

SELECT

-기본-

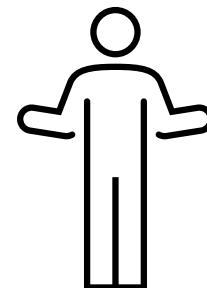
1. SQL 의미와 종류
2. SQL 실행순서
3. * 과 DISTINCT , AS
4. NULL 이란?
5. 자료형과 리터럴
6. SELECT 연산 / 함수

1. SQL 의미와 종류

SQL

Structured Query Language

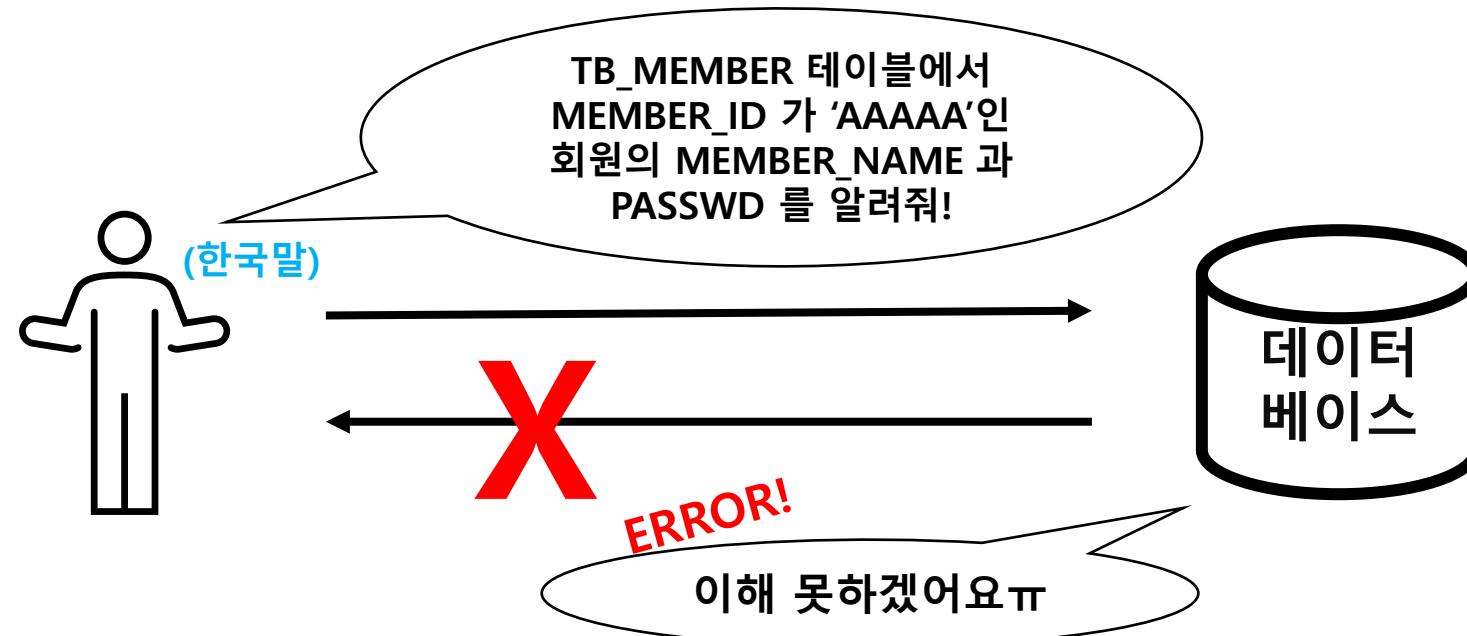
구조화된 질의 언어



데이터베이스가 이해할 수 있도록
특정 문법에 맞춰서 질의하는 것

예) 한국의 육하원칙(누가 , 무엇을 ..)
예) 영어는 주어 , 동사 , 목적어 ..

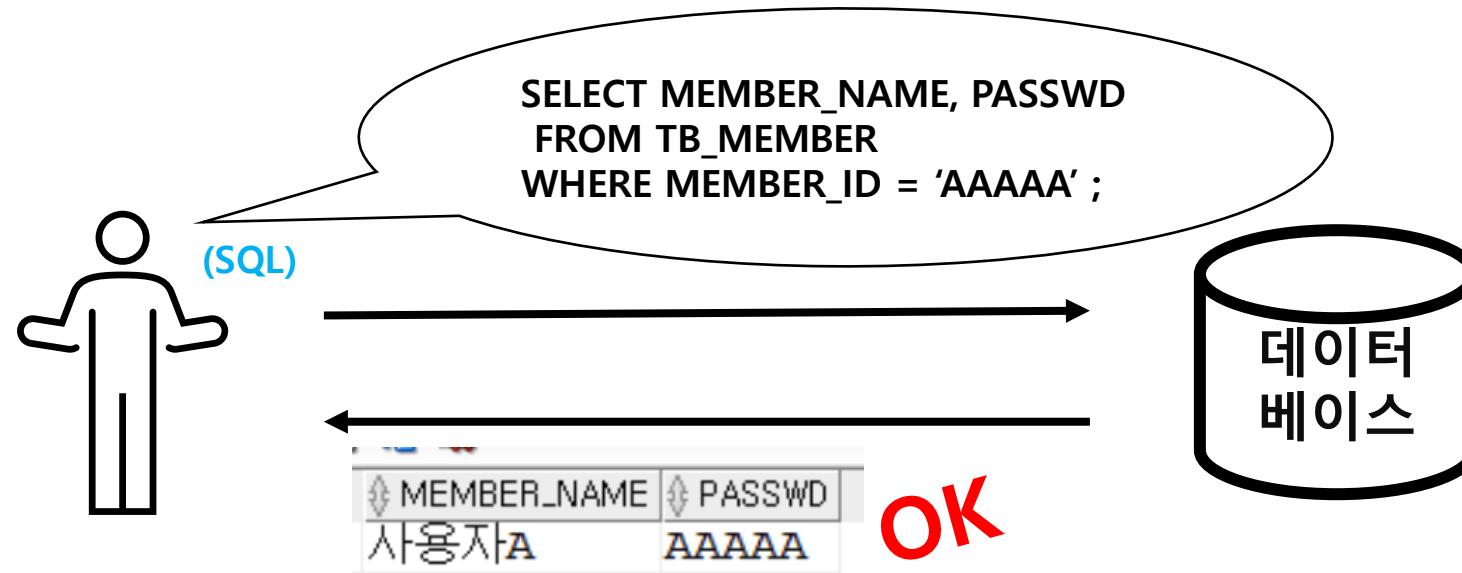
1. SQL 의미와 종류



(TB_MEMBER 테이블 내용)

MEMBER_ID	MEMBER_NAME	PASSWD	PAY_CARD_NO	JOIN_DY	GRADE_CD	GENDER	AGE	DEL_YN
AAAAAA	사용자A	AAAAAA	1111-1111-1111-1111	20200101	1	남	(null)	N
BBBBBB	사용자B	BBBBBB	2222-2222-2222-2222	20200327	2	여	25	N
CCCCCC	사용자C	CCCCCC	3333-3333-3333-3333	20210105	1	남	27	N
DDDDDD	사용자D	DDDDDD	4444-4444-4444-4444	20210630	3	여	30	N
EEEEEE	사용자E	EEEEEE	5555-5555-5555-5555	20210831	1	남	(null)	N
FFFFFF	사용자F	FFFFFF	6666-6666-6666-6666	20220216	3	여	35	N
GGGGGG	사용자G	GGGGGG	7777-7777-7777-7777	20220317	2	남	39	N
HHHHHH	사용자H	HHHHHH	8888-8888-8888-8888	20220812	5	(null)	44	N
IIIIII	사용자I	IIIIII	9999-9999-9999-9999	20230430	4	(null)	52	N

1. SQL 의미와 종류

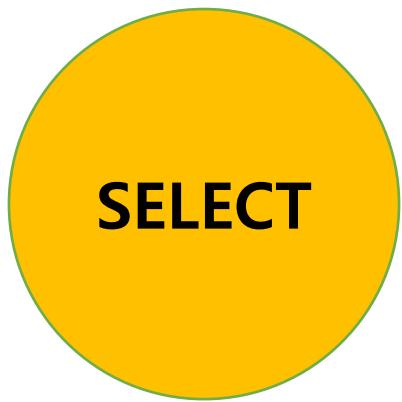


(TB_MEMBER 테이블 내용)

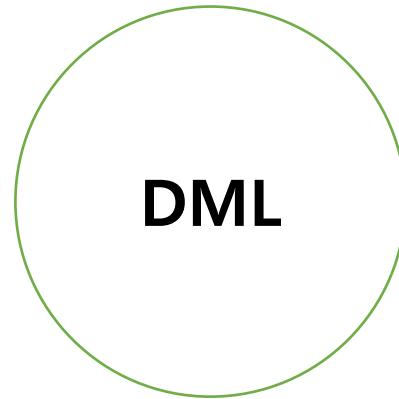
MEMBER_ID	MEMBER_NAME	PASSWD	PAY_CARD_NO	JOIN_DY	GRADE_CD	GENDER	AGE	DEL_YN
AAAAAA	사용자A	AAAAAA	1111-1111-1111-1111	20200101	1	남	(null)	N
BBBBBB	사용자B	BBBBBB	2222-2222-2222-2222	20200327	2	여	25	N
CCCCCC	사용자C	CCCCCC	3333-3333-3333-3333	20210105	1	남	27	N
DDDDDD	사용자D	DDDDDD	4444-4444-4444-4444	20210630	3	여	30	N
EEEEEE	사용자E	EEEEEE	5555-5555-5555-5555	20210831	1	남	(null)	N
FFFFFF	사용자F	FFFFFF	6666-6666-6666-6666	20220216	3	여	35	N
GGGGGG	사용자G	GGGGGG	7777-7777-7777-7777	20220317	2	남	39	N
HHHHHH	사용자H	HHHHHH	8888-8888-8888-8888	20220812	5	(null)	44	N
IIIIII	사용자I	IIIIII	9999-9999-9999-9999	20230430	4	(null)	52	N

1. SQL 의미와 종류

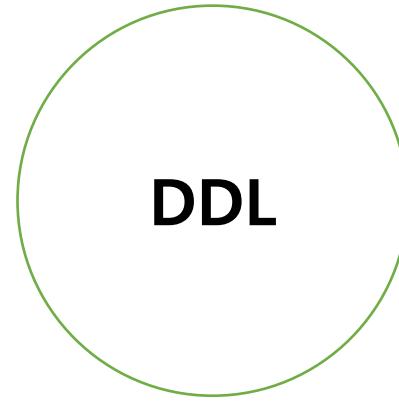
SQL 문법의 종류



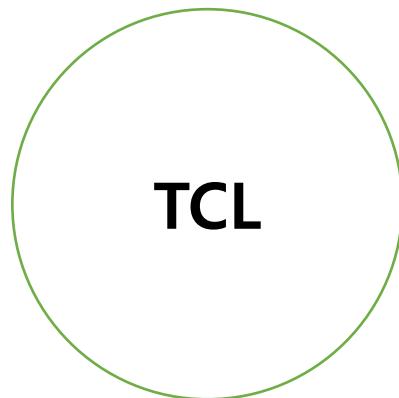
테이블에서 원하는
데이터를 조회한다.



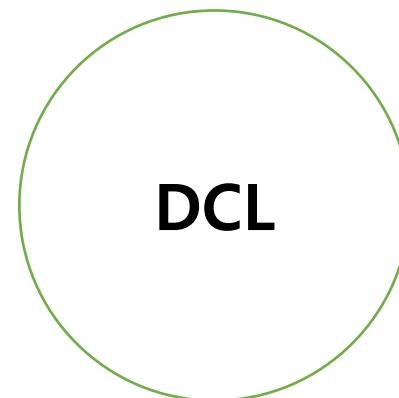
테이블에 데이터를
입력/삭제/수정한다.



테이블 같은 데이터 저장소
객체를 만들거나 수정한다.

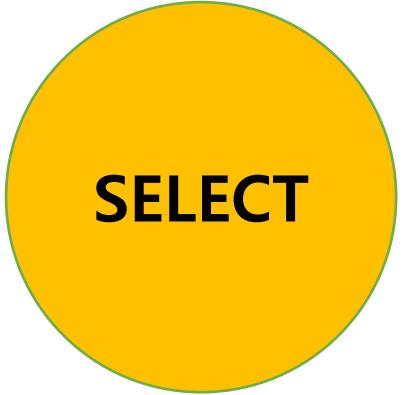


트랜잭션을 제어한다.

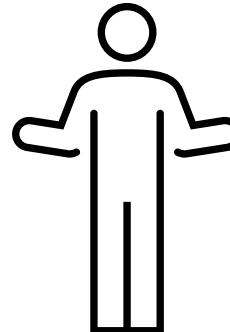
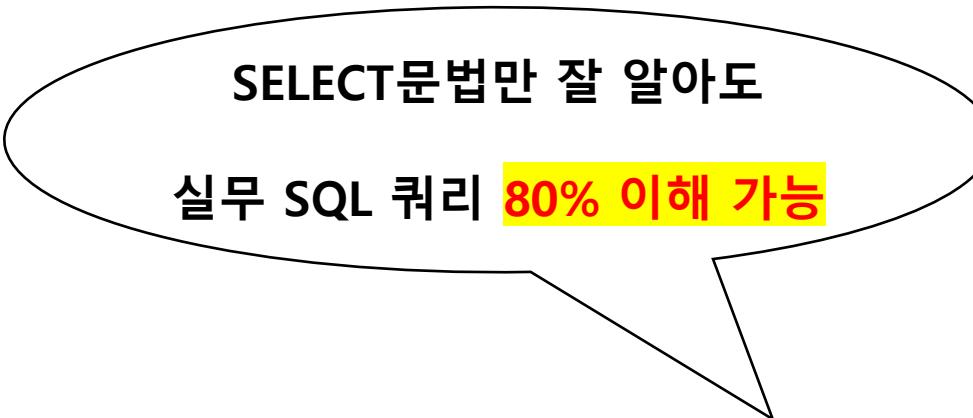


객체에 권한을 부여한다.

1. SQL 의미와 종류



테이블에서 원하는
데이터를 조회한다.



**SELECT
FROM
WHERE
GROUP BY
HAVING
ORDER BY**

테이블이란?

엔터티를 모델링함으로써 DB가 이해할 수 있게 만든 결과물로
데이터를 저장하는 저장소 기능을 가집니다.

(엔터티)

회원			
회원ID	비밀번호	이름	연락처
taewoo	Ex123!!	강태우	062-111-2222
haha12	Pwd23@	하동훈	010-2222-2222
newMan	New123!!	신입	010-6666-6666

(속성)

↓

← 인스턴스
-엔터티 안에 존재하는 각각의 데이터

회원

회원ID
비밀번호
이름
연락처

[개념적 데이터 모델링]

[논리적 데이터 모델링]

테이블이란?

엔터티를 물리적 모델링한 결과로 데이터를 저장하는 저장소 기능을 가집니다.

```
* TB_MEMBER 테이블 생성 */
CREATE TABLE TB_MEMBER (
    MEMBER_ID      VARCHAR2(30) NOT NULL, --회원ID
    MEMBER_NAME    VARCHAR2(20) NOT NULL, --회원이름
    PASSWD         VARCHAR2(50) NOT NULL, --패스워드
    PAY_CARD_NO   VARCHAR2(20) NOT NULL, --결제카드번호
    JOIN_DY        VARCHAR2(8) NOT NULL, --가입일자
    GRADE_CD      NUMBER NOT NULL, --등급코드
    GENDER         VARCHAR(5), --성별
    AGE            NUMBER(3,0) --나이
);

/* TB_MEMBER 테이블의 MEMBER_ID 컬럼을 기준으로 PRIMARY KEY 생성 */
ALTER TABLE TB_MEMBER ADD CONSTRAINT PK_MEMBER PRIMARY KEY(MEMBER_ID);

/* TB MEMBER 테이블에 테스트용 데이터 삽입 */

```



테이블 : TB_MEMBER

MEMBER_ID	MEMBER_NAME	PASSWD	PAY_CARD_NO	JOIN_DY
AAAAAA	사용자A	AAAAAA	1111-1111-1111-1111	2021-01-01
BBBBBB	사용자B	BBBBBB	2222-2222-2222-2222	2021-01-01
CCCCCC	사용자C	CCCCCC	3333-3333-3333-3333	2021-01-01
DDDDDD	사용자D	DDDDDD	4444-4444-4444-4444	2021-01-01
EEEEEE	사용자E	EEEEEE	5555-5555-5555-5555	2021-01-01
FFFFFF	사용자F	FFFFFF	6666-6666-6666-6666	2021-01-01
GGGGGG	사용자G	GGGGGG	7777-7777-7777-7777	2021-01-01
HHHHHH	사용자H	HHHHHH	8888-8888-8888-8888	2021-01-01
IIIIII	사용자I	IIIIII	9999-9999-9999-9999	2021-01-01

[물리적 데이터 모델링]

엔터티: 회원에서 테이블: TB_MEMBER로 진화!

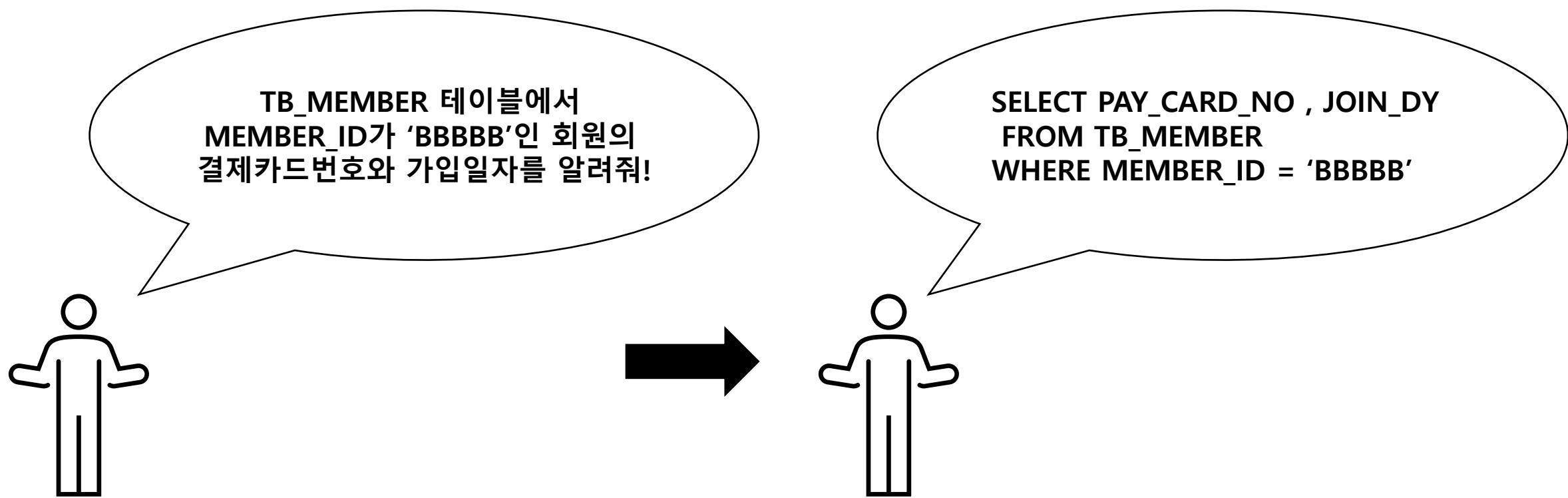
1. SQL 의미와 종류

테이블은 2차원의 행, 열 형태를 유지합니다

MEMBER_ID	MEMBER_NAME	PASSWD	PAY_CARD_NO	JOIN_DY	GRADE_CD	GENDER	AGE	DEL_YN
AAAAAA	사용자A	AAAAAA	1111-1111-1111-1111	20200101	1	남	(null)	N
BBBBBB	사용자B	BBBBBB	2222-2222-2222-2222	20200327	2	여	25	N
CCCCCC	사용자C	CCCCCC	3333-3333-3333-3333	20210105	1	남	27	N
DDDDDD	사용자D	DDDDDD	4444-4444-4444-4444	20210630	3	여	30	N
EEEEEE	사용자E	EEEEEE	5555-5555-5555-5555	20210831	1	남	(null)	N
FFFFFF	사용자F	FFFFFF	6666-6666-6666-6666	20220216	3	여	35	N
GGGGGG	사용자G	GGGGGG	7777-7777-7777-7777	20220317	2	남	39	N
HHHHHH	사용자H	HHHHHH	8888-8888-8888-8888	20220812	5	(null)	44	N
IIIIII	사용자I	IIIIII	9999-9999-9999-9999	20230430	4	(null)	52	N

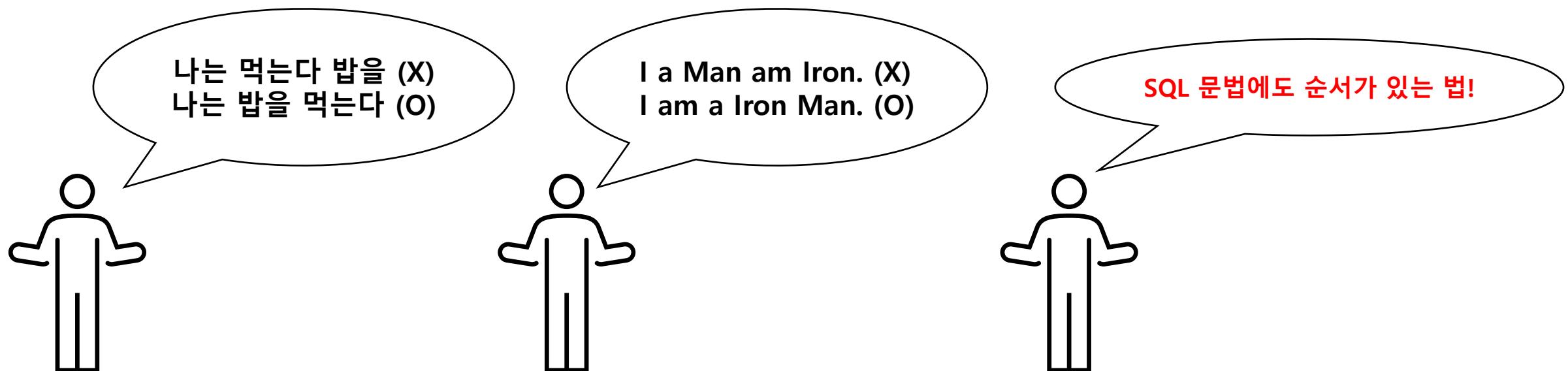
TB_MEMBER 테이블은 9개의 행(튜플) 과 9개의 열(컬럼) 으로 구성되어 있습니다.

따라 작성해봅시다.



2. SQL 실행 순서

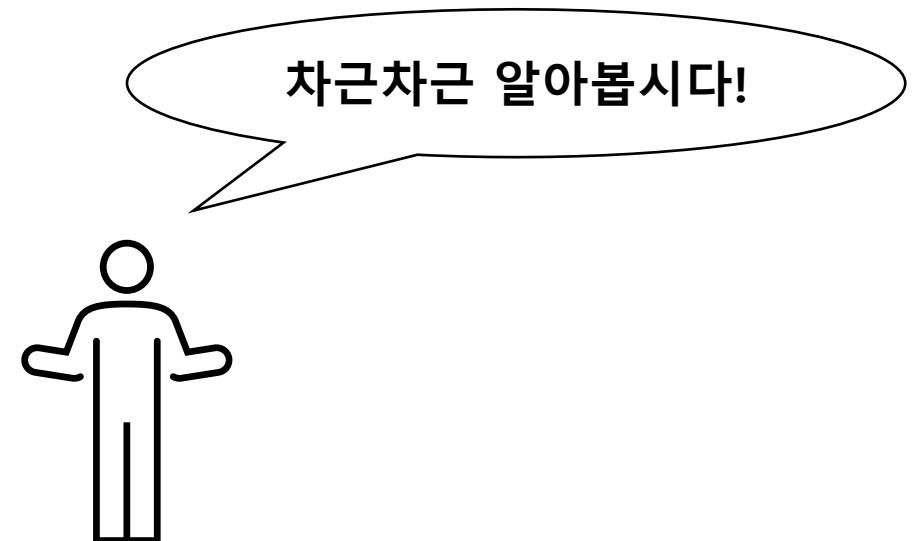
SQL 문법의 실행 순서는 매우 중요합니다!



2. SQL 실행 순서

SQL 문법 실행 순서

- 5** SELECT 출력하고 싶은 컬럼만 작성하기
- 1** FROM 데이터를 가져올 테이블 입력
- 2** WHERE 원하는 튜플만 가져오도록 필터링
- 3** GROUP BY 특정 컬럼을 기준으로 그룹화
- 4** HAVING 그룹화 상태의 데이터를 필터링
- 6** ORDER BY 특정 컬럼으로 정렬하기



2. SQL 실행 순서

따라 작성해봅시다.

- 3** `SELECT MEMBER_ID , MEMBER_NAME , GRADE_CD`
- 1** `FROM TB_MEMBER`
- 2** `WHERE GRADE_CD = 2 ;`

< 실행 순서 >

1. TB_MEMBER 테이블에서(FROM) 데이터를 가져오겠습니다.
2. TB_MEMBER 테이블에서 GRADE_CD (등급코드) 가 2 인 튜플만 가져오겠습니다.
3. 출력되는 튜플에 대해 MEMBER_ID , MEMBER_NAME , GRADE_CD 컬럼만 가져오겠습니다.

2. SQL 실행 순서

3 SELECT MEMBER_ID , MEMBER_NAME , GRADE_CD

1 FROM TB_MEMBER

2 WHERE GRADE_CD = 2 ;

< 실행 순서 >

1. TB_MEMBER 테이블에서(FROM) 데이터를 가져오겠습니다.

TB_MEMBER

MEMBER_ID	MEMBER_NAME	PASSWD	PAY_CARD_NO	JOIN_DY	GRADE_CD	GENDER	AGE	DEL_YN
AAAAAA	사용자A	AAAAAA	1111-1111-1111-1111	20200101	1	남	(null)	N
BBBBBB	사용자B	BBBBBB	2222-2222-2222-2222	20200327	2	여	25	N
CCCCCC	사용자C	CCCCCC	3333-3333-3333-3333	20210105	1	남	27	N
DDDDDD	사용자D	DDDDDD	4444-4444-4444-4444	20210630	3	여	30	N
EEEEEE	사용자E	EEEEEE	5555-5555-5555-5555	20210831	1	남	(null)	N
FFFFFF	사용자F	FFFFFF	6666-6666-6666-6666	20220216	3	여	35	N
GGGGGG	사용자G	GGGGGG	7777-7777-7777-7777	20220317	2	남	39	N
HHHHHH	사용자H	HHHHHH	8888-8888-8888-8888	20220812	5	(null)	44	N
IIIIII	사용자I	IIIIII	9999-9999-9999-9999	20230430	4	(null)	52	N

2. SQL 실행 순서

```
3 SELECT MEMBER_ID , MEMBER_NAME , GRADE_CD  
1 FROM TB_MEMBER  
2 WHERE GRADE_CD = 2 ;
```

< 실행 순서 >

2. TB_MEMBER 테이블에서
GRADE_CD (등급코드) 가 2 인 대상만 가져오겠습니다.

TB_MEMBER

MEMBER_ID	MEMBER_NAME	PASSWD	PAY_CARD_NO	JOIN_DY	GRADE_CD	GENDER	AGE	DEL_YN
AAAAAA	사용자A	AAAAAA	1111-1111-1111-1111	20200101	1	남	(null)	N
BBBBBB	사용자B	BBBBBB	2222-2222-2222-2222	20200327	2	여	25	N
CCCCCC	사용자C	CCCCCC	3333-3333-3333-3333	20210105	1	남	27	N
DDDDDD	사용자D	DDDDDD	4444-4444-4444-4444	20210630	3	여	30	N
EEEEEE	사용자E	EEEEEE	5555-5555-5555-5555	20210831	1	남	(null)	N
FFFFFF	사용자F	FFFFFF	6666-6666-6666-6666	20220216	3	여	35	N
GGGGGG	사용자G	GGGGGG	7777-7777-7777-7777	20220317	2	남	39	N
HHHHHH	사용자H	HHHHHH	8888-8888-8888-8888	20220812	5	(null)	44	N
IIIIII	사용자I	IIIIII	9999-9999-9999-9999	20230430	4	(null)	52	N

2. SQL 실행 순서

```
3 SELECT MEMBER_ID , MEMBER_NAME , GRADE_CD  
1 FROM TB_MEMBER  
2 WHERE GRADE_CD = 2 ;
```

< 실행 순서 >

2. TB_MEMBER 테이블에서
GRADE_CD (등급코드) 가 2 인 대상만 가져오겠습니다.

TB_MEMBER

MEMBER_ID	MEMBER_NAME	PASSWD	PAY_CARD_NO	JOIN_DY	GRADE_CD	GENDER	AGE	DEL_YN
BBBBB	사용자B	BBBBB	2222-2222-2222-2222	20200327	2	여	25	N
GGGGG	사용자G	GGGGG	7777-7777-7777-7777	20220317	2	남	39	N

2. SQL 실행 순서

```
3 SELECT MEMBER_ID , MEMBER_NAME , GRADE_CD  
1 FROM TB_MEMBER  
2 WHERE GRADE_CD = 2 ;
```

< 실행 순서 >

3. 출력되는 튜플에 대해

MEMBER_ID , MEMBER_NAME , GRADE_CD 컬럼만
가져오겠습니다.

TB_MEMBER

MEMBER_ID	MEMBER_NAME	PASSWD	PAY_CARD_NO	JOIN_DY	GRADE_CD	GENDER	AGE	DEL_YN
BBBBB	사용자B	BBBBB	2222-2222-2222-2222	20200327	2	여	25	N
GGGGG	사용자G	GGGGG	7777-7777-7777-7777	20220317	2	남	39	N

2. SQL 실행 순서

```
3 SELECT MEMBER_ID , MEMBER_NAME , GRADE_CD  
1 FROM TB_MEMBER  
2 WHERE GRADE_CD = 2 ;
```

< 실행 순서 >

3. 출력되는 튜플에 대해

MEMBER_ID , MEMBER_NAME , GRADE_CD 컬럼만
가져오겠습니다.

TB_MEMBER

	MEMBER_ID	MEMBER_NAME	GRADE_CD
1	BBBBBB	사용자B	2
2	GGGGGG	사용자G	2

2. SQL 실행 순서

실습해봅시다.

1. TB_MEMBER (회원) 테이블에서 MEMBER_NAME (회원이름) , GENDER (성별) , AGE (나이) 정보를 출력해주세요.
2. TB_GRADE (등급) 테이블에서 GRADE_CD (등급코드) , GRADE_NAME (등급이름) 정보를 출력해주세요.
3. TB_PRD (상품) 테이블에서 PRD_ID (상품ID) , PRD_NAME (상품이름) , PRD_INFO (상품정보) 정보를 출력해주세요.

1)

MEMBER_NAME	GENDER	AGE
사용자A	남	(null)
사용자B	여	25
사용자C	남	27
사용자D	여	30
사용자E	남	(null)
사용자F	여	35
사용자G	남	39
사용자H	(null)	44
사용자I	(null)	52

2)

GRADE_CD	GRADE_NAME
1	브론즈
2	실버
3	골드
4	VIP
5	VVIP

3)

PRD_ID	PRD_NAME	PRD_INFO
P0001	헤어드라이기	머리를 아주 잘 말려주는 헤어드라이기 입니다.
P0002	에어컨	더위를 쌈 쌈 날려주는 에어컨입니다.
P0003	세탁기	옷을 깨끗하게 세탁해주는 세탁기입니다.
P0004	건조기	빠르고 간편하게 옷을 건조할 수 있는 건조기입니다.
P0005	노트북	성능이 뛰어나고 가벼운 노트북입니다.
P0006	데스크탑	강력한 성능을 가진 데스크탑 컴퓨터입니다.
P0007	태블릿	가벼우면서도 다양한 기능을 갖춘 태블릿입니다.
P0008	애플14	다양한 기능을 가진 스마트폰입니다.
P0009	갤럭시s23	줌이 엄청나게 좋은 스마트폰입니다.
P0010	조아샴푸	머리가 개운해지는 샴푸입니다.
P0011	주전자	물을 끓이기에 적합한 주전자입니다.
P0012	전기밥솥	쌀맛이 좋아지는 전기밥솥입니다.
P0013	냄비	식재료를 끓이기에 적합한 냄비입니다.
P0014	칼	식재료를 자르기에 적합한 칼입니다.
P0015	수세미	욕실 청소에 사용되는 수세미입니다.
P0017	곰팡이제거제	곰팡이를 제거할 때 최고입니다.
P0018	샤워기	강력한 물줄기로 청결한 목욕을 즐길 수 있는 샤워기입니다.
P0019	린스	찰랑거리는 머리를 가지고 싶다면 이 제품을 기억하세요
P0020	수건	호텔에서도 사용되는 고급소재의 친환경 타올 소재의 수건입니다.

2. SQL 실행 순서

실습 답

1. TB_MEMBER (회원) 테이블에서 MEMBER_NAME (회원이름) , GENDER (성별) , AGE (나이) 정보를 출력해주세요.
2. TB_GRADE (등급) 테이블에서 GRADE_CD (등급코드) , GRADE_NAME (등급이름) 정보를 출력해주세요.
3. TB_PRD (상품) 테이블에서 PRD_ID (상품ID) , PRD_NAME (상품이름) , PRD_INFO (상품정보) 정보를 출력해주세요.

1)

```
SELECT MEMBER_NAME  
      , GENDER  
      , AGE  
FROM TB_MEMBER ;
```

2)

```
SELECT GRADE_CD  
      , GRADE_NAME  
FROM TB_GRADE ;
```

3)

```
SELECT PRD_ID  
      , PRD_NAME  
      , PRD_INFO  
FROM TB_PRD ;
```


3. * 과 DISTINCT , AS

SELECT * FROM TB_PRD ;

TB_PRD (상품) 테이블의 **모든 컬럼** 정보를 출력합니다.

(아래처럼 모든 컬럼을 직접 입력해도 결과는 같아요)

```
SELECT PRD_ID  
      , PRD_NAME  
      , PRD_TYPE  
      , PRD_INFO  
      , PRD_PRICE  
      , SELL_COMP_NAME  
      , REG_DATE  
FROM TB_PRD ;
```

***(애스터리스크)** 는 SELECT 뒤에 사용되며

테이블 내의 모든 컬럼 정보를 출력합니다.

3. * 과 DISTINCT , AS

실습해봅시다.

1. TB_MEMBER (회원) 테이블의 모든 컬럼 정보를 출력해주세요.
2. TB_PRD (상품) 테이블의 모든 컬럼 정보를 출력해주세요.

1)

MEMBER_ID	MEMBER_NAME	PASSWD	PAY_CARD_NO	JOIN_DY	GRADE_CD	GENDER	AGE	DEL_YN
AAAAAA	사용자A	AAAAAA	1111-1111-1111-1111	20200101	1	남	(null)	N
BBBBBB	사용자B	BBBBBB	2222-2222-2222-2222	20200327	2	여	25	N
CCCCCC	사용자C	CCCCCC	3333-3333-3333-3333	20210105	1	남	27	N
DDDDDD	사용자D	DDDDDD	4444-4444-4444-4444	20210630	3	여	30	N
EEEEEE	사용자E	EEEEEE	5555-5555-5555-5555	20210831	1	남	(null)	N
FFFFFF	사용자F	FFFFFF	6666-6666-6666-6666	20220216	3	여	35	N
GGGGGG	사용자G	GGGGGG	7777-7777-7777-7777	20220317	2	남	39	N
HHHHHH	사용자H	HHHHHH	8888-8888-8888-8888	20220812	5	(null)	44	N
IIIIII	사용자I	IIIIII	9999-9999-9999-9999	20230430	4	(null)	52	N

2)

PRD_ID	PRD_NAME	PRD_TYPE	PRD_INFO	PRD_PRICE	SELL_COMP_NAME	REG_DATE
P0001	헤어드라이기	가전	머리를 아주 잘 말려주는 헤어드라이기 입니다.	30000	LG	2023-05-06 21:32:33
P0002	에어컨	가전	더위를 쐐 쏘 날려주는 에어컨입니다.	1500000	삼성	2023-05-06 21:32:33
P0003	세탁기	가전	옷을 깨끗하게 세탁해주는 세탁기입니다.	600000	LG	2023-05-06 21:32:33
P0004	건조기	가전	빠르고 간편하게 옷을 건조할 수 있는 건조기입니다.	800000	삼성	2023-05-06 21:32:33
P0005	노트북	컴퓨터	성능이 뛰어나고 가벼운 노트북입니다.	1500000	Apple	2023-05-06 21:32:33
P0006	데스크탑	컴퓨터	강력한 성능을 가진 데스크탑 컴퓨터입니다.	2000000	Dell	2023-05-06 21:32:33
P0007	태블릿	컴퓨터	가벼우면서도 다양한 기능을 갖춘 태블릿입니다.	800000	삼성	2023-05-06 21:32:33
P0008	애플14	스마트폰	다양한 기능을 가진 스마트폰입니다.	1200000	Apple	2023-05-06 21:32:33
P0009	갤럭시S23	스마트폰	중이 엄청나게 좋은 스마트폰입니다.	1500000	삼성	2023-05-06 21:32:33
P0010	조아샴푸	욕실용품	머리가 개운해지는 샴푸입니다.	20000	LG	2023-05-06 21:32:34
P0011	주전자	주방용품	물을 끓이기에 적합한 주전자입니다.	20000	LG	2023-05-06 21:32:34
P0012	전기밥솥	주방용품	쌀맛이 좋아지는 전기밥솥입니다.	80000	삼성	2023-05-06 21:32:34
P0013	냄비	주방용품	식재료를 끓이기에 적합한 냄비입니다.	30000	매일유업	2023-05-06 21:32:34
P0014	칼	주방용품	식재료를 자르기에 적합한 칼입니다.	15000	매일유업	2023-05-06 21:32:34
P0015	수세미	욕실용품	욕실 청소에 사용되는 수세미입니다.	5000	3M	2023-05-06 21:32:34
P0017	곰팡이제거제	욕실용품	곰팡이를 제거할 때 최고입니다.	10000	3M	2023-05-06 21:32:34
P0018	샤워기	욕실용품	강력한 물줄기로 청결한 목욕을 즐길 수 있는 샤워기입니다.	50000	다우니	2023-05-06 21:32:34
P0019	린스	욕실용품	찰랑거리는 머리를 가지고 싶다면 이 제품을 기억하세요	20000	다우니	2023-05-06 21:32:34
P0020	수건	욕실용품	호텔에서도 사용되는 고급소재의 친환경 타올 소재의 수건입니다.	5000	다우니	2023-05-06 21:32:34

3. * 과 DISTINCT , AS

답)

1. TB_MEMBER (회원) 테이블의 모든 컬럼 정보를 출력해주세요.
2. TB_PRD (상품) 테이블의 모든 컬럼 정보를 출력해주세요.

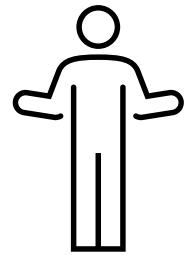
1) `SELECT *`

`FROM TB_MEMBER ;`

2) `SELECT *`

`FROM TB_PRD ;`

3. * 과 DISTINCT , AS



실무 TIP !

* 은 잘 사용하지 않습니다.

이유1) 성능이 감소합니다

이유2) 어떤 컬럼 정보를 이용하는지 눈으로 확인이 어렵습니다

-- 회원의 MEMBER_ID 와 MEMBER_NAME 을 알고싶어요

```
SELECT MEMBER_ID  
      , MEMBER_NAME  
  FROM TB_MEMBER ;
```

MEMBER_ID	MEMBER_NAME
AAAAAA	사용자A
BBBBBB	사용자B
CCCCCC	사용자C
DDDDDD	사용자D
EEEEEE	사용자E
FFFFFF	사용자F
GGGGGG	사용자G
HHHHHH	사용자H
IIIIII	사용자I

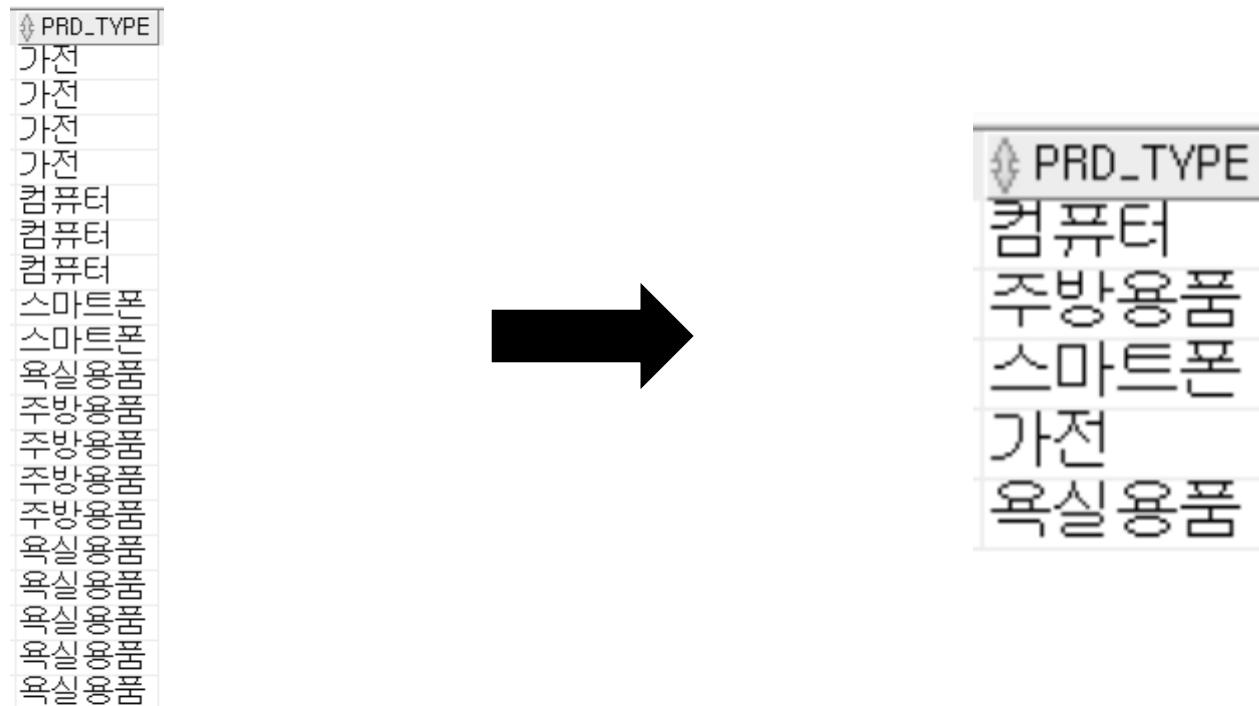
```
SELECT *  
  FROM TB_MEMBER ;
```

MEMBER_ID	MEMBER_NAME	PASSWD	PAY_CARD_NO	JOIN_DY	GRADE_CD	GENDER	AGE	DEL_VN
AAAAAA	사용자A	AAAAAA	1111-1111-1111-1111	20200101	1남	(null) N		
BBBBBB	사용자B	BBBBBB	2222-2222-2222-2222	20200327	2여	25N		
CCCCCC	사용자C	CCCCCC	3333-3333-3333-3333	20210105	1남	27N		
DDDDDD	사용자D	DDDDDD	4444-4444-4444-4444	20210630	3여	30N		
EEEEEE	사용자E	EEEEEE	5555-5555-5555-5555	20210831	1남	(null) N		
FFFFFF	사용자F	FFFFFF	6666-6666-6666-6666	20220216	3여	35N		
GGGGGG	사용자G	GGGGGG	7777-7777-7777-7777	20220317	2남	39N		
HHHHHH	사용자H	HHHHHH	8888-8888-8888-8888	20220812	5(null)	44N		
IIIIII	사용자I	IIIIII	9999-9999-9999-9999	20230430	4(null)	52N		

3. * 과 DISTINCT , AS

SELECT DISTINCT PRD_TYPE FROM TB_PRD ;

TB_PRD (상품) 테이블의 PRD_TYPE (상품타입) 값을 중복없이 출력합니다.



DISTINCT는 SELECT 뒤, 컬럼 앞에 사용되며 해당 컬럼 정보에 대해 중복을 제거합니다.

실습해봅시다

1. TB_MEMBER 테이블에서 GENDER (성별) 컬럼만 중복 없이 출력해주세요.
2. TB_MEMBER 테이블에서 GRADE_CD (등급코드) 컬럼만 중복 없이 출력해주세요.

* 출력되는 순서는 상관없습니다.

1)

GENDER
(null)
여
남

2)

GRADE_CD
1
2
5
4
3

답)

1. TB_MEMBER 테이블에서 GENDER (성별) 컬럼만 중복 없이 출력해주세요.
2. TB_MEMBER 테이블에서 GRADE_CD (등급코드) 컬럼만 중복 없이 출력해주세요.

* 출력되는 순서는 상관없습니다.

1)

```
SELECT DISTINCT GENDER  
FROM TB_MEMBER ;
```

2)

```
SELECT DISTINCT GRADE_CD  
FROM TB_MEMBER ;
```

3. * 과 DISTINCT , AS

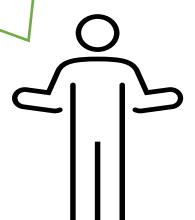
AS 는 SELECT 부분에서 출력하려는 컬럼에 대해 새로운 별명(ALIAS)를 부여할 수 있습니다.
(약간 힙합의 A.K.A)

```
SELECT MEMBER_ID      AS M_ID  
     , MEMBER_NAME    AS M_NAME  
     , PAY_CARD_NO   AS P_CARD  
FROM TB_MEMBER ;
```

AS 사용시 주의사항

- (1) 띄어쓰기 불가
- (2) 영문자로 시작해야함
- (3) 예약어 불가
- (4) 특수문자는 \$, _ , # 만 가능

*자바의 명명규칙과 거의 동일합니다.



4. NULL 이란?

```
SELECT *
FROM TB_MEMBER
WHERE MEMBER_ID = 'HHHHH';
```

MEMBER_ID	MEMBER_NAME	PASSWD	PAY_CARD_NO	JOIN_DY	GRADE_CD	GENDER	AGE	DEL_YN
HHHHH	사용자H	HHHHH	8888-8888-8888-8888	20220812	5	(null)	44	N

MEMBER_ID 가 'HHHHH' 인 회원의 MEMBER_NAME(회원이름) 은 사용자H 입니다.

결제할 수 있는 카드번호인 PAY_CARD_NO 의 값은 8888-8888-8888-8888 입니다.

이 사용자의 GENDER (성별) 은 아직 정해지지 않았습니다.

4. NULL 이란?

NULL이 필요한 이유

(1) 아직 어떤 값이 들어오지 않았음을 표현할 수 있다.

새로운 회원가입 요청이 들어왔습니다.

MEMBER_ID : JJJJJ

MEMBER_NAME : 새로운자

PASSWD : JJJJJ

PAY_CARD_NO : AAAA-AAAA-AAAA-AAAA

JOIN_DY : 20230519

GRADE_CD : 1

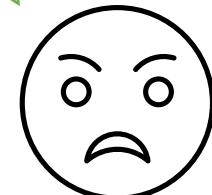
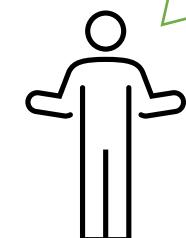
AGE : 30

DEL_YN : N

MEMBER_ID	MEMBER_NAME	PASSWD	PAY_CARD_NO	JOIN_DY	GRADE_CD	GENDER	AGE	DEL_YN
JJJJJ	새로운자	JJJJJ	AAAA-AAAA-AAAA-AAAA	20230519	1		30	N

GENDER(성별) 정보가 없는데

이거 입력해도 되는거에요?



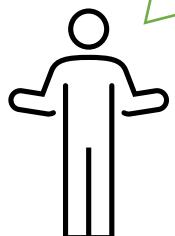
4. NULL 이란?

NULL이 필요한 이유

(1) 아직 어떤 값이 들어오지 않았음을 표현할 수 있다.

새로운 회원가입 요청이 들어왔습니다.

MEMBER_ID : JJJJJ
MEMBER_NAME : 새로운자
PASSWD : JJJJJ
PAY_CARD_NO : AAAA-AAAA-AAAA-AAAA
JOIN_DY : 20230519
GRADE_CD : 1
AGE : 30
DEL_YN : N



복사해서 실행해봅시다!

```
INSERT INTO TB_MEMBER (
    MEMBER_ID
    , MEMBER_NAME
    , PASSWD
    , PAY_CARD_NO
    , JOIN_DY
    , GRADE_CD
    , AGE
    , DEL_YN
) VALUES (
    'JJJJJ'
    , '새로운자'
    , 'JJJJJ'
    , 'AAAA-AAAA-AAAA-AAAA'
    , '20230519'
    , 1
    , 30
    , 'N'
);
```

4. NULL 이란?

NULL이 필요한 이유

(2) 테이블의 특성상 행 X 열 형태를 유지를 해야합니다.

새로운 회원가입 요청이 들어왔습니다.

MEMBER_ID : JJJJJ

MEMBER_NAME : 새로운자

PASSWD : JJJJJ

PAY_CARD_NO : AAAA-AAAA-AAAA-AAAA

JOIN_DY : 20230519

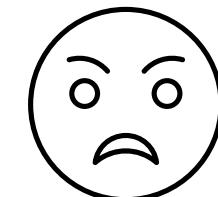
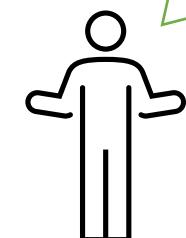
GRADE_CD : 1

AGE : 30

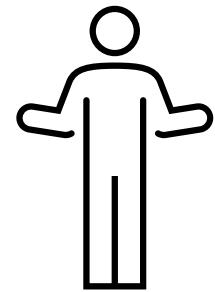
DEL_YN : N

MEMBER_ID	MEMBER_NAME	PASSWD	PAY_CARD_NO	JOIN_DY	GRADE_CD	GENDER	AGE	DEL_YN
JJJJJ	새로운자	JJJJJ	AAAA-AAAA-AAAA-AAAA	20230519	1		30	N

저거 하나 때문에 데이터를 안 넣을 수도 없고..
Null 값을 넣어서 행x열 형태를 유지하자!



5. 자료형과 리터럴



아래에 3개의 바구니가 있습니다.

각각의 통에는 조류 , 포유류 , 양서류 만 들어갈 수 있습니다.



조류

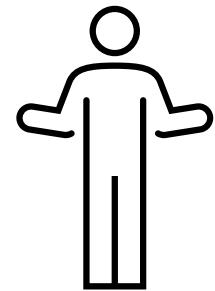


포유류



양서류

5. 자료형과 리터럴



아래에 3개의 바구니가 있습니다.
각각의 통에는 숫자형, 문자형, 날짜형만 들어갈 수 있습니다.



숫자형

1
1.42

5000



문자형

'A'
'3살'

'강태우'

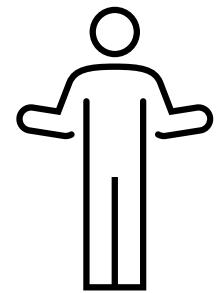


날짜형

2023/01/12

2023-01-12 14:12:12

5. 자료형과 리터럴



이 때 입력되는 값들을 데이터이자 **리터럴**이라고 부르며
값들을 보관하는 **바구니는 특정한 자료형을 가지게 됩니다.**



숫자형

1
1.42

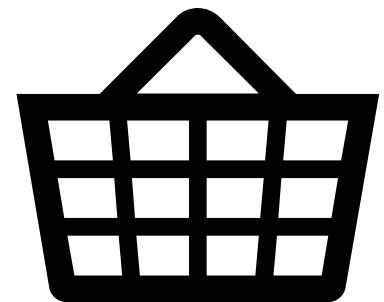
5000



문자형

'A'
'안녕하세요'

'강태우'
'3살'

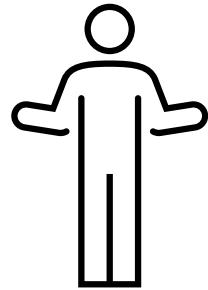


날짜형

2023/01/12

2023-01-12 14:12:12

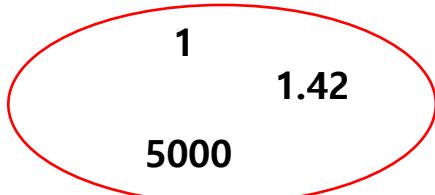
5. 자료형과 리터럴



이 때 입력되는 값들을 데이터이자 **리터럴**이라고 부르며
값들을 보관하는 **바구니는 특정한 자료형을 가지게 됩니다.**



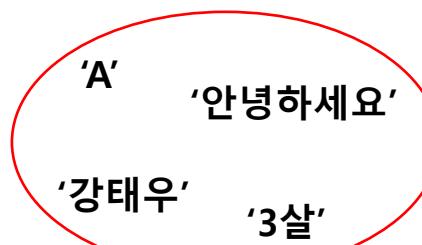
자료형 -> 숫자형



숫자형 리터럴



자료형 -> 문자형



문자형 리터럴



자료형 -> 날짜형



날짜형 리터럴

5. 자료형과 리터럴

테이블을 만들 때 각 속성(컬럼)에 대해 특정 자료형을 부여합니다.

이렇게 하면 일관성 있게 값을 입력할 수 있습니다. (DDL에서 추가로 배울 예정)

```
/* TB_PRD 테이블 생성*/
CREATE TABLE TB_PRD (
    PRD_ID           문자형      --상품 정보가 들어있는 테이블
    VARCHAR2(30)     NOT NULL   ,
    PRD_NAME         VARCHAR2(50)  NOT NULL   ,
    PRD_TYPE         VARCHAR2(50)  NOT NULL   ,
    PRD_INFO         VARCHAR2(4000) NOT NULL   ,
    PRD_PRICE        NUMBER      숫자형      NOT NULL   ,
    SELL_COMP_NAME   VARCHAR2(50)  NOT NULL   ,
    REG_DATE         DATE       날짜형      NOT NULL
)
;
```

PRD_ID	PRD_NAME	PRD_TYPE	PRD_INFO	PRD_PRICE	SELL_COMP_NAME	REG_DATE
P0001	헤어드라이기	가전	머리를 아주 잘 말려주는 헤어드라이기 입니다.	30000	LG	2023-05-06 21:32:33
P0002	에어컨	가전	더위를 싹 싹 날려주는 에어컨입니다.	1500000	삼성	2023-05-06 21:32:33
P0003	세탁기	가전	옷을 깨끗하게 세탁해주는 세탁기입니다.	600000	LG	2023-05-06 21:32:33
P0004	거조기	가전	빠르고 가파르게 오르 거조하스 이느 거조기입니다.	800000	사서	2023-05-06 21:32:33

실습해봅시다.

자료형과 리터럴 테스트를 위해 테이블을 만들어보겠습니다.
아래 쿼리(SQL)을 복사해주세요.

```
CREATE TABLE 자료형테스트 (    --자료형테스트 라는 이름의 테이블을 만듭니다.
    문자컬럼 VARCHAR2(10) ,      --문자컬럼 이라는 컬럼에 문자형 자료형(VARCHAR2)부여
    숫자컬럼 NUMBER      ,      --숫자컬럼 이라는 컬럼에 숫자형 자료형(NUMBER) 부여
    날짜컬럼 DATE ) ;        --날짜컬럼 이라는 컬럼에 날짜형 자료형(DATE) 부여
```

실습해봅시다. (아래 데이터를 복사)

INSERT INTO 자료형테스트 (문자컬럼 , 숫자컬럼 , 날짜컬럼) VALUES ('A' , 1 , SYSDATE); --될까요?

INSERT INTO 자료형테스트 (문자컬럼 , 숫자컬럼 , 날짜컬럼) VALUES ('A' , '3살' , SYSDATE); --될까요?

INSERT INTO 자료형테스트 (문자컬럼 , 숫자컬럼 , 날짜컬럼) VALUES ('A' , 3 , 1); --될까요?

INSERT INTO 자료형테스트 (문자컬럼 , 숫자컬럼 , 날짜컬럼) VALUES ('BB' , '3' , SYSDATE); --될까요?

SELECT

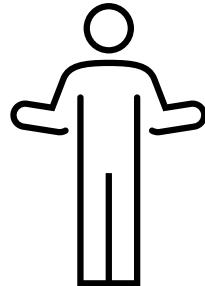
-연산&함수-

1. SELECT에 리터럴 사용하기
2. SELECT에서 사칙연산하기
3. SELECT에서 연결연산하기
4. 내장형함수 & 형변환함수 사용하기
5. NULL 함수 사용하기

1. SELECT에 리터럴 사용하기

```
SELECT PRD_ID  
      , PRD_NAME  
      , PRD_PRICE  
      , 10  
FROM TB_PRD;
```

TB_PRD 테이블에 있는 컬럼이 아니라
숫자형 리터럴(10)을 SELECT에 입력하면
어떻게 될까요?



1. SELECT에 리터럴 사용하기

```
SELECT PRD_ID  
      , PRD_NAME  
      , PRD_PRICE  
      , 10  
FROM TB_PRD;
```

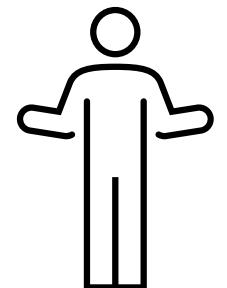
◆ PRD_ID	◆ PRD_NAME	◆ PRD_PRICE	◆ 10
P0001	헤어드라이기	30000	10
P0002	에어컨	1500000	10
P0003	세탁기	600000	10
P0004	건조기	800000	10
P0005	노트북	1500000	10
P0006	데스크탑	2000000	10
P0007	태블릿	800000	10
P0008	애플14	1200000	10
P0009	갤럭시S23	1500000	10
P0010	조아샴푸	20000	10
P0011	주전자	20000	10
P0012	전기밥솥	80000	10
P0013	냄비	30000	10
P0014	칼	15000	10
P0015	수세미	5000	10
P0017	곰팡이제거제	10000	10
P0018	샤워기	50000	10
P0019	린스	20000	10
P0020	수건	5000	10

정답 :

TB_PRD 테이블에서 출력될 튜플 수 만큼

똑같은 값이 반복 출력됩니다.

왜냐면 행X열 형태를 맞춰야 해서음

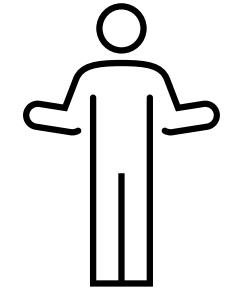


1. SELECT에 리터럴 사용하기

```
SELECT PRD_ID  
      , PRD_NAME  
      , PRD_PRICE  
      , 10 AS 리터럴  
FROM TB_PRD;
```

PRD_ID	PRD_NAME	PRD_PRICE	리터럴
P0001	헤어드라이기	30000	10
P0002	에어컨	1500000	10
P0003	세탁기	600000	10
P0004	건조기	800000	10
P0005	노트북	1500000	10
P0006	데스크탑	2000000	10
P0007	태블릿	800000	10
P0008	애플14	1200000	10
P0009	갤럭시S23	1500000	10
P0010	조아샴푸	20000	10
P0011	주전자	20000	10
P0012	전기밥솥	80000	10
P0013	냄비	30000	10
P0014	칼	15000	10
P0015	수세미	5000	10
P0017	곰팡이제거제	10000	10
P0018	샤워기	50000	10
P0019	린스	20000	10
P0020	수건	5000	10

또한 리터럴에도
AS(별칭)를 부여 가능합니다.

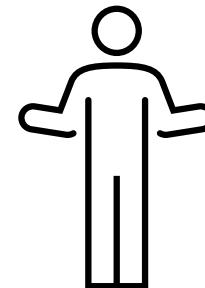


2. SELECT 에서 사칙연산하기

```
SELECT PRD_ID  
      , PRD_NAME  
      , PRD_PRICE  
      , 5000  
      , PRD_PRICE + 5000  
FROM TB_PRD ;
```

이번에는 SELECT 에서 사칙연산을 해보겠습니다.

PRD_PRICE (상품가격) 이라는 컬럼에 5000 을 더해보면(+) 어떻게 될까요?

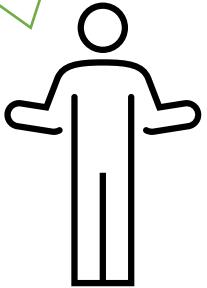


2. SELECT 에서 사칙연산하기

```
SELECT PRD_ID  
      , PRD_NAME  
      , PRD_PRICE  
      , 5000  
      , PRD_PRICE + 5000  
FROM TB_PRD ;
```

PRD_ID	PRD_NAME	PRD_PRICE	5000	PRD_PRICE+5000
P0001	헤어드라이기	30000	5000	35000
P0002	에어컨	1500000	5000	1505000
P0003	세탁기	600000	5000	605000
P0004	건조기	800000	5000	805000
P0005	노트북	1500000	5000	1505000
P0006	데스크탑	2000000	5000	2005000
P0007	태블릿	800000	5000	805000
P0008	애플14	1200000	5000	1205000
P0009	갤럭시S23	1500000	5000	1505000
P0010	조아샴푸	20000	5000	25000
P0011	주전자	20000	5000	25000
P0012	전기밥솥	80000	5000	85000
P0013	냄비	30000	5000	35000
P0014	칼	15000	5000	20000
P0015	수세미	5000	5000	10000
P0017	곰팡이제거제	10000	5000	15000
P0018	샤워기	50000	5000	55000
P0019	린스	20000	5000	25000
P0020	수건	5000	5000	10000

각각의 투플에 있는 PRD_PRICE 컬럼에
5000이 더해진 결과물이
새롭게 출력이 되었습니다.



2. SELECT 에서 사칙연산하기

문제)

상품 가격의 VAT (10%) 를 적용한 가격을 출력해야 합니다.

기존 상품 가격의 10% 의 값을 표현해 부가가치세 라는 명칭을 AS 로 부여하고 ,

기존 상품 가격 + (기존 상품 가격의 10%) 의 값을 이용해서 판매가격

이라는 이름으로 부여하세요.

(오른쪽 테이블처럼 결과가 나와야 합니다)

◆ PRD_ID	◆ PRD_NAME	◆ PRD_PRICE	◆ 부가가치세	◆ 판매가격
P0001	헤어드라이기	30000	3000	33000
P0002	에어컨	1500000	150000	1650000
P0003	세탁기	600000	60000	660000
P0004	건조기	800000	80000	880000
P0005	노트북	1500000	150000	1650000
P0006	데스크탑	2000000	200000	2200000
P0007	태블릿	800000	80000	880000
P0008	애플14	1200000	120000	1320000
P0009	갤럭시s23	1500000	150000	1650000
P0010	조아샴푸	20000	2000	22000
P0011	주전자	20000	2000	22000
P0012	전기밥솥	80000	8000	88000
P0013	냄비	30000	3000	33000
P0014	칼	15000	1500	16500
P0015	수세미	5000	500	5500
P0017	곰팡이제거제	10000	1000	11000
P0018	샤워기	50000	5000	55000
P0019	런스	20000	2000	22000
P0020	수건	5000	500	5500

2. SELECT 에서 사칙연산하기

문제)

상품 가격의 VAT (10%) 를 적용한 가격을 출력해야 합니다.

기존 상품 가격의 10% 의 값을 표현해 부가가치세 라는 명칭을 AS 로 부여하고 ,

기존 상품 가격 + (기존 상품 가격의 10%) 의 값을 이용해서 판매가격

이라는 이름으로 부여하세요.

(오른쪽 테이블처럼 결과가 나와야 합니다)

정답

```
SELECT PRD_ID  
      , PRD_NAME  
      , PRD_PRICE  
      , PRD_PRICE * 0.1 AS 부가가치세  
      , PRD_PRICE + ( PRD_PRICE * 0.1) AS 판매가격  
FROM TB_PRD ;
```

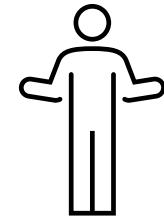
◆ PRD_ID	◆ PRD_NAME	◆ PRD_PRICE	◆ 부가가치세	◆ 판매가격
P0001	헤어드라이기	30000	3000	33000
P0002	에어컨	1500000	150000	1650000
P0003	세탁기	600000	60000	660000
P0004	건조기	800000	80000	880000
P0005	노트북	1500000	150000	1650000
P0006	데스크탑	2000000	200000	2200000
P0007	태블릿	800000	80000	880000
P0008	애플14	1200000	120000	1320000
P0009	갤럭시s23	1500000	150000	1650000
P0010	조아샴푸	20000	2000	22000
P0011	주전자	20000	2000	22000
P0012	전기밥솥	80000	8000	88000
P0013	냄비	30000	3000	33000
P0014	칼	15000	1500	16500
P0015	수세미	5000	500	5500
P0017	곰팡이제거제	10000	1000	11000
P0018	샤워기	50000	5000	55000
P0019	런스	20000	2000	22000
P0020	수건	5000	500	5500

3. SELECT 에서 연결(||)연산하기

```
SELECT GRADE_NAME || ' (등급레벨:' || GRADE_CD || ') ' AS 등급명과레벨  
FROM TB_GRADE ;
```

◆ 등급명과레벨
브론즈 (등급레벨: 1)
실버 (등급레벨: 2)
골드 (등급레벨: 3)
VIP (등급레벨: 4)
VVIP (등급레벨: 5)

연결 연산(||) 을 이용하면
각 튜플을 기준으로
컬럼과 리터럴을 조합해 연결할 수 있습니다.



3. SELECT 에서 연결(||)연산하기

문제)

아래 결과와 같이 데이터를 출력해보세요. (연결 연산 활용)

힌트 : TB_MEMBER 테이블의 MEMBER_ID 와 PAY_CARD_NO 컬럼 활용

회원의카드번호	
AAAAAA	회원의 카드번호는 1111-1111-1111-1111 입니다
BBBBBB	회원의 카드번호는 2222-2222-2222-2222 입니다
CCCCCC	회원의 카드번호는 3333-3333-3333-3333 입니다
DDDDDD	회원의 카드번호는 4444-4444-4444-4444 입니다
EEEEEE	회원의 카드번호는 5555-5555-5555-5555 입니다
FFFFFF	회원의 카드번호는 6666-6666-6666-6666 입니다
GGGGGG	회원의 카드번호는 7777-7777-7777-7777 입니다
HHHHHH	회원의 카드번호는 8888-8888-8888-8888 입니다
IIIIII	회원의 카드번호는 9999-9999-9999-9999 입니다
JJJJJJ	회원의 카드번호는 AAAA-AAAA-AAAA-AAAA 입니다

3. SELECT 에서 연결(||)연산하기

답)

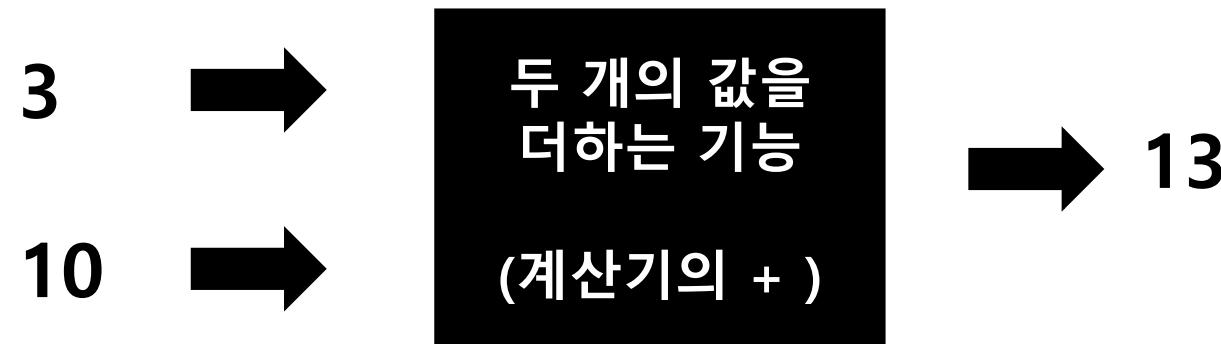
```
SELECT MEMBER_ID || ' 회원의 카드번호는 ' || PAY_CARD_NO || ' 입니다' AS 회원의카드번호  
FROM TB_MEMBER ;
```

회원의카드번호	
AAAAA	회원의 카드번호는 1111-1111-1111-1111 입니다
BBBBB	회원의 카드번호는 2222-2222-2222-2222 입니다
CCCCC	회원의 카드번호는 3333-3333-3333-3333 입니다
DDDDD	회원의 카드번호는 4444-4444-4444-4444 입니다
EEEEE	회원의 카드번호는 5555-5555-5555-5555 입니다
FFFFF	회원의 카드번호는 6666-6666-6666-6666 입니다
GGGGG	회원의 카드번호는 7777-7777-7777-7777 입니다
HHHHH	회원의 카드번호는 8888-8888-8888-8888 입니다
IIIII	회원의 카드번호는 9999-9999-9999-9999 입니다
JJJJJ	회원의 카드번호는 AAAA-AAAA-AAAA-AAAA 입니다

4. 내장형함수 & 형변환함수 사용하기

함수에 대해 알아봅시다.

함수는 입력 값을 넣어 특정한 기능을 통해 결과 값을 출력해줍니다.

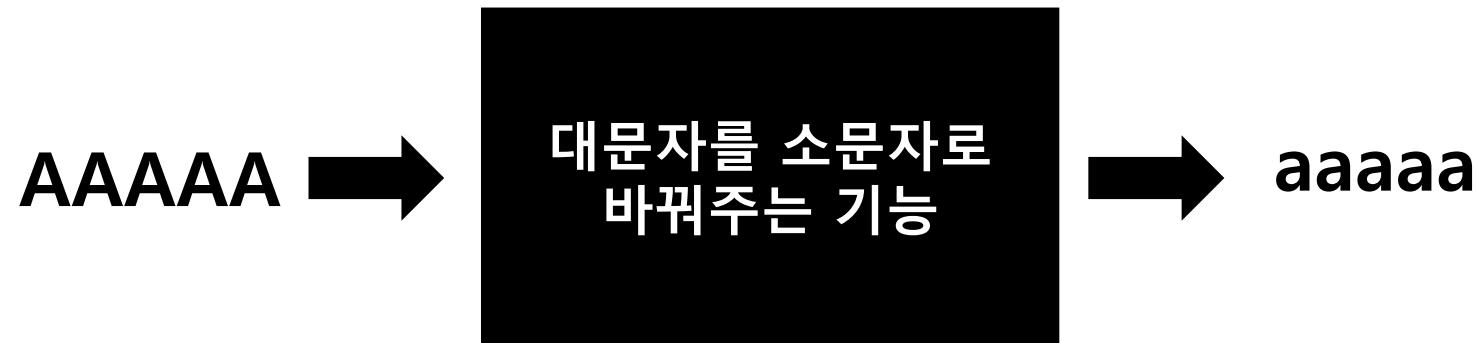


계산기의 **더하기기능은 두 개의 숫자를 입력받아 더한 결과를 출력합니다.**

4. 내장형함수 & 형변환함수 사용하기

함수에 대해 알아봅시다.

함수는 입력 값을 넣어 특정한 기능을 통해 결과 값을 출력해줍니다.

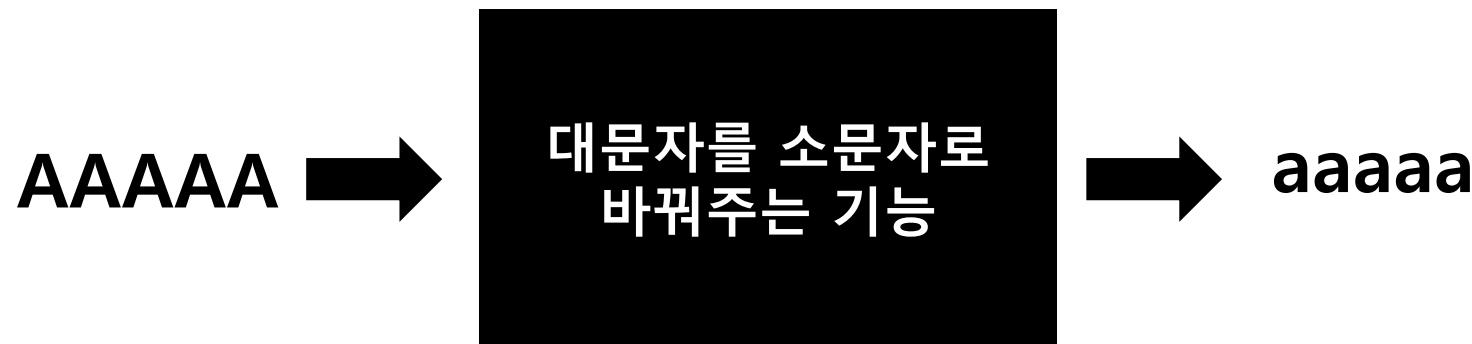


대문자를 소문자로 바꿔주는 기능은 대문자를 입력받아 소문자를 출력합니다.

4. 내장형함수 & 형변환함수 사용하기

함수의 특징

1. 블랙 박스가 어떻게(how) 만들어져 있는지는 알 필요 없습니다.
2. 입력 개수와 출력 개수는 함수를 만든 사람의 마음입니다.
3. 원하는 함수를 사용하고 싶다면 입력 개수, 출력 개수, 기능(what)만 알면 됩니다



4. 내장형함수 & 형변환함수 사용하기

내장형(built-in)함수란 미리 만들어 놓은 함수입니다.

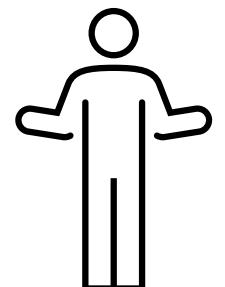
(우리는 사용만 하면 됩니다. ㅋ꿀)

문자형 함수 : LOWER , SUBSTR , TRIM , REPLACE

숫자형 함수 : MOD , ROUND

날짜형 함수 : SYSDATE

전부 외울 필요 없습니다.
필요할 때마다 인터넷에
구글링 해보세요
ex) 오라클 내장형 함수



4. 내장형함수 & 형변환함수 사용하기

문자형 함수 : **LOWER** , **SUBSTR** , **TRIM** , **REPLACE**

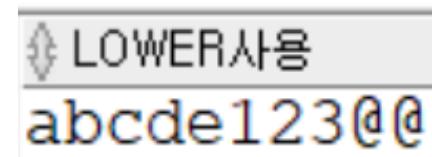
입력 값 : 문자형 리터럴 1개 혹은 문자형 컬럼

출력 값 : 문자형 리터럴 1개

기능 : 입력받은 문자열에서 대문자를 소문자로 바꿔준다

```
SELECT LOWER('ABCDE123@@') AS LOWER사용
```

```
FROM DUAL ; --dual은 테스트 할 때 쓰는 테이블
```



```
SELECT MEMBER_ID , PASSWD , LOWER(PASSWD) AS 패스워드소문자
```

```
FROM TB_MEMBER ;
```

MEMBER_ID	PASSWD	패스워드소문자
AAAAAA	AAAAAA	aaaaaa
BBBBBB	BBBBBB	bbbbbb
CCCCCC	CCCCCC	cccccc
DDDDDD	DDDDDD	dddddd
EEEEEE	EEEEEE	eeeeee
FFFFFF	FFFFFF	ffffff
GGGGGG	GGGGGG	gggggg
HHHHHH	HHHHHH	hhhhh
IIIIII	IIIIII	iiiii
JJJJJJ	JJJJJJ	jjjjjj

4. 내장형함수 & 형변환함수 사용하기

문자형 함수 : LOWER , SUBSTR , TRIM , REPLACE

입력 값 : 문자형 리터럴 1개 , 시작위치 , 길이

출력 값 : 문자형 리터럴 1개

기능 : 입력받은 문자형 리터럴에서 시작위치에서 길이만큼 잘라낸다

```
SELECT PRD_ID  
      , PRD_INFO  
      , SUBSTR(PRD_INFO , 1 , 5) || '...' AS 상품설명생략  
  FROM TB_PRD  
 WHERE PRD_TYPE = '가전';
```

PRD_ID	PRD_INFO	상품설명생략
P0001	머리를 아주 잘 말려주는 헤어드라이기 입니다.	머리를 아...
P0002	더위를 싹 싹 날려주는 에어컨입니다.	더위를 싹...
P0003	옷을 깨끗하게 세탁해주는 세탁기입니다.	옷을 깨끗...
P0004	빠르고 간편하게 옷을 건조할 수 있는 건조기입니다.	빠르고 간...

4. 내장형함수 & 형변환함수 사용하기

문자형 함수 : LOWER , SUBSTR , TRIM , REPLACE

입력 값 : 문자형 리터럴 1개 (기본적으로 쓰는 방식 기준),

출력 값 : 문자형 리터럴 1개

기능 : 입력받은 문자형 리터럴의 양 끝의 공백을 제거한다. (단, 문자 중간의 공백은 제거 X)

```
SELECT TRIM(' 안녕하세요 ') , TRIM( '안 넣 하 세 요 ')  
FROM DUAL ;
```

TRIM('안녕하세요')	TRIM('안녕하세요')_1
안녕하세요	안 넣 하 세 요

4. 내장형함수 & 형변환함수 사용하기

문자형 함수 : LOWER , SUBSTR , TRIM , REPLACE

입력 값 : 문자형 리터럴 1개 , 바뀔값, 바꿀값

출력 값 : 문자형 리터럴 1개

기능 : 입력받은 문자형 리터럴 안에있는 바뀔값을 바꿀값으로 변경하여 출력한다.

```
SELECT MEMBER_ID  
      , PAY_CARD_NO  
      , REPLACE(PAY_CARD_NO , '-' , '') AS 결제카드번호특수제외  
  FROM TB_MEMBER  
 WHERE GRADE_CD = 2;
```

MEMBER_ID	PAY_CARD_NO	결제카드번호특수제외
BBBBB	2222-2222-2222-2222	2222222222222222
GGGGG	7777-7777-7777-7777	7777777777777777

문자형 함수 문제를 풀어봅시다

1. 아래 빈 문항과 ?(물음표)에 알맞은 값을 넣어 smhrd라는 출력값만 나오도록 해주세요.

```
SELECT [REDACTED] ('https://smhrd.or.kr/' , ? , ? )  
FROM DUAL ;
```

결과
smhrd

2. TB_MEMBER 테이블을 이용해서 다음과 같이 데이터를 출력해주세요.
이때 JOIN_DY 컬럼을 활용해서 앞자리 4자리 년도만 출력하도록 해주세요.

MEMBER_ID	MEMBER_NAME	JOIN_DY	가입년도
AAAAAA	사용자A	20200101	2020
BBBBBB	사용자B	20200327	2020
CCCCCC	사용자C	20210105	2021
DDDDDD	사용자D	20210630	2021
EEEEEE	사용자E	20210831	2021
FFFFFF	사용자F	20220216	2022
GGGGGG	사용자G	20220317	2022
HHHHHH	사용자H	20220812	2022
IIIIII	사용자I	20230430	2023
JJJJJJ	새로운자	20230519	2023

3. TB_MEMBER 테이블을 이용해 다음과 같이 데이터를 출력하되,
PAY_CARD_NO 컬럼을 이용해 구분자를 '-'에서 '/'로 바꾸어
구분자변경이라는 명칭으로 출력해주세요.

MEMBER_ID	PAY_CARD_NO	구분자변경
AAAAAA	1111-1111-1111-1111	1111/1111/1111/1111
BBBBBB	2222-2222-2222-2222	2222/2222/2222/2222
CCCCCC	3333-3333-3333-3333	3333/3333/3333/3333
DDDDDD	4444-4444-4444-4444	4444/4444/4444/4444
EEEEEE	5555-5555-5555-5555	5555/5555/5555/5555
FFFFFF	6666-6666-6666-6666	6666/6666/6666/6666
GGGGGG	7777-7777-7777-7777	7777/7777/7777/7777
HHHHHH	8888-8888-8888-8888	8888/8888/8888/8888
IIIIII	9999-9999-9999-9999	9999/9999/9999/9999
JJJJJJ	AAAA-AAAA-AAAA-AAAA	AAAA/AAAA/AAAA/AAAA

문자형 함수 문제를 풀어봅시다(답)

1. 아래 빈 문항과 ?(물음표)에 알맞은 값을 넣어 smhrd라는 출력값만 나오도록 해주세요.

```
SELECT SUBSTR('https://smhrd.or.kr/' , 9 , 5 ) AS 결과  
FROM DUAL ;
```

결과
smhrd

2. TB_MEMBER 테이블을 이용해서 다음과 같이 데이터를 출력해주세요.
이때 JOIN_DY 컬럼을 활용해서 앞자리 4자리 년도만 출력하도록 해주세요.

```
SELECT MEMBER_ID  
      , MEMBER_NAME  
      , JOIN_DY  
      , SUBSTR(JOIN_DY , 1, 4 ) AS 가입년도  
FROM TB_MEMBER ;
```

MEMBER_ID	MEMBER_NAME	JOIN_DY	가입년도
AAAAAA	사용자A	20200101	2020
BBBBBB	사용자B	20200327	2020
CCCCCC	사용자C	20210105	2021
DDDDDD	사용자D	20210630	2021
EEEEEE	사용자E	20210831	2021
FFFFFF	사용자F	20220216	2022
GGGGGG	사용자G	20220317	2022
HHHHHH	사용자H	20220812	2022
IIIIII	사용자I	20230430	2023
JJJJJJ	새로운자	20230519	2023

3. TB_MEMBER 테이블을 이용해 다음과 같이 데이터를 출력하되,
PAY_CARD_NO 컬럼을 이용해 구분자를 '-'에서 '/'로 바꾸어
구분자변경이라는 명칭으로 출력해주세요.

```
SELECT MEMBER_ID  
      , PAY_CARD_NO  
      , REPLACE(PAY_CARD_NO , '-' , '/') AS 구분자변경  
FROM TB_MEMBER ;
```

MEMBER_ID	PAY_CARD_NO	구분자변경
AAAAAA	1111-1111-1111-1111	1111/1111/1111/1111
BBBBBB	2222-2222-2222-2222	2222/2222/2222/2222
CCCCCC	3333-3333-3333-3333	3333/3333/3333/3333
DDDDDD	4444-4444-4444-4444	4444/4444/4444/4444
EEEEEE	5555-5555-5555-5555	5555/5555/5555/5555
FFFFFF	6666-6666-6666-6666	6666/6666/6666/6666
GGGGGG	7777-7777-7777-7777	7777/7777/7777/7777
HHHHHH	8888-8888-8888-8888	8888/8888/8888/8888
IIIIII	9999-9999-9999-9999	9999/9999/9999/9999
JJJJJJ	AAAA-AAAA-AAAA-AAAA	AAAA/AAAA/AAAA/AAAA

4. 내장형함수 & 형변환함수 사용하기

숫자형 함수 : MOD , ROUND

입력 값 : 숫자형 리터럴 2개 (피제수 , 제수) -- 예시) 피제수 : 10 , 제수 : 3

출력 값 : 숫자형 리터럴 1개 -- 예시) $10 \% 3 = 1$

기능 : 피제수를 제수로 나눈 나머지 숫자형 리터럴을 출력한다.

```
SELECT MOD(100, 8) AS 나머지값  
      , MOD(30, 10) AS 나머지값2  
FROM DUAL ;
```

나머지값	나머지값2
4	0

4. 내장형함수 & 형변환함수 사용하기

숫자형 함수 : MOD , ROUND

입력 값 : 숫자형 리터럴 2개 (실수 , 소수점자릿수)

출력 값 : 숫자형 리터럴 값

기능 : 실수를 소수점 자릿수까지 반올림한 결과를 출력한다.

```
SELECT ROUND(1.452, 2), ROUND(1.452, 1)  
FROM DUAL
```

ROUND(1.452,2)	ROUND(1.452,1)
1.45	1.5

4. 내장형함수 & 형변환함수 사용하기

날짜형 함수 : SYSDATE

SYSDATE 는 입력 시 바로 출력이 되며, 현재 시간을 초단위까지 날짜형 리터럴로 출력합니다.

```
SELECT SYSDATE  
FROM DUAL ;
```

SYSDATE
2023-05-06 23:25:18

4. 내장형함수 & 형변환함수 사용하기

날짜형 함수 : SYSDATE

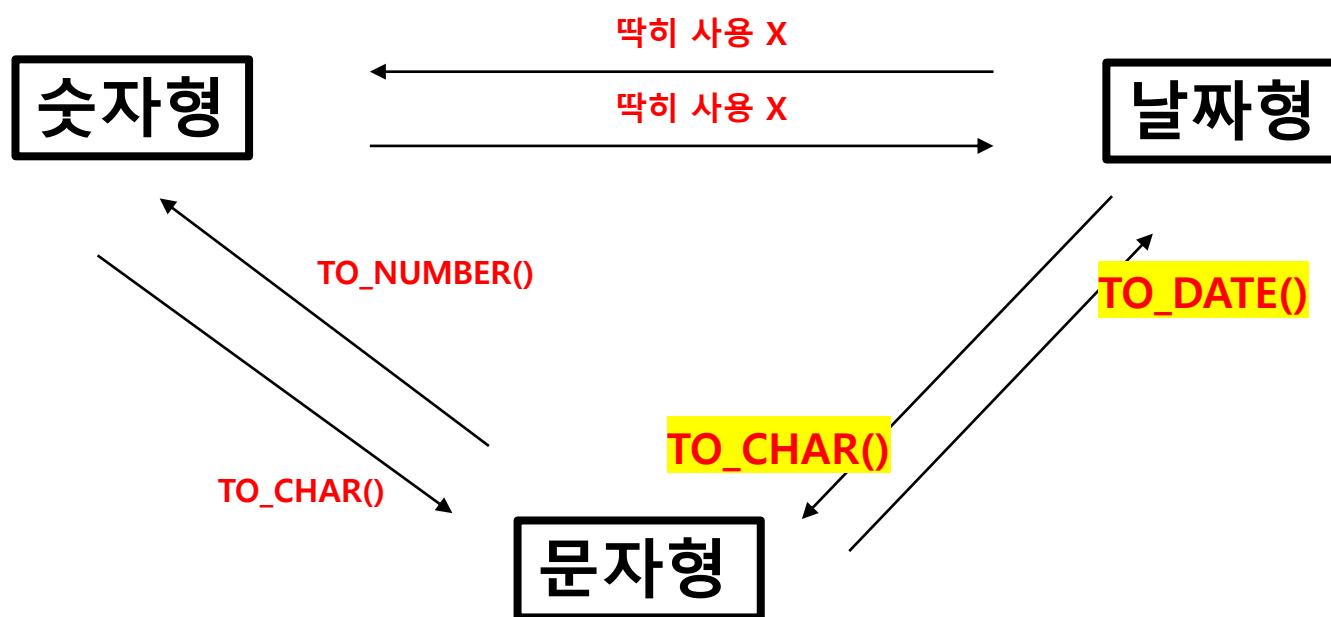
실무에서는 현재 시간을 입력하거나, 아래와 같이 날짜를 조작해 사용하기도 합니다.

```
SELECT SYSDATE + 1 AS 하루더함  
      , SYSDATE + 1/24 AS 한시간더함  
      , SYSDATE + 1/24/60 AS 일분더함  
      , SYSDATE + 1/24/60/60 AS 일초더함  
FROM DUAL ;
```

하루더함	한시간더함	일분더함	일초더함
2023-05-07 23:26:13	2023-05-07 00:26:13	2023-05-06 23:27:13	2023-05-06 23:26:14

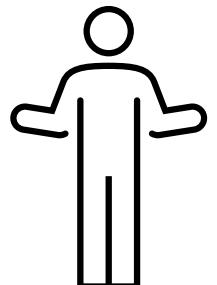
4. 내장형함수 & 형변환함수 사용하기

형변환함수는 특정 자료형으로 값을 변경합니다.



실무 팁!

날짜형 -> 문자형 변환과
문자형 -> 날짜형 변환은
자주 사용하므로 익숙해지기를
추천합니다.



다음을 실행해봅시다.

1. SELECT TO_NUMBER('1') FROM DUAL ; -- 문자형('1')을 숫자형(1)로 형변환해 출력
2. SELECT TO_CHAR(1) FROM DUAL ; -- 숫자형(1)을 문자형('1')으로 형변환해 출력
3. SELECT TO_CHAR(SYSDATE , 'YYYY/MM/DD HH24:MI:SS') FROM DUAL ;
4. SELECT TO_CHAR(SYSDATE , 'YYYYMMDD') FROM DUAL ;
5. SELECT TO_DATE('20230101' , 'YYYY/MM/DD') FROM DUAL ;
6. SELECT TO_DATE('20230101141212' , 'YYYY/MM/DD HH24:MI:SS') FROM DUAL;

/*

3번 문제는 날짜형(SYSDATE)을 문자형으로 변경합니다. **(포맷의미는 꼭 외웁시다!)**

5번 문제는 문자형('2023010114')을 날짜형 (2023/01/01 14:00:00)으로 변경합니다.

6번 문제는 문자형('20230101141212')을 날짜형(2023/01/01 14:12:12)으로 변경합니다.

*/

실습문제

1. TB_PRD (상품) 테이블에는 해당 상품을 등록한 시점인 REG_DATE (등록일시) 컬럼이 존재합니다. 아래와 같이 데이터를 출력해주세요. (TO_CHAR 활용)

* REG_DATE 컬럼은 날짜형이며, 출력된 등록년월일 컬럼은 문자형임

PRD_NAME	PRD_PRICE	SELL_COMP_NAME	등록년월일
헤어드라이기	30000	LG	20230506
에어컨	1500000	삼성	20230506
세탁기	600000	LG	20230506
건조기	800000	삼성	20230506
노트북	1500000	Apple	20230506
데스크탑	2000000	Dell	20230506
태블릿	800000	삼성	20230506
애플14	1200000	Apple	20230506
갤럭시s23	1500000	삼성	20230506
조아샴푸	20000	LG	20230506
주전자	20000	LG	20230506
전기밥솥	80000	삼성	20230506
냄비	30000	매일유업	20230506
칼	15000	매일유업	20230506
수세미	5000	3M	20230506
곰팡이제거제	10000	3M	20230506
샤워기	50000	다우니	20230506
린스	20000	다우니	20230506
수건	5000	다우니	20230506

실습문제 답

1. TB_PRD (상품) 테이블에는 해당 상품을 등록한 시점인 REG_DATE (등록일시) 컬럼이 존재합니다. 아래와 같이 데이터를 출력해주세요. (TO_CHAR 활용)

* REG_DATE 컬럼은 날짜형이며, 출력된 등록년월일 컬럼은 문자형임

```
SELECT PRD_NAME  
      , PRD_PRICE  
      , SELL_COMP_NAME  
      , TO_CHAR(REG_DATE , 'YYYYMMDD') AS 등록년월일  
FROM TB_PRD ;
```

PRD_NAME	PRD_PRICE	SELL_COMP_NAME	등록년월일
헤어드라이기	30000	LG	20230506
에어컨	1500000	삼성	20230506
세탁기	600000	LG	20230506
건조기	800000	삼성	20230506
노트북	1500000	Apple	20230506
데스크탑	2000000	Dell	20230506
태블릿	800000	삼성	20230506
애플14	1200000	Apple	20230506
갤럭시S23	1500000	삼성	20230506
조아샴푸	20000	LG	20230506
주전자	20000	LG	20230506
전기밥솥	80000	삼성	20230506
냄비	30000	매일유업	20230506
칼	15000	매일유업	20230506
수세미	5000	3M	20230506
곰팡이제거제	10000	3M	20230506
샤워기	50000	다우니	20230506
린스	20000	다우니	20230506
수건	5000	다우니	20230506

아래 쿼리는 실행이 될까요?

```
SELECT MEMBER_NAME + 1000 FROM TB_MEMBER;
```

(MEMBER_NAME은 문자형 컬럼이고 1000은 숫자형 값이라서 서로 데이터의 형이 다른 상황)

4. 내장형함수 & 형변환함수 사용하기

연산을 하기 전에 먼저 **형변환이 발생합니다.**

형 변환 우선순위 : 날짜형 > 숫자형 > 문자형

```
SELECT MEMBER_NAME + 1000 FROM TB_MEMBER;
```

문자형



숫자형



4. 내장형함수 & 형변환함수 사용하기

연산을 하기 전에 먼저 **형변환이 발생합니다.**

형변환 시 자동적으로 형변환 함수를 적용하게 됩니다.

```
SELECT TO_NUMBER(MEMBER_NAME) + 1000 FROM TB_MEMBER ;
```

문자형 -> 숫자형 형변환 시도

MEMBER_NAME
사용자A
사용자B
사용자C
사용자D
사용자E
사용자F
사용자G
사용자H
사용자I
새로운자

TO_NUMBER('사용자A')

'사용자A' 는 숫자로 변환이
불가능 합니다!

숫자형

ORA-01722: 수치가 부적합합니다
01722, 00000 - "invalid number"

ORA-01722: invalid number
01722, 00000 - "invalid number"

4. 내장형함수 & 형변환함수 사용하기

연산을 하기 전에 먼저 **형변환이 발생합니다.**

형변환 시 자동적으로 형변환 함수를 적용하게 됩니다.

```
SELECT '1' + 1 FROM DUAL ;
```

문자형 -> 숫자형 형변환 시도

```
TO_NUMBER('1')
```

'1' 은 숫자로 변환 가능합니다!

문제를 풀어봅시다

1. 2023년이 되었는데 TB_MEMBER 테이블의 AGE 정보가 수정되지 않았습니다.
현재 AGE에서 +1 을 하여 아래와 같이 정보를 출력해주세요.

MEMBER_ID	AGE	출해나이
AAAAAA	(null)	(null)
BBBBBB	25	26
CCCCCC	27	28
DDDDDD	30	31
EEEEEE	(null)	(null)
FFFFFF	35	36
GGGGGG	39	40
HHHHHH	44	45
IIIIII	52	53
JJJJJJ	30	31

답)

1. 2023년이 되었는데 TB_MEMBER 테이블의 AGE 정보가 수정되지 않았습니다.
현재 AGE에서 +1 을 하여 아래와 같이 정보를 출력해주세요.

```
SELECT MEMBER_ID  
      , AGE  
      , AGE + 1 AS 올해나이  
FROM TB_MEMBER ;
```

MEMBER_ID	AGE	올해나이
AAAAAA	(null)	(null)
BBBBBB	25	26
CCCCCC	27	28
DDDDDD	30	31
EEEEEE	(null)	(null)
FFFFFF	35	36
GGGGGG	39	40
HHHHHH	44	45
IIIIII	52	53
JJJJJJ	30	31

5. NULL 함수 사용하기

NULL은 산술,비교 연산이 불가능합니다.

◆ MEMBER_ID	◆ AGE	◆ 촐해나이
AAAAAA	(null) (null)	
BBBBBB	25	26
CCCCCC	27	28
DDDDDD	30	31
EEEEEE	(null) (null)	
FFFFFF	35	36
GGGGGG	39	40
HHHHHH	44	45
IIIIII	52	53
JJJJJJ	30	31



NULL 인 데이터에 1을 더했더니 NULL이 나왔습니다.
(NULL은 정해지지 않은 값 , 숫자 0과는 완전 다름!)

5. NULL 함수 사용하기

NULL함수란 NULL값을 대체할 수 있는 함수입니다.

NVL(data1 , data2)

NVL2(data1 , data2 , data3)

DECODE(data1 , data2 , data3 , data4)

COALESCE (data1 , data2 , data3 , , dataN)

실무 팁!

NULL 함수 중
NVL 와 **DECODE** 는
정말 자주 사용됩니다.

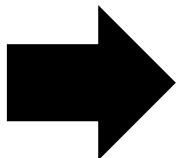


5. NULL 함수 사용하기

NVL(data1 , data2)

data1에 NULL 값이 들어오면 data2를 출력하고 NULL이 아니면 data1을 출력합니다.

```
SELECT MEMBER_ID  
      , AGE  
      , NVL( AGE , 0 )  
FROM TB_MEMBER ;
```



MEMBER_ID	AGE	NVL(AGE,0)
AAAAAA	(null)	0
BBBBBB	25	25
CCCCCC	27	27
DDDDDD	30	30
EEEEEE	(null)	0
FFFFFF	35	35
GGGGGG	39	39
HHHHHH	44	44
IIIIII	52	52
JJJJJJ	30	30

5. NULL 함수 사용하기

DECODE(data1 , data2 , data3 , data4 ...)

data1 과 data2 가 동일하면 data3을 출력하고 , 그렇지 않으면 data4를 출력합니다.

```
SELECT MEMBER_NAME  
      , AGE  
      , DECODE ( AGE , NULL , 0 , AGE )  
FROM TB_MEMBER ;
```



MEMBER_NAME	AGE	DECODE(AGE,NULL,0,AGE)
사용자A	(null)	0
사용자B	25	25
사용자C	27	27
사용자D	30	30
사용자E	(null)	0
사용자F	35	35
사용자G	39	39
사용자H	44	44
사용자I	52	52
새로운자	30	30

5. NULL 함수 사용하기

DECODE(data1 , data2 , data3 , data4 ...)

data1 과 data2 가 동일하면 data3을 출력하고 , 그렇지 않으면 data4를 출력합니다.

```
SELECT MEMBER_NAME  
      , GRADE_CD  
      , DECODE ( GRADE_CD , 1 , '브론즈' , 2 , '실버' , 3 , '골드' , 4 , 'VIP' , 'VVIP' ) AS 등급  
FROM TB_MEMBER ;
```



MEMBER_NAME	GRADE_CD	등급
사용자A	1	브론즈
사용자B	2	실버
사용자C	1	브론즈
사용자D	3	골드
사용자E	1	브론즈
사용자F	3	골드
사용자G	2	실버
사용자H	5	VVIP
사용자I	4	VIP
새로운자	1	브론즈

문제를 풀어봅시다

1. TB_MEMBER 테이블에서 정보를 출력하되 AGE가 정해지지 않은 대상은 일단 20 으로 출력되도록 해주세요. (NVL 활용)

MEMBER_ID	PASSWD	AGE
AAAAAA	AAAAAA	20
BBBBBB	BBBBBB	25
CCCCCC	CCCCCC	27
DDDDDD	DDDDDD	30
EEEEEE	EEEEEE	20
FFFFFF	FFFFFF	35
GGGGGG	GGGGGG	39
HHHHHH	HHHHHH	44
IIIIII	IIIIII	52
JJJJJJ	JJJJJJ	30

2. TB_MEMBER 테이블에서 GENDER 의 값이 '남' 이면 'MAN' 를 출력, '여' 이면 'WOMAN' , 둘 다 아니면 'ELSE' 를 출력해주세요. (DECODE 활용)

MEMBER_ID	GENDER	성별
AAAAAA	남	MAN
BBBBBB	여	WOMAN
CCCCCC	남	MAN
DDDDDD	여	WOMAN
EEEEEE	남	MAN
FFFFFF	여	WOMAN
GGGGGG	남	MAN
HHHHHH	(null)	ELSE
IIIIII	(null)	ELSE
JJJJJJ	(null)	ELSE

답)

1. TB_MEMBER 테이블에서 정보를 출력하되 AGE가 정해지지 않은 대상은 일단 20 으로 출력되도록 해주세요. (NVL 활용)

```
SELECT MEMBER_ID  
      , PASSWD  
      , NVL(AGE, 20) AS AGE  
FROM TB_MEMBER ;
```

MEMBER_ID	PASSWD	AGE
AAAAAA	AAAAAA	20
BBBBBB	BBBBBB	25
CCCCCC	CCCCCC	27
DDDDDD	DDDDDD	30
EEEEEE	EEEEEE	20
FFFFFF	FFFFFF	35
GGGGGG	GGGGGG	39
HHHHHH	HHHHHH	44
IIIIII	IIIIII	52
JJJJJJ	JJJJJJ	30

2. TB_MEMBER 테이블에서 GENDER 의 값이 '남' 이면 'MAN' 를 출력, '여' 이면 'WOMAN' ,
둘 다 아니면 'ELSE' 를 출력해주세요. (DECODE 활용)

```
SELECT MEMBER_ID  
      , GENDER  
      , DECODE ( GENDER, '남', 'MAN', '여', 'WOMAN', 'ELSE' ) AS 성별  
FROM TB_MEMBER ;
```

MEMBER_ID	GENDER	성별
AAAAAA	남	MAN
BBBBBB	여	WOMAN
CCCCCC	남	MAN
DDDDDD	여	WOMAN
EEEEEE	남	MAN
FFFFFF	여	WOMAN
GGGGGG	남	MAN
HHHHHH	(null)	ELSE
IIIIII	(null)	ELSE
JJJJJJ	(null)	ELSE

SELECT END