KB금융그룹 | 국민의 평생 금융파트너

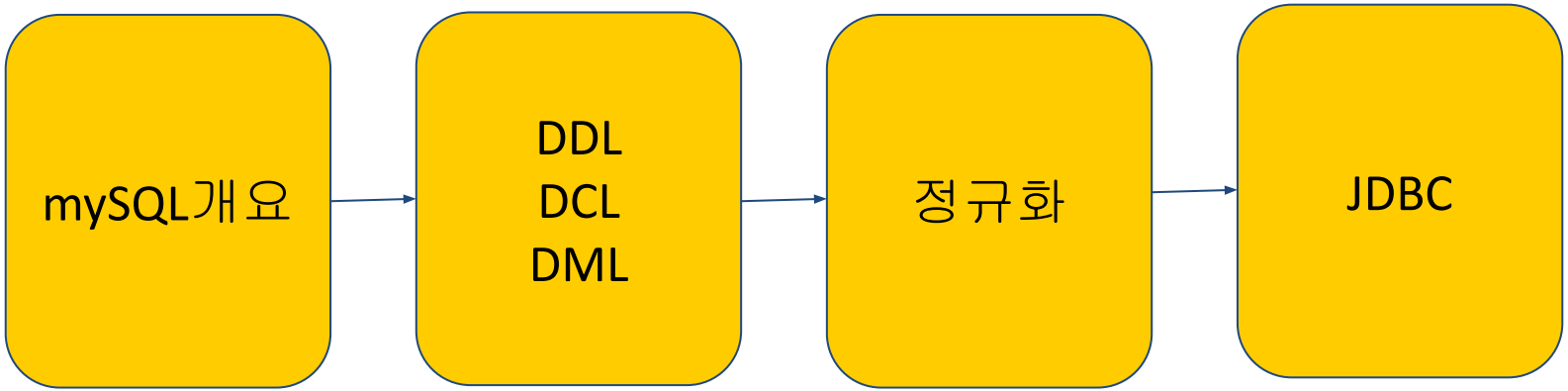
2024년 상반기 K-디지털 트레이닝

DML 기본

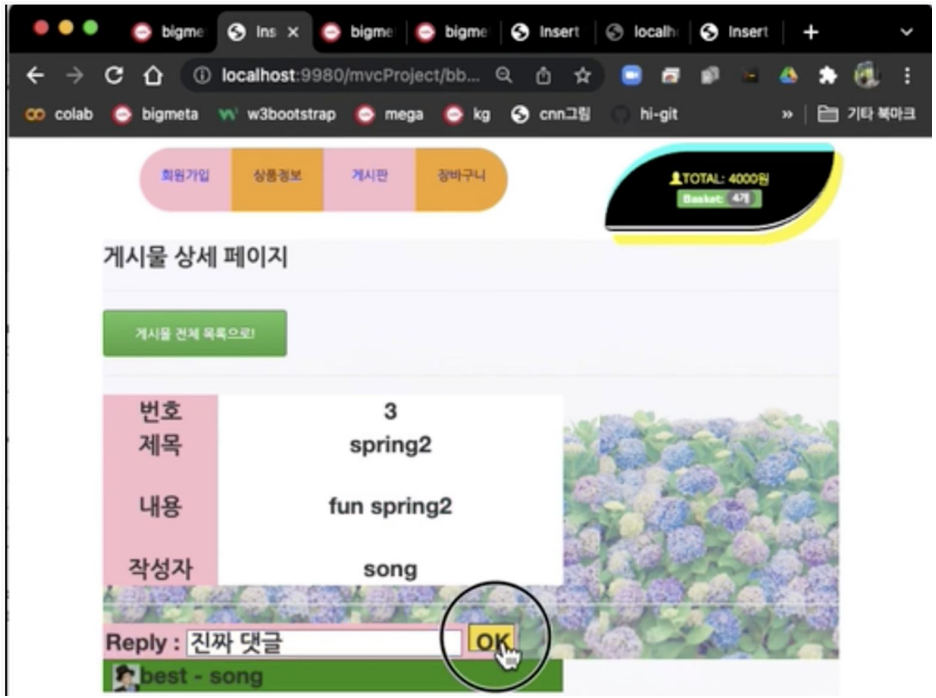
[KB] IT's Your Life

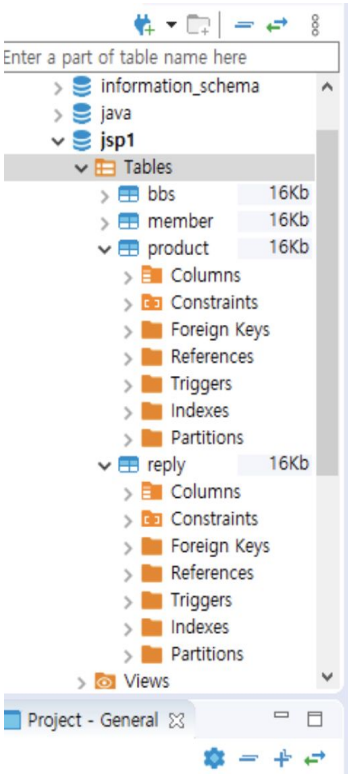
KB 국민은행

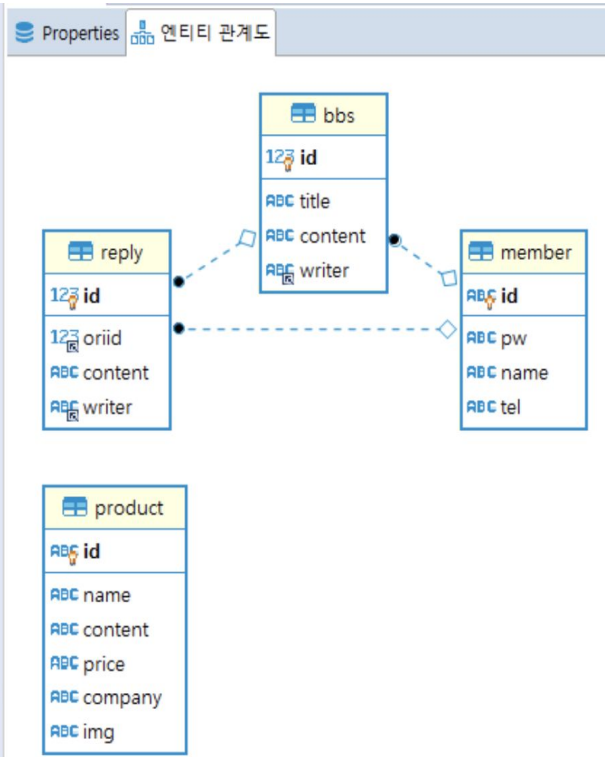
커리큘럼



학습 목표


localhost:9980/mvcProject/bb...
colab
bigmeta
w3bootstrap
mega
kg
cnn그림
hi-git
회원가입
상품정보
게시판
장바구니
TOTAL: 4000원
게시물 상세 페이지
게시물 전체 목록으로!
번호
3
제목
spring2
내용
fun spring2
작성자
song
Reply : 진짜 댓글
OK
best - song


Enter a part of table name here
information_schema
java
jsp1
Tables
bbs 16Kb
member 16Kb
product 16Kb
Columns
Constraints
Foreign Keys
References
Triggers
Indexes
Partitions
reply 16Kb
Columns
Constraints
Foreign Keys
References
Triggers
Indexes
Partitions
Views
Project - General


Properties
엔티티 관계도
bbs
id
title
content
writer
reply
id
oriid
content
writer
member
id
pw
name
tel
product
id
name
content
price
company
img

DML 개요

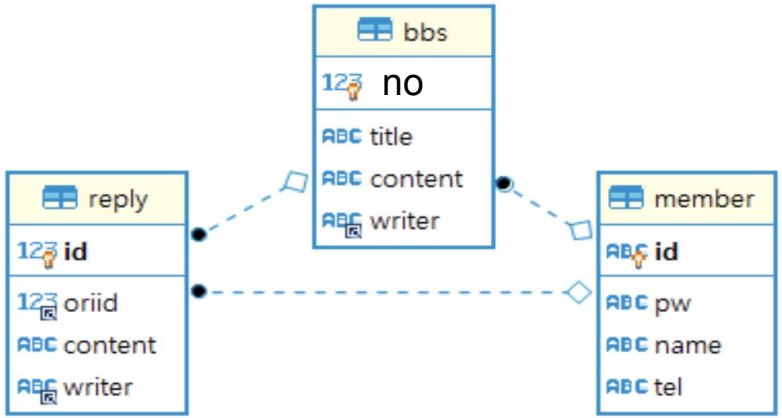
지금까지 학습내용

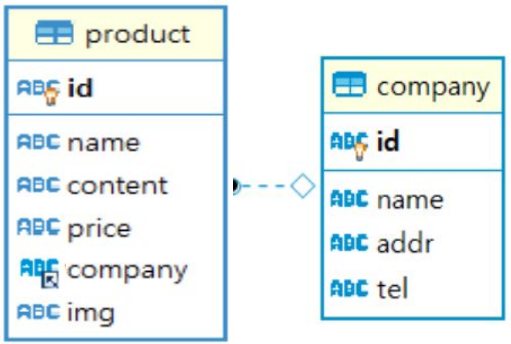
- DBMS
- 용어 중요!! entity, Schema, item, instance, domain, relationship
- Fk, Pk로 관계성 성립
- entity와 relation에 의한 결과는 table로 생성됨
- SQL
  - DDL, DML, DCL

SQL-명령어의 역할에 따른 분류

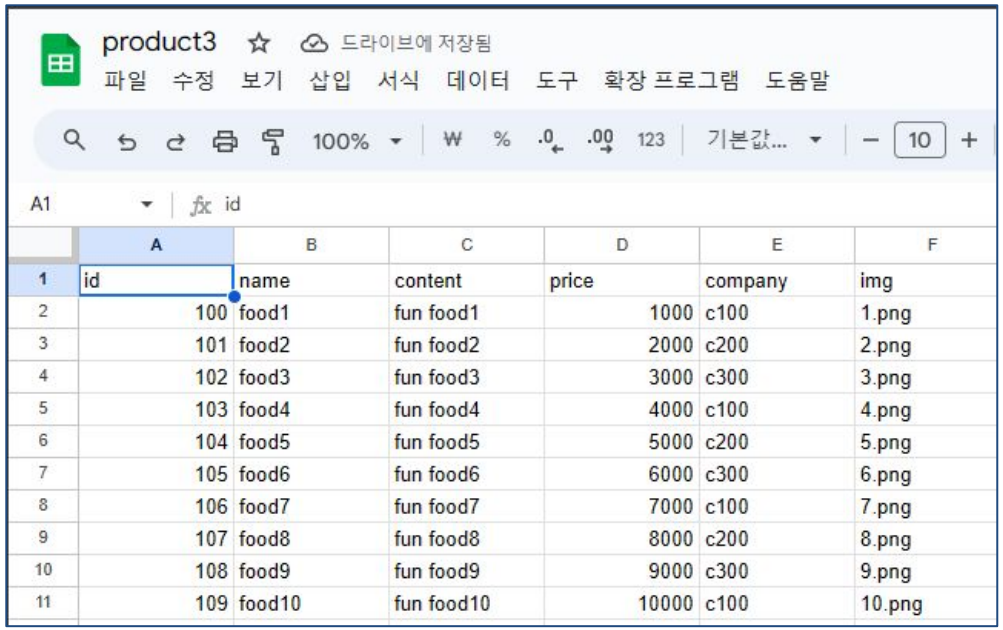
용어	Data Definition Language (DDL)	Data Manipulation Language (DML)	Data Control Language (DCL)	Transaction Control Language (TCL)
역할	데이터 항목 정의	데이터 조작	DBMS 제어 (계정승인, 권한부여/회수)	트랜잭션 제어
SQL 명령어	CREATE, ALTER, DROP, TRUNCATE	INSERT, SELECT, UPDATE, DELETE	GRANT, REVOKE	COMMIT, ROLLBACK

쇼핑몰 ERD

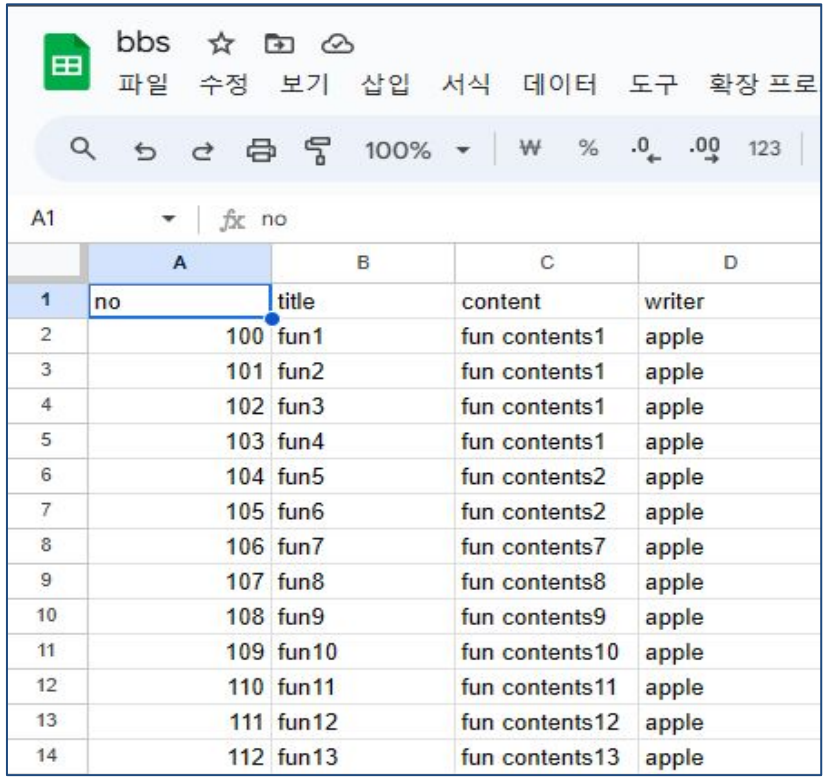

bbs
no
title
content
writer
reply
id
oriid
content
writer
member
id
pw
name
tel


product
id
name
content
price
company
img
company
id
name
addr
tel

csv-data(download link)



	A	B	C	D	E	F
1	id	name	content	price	company	img
2	100	food1	fun food1	1000	c100	1.png
3	101	food2	fun food2	2000	c200	2.png
4	102	food3	fun food3	3000	c300	3.png
5	103	food4	fun food4	4000	c100	4.png
6	104	food5	fun food5	5000	c200	5.png
7	105	food6	fun food6	6000	c300	6.png
8	106	food7	fun food7	7000	c100	7.png
9	107	food8	fun food8	8000	c200	8.png
10	108	food9	fun food9	9000	c300	9.png
11	109	food10	fun food10	10000	c100	10.png



	A	B	C	D
1	no	title	content	writer
2	100	fun1	fun contents1	apple
3	101	fun2	fun contents1	apple
4	102	fun3	fun contents1	apple
5	103	fun4	fun contents1	apple
6	104	fun5	fun contents2	apple
7	105	fun6	fun contents2	apple
8	106	fun7	fun contents7	apple
9	107	fun8	fun contents8	apple
10	108	fun9	fun contents9	apple
11	109	fun10	fun contents10	apple
12	110	fun11	fun contents11	apple
13	111	fun12	fun contents12	apple
14	112	fun13	fun contents13	apple

member table

- **FK**가 없는 독립적인 테이블을 먼저 생성

bbs shop - Schema bbs member2 bbs bbs - Table bbs - Table bbs - Table

Table Name: member

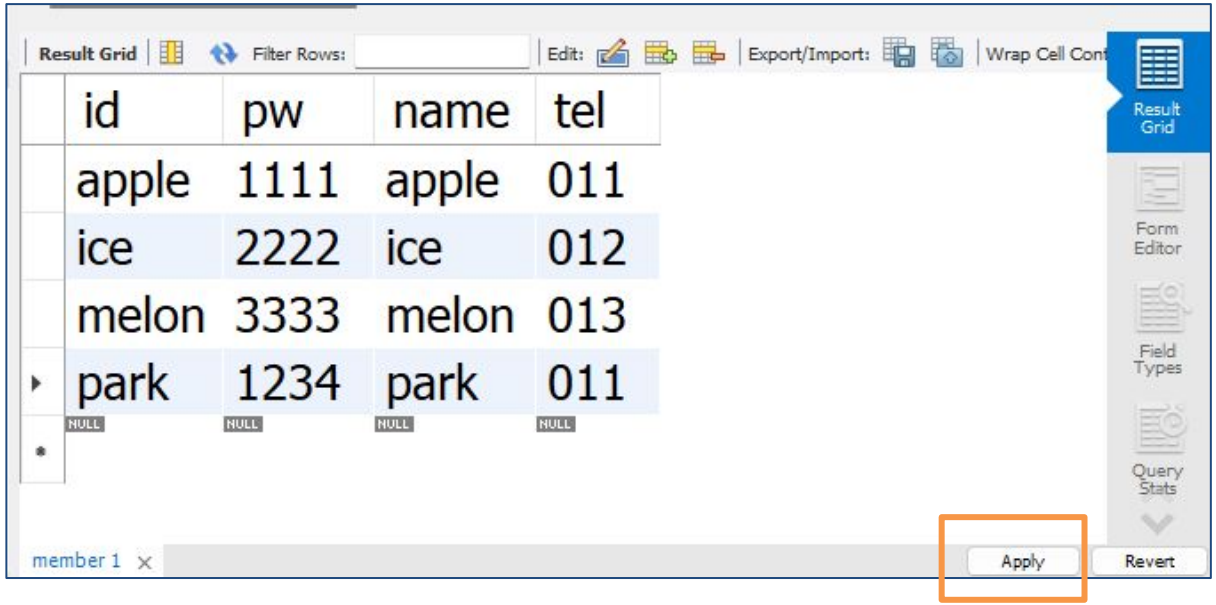
Charset/Collation: utf8mb4 utf8mb4_0900_ai_ci

Comments:

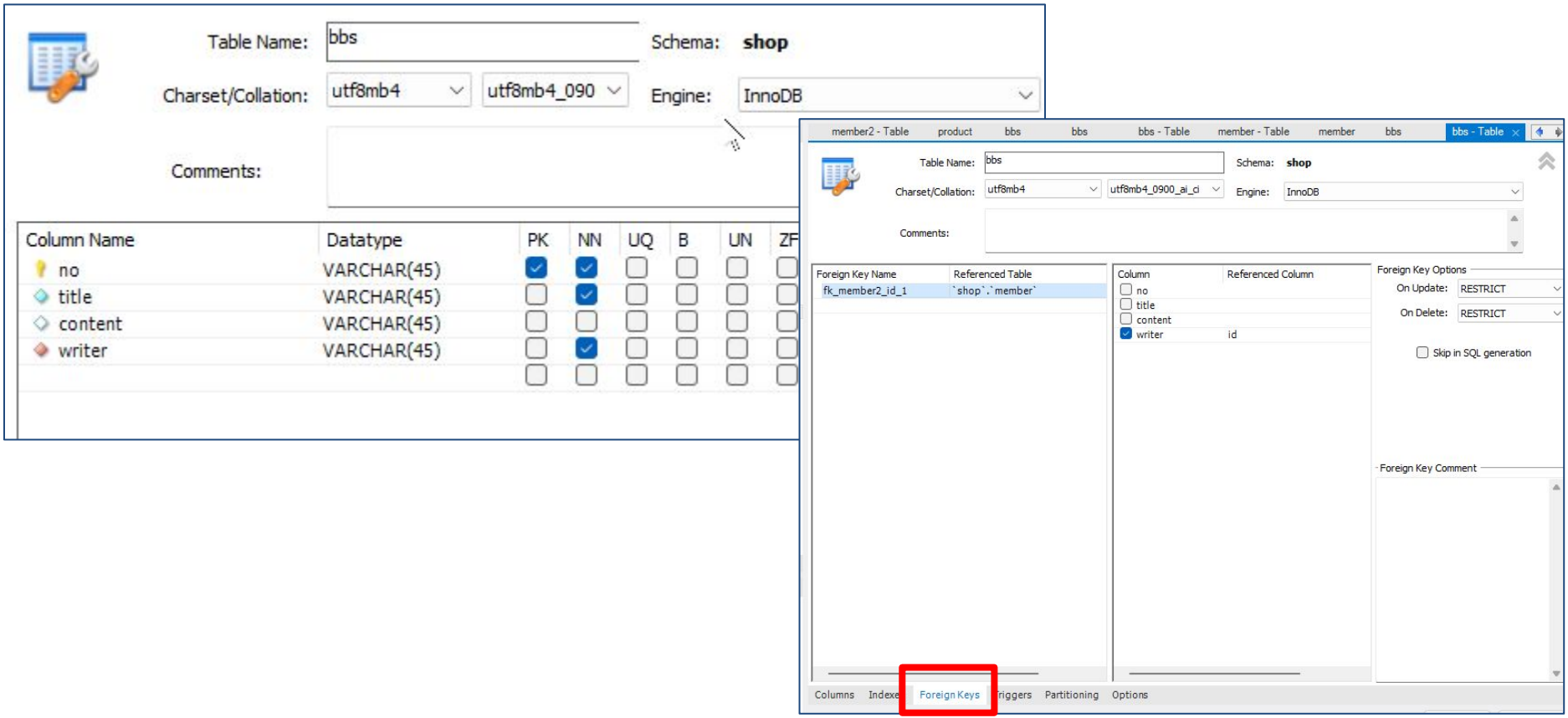
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	VARCHAR(45)	☑	☑	☐	☐	☐	☐	☐	☐	
pw	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
name	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
tel	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
		☐	☐	☐	☐	☐	☐	☐	☐	

member table

추가후, 반드시 **apply!**

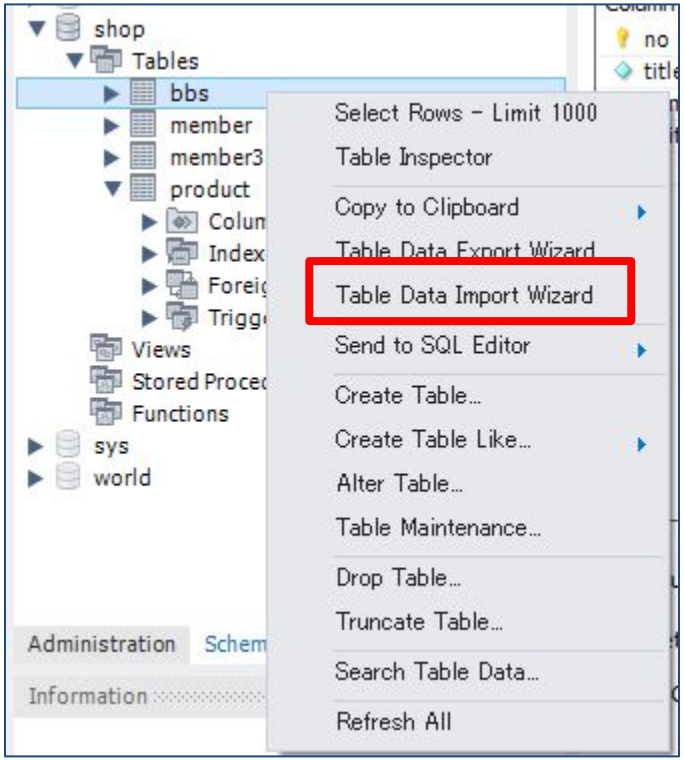


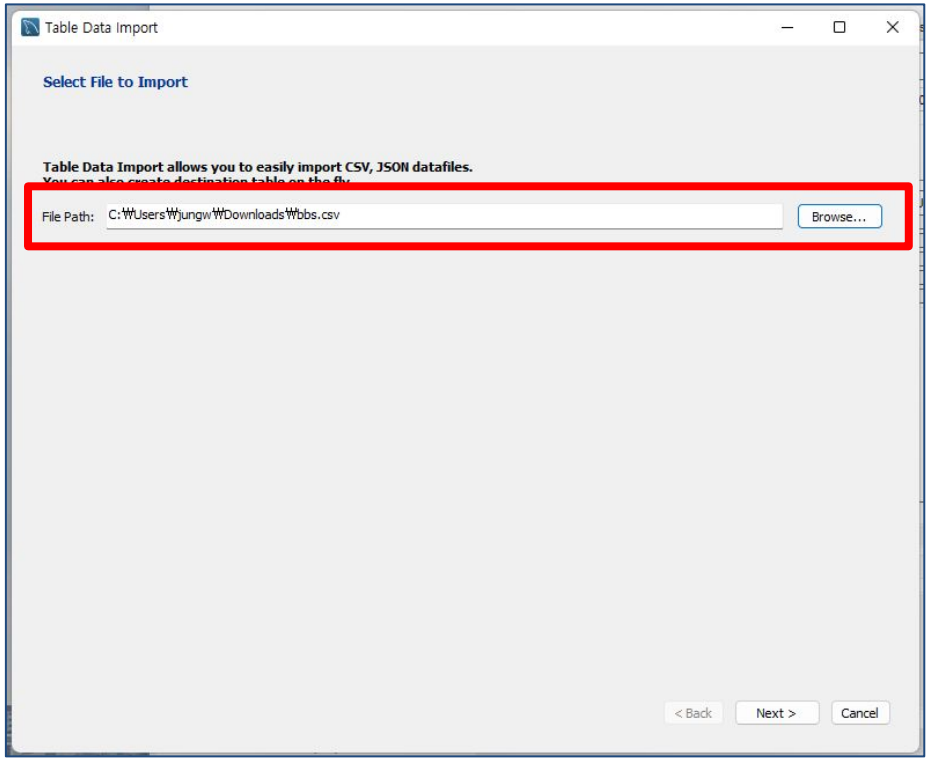
bbs table-fk설정



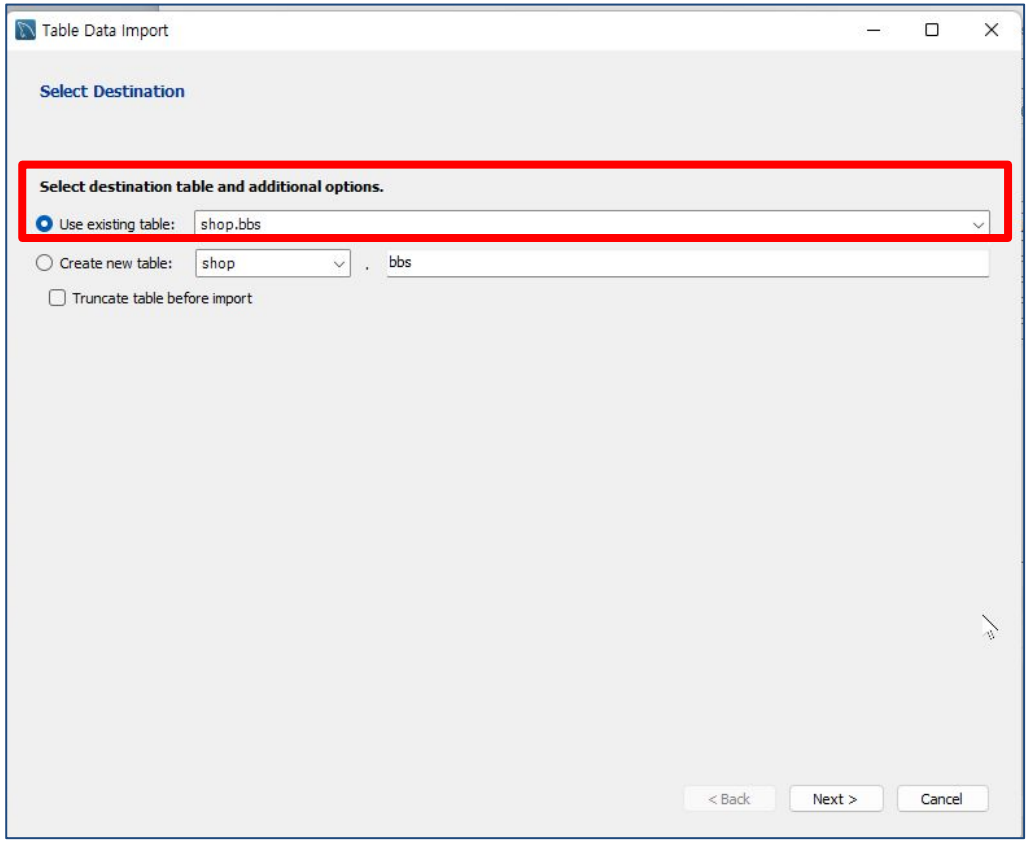
bbs table(데이터 다운로드해서 읽어오기)

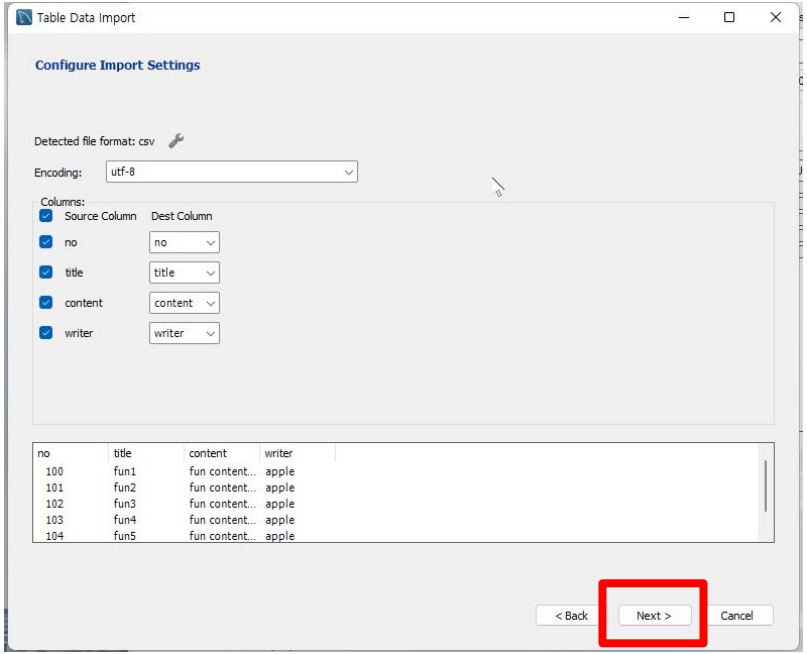
- 실습 데이터 csv다운로드



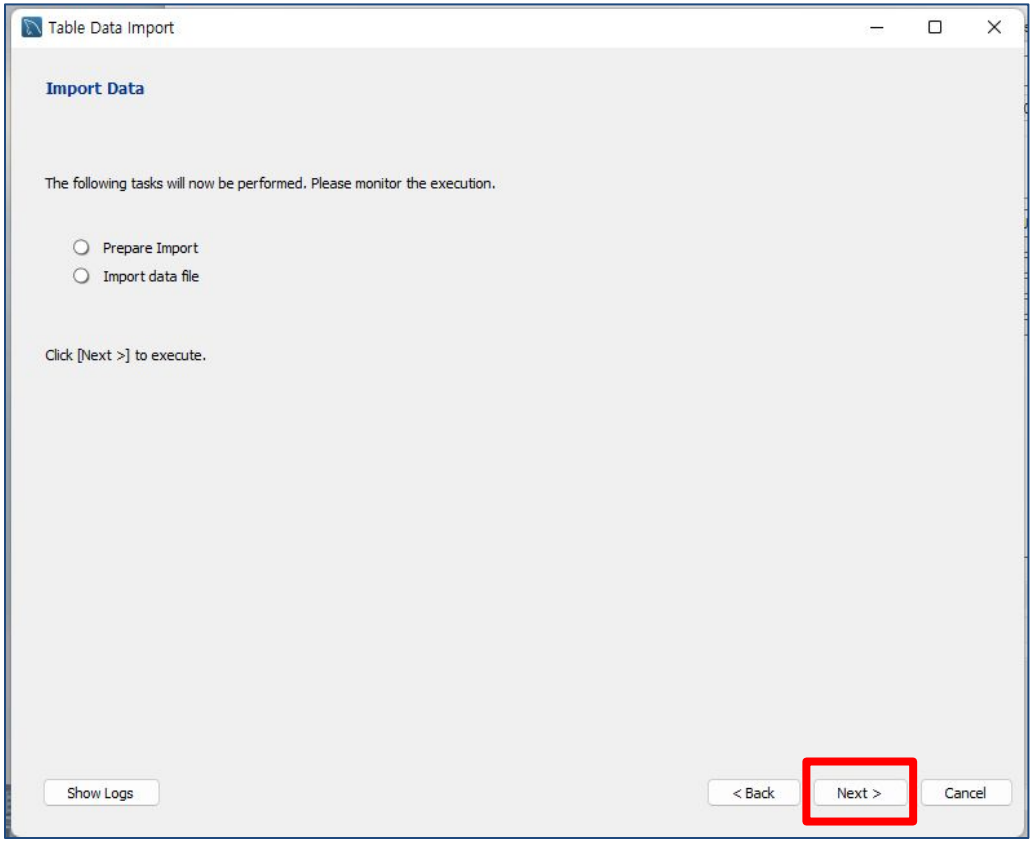


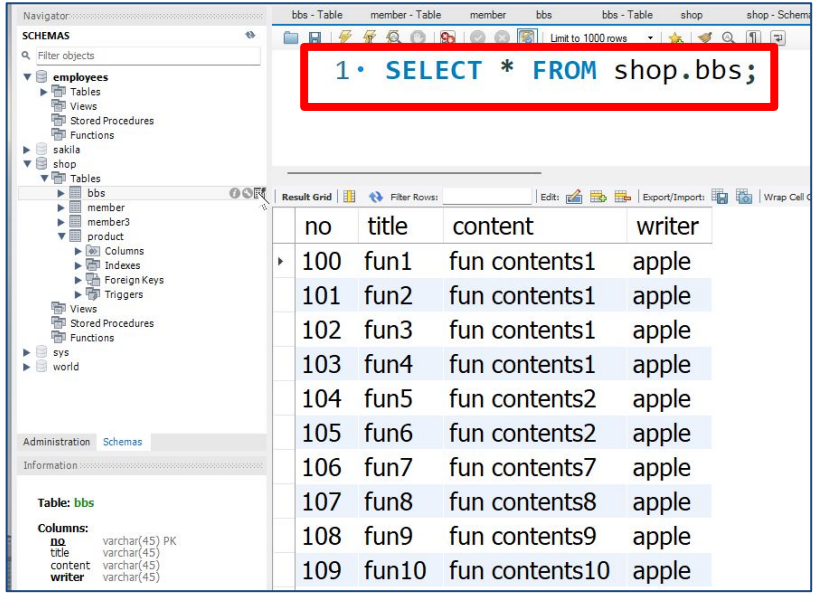
bbs table(csv데이터 읽어오기)





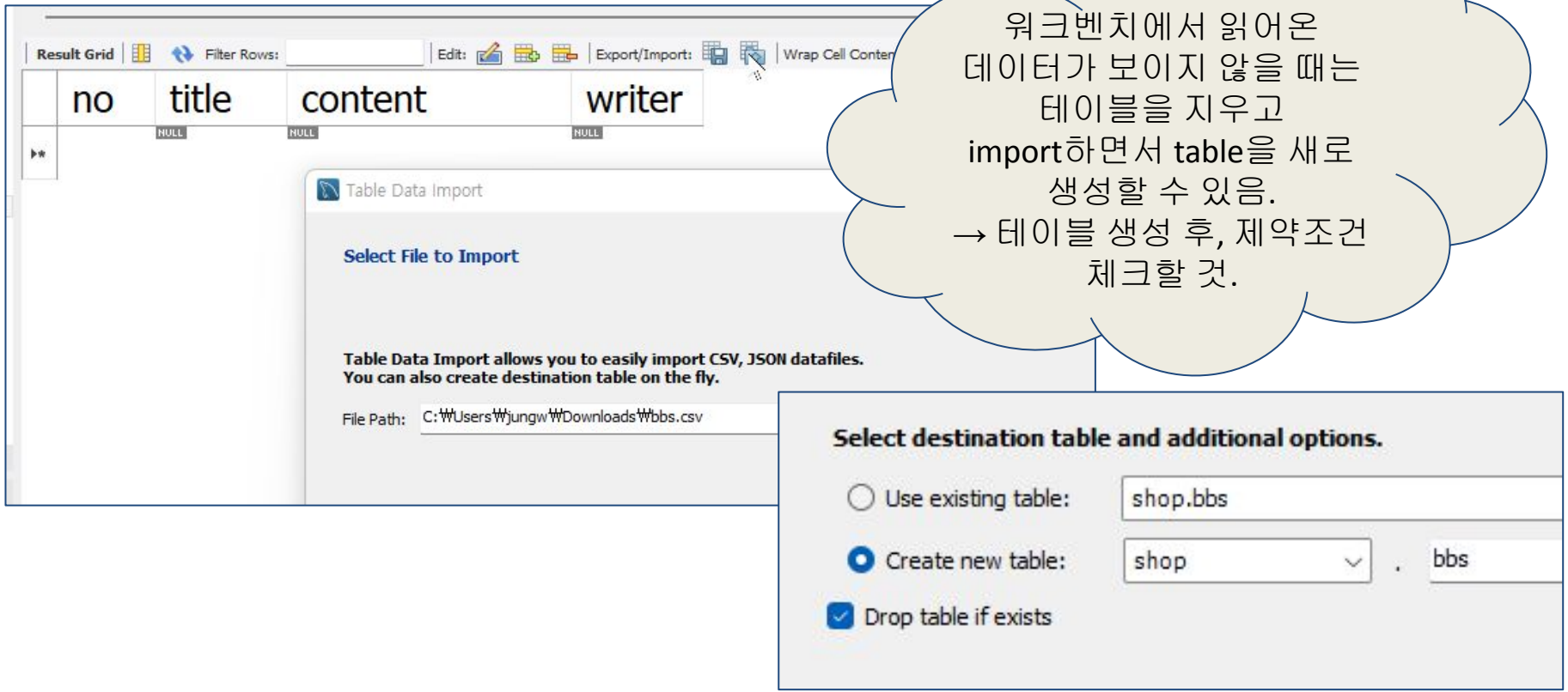
bbs table(csv데이터 읽어오기)





[샘플 파일 위치]

bbs table(csv데이터 읽어오기)

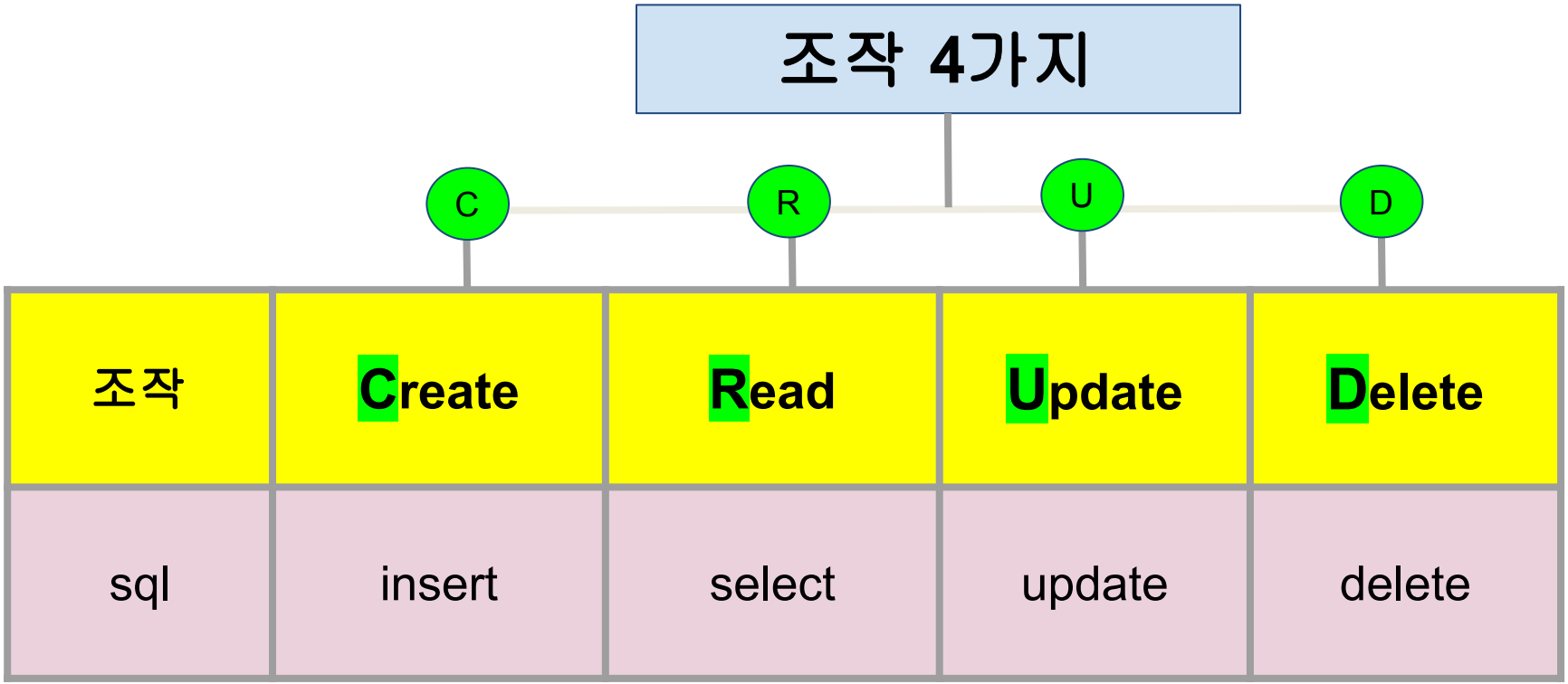


워크벤치에서 읽어온 데이터가 보이지 않을 때는 테이블을 지우고 import하면서 table을 새로 생성할 수 있음.
→ 테이블 생성 후, 제약조건 체크할 것.

DML

(Data Manipulation Language)

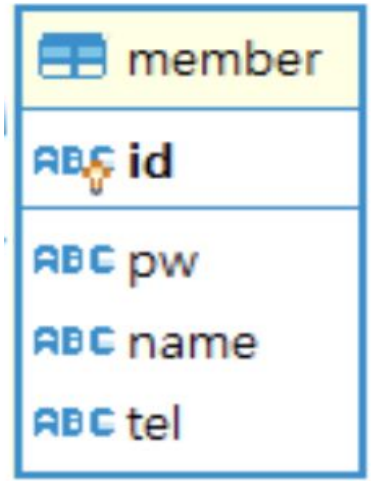
DML-CRUD, 크루드



조작	Create	Read	Update	Delete
sql	insert	select	update	delete

Create(insert) - 특정 컬럼만 선정하여 삽입가능

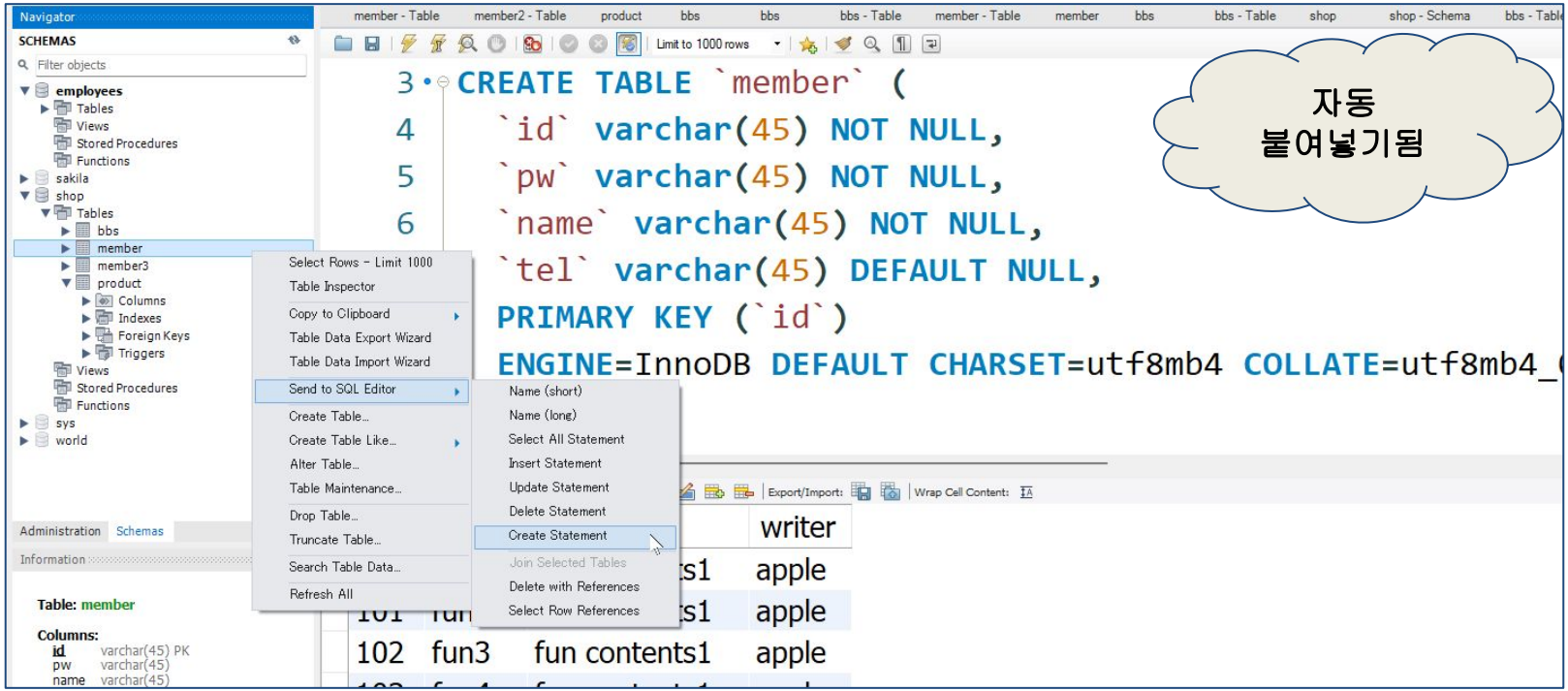
- **insert into member values (모든 컬럼 값);**
- **insert into member(id, pw, name) values (‘555’, ‘555’, ‘song’);**
  - 앞에서 명시한 컬럼의 순서대로 값을 넣어주면 됨.
- **insert into member values (‘555’, ‘555’, ‘song’, null);**



방법1) 테이블복사(스키마+데이터)

- **sql**을 사용하는 것이 더 간단

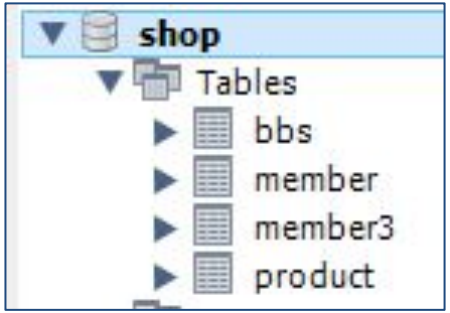
방법2) 테이블복사(스키마+데이터)

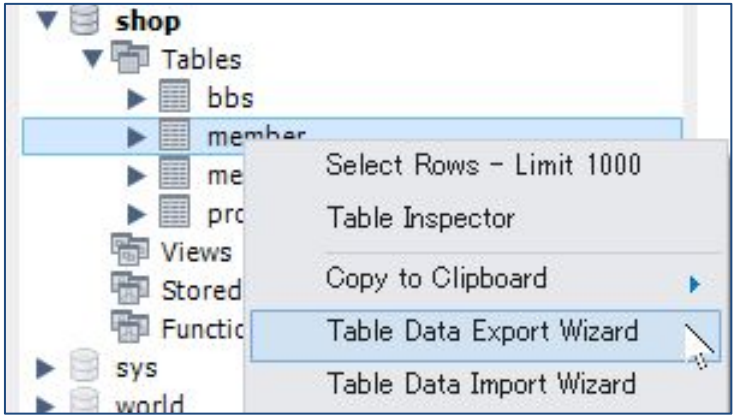


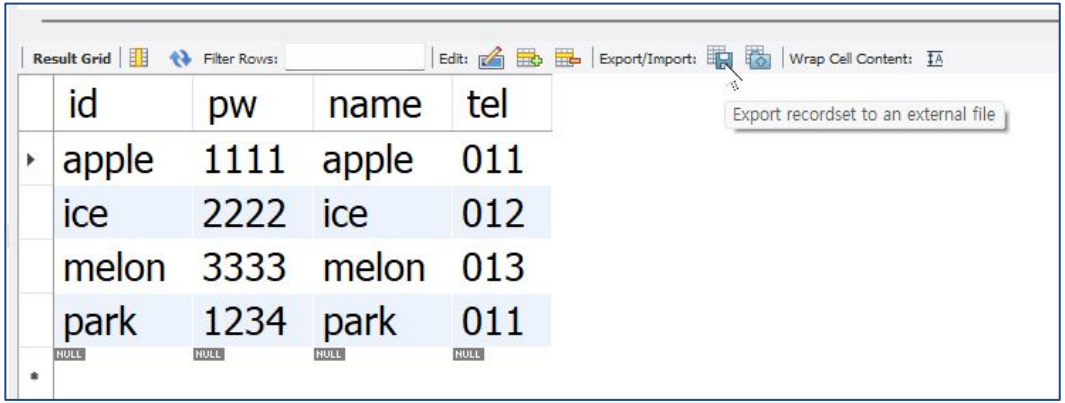
방법2) 테이블복사(스키마+데이터)

```
CREATE TABLE `member3` (
  `id` varchar(45) NOT NULL,
  `pw` varchar(45) NOT NULL,
  `name` varchar(45) NOT NULL,
  `tel` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai
```

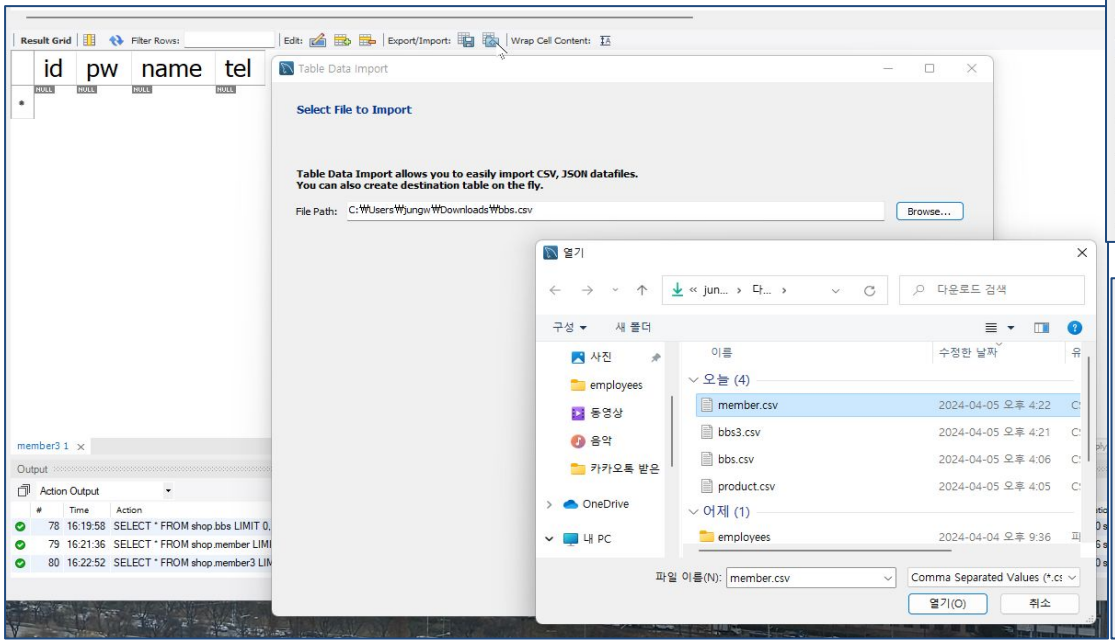
member3로
변경후 실행

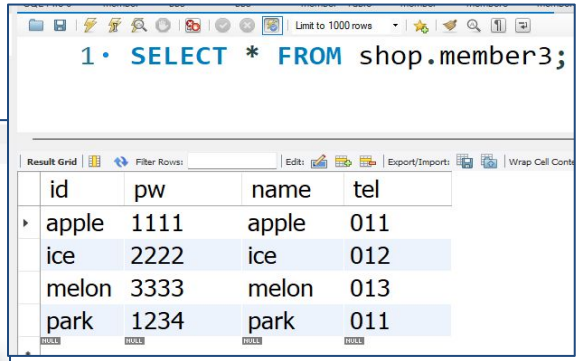


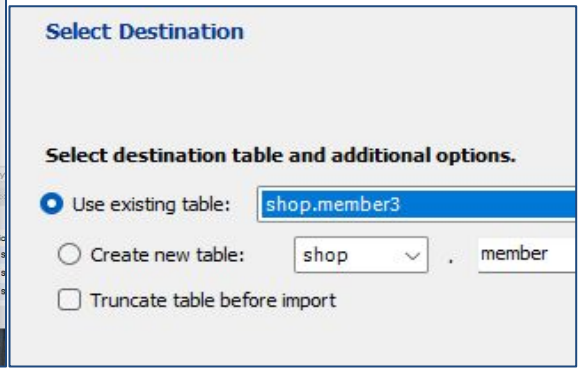




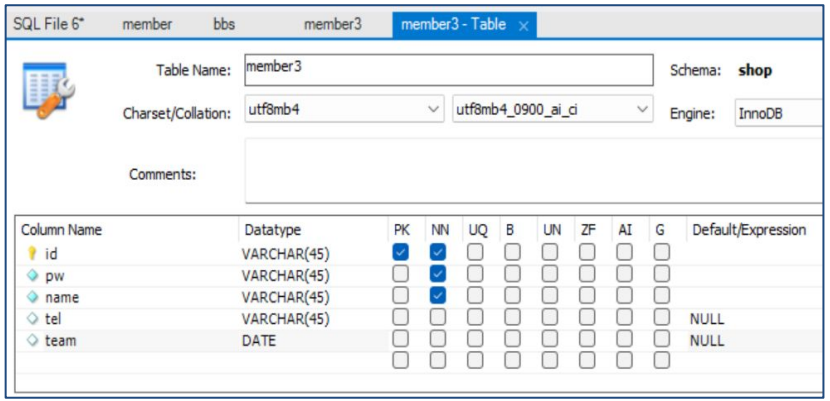
방법2) 테이블복사(스키마+데이터)

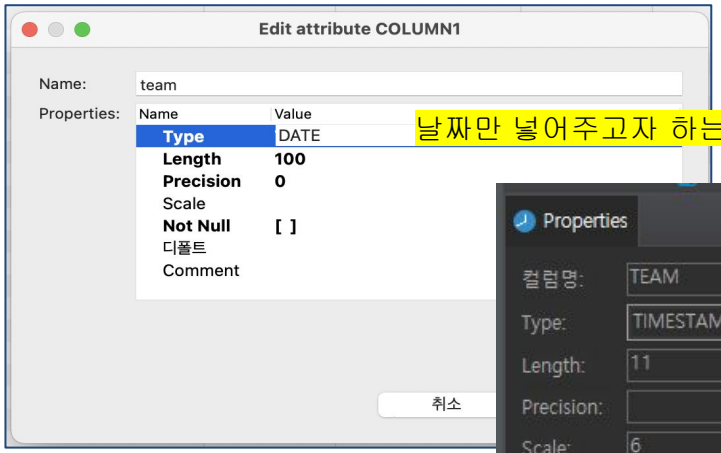




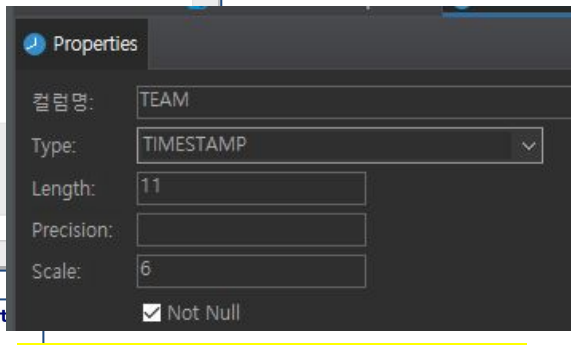


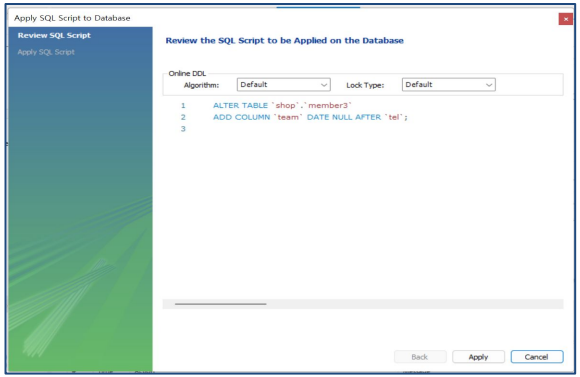
Create(insert)





날짜만 넣어주고자 하는 경우






```
INSERT INTO MEMBER3 VALUES ('555', '555', 'jung', '011', sysdat
SELECT * FROM MEMBER3
```

날짜와 시각까지 넣어주고자 하는 경우

SELECT * FROM MEMBER3 | Enter a SQL expression to filter results (use Ctrl+Space)

	ID	PW	NAME	TEL	TEAM
1	park	1234	park	011	[NULL]
2	apple	1111	apple	011	[NULL]
3	ice	2222	ice	012	[NULL]
4	melon	3333	melon	013	[NULL]
5	555	555	jung	011	2022-10-26 13:08:33

date() 함수 이용

```
14 • insert into member3 values (
15       5,5, 'name', 'tel' , now());
16
17 • select * from member3;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	pw	name	tel	team
5	5	name	tel	2024-04-05
apple	1111	apple	011	NULL
ice	2222	ice	012	NULL
melon	3333	melon	013	NULL
park	1234	park	011	NULL
NULL	NULL	NULL	NULL	NULL

```
14 • insert into member3 values (
15       6, 5, 'name', 'tel' ,  date('2021-01-02')) ;
16
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	pw	name	tel	team
5	5	name	tel	2024-04-05
6	5	name	tel	2021-01-02
7	5	name	tel	2021-01-02
apple	1111	apple	011	NULL

member3 5

mysql

컬럼명	#	Data Type	Not Null	Auto Increm
id	1	varchar(10)	[ ]	[ ]
pw	2	varchar(10)	[ ]	[ ]
name	3	varchar(10)	[ ]	[ ]
tel	4	varchar(10)	[ ]	[ ]
team	5	datetime	[ ]	[ ]

날짜와 시각까지 넣어주고 하는 경우

```
-- mysql team datatype ==> datetime
SELECT * FROM member3


insert into member3
values ('5555', '5555', 'jung', '011', now())
```

member3 1

SELECT * FROM member3 | Enter a SQL expression to filter results (use Ctrl+Space)

그리드

	id	pw	name	tel	team
1	555	555	jung	011	2022-10-27 00:00:00
2	5555	5555	jung	011	2022-10-27 11:22:00

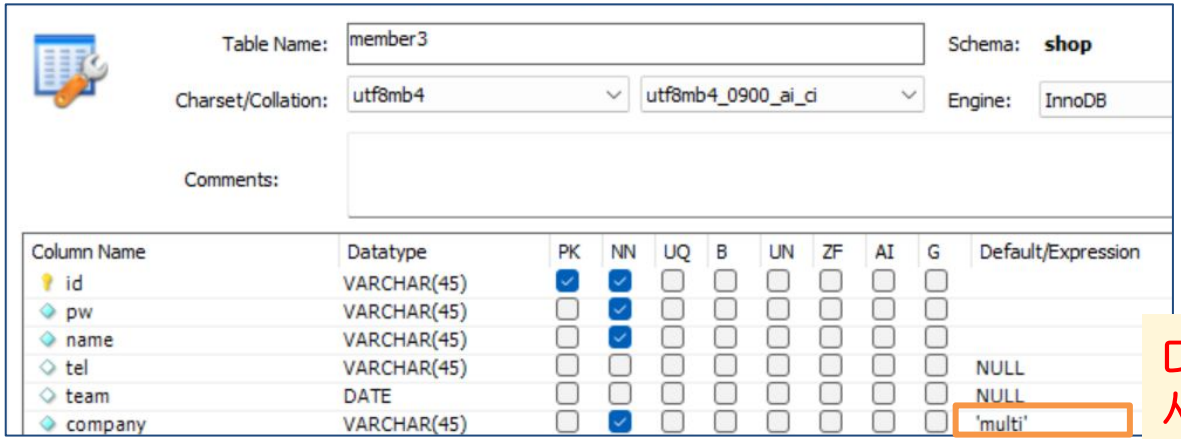
텍스트

Value

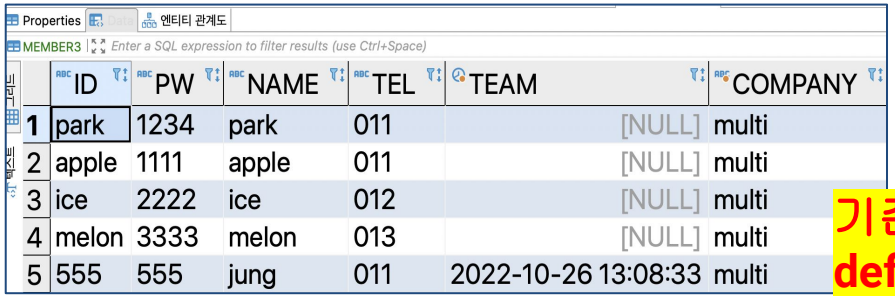
555

Create(insert)-default

- 입력하지 않으면 기본으로 넣어주는 값 설정



디폴트값 설정시 '(싱글따옴표)를 사용해야함.



기존에 있던 값들은 **default값이 multi로 모두 채워짐.**

Create(insert)-default

- `created_at`은 행이 처음 생성될 때 현재의 타임스탬프를 기본값으로 사용
- `updated_at`은 행이 업데이트될 때마다 자동으로 현재 타임스탬프로 갱신

```
CREATE TABLE example1 (
    id INT AUTO_INCREMENT,
    name VARCHAR(50),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    PRIMARY KEY (id)
);
```

Create(insert)-default

- **TIMESTAMP** 컬럼에 명시적으로 값 삽입 가능

  INSERT INTO example1 (name, created_at, updated_at)

  VALUES ('John Doe', '2022-07-01 12:34:56', '2022-07-01 12:34:56');

- 기본 **TIMESTAMP** 값 사용 가능

  INSERT INTO example1 (name) VALUES ('Jane Smith');

Create(insert)-default

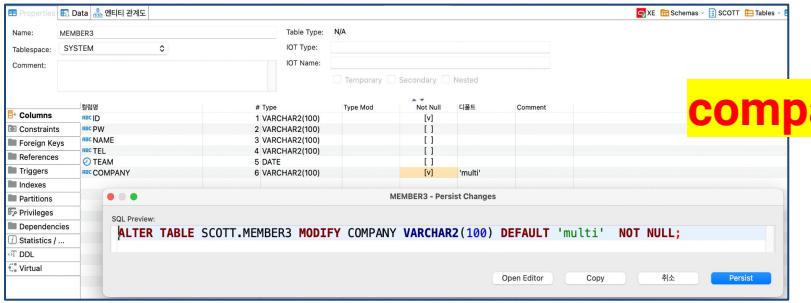

```
CREATE TABLE example3 (
    id INT AUTO_INCREMENT PRIMARY KEY,
    description TEXT,
    updated_at TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP
);

INSERT INTO example3 (description) VALUES ('이것은 설명입니다.');
-- 'updated_at'은 기본값인 CURRENT_TIMESTAMP를 사용하여 채워짐.

INSERT INTO example3 (description, updated_at) VALUES ('또 다른 설명', NULL);
-- 'updated_at'은 명시적으로 NULL을 할당 가능
```

Create(insert)- now()와 비교

특성	NOW()	CURRENT_TIMESTAMP
형식	함수	키워드/함수
호출 방식	소괄호를 사용 (NOW())	소괄호 사용 가능 (CURRENT_TIMESTAMP() 또는 CURRENT_TIMESTAMP)
반환 타입	DATETIME	TIMESTAMP
시간대 변동	시간대에 따른 값 변동 적용	시간대에 따른 값 변동 적용
SQL 표준 준수	MySQL/MariaDB 특화	ANSI SQL 표준 준수
범용성	주로 MySQL 및 MariaDB에서 사용	다양한 SQL 데이터베이스 시스템에서 일관된 사용 가능

Oracle과 Microsoft SQL Server에서도 사용 가능

Create(insert)-default



company → not null check!!

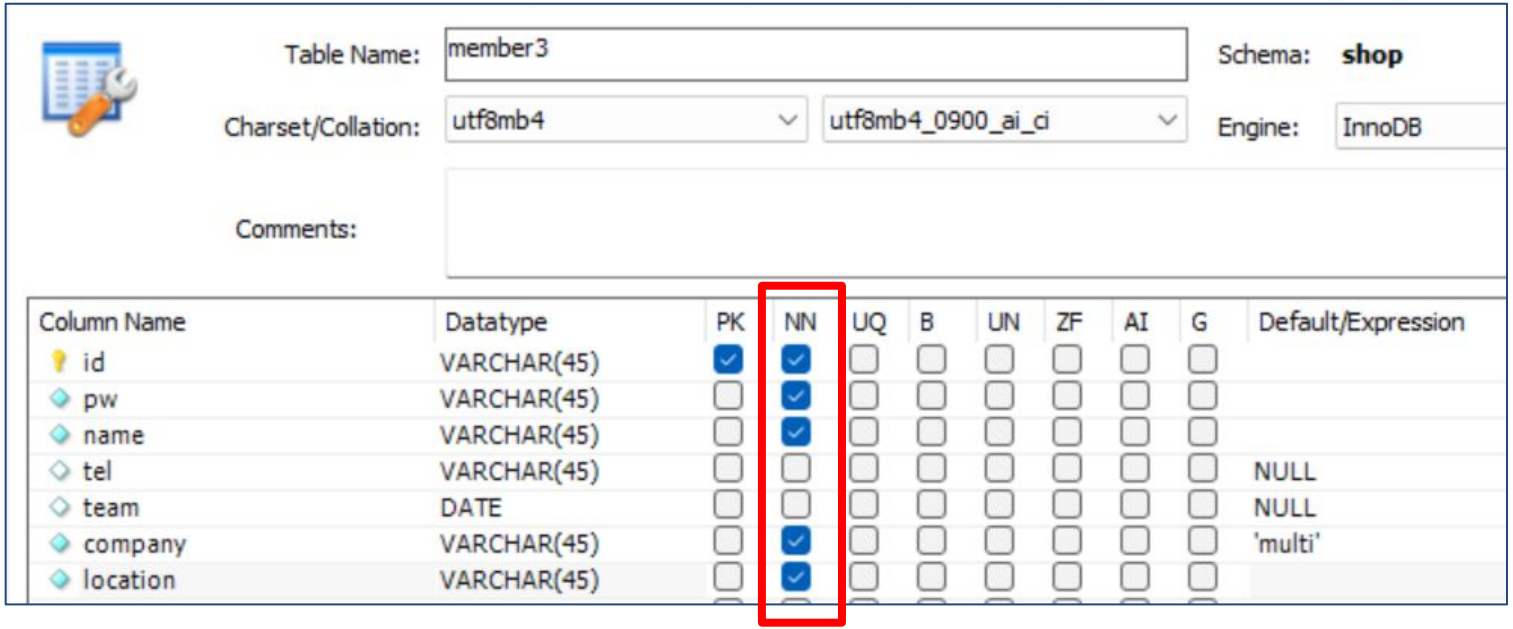
```
INSERT INTO MEMBER3(id, pw) VALUES ('999', '999')

SELECT * FROM MEMBER3
```

ELECT * FROM MEMBER3 | Enter a SQL expression to filter results (use Ctrl+Space)

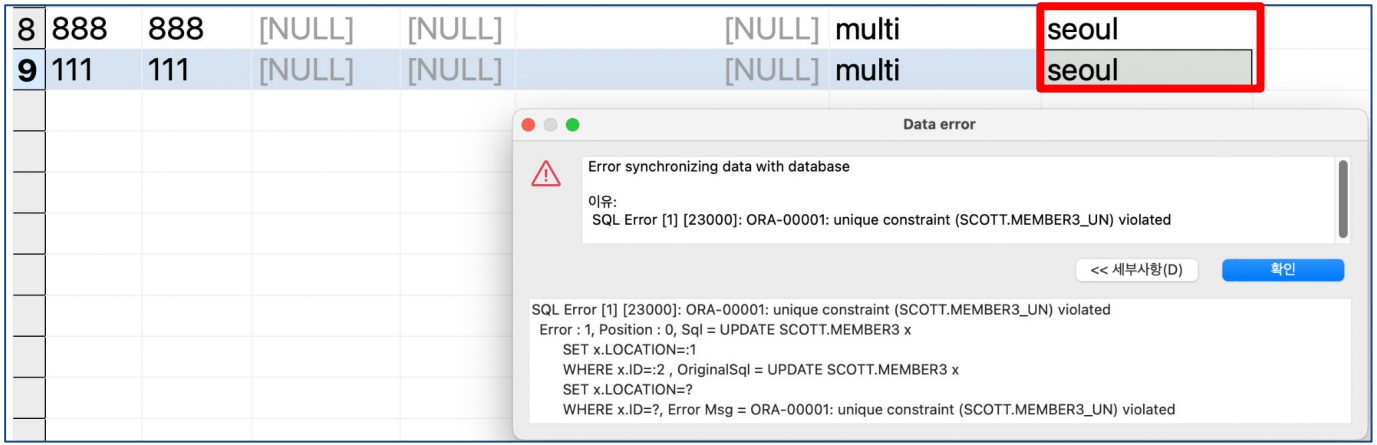
	ID	PW	NAME	TEL	TEAM	COMPANY
1	park	1234	park	011	[NULL]	multi
2	apple	1111	apple	011	[NULL]	multi
3	ice	2222	ice	012	[NULL]	multi
4	melon	3333	melon	013	[NULL]	multi
5	555	555	jung	011	2022-10-26 13:08:33	multi
6	777	777	song	015	[NULL]	lg
7	999	999	[NULL]	[NULL]	[NULL]	multi

not null인 것만 insert해보자
→ default값을 그대로 사용하고자 하는 경우에는 입력안해도 됨.

Create(insert)-unique

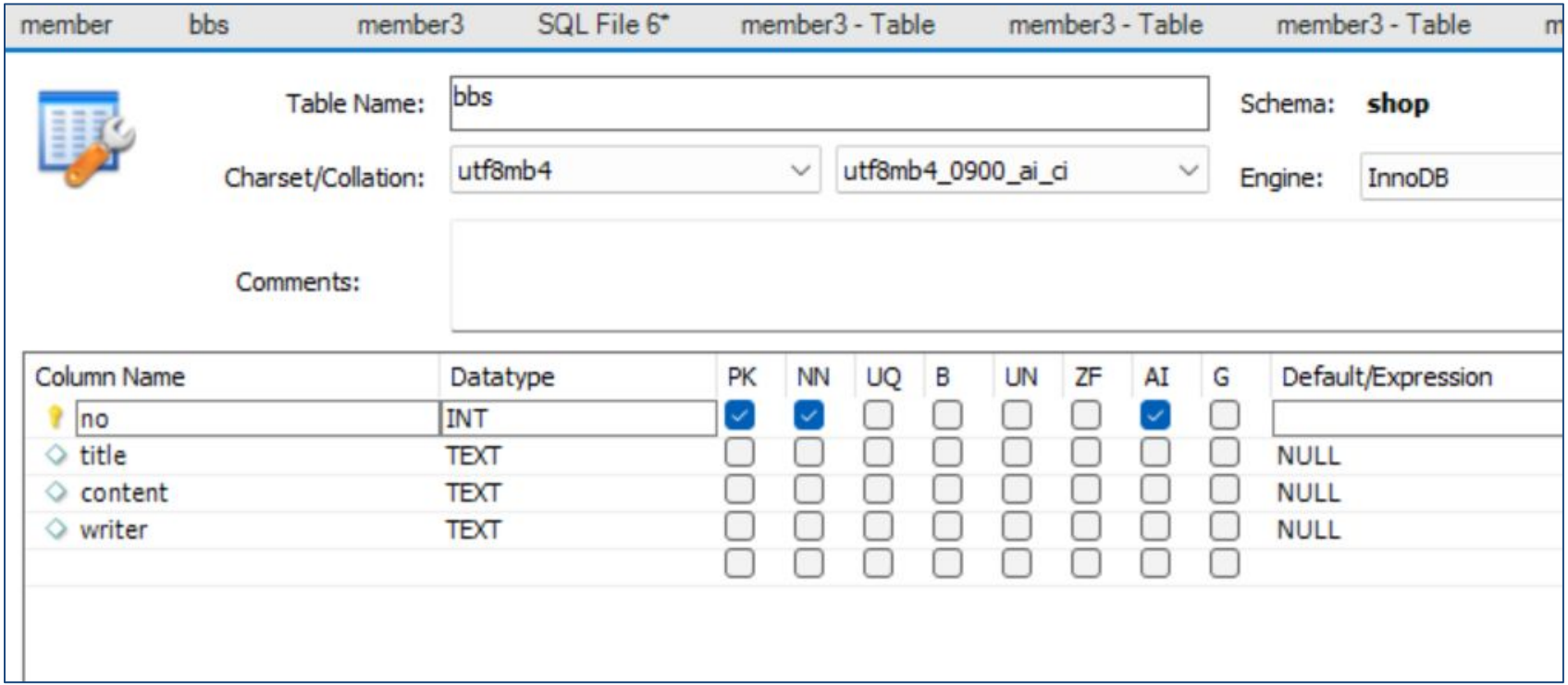


Create(insert)-unique



mysql에서는

- 1)pk설정 후, 2)autoincremet체크, data type은 int



mysql에서는

- **insert**시 해당 자동증가 컬럼값은 넣지 않음.

```
insert into bbs (null, “title”, “content”, “apple”);
```

member | bbs | member3 | SQL File 6* | member3 - Table | member3 - Table | member3 - Table | member3 | member3 - Table | bbs - Ta

Limit to 1000 rows

```
26  insert into bbs
27  values (null, 'title', 'content', 'apple');
28
29  select * from bbs;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

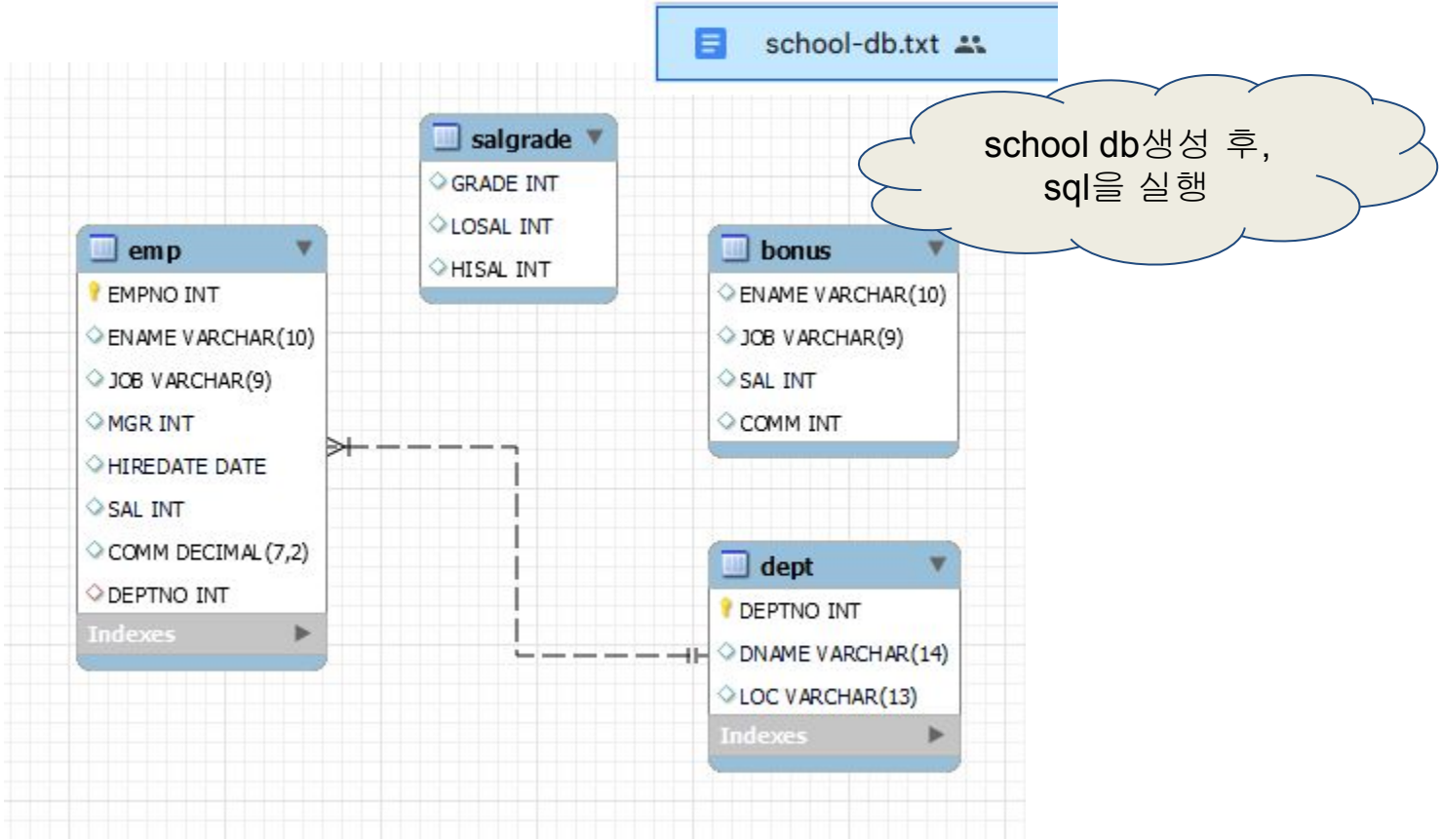
no	title	content	writer
108	fun9	fun contents9	apple
109	fun10	fun contents10	apple
110	fun11	fun contents11	apple
111	fun12	fun contents12	apple
112	fun13	fun contents13	apple
113	title	content	apple

위 식을 정리(제약조건을 넣어 테이블 생성 시)

```
CREATE TABLE "MEMBER4"
  (    ID VARCHAR(100) PRIMARY KEY,
       PW VARCHAR(100),
       NAME VARCHAR(100),
       TEL VARCHAR(100),
       TEAM DATE, - - 나중에 sysdate를 넣을 항목
       COMPANY VARCHAR(100) DEFAULT 'multi' NOT NULL,
       LOCATION VARCHAR(100) UNIQUE
)
```

→ MEMBER4를 만들고, instance를 넣어 적용한 제약조건이 맞는지 확인해보자.

실습파일 다운로드


school-db.txt
school db생성 후,
sql을 실행
salgrade
GRADE INT
LOSAL INT
HISAL INT
emp
EMPNO INT
ENAME VARCHAR(10)
JOB VARCHAR(9)
MGR INT
HIREDATE DATE
SAL INT
COMM DECIMAL(7,2)
DEPTNO INT
Indexes
bonus
ENAME VARCHAR(10)
JOB VARCHAR(9)
SAL INT
COMM INT
dept
DEPTNO INT
DNAME VARCHAR(14)
LOC VARCHAR(13)
Indexes

실습파일 다운로드

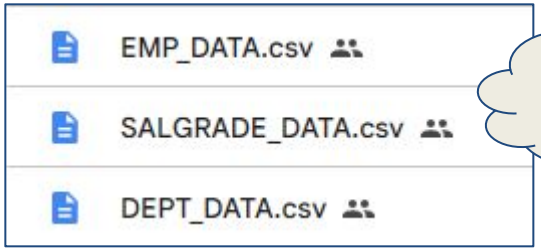
```
CREATE TABLE DEPT
(
        DEPTNO int(2),
        DNAME VARCHAR(14),
        LOC VARCHAR(13)
);

CREATE TABLE EMP(
        EMPNO int(11),
        ENAME VARCHAR(10),
        JOB VARCHAR(9),
        MGR int(11),
        HIREDATE DATE,
        SAL int(11),
        COMM decimal(7,2),
        DEPTNO int(2)
);
```

```
CREATE TABLE SALGRADE
(
        GRADE int(11),
        LOSAL  int(11),
        HISAL  int(11)
);

CREATE TABLE BONUS
(
        ENAME VARCHAR(10),
        JOB VARCHAR(9),
        SAL int(11),
        COMM int(11)
);
```

실습파일 다운로드

```
ALTER TABLE DEPT ADD (
CONSTRAINT PK_DEPT
PRIMARY KEY(DEPTNO));

ALTER TABLE EMP ADD (
CONSTRAINT PK_EMP
PRIMARY KEY(EMPNO));

ALTER TABLE EMP ADD (
CONSTRAINT FK_DEPTNO
FOREIGN KEY (DEPTNO)
REFERENCES DEPT (DEPTNO));
```

실습파일 다운로드



data import

```
SELECT * FROM school.emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7499	ALLEN	SALESMAN	7698	1981-02-20	1600	300.00	30
7521	WARD	SALESMAN	7698	1981-02-22	1250	500.00	30
7654	MARTIN	SALESMAN	7698	1981-09-28	1250	1400.00	30
7844	TURNER	SALESMAN	7698	1981-09-08	1500	0.00	30

```
SELECT * FROM school.salgrade;
```

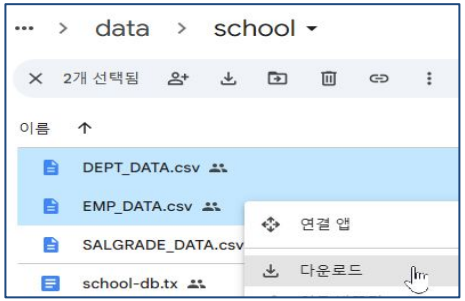
GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

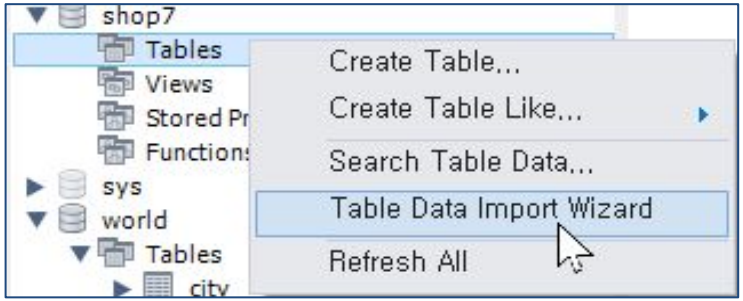
```
SELECT * FROM school.dept;
```

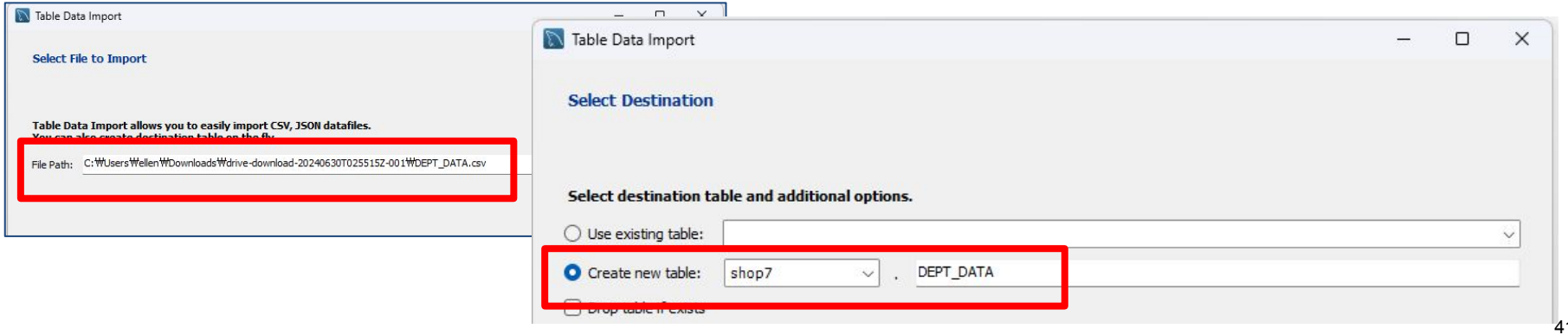
DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

import data가 모두 되지 않는 경우

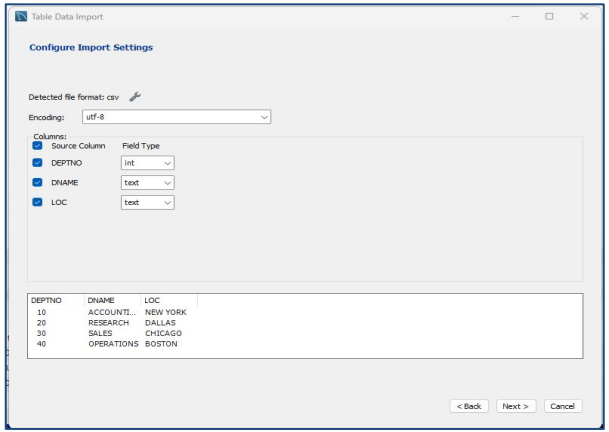
- 1) 외래키 삭제 2) 기존 **emp**테이블을 삭제 후, 3)다음과 같이 **import**하여 테이블 명 수정

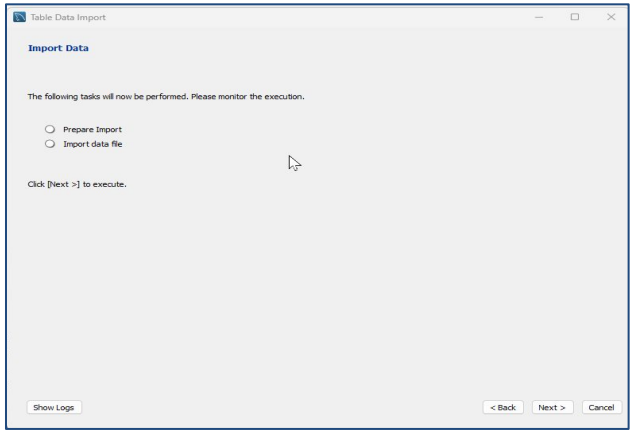


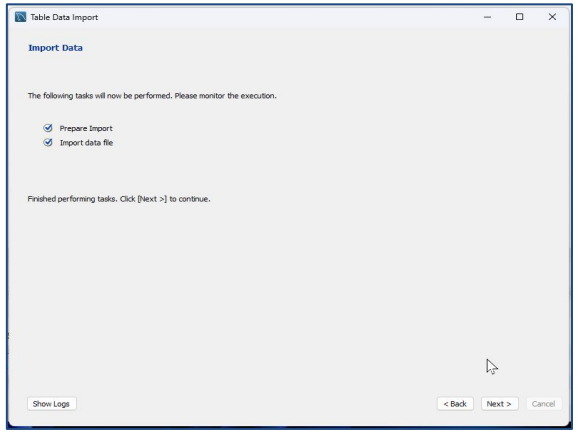


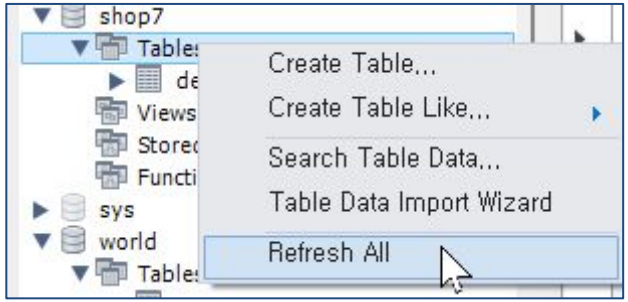


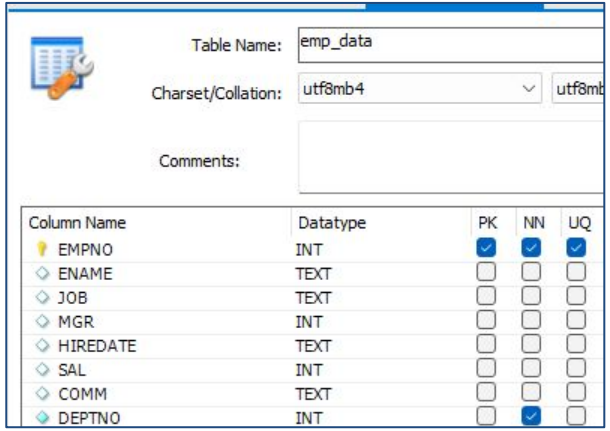
import data가 모두 되지 않는 경우







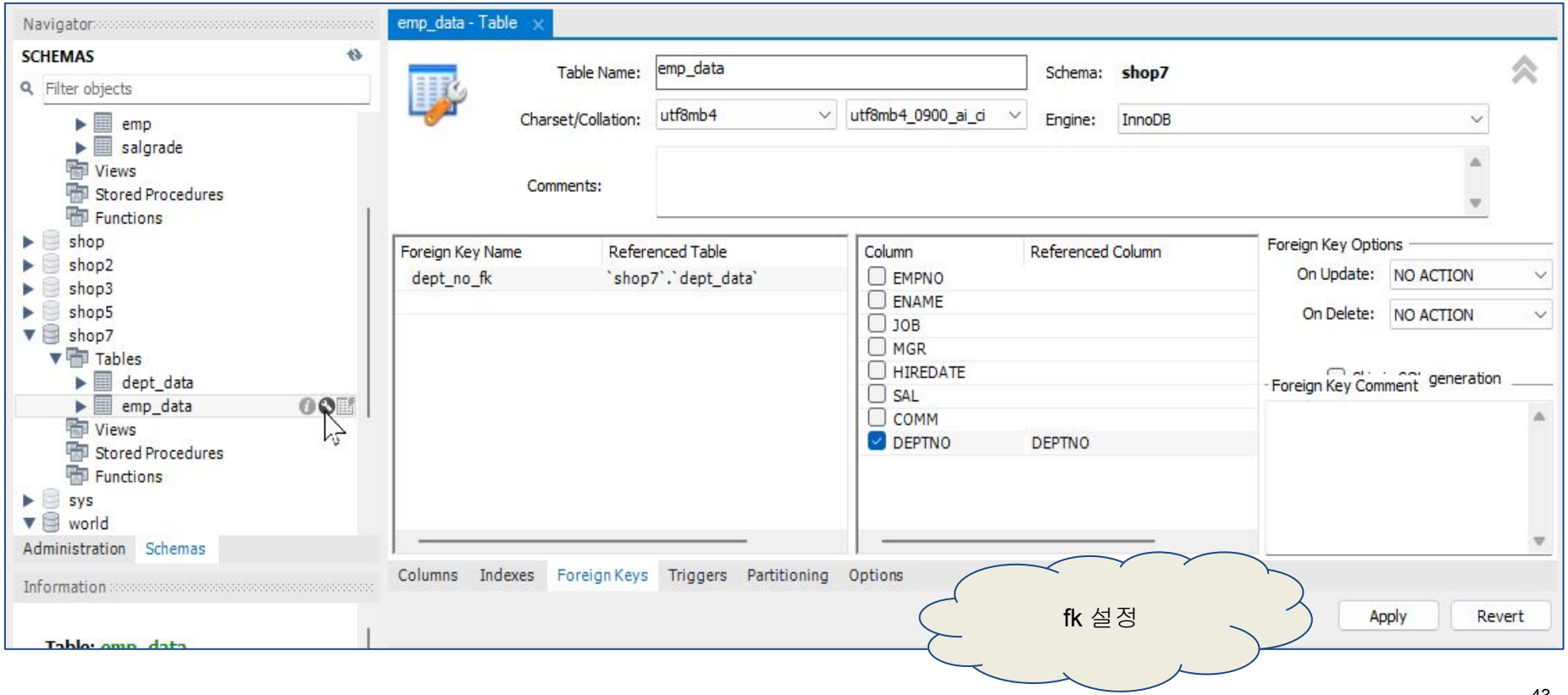




pk, NOT NULL설정

import data가 모두 되지 않는 경우

- 외래키 설정(emp_data table의 deptno와 dept_data table의 deptno)



Read(select)-행단위, 열단위

- **emp테이블을 이용**
  - 행단위 검색: selection
  - 열단위 검색: projection
- **select * from emp;**
- **select deptno from emp; 특정컬럼**
- **select distinct deptno from emp; 중복제거**
- **select ename, sal * 12 as yearsal from emp; as(alias, 별칭)**
- **select * from emp order by sal; sal컬럼의 값 순서대로 정렬(오름차순)**
- **select * from emp order by sal desc; sal컬럼의 값 순서대로 정렬(내림차순)**

Read(select)-where(조건검색)

- select * from emp where deptno = 30;
- select * from emp where not deptno = 30;
- select * from emp where deptno = 30 and job = ‘salesman’;
- select * from emp where deptno = 30 or job = ‘salesman’;
- select * from emp where sal >= 3000;
- select * from emp where sal != 3000;
- select * from emp where job in (‘MANAGER’, ‘SALESMAN’, ‘CLERK’); or연산, 포함
- select * from emp where job not in (‘MANAGER’, ‘SALESMAN’, ‘CLERK’);
- select * from emp where sal between 2000 and 3000; 사이값
- select * from emp where sal not between 2000 and 3000;

Read(select)-where(조건검색, %(0~무한대) _(한글자))

- select * from emp where ename like '_L%'; L앞에는 한글자, 뒤에는 0~무한대
- select * from emp where ename like '%AM%'; AM이 앞에오거나, 가운데 오거나, 맨뒤에 와도 됨
- select * from emp where ename not like '%AM%';
- select * from emp where comm is null;
- select * from emp where comm is not null;
- SELECT * FROM EMP WHERE HIREDATE > '1981-01-01';

Update(update), Delete(delete)

- dept테이블을 dept_temp2로 복사
- update dept_temp2 set loc = ‘SEOUL’; 전체수정
- update dept_temp2 set dname = ‘DATABASE’, loc = ‘SEOUL’ where deptno = 40;
- delete from dept_temp2 where job = ‘MANAGER’;

제약조건 추가

- **EMP 테이블 추가 제약조건**

1. **참조 무결성 강화 :** 매니저가 실제로 존재하는 직원인지 검증

```
ALTER TABLE EMP ADD CONSTRAINT FK_MGR
FOREIGN KEY (MGR) REFERENCES EMP(EMPNO);
```

2. **데이터 유효성 검사 :** SAL (급여)이나 COMM (커미션)과 같은 금액 관련 필드에 최소값을 설정하여 음수 값이 입력되지 않도록 설정

```
ALTER TABLE EMP
ADD CONSTRAINT CHK_SAL CHECK (SAL >= 0),
ADD CONSTRAINT CHK_COMM CHECK (COMM >= 0);
```

제약조건 추가

- **EMP 테이블 추가 제약조건**

  3. **데이터 유효성 검사 :** JOB 필드가 가질 수 있는 값으로
     'MANAGER', 'SALESMAN', 'CLERK', 'ANALYST'를 지정

```
-- EMP 테이블의 JOB 컬럼을 ENUM 타입으로 변경
ALTER TABLE EMP MODIFY JOB ENUM('MANAGER', 'SALESMAN', 'CLERK', 'ANALYST')
NOT NULL;

INSERT INTO EMP (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)
VALUES
(7902, 'FORD', 'ANALYST', 7566, '1981-12-03', 3000, 0.00, 20);
```

제약조건 추가

- **DEPT 테이블 추가 제약조건**

1. **유일성 강화** : DNAME (부서 이름)이 유니크하다는 제약을 추가하여, 같은 이름의 부서가 중복되어 생성되지 않도록 설정

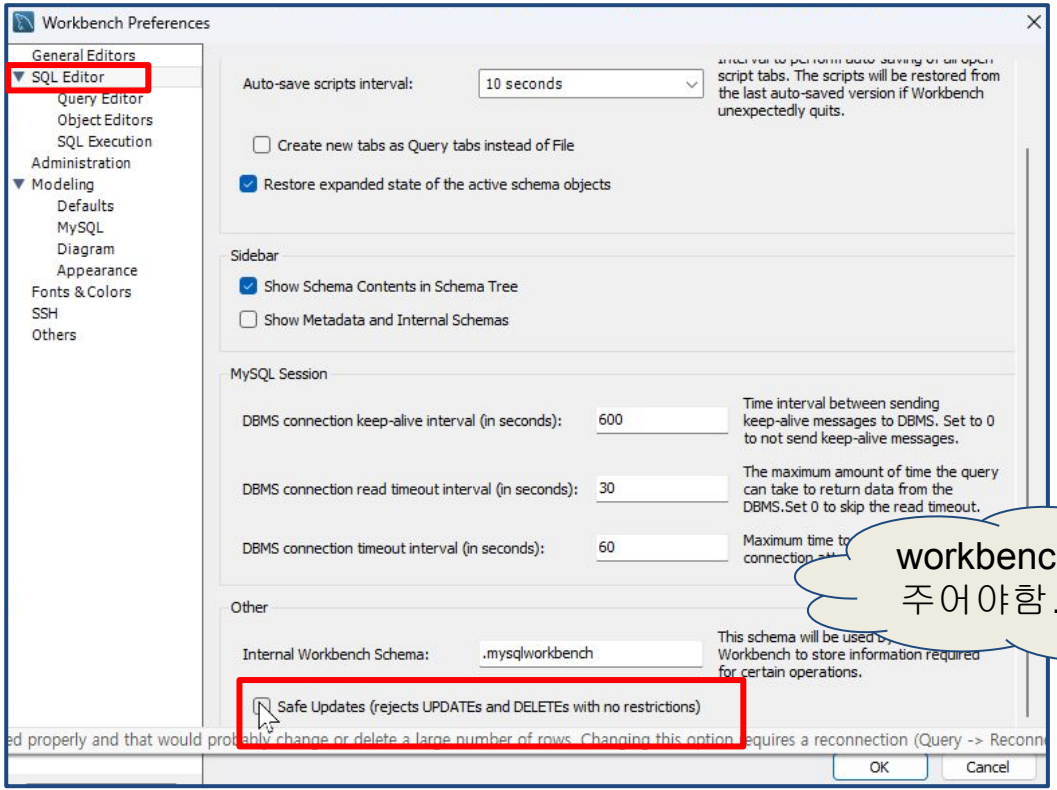
```
ALTER TABLE DEPT ADD CONSTRAINT UNQ_DNAME UNIQUE (DNAME);
```

- **SALGRADE 테이블 추가 제약조건**

1. **데이터 유효성 검사 :** LOSAL과 HISAL이 올바른 범위 내에 있도록 하고, LOSAL이 HISAL보다 항상 작거나 같아야 한다는 제약을 추가

```
ALTER TABLE SALGRADE
ADD CONSTRAINT CHK_SALGRADE_RANGE CHECK (LOSAL <= HISAL);
```

safe mode 해제



workbench재시작해 주어야함. → restart

핵심 정리

- DML, CRUD
- Create : now(), default, sequence, unique
- Read : where, distinct, order by, and, or, not, in
- Update : where, cell단위(칸)
- Delete : where, record단위(행), commit/rollback 대상(취소가능, truncate와 비교-취소불가능)