

# 7장 함수

강사 김영석

A top-down view of a wooden desk. In the upper right, a portion of a silver laptop is visible, showing the keyboard. To its left is a small green succulent in a pot. Below the succulent is a pair of black-rimmed glasses. In the lower center, there is a white mug filled with dark coffee. The bottom edge of the frame shows the corner of a dark-colored tablet or laptop.

# CONTENT

001 함수

002 단일 행 함수

003 그룹 함수

# 001 함수

## ❖ 함수란 ?

- 오라클 데이터베이스 시스템에서 제공하는 함수는 미리 정의된 기능을 통해 데이터를 좀더 편리하게 조작할 수 있도록 도와준다.
- 함수란 사용자가 입력 값을 넣으면 정해 놓은 출력 값이 나오는 개념이다.
- 함수를 이용하면 문자, 숫자, 날짜 값 등을 조작할 수 있으며, 각 데이터 타입끼리 변환 할 수도 있다. (단일 행 함수 기능)
- 복수의 행을 조합하여 그룹 당 하나의 결과로도 출력 할 수 있다. (그룹 함수 기능)

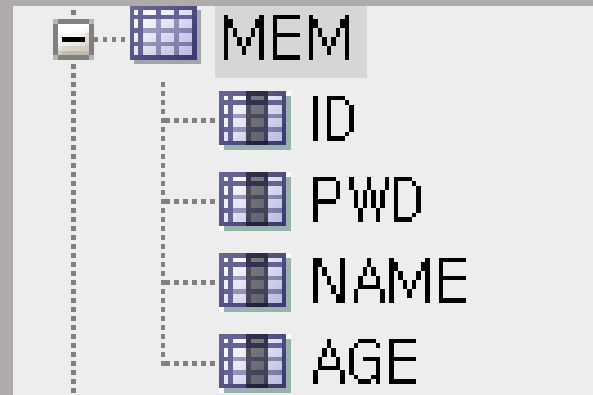
# 001 함수

## ❖ 단일 행 함수

- 오라클 데이터베이스는 다양한 데이터 타입을 지원한다.
- 테이블의 열은 한 가지 데이터 타입으로 지정되어 있으며 지정된 데이터 타입과 일치하는 데이터 값만 저장할 수 있다.
- 실무에서 가장 많이 쓰는 데이터 타입은 숫자, 문자, 날짜 타입이다.

저장 데이터	데이터 타입	설명
문자	char(n)	N 크기 만큼 고정 길이의 문자 타입을 저장한다. 최대 2,000바이트까지 저장할 수 있다.
문자	varchar2(n)	N 크기 만큼 가변 길이의 문자 타입을 저장한다. 최대 4,000바이트까지 저장할 수 있다.
숫자	number(p, s)	숫자 타입을 저장한다(p : 정수 자릿수, s : 소수 자릿수)
날짜	date	날짜 타입을 저장한다. 9999년 12월 31일까지 저장할 수 있다.

# 001 함수



	⌘ COLUMN_NAME	⌘ DATA_TYPE	⌘ NULLABLE	DATA_DEFAULT	⌘ COLUMN_ID	⌘ COMMENTS
1	ID	VARCHAR2(10 BYTE)	No	(null)	1	(null)
2	PWD	VARCHAR2(20 BYTE)	No	(null)	2	(null)
3	NAME	VARCHAR2(10 BYTE)	No	(null)	3	(null)
4	AGE	NUMBER(38,0)	Yes	(null)	4	(null)



# 002 단일 행 함수

## ❖ 단일 행 함수 종류

종류	설명
문자 타입 함수	문자를 입력 받아 문자와 숫자를 반환한다.
숫자 타입 함수	숫자를 입력 받아 숫자를 반환한다.
날짜 타입 함수	날짜에 대해 연산한다. 숫자를 반환하는 MONTHS_BETWEEN 함수를 제외한 모든 날짜 타입 함수는 날짜 값을 반환한다.
변환 타입 함수	임의의 데이터 타입의 값을 다른 데이터 타입으로 변환한다.
일반 함수	그 외 NVL, DECODE, CASE WHEN, 순위 함수 등

# 002 단일 행 함수

## ❖ 문자 타입 함수

- 문자 타입 함수는 주로 데이터 조작에 쓰이며 종류는 다음과 같다
- 문자나 문자열 데이터는 작은 따옴표('')로 묶어서 문자 타입으로 표현한다.

# 002 단일 행 함수

함수	설명	예	결과
LOWER	값을 소문자로 변환한다.	LOWER('ABCD')	abcd
UPPER	값을 대문자로 변환한다.	UPPER('abcd')	ABCD
INITCAP	첫 번째 글자만 대문자로 변환한다.	INITCAP('abcd')	Abcd
SUBSTR	문자열 중 일부분을 선택한다.	SUBSTR('ABC', 1, 2)	AB
REPLACE	특정 문자열을 찾아 바꾼다.	REPLACE('AB', 'A', 'E')	EB
CONCAT	두 문자열을 연결한다. (   연산자와 같다)	CONCAT('A', 'B')	AB
LENGTH	문자열의 길이를 구한다.	LENGTH('AB')	2
INSTR	명명된 문자의 위치를 구한다.	INSTR('ABCD', 'D')	4
LPAD	왼쪽부터 특정 문자로 자리를 채운다.	LPAD('ABCD', 6, '*')	**ABCD
RPAD	오른쪽부터 특정 문자로 자리를 채운다.	RPAD('ABCD', 6, '*')	ABCD**
LTRIM	주어진 문자열의 왼쪽 문자를 지운다.	LTRIM('ABCD', 'AB')	CD
RTRIM	주어진 문자열의 오른쪽 문자를 지운다.	RTRIM('ABCD', 'CD')	AB



# 002 단일 행 함수

## ❖ LOWER, UPPER, INITCAP 함수

- LOWER 함수는 데이터 값을 모두 소문자로, UPPER 함수는 데이터 값을 모두 대문자로 변환할 때 사용한다.
- INITCAP 함수는 데이터 값의 첫 번째 문자만 대문자로 변환할 때 사용한다.
- SQL은 데이터 값의 대소문자를 구분하기 때문에 데이터 출력 값을 표준화 할 때 유용하다.

LOWER ('문자열' 또는 [열 이름])

UPPER ('문자열' 또는 [열 이름])

INITCAP ('문자열' 또는 [열 이름])

## 002 단일 행 함수

```
SELECT LOWER('ABCD') AS LOWER적용, UPPER(id) AS UPPER적용,  
INITCAP(id) AS INITCAP적용 FROM mem;
```



	❖ LOWER적용	❖ UPPER적용	❖ INITCAP적용
1	abcd	HONG	Hong
2	abcd	LEE	Lee
3	abcd	YOU	You
4	abcd	HONG1	Hong1
5	abcd	SIN	Sin

# 002 단일 행 함수

## ❖ SUBSTR 함수

- SUBSTR 함수는 데이터에서 지정된 길이만큼 문자열을 추출할 때 사용한다.
- 데이터 값이 이미 가공되어 데이터베이스에 저장되어 있는 경우라도 일부 문자열을 잘라내 가공해야 할 때가 자주 있다.

SUBSTR('문자열' 또는 [열 이름], 시작 위치, 길이)

```
SELECT SUBSTR(id, 1, 2) FROM mem;
```



	⚡ SUBSTR(ID,1,2)
1	ho
2	le
3	yo
4	ho
5	si

# 002 단일 행 함수

## ❖ REPLACE 함수

- 특정 문자를 찾아 바꾸어 주는 함수이다.
- 사용자가 바꾸고자 하는 문자나 문자열을 지정하면 지정한 형태로 문자나 문자열이 바뀐다. (값이 바뀌는 것은 아니다.)

REPLACE ('문자열' 또는 [열 이름], '변경전 문자열', '변경후 문자열')

```
SELECT id, REPLACE(id, 'hong1', 'Hong') AS 변경 FROM mem;
```



	ID	변경
1	hong	hong
2	lee	lee
3	you	you
4	hong1	Hong
5	sin	sin

# 002 단일 행 함수

## ❖ LPAD, RPAD 함수

- LPAD는 왼쪽 부터 특정 문자로 자릿수를 채우는 함수이고, RPAD는 오른쪽 부터 특정 문자로 자릿수를 채우는 함수이다.
- 리포트나 프로그래밍에서 데이터 값의 자릿수를 맞춰야 할 때 유용하다.

LPAD('문자열' 또는 [열 이름], 고정 길이 값, '채워질 문자')

RPAD('문자열' 또는 [열 이름], 고정 길이 값, '채워질 문자')



# 002 단일 행 함수

```
SELECT id, LPAD(id, 6, '*') FROM mem;
```



ID	LPAD(ID,6,'*')
1 hong	**hong
2 lee	***lee
3 you	***you
4 hong1	*hong1
5 sin	***sin

```
SELECT id, RPAD(id, 6, '*') FROM mem;
```



ID	RPAD(ID,6,'*')
1 hong	hong**
2 lee	lee***
3 you	you***
4 hong1	hong1*
5 sin	sin***

# 002 단일 행 함수

## ❖ LTRIM, RTRIM 함수

- LTRIM은 왼쪽부터 지정한 문자를 지우는 함수이고, RTRIM은 오른쪽부터 지정한 문자를 지우는 함수이다.
- 지정한 문자의 순서는 중요하지 않으며 지정한 문자 외의 값을 만나면 진행을 중단한다.
- '삭제할 문자' 옵션을 주지 않으면 공백을 제거한다.

LTRIM ('문자열' 또는 [열 이름], '삭제할 문자')

RTRIM ('문자열' 또는 [열 이름], '삭제할 문자')

# 002 단일 행 함수

```
SELECT id, LTRIM(id, 'h') FROM mem;  
SELECT id, RTRIM(id, 'g') FROM mem;
```



	ID	LTRIM(ID, 'H')
1	hong	ong
2	lee	lee
3	you	you
4	hong1	ong1
5	sin	sin

	ID	RTRIM(ID, 'G')
1	hong	hon
2	lee	lee
3	you	you
4	hong1	hong1
5	sin	sin

# 002 단일 행 함수

## ❖ TRIM 함수

- TRIM 함수는 공백을 제거하는데 사용한다.
- 단, 문자열 중간에 있는 공백은 제거 할 수 없다.

TRIM ('문자열' 또는 [열 이름])

```
SELECT TRIM(' AB CD ') AS 공백제거 FROM dual;
```



	공백제거
1	AB CD

# 002 단일 행 함수

## ❖ DUAL 테이블

- DUAL 테이블은 더미(Dummy) 라는 하나의 열과 하나의 데이터 값을 갖고 있는 테이블이다.
- 임의의 값을 알고자 하거나 특정 테이블을 참고하지 않아도 될 때 유용한 테이블 이다.
- 즉, 오라클 데이터베이스 시스템에서 테스트용으로 사용할 수 있는 테이블이다.



# 002 단일 행 함수

## ❖ 숫자 타입 함수

- 숫자 타입 함수는 주로 숫자를 계산하거나 계산이 끝난 후에 추가로 가공 처리 할 때 사용한다.
- SQL 은 다양한 숫자 타입 함수를 제공하는데 그중 가장 많이 쓰는 ROUND와 TRUNC 함수를 알아보자.

# 002 단일 행 함수

함수	설명	예	결과
<b>ROUND</b>	숫자를 반올림한다. 0이 소수점 첫째 자리다.	ROUND(15.351, 0)	15
<b>TRUNC</b>	숫자를 절삭한다. 0이 소수점 첫째 자리다.	TRUNC(15.351, 1)	15.3
<b>MOD</b>	나누기 후 나머지를 구한다.	MOD(15, 2)	1
<b>CEIL</b>	숫자를 정수로 올림한다.	CEIL(15.351)	16
<b>FLOOR</b>	숫자를 정수로 내림한다.	FLOOR(15.351)	15
<b>SIGN</b>	양수(1), 음수(-1), 0 인지를 구분하여 출력한다.	SIGN(15)	1
<b>POWER</b>	거듭제곱을 출력한다.	POWER(2, 3)	8
<b>SQRT</b>	제곱근을 출력한다.	SQRT(4)	2

# 002 단일 행 함수

## ❖ ROUND 함수

- ROUND 함수는 지정한 자리에서 반올림 하는 함수이다.
- 반올림할 자리 값은 생략 할 수 있으며 기본값은 0이다.
- 반올림할 자리 값이 양수면 소수 자리에서 반올림하고 음수면 정수 자리에서 반올림한다.
- 0이 소수점 첫째 자리이기 때문에 1은 소수점 둘째 자리에서 반올림한다. -1은 정수 첫째 자리에서 반올림하겠다는 의미이다.

ROUND(숫자 또는 [열 이름], 반올림할 자리 값)

```
SELECT 1234.5678, ROUND(1234.5678,0),ROUND(1234.5678,1),ROUND(1234.5678,2) FROM dual;
```

	1234.5678	ROUND(1234.5678,0)	ROUND(1234.5678,1)	ROUND(1234.5678,2)
1	1234.5678	1235	1234.6	1234.57

# 002 단일 행 함수

## ❖ TRUNC 함수

- TRUNC 함수는 지정한 숫자 자리에서 숫자를 버리는 함수 이다.
- 기본 문법은 ROUND 함수와 동일하다.
- 버리는 자리 값은 0은 소수점 첫째 자리에서 버린다는 의미이다.
- 양수는 소수 자리, 음수는 정수 자리에서 값을 버린다.

TRUNC(숫자 또는 [열 이름], 버리는 자리 값)

```
SELECT 1234.5678, TRUNC(1234.5678,0), TRUNC(1234.5678,1), TRUNC(1234.5678,2) FROM dual;
```



	1234.5678	TRUNC(1234.5678,0)	TRUNC(1234.5678,1)	TRUNC(1234.5678,2)
1	1234.5678	1234	1234.5	1234.56

# 002 단일 행 함수

## ❖ 날짜 타입 함수

- 데이터를 다루다 보면 날짜를 계산하고 처리해야 하는 경우가 많다.
- 날짜 타입 함수는 날짜를 연산하여 숫자로 출력하는 MONTHS\_BETWEEN 외에는 모든 결과를 날짜 타입으로 출력한다.

날짜 연산	설명	반환값
<b>Date + Number</b>	날짜에 일수를 더한다.	Date
<b>Date - Number</b>	날짜에 일수를 뺀다.	Date
<b>Date - Date</b>	날짜에서 날짜를 뺀다.	일수
<b>Date + Number / 24</b>	날짜에 시간을 더할 때는 시간 24로 나누어서 날짜를 더한다.	Date



## 002 단일 행 함수

```
SELECT to_CHAR(SYSDATE, 'YY/MM/DD/HH24:MI') AS 오늘날짜,  
SYSDATE - 1 AS 어제날짜, SYSDATE + 1 AS 내일날짜,  
TO_DATE('20200131') - TO_DATE('20200101') AS 날짜빼기,  
SYSDATE + 20 / 24 시간더하기  
FROM dual;
```



	🔗 오늘날짜	🔗 어제날짜	🔗 내일날짜	🔗 날짜빼기	🔗 시간더하기
1	21/01/10/04:19	21/01/09	21/01/11	30	21/01/11

# 002 단일 행 함수

날짜 함수	설명	예	결과
<b>MONTHS_BETWEEN</b>	두 날짜 사이의 월수를 계산한다.	MONTHS_BETWEEN(SYSDATE, HIRE_DATE)	171.758
<b>ADD_MONTHS</b>	월을 날짜에 더한다.	ADD_MONTHS(HIRE_DATE, 5)	03/11/17
<b>NEXT_DAY</b>	명시된 날짜부터 돌아오는 요일에 대한 날짜를 출력한다.(SUNDAY:1, MONDAY:2, ... )	NEXT_DAY(HIRE_DATE,1)	03/06/22
<b>LAST_DAY</b>	월의 마지막 날을 계산한다.	LAST_DAY(HIRE_DATE)	03/06/30
<b>ROUND</b>	날짜를 가장 가까운 연도 또는 월로 반올림한다. (YEAR 또는 MONTH)	ROUND(HIRE_DATE, 'MONTH')	03/07/01
<b>TRUNC</b>	날짜를 가장 가까운 연도 또는 월로 절삭한다. (YEAR 또는 MONTH)	TRUNC(HIRE_DATE, 'MONTH')	03/06/1

# 002 단일 행 함수

## ❖ MONTHS\_BETWEEN 함수

- MONTHS\_BETWEEN 함수는 날짜와 날짜 사이의 개월 수를 계산한다.
- 결과는 음수나 양수가 될 수 있다. (소수는 한달이 되지 못한 날이다)

MONTHS\_BETWEEN(날짜, 날짜)

```
SELECT SYSDATE, '22/01/31', MONTHS_BETWEEN(SYSDATE, '22/01/31'),  
MONTHS_BETWEEN(SYSDATE, '20/01/01') FROM dual;
```



	SYSDATE	'22/01/31'	MONTHS_BETWEEN(SYSDATE,'22/01/31')	MONTHS_BETWEEN(SYSDATE,'20/01/01')
1	21/01/10	22/01/31	-12.6710958034647550776583034647550776583	12.29664613201911589008363201911589008363

# 002 단일 행 함수

## ❖ ADD\_MONTHS 함수

- ADD\_MONTHS 함수는 날짜에 월을 빼거나 더하는 함수이다.
- 결과는 날짜 타입으로 출력 된다.

ADD\_MONTHS(날짜, 숫자)

```
SELECT ADD_MONTHS(SYSDATE, 3), ADD_MONTHS(SYSDATE, -3) FROM dual;
```



	ADD_MONTHS(SYSDATE,3)	ADD_MONTHS(SYSDATE,-3)
1	21/04/10	20/10/10

# 002 단일 행 함수

## ❖ NEXT\_DAY 함수

- NEXT\_DAY 함수는 지정된 요일의 돌아오는 날짜가 언제인지 계산하는 함수이다.
- 문자로 '일요일', '월요일'과 같이 요일을 기술하면 된다.
- 숫자로 일요일은 1, 월요일은 2와 같이 기술 할 수 도 있다.

NEXT\_DAY(날짜, '요일' 또는 숫자)

```
SELECT NEXT_DAY(SYSDATE, 2), NEXT_DAY(SYSDATE, '월요일') FROM dual;
```



	NEXT_DAY(SYSDATE,2)	NEXT_DAY(SYSDATE,'월요일')
1	21/01/11	21/01/11



# 002 단일 행 함수

## ❖ LAST\_DAY 함수

- LAST\_DAY 함수는 월의 마지막 날짜를 계산해서 출력하는 함수이다.

LAST\_DAY(날짜)

```
SELECT LAST_DAY(SYSDATE) FROM dual;
```



	LAST_DAY(SYSDATE)
1	21/01/31

# 002 단일 행 함수

## ❖ ROUND, TRUNC 함수

- ROUND 함수는 지정된 값을 기준으로 반올림하는 함수이고, TRUNC는 지정 값을 기준으로 월 또는 연도로 절삭하는 함수 이다.
- ROUND 함수를 사용하면 날짜를 가장 가까운 월 또는 연도로 반올림 할 수 있다.
- TRUNC 함수를 사용하면 월을 절삭하면 해당 월의 1일이 되고 연도를 절삭하면 해당 연도의 1월 1일이 된다.
- 날짜별 예약 접수 현황, 월 단위 또는 주 단위에 가장 많이 팔린 상품, 고객의 최초 등록 날짜로부터 지금까지 거래한 총 기간 등 날짜와 관계된 데이터 값을 날짜 함수를 이용해 연산한다.

ROUND 또는 TRUNC (날짜, 지정 값)

# 002 단일 행 함수

```
SELECT ROUND(SYSDATE, 'MONTH'),  
ROUND(SYSDATE, 'YEAR'),  
TRUNC(SYSDATE, 'MONTH'),  
TRUNC(SYSDATE, 'YEAR')  
FROM dual;
```

	ROUND(SYSDATE,'MONTH')	ROUND(SYSDATE,'YEAR')	TRUNC(SYSDATE,'MONTH')	TRUNC(SYSDATE,'YEAR')
1	21/01/01	21/01/01	21/01/01	21/01/01

```
SELECT ROUND(TO_DATE('21-08-20'), 'MONTH'),  
ROUND(TO_DATE('21-08-20'), 'YEAR'),  
TRUNC(TO_DATE('21-08-20'), 'MONTH'),  
TRUNC(TO_DATE('21-08-20'), 'YEAR')  
FROM dual;
```

	ROUND(TO_DATE('21-08-20'),'MONTH')	ROUND(TO_DATE('21-08-20'),'YEAR')	TRUNC(TO_DATE('21-08-20'),'MONTH')	TRUNC(TO_DATE('21-08-20'),'YEAR')
1	21/09/01	22/01/01	21/08/01	21/01/01

# 002 단일 행 함수

## ❖ 변환 함수

- 오라클 데이터베이스 시스템은 각 열에 대해 데이터 타입을 규정하고 있다.
- SQL 문을 실행하기 위해 데이터 값의 데이터 타입을 변환해야 할 때도 있다. 이럴 때 사용하는 것이 바로 변환 함수이다.
- 시스템에서 정해진 데이터 형식에 대해 다른 데이터 타입의 데이터를 사용하는 것을 일부 허용한다.
- 데이터 타입 변환은 오라클 데이터베이스 시스템에 의해 자동으로(암시적으로) 혹은 사용자에게 의해서 수동으로(명시적으로) 실행 될 수 있다.

# 002 단일 행 함수

## ❖ 자동 데이터 타입 변환

- SQL 문을 조작할 때 오라클 데이터베이스 시스템은 특정한 경우에 데이터 타입을 자동으로 변환한다.
- VARCHAR2 타입으로 입력된 데이터 값 100은 NUMBER 타입으로 자동 변환 되어 산술 연산이 될 수 있고, 숫자 타입 데이터 값 100은 VARCHAR2 타입으로 자동 변환되어 저장될 수 있다.
- 데이터 타입을 원하는 값의 데이터 타입으로 변환 할 수 있는 경우에만 SQL 문이 올바르게 수행된다.
- 자동 데이터 타입 변환을 이용할 수 있더라도 SQL문의 성능과 안정성을 위해 수동 데이터 타입 변환으로 수행 할 것을 권장한다.

# 002 단일 행 함수

FROM	TO
VARCHAR2 혹은 CHAR	NUMBER(숫자)
VARCHAR2 혹은 CHAR	DATE(날짜)
NUMBER	VARCHAR2(문자)
DATE	VARCHAR2(문자)

```
SELECT 1 + '2' FROM dual;
```



	1 + '2'
1	3



# 002 단일 행 함수

## ❖수동 데이터 타입 변환

- SQL 은 사용자가 데이터 타입의 값을 다른 데이터 타입의 값으로 변환할 수 있도록 변환 함수를 제공한다.

함수	설명
TO_CHAR	숫자, 문자, 날짜 값을 지정 형식의 VARCHAR2 타입으로 변환한다.
TO_NUMBER	문자를 숫자 타입으로 변환한다.
TO_DATE	날짜를 나타내는 문자열을 지정 형식의 날짜 타입으로 변환한다.

# 002 단일 행 함수

## ❖ TO\_CHAR 함수

- TO\_CHAR 함수는 날짜, 숫자, 문자 값을 지정한 형식의 VARCHAR2 타입 문자열로 변환하는 함수이다.

TO\_CHAR(날짜 데이터 타입, '지정 형식')

지정 형식	설명	예	결과
CC	세기	TO_CHAR(SYSDATE, 'CC')	21
YYY or YYYY or YY or Y	연도	TO_CHAR(SYSDATE, 'YYYY')	2017, 017, 17, 7
Y,YYY	콤마가 있는 연도	TO_CHAR(SYSDATE, 'Y,YYY')	2,017
YEAR	문자로 표현된 연도	TO_CHAR(SYSDATE, 'YEAR')	TWENTY TWENTY-ONE
BC or AD	BC/AD 지시자	TO_CHAR(SYSDATE, 'AD')	서기
Q	분기	TO_CHAR(SYSDATE, 'Q')	1

# 002 단일 행 함수

지정 형식	설명	예	결과
MM	두 자리 값의 월	TO_CHAR(SYSDATE, 'MM')	01
MONTH	아홉 자리를 위해 공백을 추가한 월 이름	TO_CHAR(SYSDATE, 'MONTH')	1월
MON	세 자리의 약어로 된 월 이름(영문 설정일 경우)	TO_CHAR(SYSDATE, 'MON')	1월
RM	로마 숫자 월	TO_CHAR(SYSDATE, 'RM')	I
WW or W	연, 월의 주	TO_CHAR(SYSDATE, 'WW')	02, 2
DDD or DD or D	연, 월, 주의 일	TO_CHAR(SYSDATE, 'DD')	010, 10, 1
DAY	아홉 자리를 위해 공백을 추가한 요일 이름	TO_CHAR(SYSDATE, 'DAY')	일요일
DY	세 자리 약어로 된 요일 이름 (영문 설정일 경우)	TO_CHAR(SYSDATE, 'DY')	토
J	Julian day, BC 4713년 12월 31일 이후의 요일 수	TO_CHAR(SYSDATE, 'J')	2459225

# 002 단일 행 함수

```
SELECT TO_CHAR(SYSDATE, 'YYYY'),  
       TO_CHAR(SYSDATE, 'YY'),  
       TO_CHAR(SYSDATE, 'MM'),  
       TO_CHAR(SYSDATE, 'MON'),  
       TO_CHAR(SYSDATE, 'YYYYMMDD') AS 응용1,  
       TO_CHAR(TO_DATE('2021/01/01'), 'YYYY-MM-DD') AS 응용2  
FROM dual;
```

	TO_CHAR(SYSDATE, 'YYYY')	TO_CHAR(SYSDATE, 'YY')	TO_CHAR(SYSDATE, 'MM')	TO_CHAR(SYSDATE, 'MON')
1	2021	21	01	1월

응용1	응용2
20210110	2021-01-01

# 002 단일 행 함수

## ❖ 시간 지정 변환 형식

- 시간 지정 형식은 날짜의 시간 부분에 대해 표현 한다.

지정 형식	설명
AM or PM	오전 또는 오후 표시
HH / HH12 or HH24	시간표형 (1 ~ 12시 또는 0 ~ 23시)
MI	분(0 ~ 59분)
SS	초(0 ~ 59초)

```
SELECT TO_CHAR(SYSDATE, 'HH:MI:SS PM') AS 시간형식,  
TO_CHAR(SYSDATE, 'YYYY-MM-DD HH:MI:SS PM') AS 날짜시간형식  
FROM dual;
```

	❖ 시간형식	❖ 날짜시간형식
1	02:07:26 오후	2021-01-10 02:07:26 오후

## 002 단일 행 함수

- 시간 또는 날짜는 '/', '::', '-', ':' 등의 문자로 날짜와 시간을 꾸밀 수 있다.
- 하지만 긴 문자열을 넣으려고 한다면 꼭 큰 따옴표(") 안에 문자를 넣어야 된다.

```
SELECT TO_CHAR(SYSDATE, 'HH:MI:SS PM') AS 시간형식,  
TO_CHAR(SYSDATE, ' "날짜 : " YYYY-MM-DD, " 시간 : " HH:MI:SS PM') AS 날짜시간형식  
FROM dual;
```



	⌘ 시간형식	⌘ 날짜시간형식
1	02:11:45 오후	날짜 : 2021-01-10, 시간 : 02:11:45 오후



# 002 단일 행 함수

## ❖ TO\_CHAR 함수

- TO\_CHAR 함수는 날짜 뿐만 아니라 숫자 값도 지정한 형식의 문자열로 변환할 수 있다.

TO\_CHAR(숫자 데이터 타입, '지정 형식')

지정 형식	설명	예	결과
9	9로 출력 자릿수 지정	TO_CHAR(12345,'999999999')	12345
0	자릿수만큼 0을 출력	TO_CHAR(12345,'099999999')	00012345
\$	달러 기호	TO_CHAR(12345,'\$99999999')	\$12.345
L	지역 화폐 기호(원)	TO_CHAR(12345,':L:99999999')	₩12,345
.	명시한 위치에 소수점	TO_CHAR(12345,'9999999.99')	12345.00
,	명시한 위치에 쉼표	TO_CHAR(12345,'999,999,999')	12,345

## 002 단일 행 함수

```
SELECT TO_CHAR(12345, '999999999'),  
       TO_CHAR(12345, '099999999'),  
       TO_CHAR(12345, '$99999'),  
       TO_CHAR(12345, 'L99999'),  
       TO_CHAR(12345, '99999.99'),  
       TO_CHAR(12345, '99,999')  
FROM dual;
```

TO_CHAR(12345,'999999999')	TO_CHAR(12345,'099999999')	TO_CHAR(12345,'\$99999')
12345	000012345	\$12345

TO_CHAR(12345,'L99999')	TO_CHAR(12345,'99999.99')	TO_CHAR(12345,'99,999')
₩12345	12345.00	12,345

# 002 단일 행 함수

## ❖ TO\_NUMBER 함수

- TO\_NUMBER 함수는 숫자 타입의 문자열을 숫자 데이터 타입으로 변환하는 함수로 출력 결과는 변하지 않고 데이터 타입만 바꾼다.
- 데이터 타입 바뀐 데이터 값은 숫자 타입 함수에서 사용할 수 있다.

TO\_NUMBER(number)

```
SELECT TO_NUMBER('123'), 123+TO_NUMBER('123') FROM dual;
```

	TO_NUMBER('123')	123+TO_NUMBER('123')
1	123	246

# 002 단일 행 함수

## ❖ TO\_DATE 함수

- TO\_DATE 함수는 날짜를 나타내는 문자열을 명시된 날짜로 변환하는 함수이며 주로 출력 값을 명시된 형태로 나타나게 하거나 날짜를 계산할 때 사용한다.

TO\_DATE(문자열, '지정 형식')

```
SELECT TO_DATE('2020-01-01') FROM dual;
```



	TO_DATE('2020-01-01')
1	20/01/01

# 002 단일 행 함수

## ❖ NVL 함수

- NVL 함수는 NULL 값을 지정한 값으로 치환해 주는 함수이다.
- NVL2 함수는 조회한 행의 값이 NULL 값이 아닌 경우 지정한 값1을 NULL 값이면 지정한 값2로 치환해 주는 함수이다.
- 유용한 함수이므로 알고 있으면 좋다.

NVL(데이터 값, '지정한 값')

NVL2(데이터 값, '지정한 값1', '지정한 값2')

```
SELECT age, NVL(age, 20), NVL2(age, 'Y', 'N'),  
name, NVL2(name, 'Y', 'N')  
FROM mem;
```

	AGE	NVL(AGE,20)	NVL2(AGE,'Y','N')	NAME	NVL2(NAME,'Y','N')
1	20	20	Y	홍길동	Y
2	22	22	Y	이순신	Y
3	20	20	Y	유관순	Y
4	18	18	Y	홍길동	Y
5	(null)	20	N	신동엽	Y

# 002 단일 행 함수

## ❖ DECODE 함수

- DECODE 함수는 프로그래밍을 해 본 사람이라면 들어 본적이 있을 IF-THEN-ELSE-END의 조건 논리를 가능하게 하는 함수이다.
- 데이터 값이 조건 값과 일치하면 치환 값을 출력하고 일치 하지 않으면 기본값을 출력해준다.
- SQL 문을 작성할 때 매우 자주 쓰는 함수이다.

DECODE(열 이름, 조건 값, 치환 값, 기본값)

```
SELECT DECODE(age, 20, age+1, age) AS 나이,  
       DECODE(age, 20, '나이증가', '나이유지') AS 나이증가여부  
FROM mem;
```

	나이	나이증가여부
1	21	나이증가
2	22	나이유지
3	21	나이증가
4	18	나이유지
5	(null)	나이유지



# 002 단일 행 함수

## ❖ CASE 함수

- CASE 함수는 복잡한 조건 논리를 처리 할 수 있다.
- DECODE 함수 보다 복잡한 조건식을 여러 개 적용할 때 CASE 표현식을 이용하는 것이 편하다.
- DECODE 함수는 데이터 값이 정확히 맞거나 틀린 조건만 처리 할 수 있고, CASE 함수는 조건의 범위가 다양한 경우에 쉽게 처리 할 수 있다.

# 002 단일 행 함수

CASE

WHEN 조건 1 THEN 출력 값1

WHEN 조건 2 THEN 출력 값2

...

ELSE 출력 값3

END

# 002 단일 행 함수

```
SELECT age,  
CASE  
    WHEN age >= 20 THEN '성인'  
    WHEN age < 20 THEN '미성년자'  
    ELSE '나이없음'  
END AS 구분  
FROM mem ORDER BY id DESC;
```



	AGE	구분
1	20	성인
2	(null)	나이없음
3	22	성인
4	18	미성년자
5	20	성인

# 002 단일 행 함수

## ❖ RANK, DENSE\_RANK, ROW\_NUMBER 함수

- RANK, DENSE\_RANK, ROW\_NUMBER 함수는 데이터 값에 순위를 매기는 함수이다.
- 순위를 매기는 것은 동일하지만 사용방법이 조금씩 다르다.
- 즉, 공통 순위가 있을 때 출력을 어떻게 하느냐에 따라 용도가 달라진다.

함수	설명	순위 예
RANK	공통 순위를 출력하되 공통 순위만큼 건너뛰어 다음 순위를 출력한다.	1,2,2,4,....
DENSE_RANK	공통 순위를 출력하되 건너뛰지 않고 바로 다음 순위를 출력 한다.	1,2,2,3,....
ROW_NUMBER	공통 순위를 없이 출력한다.	1,2,3,4,...

# 002 단일 행 함수

RANK () OVER([PARTITION BY 열 이름] ORDER BY [열 이름])

```
SELECT name,  
RANK() OVER(ORDER BY age DESC) AS RANK나이순,  
DENSE_RANK() OVER(ORDER BY age DESC) AS DENSE나이순,  
ROW_NUMBER() OVER(ORDER BY age DESC) AS ROW나이순  
FROM mem;
```



NAME	RANK나이순	DENSE나이순	ROW나이순
신동엽	1	1	1
이순신	2	2	2
홍길동	3	3	3
유관순	3	3	4
홍길동	5	4	5

# 003 그룹 함수

## ❖ 그룹 함수

- 그룹 함수는 단일 행 함수와 달리 여러 행에 대해 함수가 적용되어 하나의 결과를 나타내는 함수이다. 이런 함수를 집계 함수라고도 부른다.
- 기준 열에 대해 같은 데이터 값끼리 그룹을 묶고 묶은 행의 집합에 대해 그룹 함수 연산이 필요하다면 GROUP BY 절을 이용하여 처리할 수 있다.
- 묶은 그룹에 대해 조건이 필요하다면 HAVING 절을 이용한다.

```
SELECT 그룹 함수(열 이름) FROM [테이블 이름]  
[WHERE 조건식] [ORDER BY 열 이름];
```



# 003 그룹 함수

함수	설명	예	null 처리
COUNT	행의 총 개수를 센다.	COUNT(age)	(*)의 경우 null 값도 개수로 포함한다.
SUM	합계	SUM(age)	null 값을 제외하고 연산
AVG	평균	AVG(age)	
MAX	최대값	MAX(age)	
MIN	최솟값	MIN(age)	
STDDEV	표준편차	STDDEV(age)	
VARIANCE	분산	VARIANCE(age)	

# 003 그룹 함수

## ❖ COUNT 함수

- COUNT 함수는 지정한 열의 행 갯수를 세는 함수이다.

COUNT(열 이름)

```
SELECT COUNT(id) AS 가입자수 FROM mem;
```



	가입자수
1	5

```
SELECT COUNT(age) AS 나이입력수1,  
COUNT(*) AS 나이입력수2 FROM mem;
```



	나이입력수1	나이입력수2
1	4	5

# 003 그룹 함수

## ❖ SUM, AVG 함수

- SUM 함수는 열의 데이터의 합계를 구하는 함수이고, AVG 함수는 열의 데이터의 평균을 구하는 함수이다.

SUM(열 이름), AVG(열 이름)

```
SELECT SUM(age) 나이총합, AVG(age) 나이평균1,  
SUM(age)/COUNT(age) 나이평균2,  
AVG(NVL(age, 0)) Null포함 FROM mem;
```



	❖ 나이총합	❖ 나이평균1	❖ 나이평균2	❖ NULL포함
1	80	20	20	16

# 003 그룹 함수

## ❖ MAX, MIN 함수

- MAX 함수는 최댓값을 출력하는 함수이고 MIN 함수는 최솟값을 출력하는 함수 이다.
- 모든 데이터 타입에 대해서 MAX 함수와 MIN 함수를 사용할 수 있다.
- 영어는 알파벳 순, 한글은 가나다 순, 숫자와 날짜는 숫자로 연산하여 결과를 출력 한다.

MAX (열 이름), MIN (열 이름)

```
SELECT MAX(id), MIN(id), MAX(age), MIN(age) FROM mem;
```



	MAX(ID)	MIN(ID)	MAX(AGE)	MIN(AGE)
1	you	hong	22	18

# 003 그룹 함수

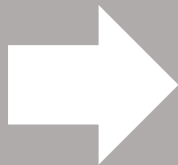
## ❖ GROUP BY 절

- SQL에서는 같은 데이터 값을 갖는 행끼리 묶어서 그룹화한 다음, 그에 해당하는 다른 열의 데이터 집합을 그룹 함수에 전달하여 연산할 수 있다. 이런 경우 GROUP BY 절을 사용한다.
- GROUP BY 절은 기존 열을 지정하여 그룹화하는 명령어이다.
- SELECT 절에 열 이름과 그룹 함수를 함께 기술했다면 GROUP BY 절을 반드시 사용해야 한다.
- GROUP BY 절 열 이름 순서대로 그룹화가 된다. 즉, GROUP BY 대그룹, 중그룹, 소그룹 형태가 된다.

SELECT 기준 열, 그룹 함수(열 이름) FROM [테이블 이름]  
[WHERE 조건식] GROUP BY [열 이름] [ORDER BY 열 이름];

# 003 그룹 함수

```
SELECT name, SUM(age) FROM mem GROUP BY name;
```



	NAME	SUM(AGE)
1	이순신	22
2	유관순	20
3	홍길동	38
4	신동엽	(null)



# 003 그룹 함수

## ❖HAVING 절

- HAVING 절은 그룹화 된 값에 조건식을 적용할 때 사용한다. 즉, WHERE 절에서는 그룹 함수를 사용할 수 없으므로 HAVING 절을 사용해 그룹 함수의 결과값에 대해 조건식을 적용할 수 있다.
- 일반적으로 HAVING 절은 GROUP BY 절 다음에 기술하는 것이 좋다.

```
SELECT [열 이름], 그룹 함수(열 이름) FROM [테이블 이름]  
[WHERE 조건식] GROUP BY [열 이름]  
[HAVING 조건식] [ORDER BY 열 이름]
```

# 003 그룹 함수

```
SELECT name, AVG(age) FROM mem GROUP BY name HAVING SUM(age) > 20;
```



	NAME	AVG(AGE)
1	이순신	22
2	홍길동	19

A photograph of a server room with rows of server racks on both sides of a central aisle. The racks have glass doors and internal components are visible, with many small blue lights glowing. The ceiling has several long, rectangular light fixtures. The overall atmosphere is dimly lit, emphasizing the blue light from the servers.

수고하셨습니다.