자로 표시
DY	요일을 3자리 약자로 표시
DAY	요일을 완전한 이름으로 표시
DD	월의 날짜를 숫자로 표시

```
SELECT last_name, TO_CHAR(hire_date, 'DD Month YYYY') AS HIREDATE
FROM employees;
```

Elements of the Date Format Model

요소	설명
SCC or CC	세기 , 기원전 날짜에는 -를 접두어로 붙임
날짜의 연도 YYYY or SYYYY	연도 , 기원전 날짜에는 -를 접두어로 붙임
YYY or YY or Y	연도의 마지막 세자리, 두 자리 또는 한자리
Y,YYY	해당 위치에 쉼표가 있는 연도
IYYY,IYY,IY,I	ISO표준을 따르는 네 자리, 세 자리, 두 자리 또는 한 자리 연도
SYEAR or YEAR	연도(문자) , 기원전 날짜에는 -를 접두어로 붙임
BC or AD	B.C. / A.D. 표시자
B.C. or A.D.	마침표가 있는 B.C. / A.D. 표시자
Q	일년 중의 분기
WW or W	일년 중의 주(week) 또는 한달 중의 주
DDD or DD or D	일년 중의 일(day), 한달 중의 일 또는 한 주 중의 일
J	율리우스력의 일 , 기원전 4713년 12월 31일부터의 일 수
RM	로마 숫자 달

Elements of the Date Format Model

- 시간 형식

  표에 있는 형식을 사용하여 시간 정보 및 리터럴을 표시하고 숫자를 문자들로 변경합니다.

요소	설명
AM or PM	오전/오후 표시자
A.M. or P.M.	마침표가 포함된 오전/오후 표시자
HH or HH12 or HH24	하루 중의 시 또는 1-12 또는 0-23으로 표시되는 시간
MI	분(0-59)
SS	초(0-59)
SSSS	자정부터의 초(0-86399)

Elements of the Date Format Model

- 기타 형식

요소	설명
/ . ,	구두점을 결과에 재사용
“of the”	인용 문자열을 결과에서 재사용

- 숫자 표시에 영향을 주는 접미어 지정

요소	설명
TH	서수(예를 들어, 4 TH의 경우 DDTH)
SP	문자로 표현한 숫자(예를 들어, FOUR의 경우 DDSP)
SPTH or THSP	문자로 표현한 서수(예를 들어, FOURTH의 경우 DDSPTH)

TO_CHAR 함수

- TO_CHAR 함수를 사용하여 숫자 값을 문자로 표시할 수 있는 몇가지 형식 요소가 있다.

9	숫자를 나타낸다. (9의 수는 표시의 폭을 결정)
0	강제로 0을 표시하게 된다.
$	$ 기호를 표시한다.
L	지역 화폐 기호를 표시한다.
.	소수점을 표시한다.
,	1000 단위 기호를 표시한다.

```
SELECT TO_CHAR(salary, '$99,999.00') SALARY
FROM employees ;
```

TO_NUMBER , TO_DATE 함수

- 문자열을 TO_NUMBER 함수를 사용하여 숫자 형식으로 변환한다.

```
TO_NUMBER(char[, 'format_model'])
```

- 문자열을 TO_DATE 함수를 사용하여 날짜 형식으로 변환한다.

```
TO_DATE(char[, 'format_model'])
```

```
SELECT last_name, TO_CHAR(hire_date, 'DD-MON-YYYY')
FROM employees
WHERE hire_date < TO_DATE('01-01-90', 'DD-MM-RR');
```

LAST_NAME	TO_CHAR(HIR
King	17-JUN-1987
Kochhar	21-SEP-1989
Whalen	17-SEP-1987

중첩 함수

- 단일 행 함수는 여러 번 중첩될 수 있다.
- 중첩 함수는 가장 안쪽부터 바깥쪽 순으로 계산된다.

```
SELECT      last_name,
            NVL(TO_CHAR(manager_id) , 'No Manager')
FROM        employees
WHERE       manager_id IS NULL;
```

일반 함수

NVL(expr1, expr2)	expr1 값이 널인 경우 expr2 값을 반환하며 그렇지 않은 경우 expr1 값을 반환한다.
NVL2(expr1, expr2, expr3)	expr1이 널이 아닌 경우 , NVL2는 expr2를 반환한다. expr1이 널인 경우, NVL2는 expr3을 반환한다. 인수 expr1에는 모든 데이터 유형을 사용할 수 있다.
NULLIF(expr1, expr2)	두 표현식을 비교하여 동일한 경우 널을 반환하고 동일하지 않은 경우 첫 번째 표현식을 반환한다.
COALESCE(expr1, expr2, … , exprn)	표현식 목록에서 널이 아닌 첫 번째 표현식을 반환한다.

NVL 함수

- Null 값을 실제 값으로 변환한다.
- 데이터 형식은 date, character, number가 사용될 수 있다.
- 데이터 형식은 일치해야 한다.
  - NVL(commission_pct, 0)
  - NVL(hire_date, to_date('01/01/97', 'dd/mm/yy'))
  - NVL(job_id, 'No Job Yet')

```
SELECT last_name, salary, NVL(commission_pct, 0),
       (salary*12) +  (salary*NVL(commission_pct, 0)) AN_SAL
FROM employees;
```

LAST_NAME	SALARY	NVL(COMMISSION_PCT,0)	AN_SAL
King	24000	0	288000
Kochhar	17000	0	204000
De Haan	17000	0	204000

NVL2 함수

- 첫 번째 표현식이 널이 아닌 경우, 두 번째 표현식을 반환하며 첫 번째 표현식이 널인 경우, 세 번째 표현식을 반환한다.

```
SELECT last_name, salary, commission_pct,
       NVL2( commission_pct, 'SAL+COM', 'SAL') income
FROM employees
WHERE department_id  IN  (50 , 80);
```

NULLIF 함수

- 두 표현식을 비교하여 동일한 경우 널을 반환하고 동일하지 않은 경우 첫 번째 표현식을 반환. 첫 번째 표현식에 널 리터럴을 지정할 수 없다.

```
SELECT  first_name,  LENGTH(first_name)  “expr1”,
        last_name,  LENGTH(last_name)  “expr2”,
        NULLIF(LENGTH(first_name) ,  LENGTH(last_name) )  result
FROM  employees;
```

COALESCE 함수

- 여러 대체 값을 사용할 수 있다는 장점이 있다.
- 첫 번째 표현식이 널이 아닌 경우 해당 표현식을 반환하며, 널인 경우에는 나머지 표현식에 대해 COALESCE 함수를 적용한다.

```
SELECT  last_name,
        COALESCE( commission_pct,  salary,  10)  comm
FROM  employees
ORDER  BY  commission_pct;
```

Conditional Expressions

- SQL문 안에 IF-THEN-ELSE 논리의 사용을 제공한다.
- 두 가지의 메소드를 사용 :
  - CASE 표현
  - DECODE 함수

```
CASE job_id  WHEN 'IT_PROG'      THEN 1.10*salary
             WHEN 'SH_CLERK'    THEN 1.15*salary
             WHEN 'SA_REP'      THEN 1.20*salary
             ELSE                 salary
END AS "REVISED_SALARY"


DECODE(job_id, 'IT_PROG', 1.10*salary,
               'SH_CLERK',  1.15*salary,
               'SA_REP', 1.20*salary,
               salary)
```

실습

1. 현재 날짜를 표시하는 질의를 작성하고 열 레이블을 Data로 지정하십시오.
2. 각 사원에 대해 사원 번호, 이름 , 급여 및 15% 인상된 급여를 정수로 표시하십시오. 인상된 급여 열의 레이블을 New Salary 로 지정하십시오.
3. 이름이 J, A또는 M으로 시작하는 모든 사원의 이름(첫 글자는 대문자로, 나머지 글자는 소문자로 표시) 및 이름 길이를 표시하는 질의를 작성하고 각 열에 적합한 레이블을 지정하십시오. 결과를 사원의 이름에 따라 정렬하십시오.
4. 각 사원의 이름을 표시하고 근무 달 수(입사일로부터 현재까지의 달 수)를 계산하여 열 레이블을 MONTHS_WORKED로 지정하십시오. 결과는 정수로 반올림하여 표시하고 근무 달 수를 기준으로 정렬하십시오.
5. 각 사원에 대해 다음 항목을 생성하는 질의를 작성하십시오. <employee last name> earn <salary> monthly but wants <3 times salary> 열 레이블을 Dream Salaries로 지정하십시오.

실습

6.모든 사원의 이름과 급여를 표시하는 질의를 작성하십시오. 급여는 15자 길이로 왼쪽에 $기호가 채워진 형식으로 표기하고 열 레이블을 SALARY로 지정하십시오.

7.사원의 이름, 입사일 및 급여 검토일을 표시하십시오. 급여 검토일은 여섯 달이 경과한 후 첫 번째 월요일입니다. 열 레이블을 REVIEW로 지정하고 날짜는 “Monday, the Thirty-First of July, 2000”과 같은 형식으로 표시되도록 지정하십시오.

8. 이름, 입사일 및 업무 시작 요일을 표시하고 열 레이블을 DAY로 지정하십시오. Monday를 시작으로 해서 요일을 기준으로 결과를 정렬하십시오.

9.사원의 이름과 커미션 합계를 표시하는 질의를 작성하십시오. 커미션을 받지 않는 사원일 경우 “No Commission”을 표시하십시오. 열 레이블은 COMM으로 지정하십시오.

10.사원의 이름을 표시하고 급여 총액을 별표(*)로 나타내는 질의를 작성하십시오. 각 별표는 1,000달러를 나타냅니다. 급여를 기준으로 데이터를 내림차순으로 정렬하고 열 레이블을 EMPLOYEES_AND_THEIR_SALARIES로 지정하십시오.

실습

11. DECODE 함수를 사용하여 다음 데이터에 따라 JOB_ID 열의 값을 기준으로 모든 사원의 등급을 표시하는 질의를 작성하십시오.

업무	등급
AD_PRES	A
ST_MAN	B
IT_PROG	C
SA_REP	D
ST_CLERK	E
기타	0

12. 12번 문제의 명령문을 CASE 구문을 사용하여 재작성하십시오.