



8장. MySQL Workbench

- ☐ Data manipulation
- ☐ Backup & Recovery
- ☐ 테이블 생성

수업 전에

- 이전 실습에서 OFF 로 했던 autocommit 변수를 다시 ON 으로 한다.

↳ 업데이트가 바로 반영

```
SET autocommit = ON;  
SHOW VARIABLES LIKE 'autocommit';
```

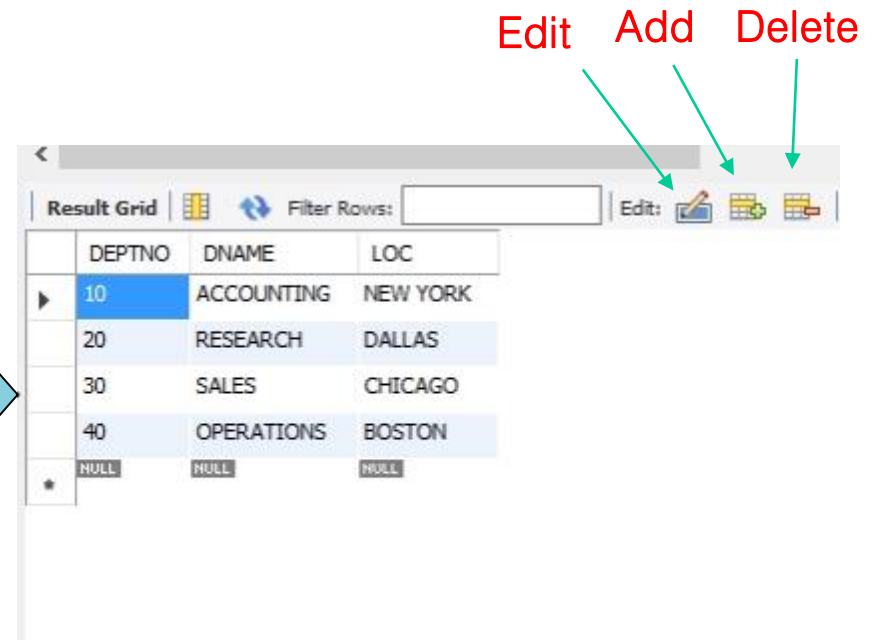
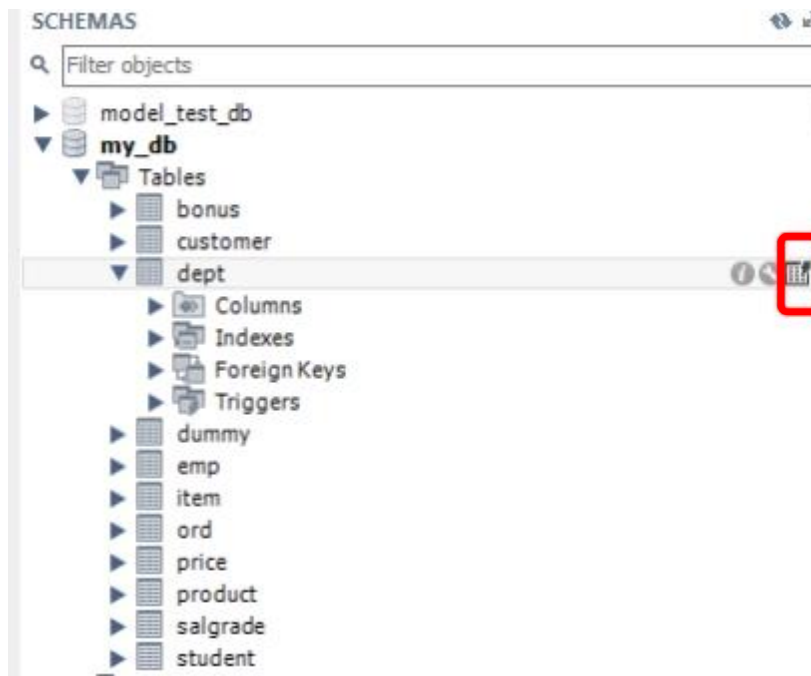
| | Variable_name | Value |
|---|---------------|-------|
| ▶ | autocommit | ON |

Data Manipulation

1. 테이블에 데이터 넣기

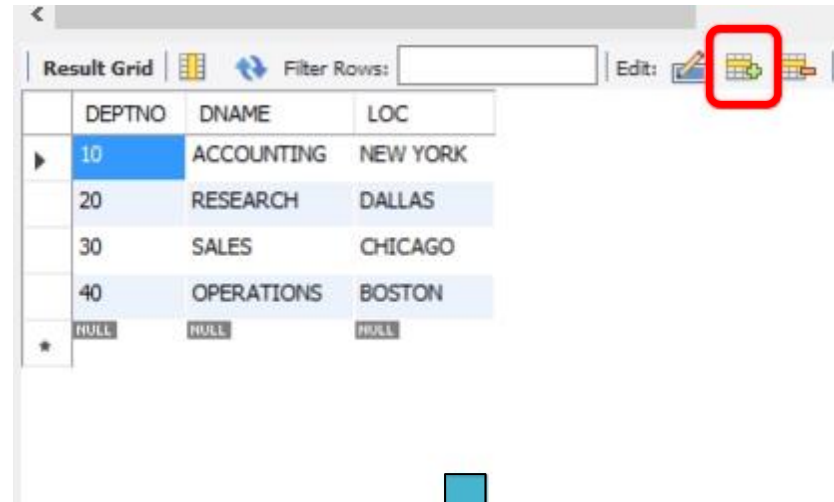
❑ dept 테이블에 50 번 부서를 추가해 보자

(1) dept 테이블을 불러온다

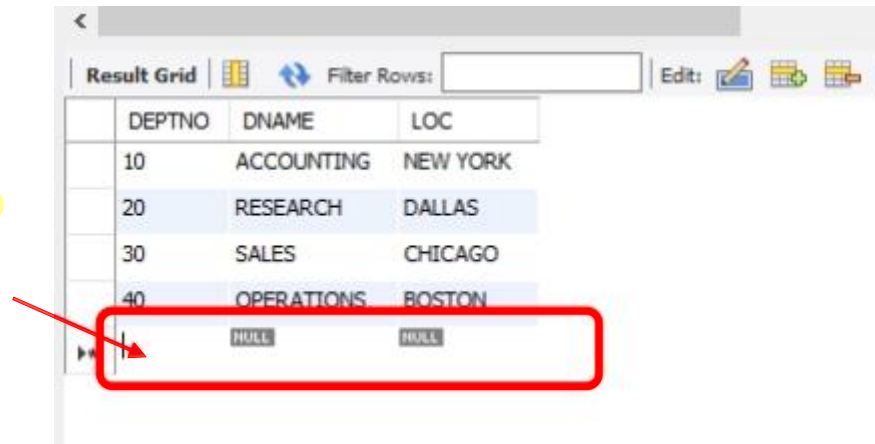


1. 테이블에 데이터 넣기

(2) 행 추가 버튼 클릭

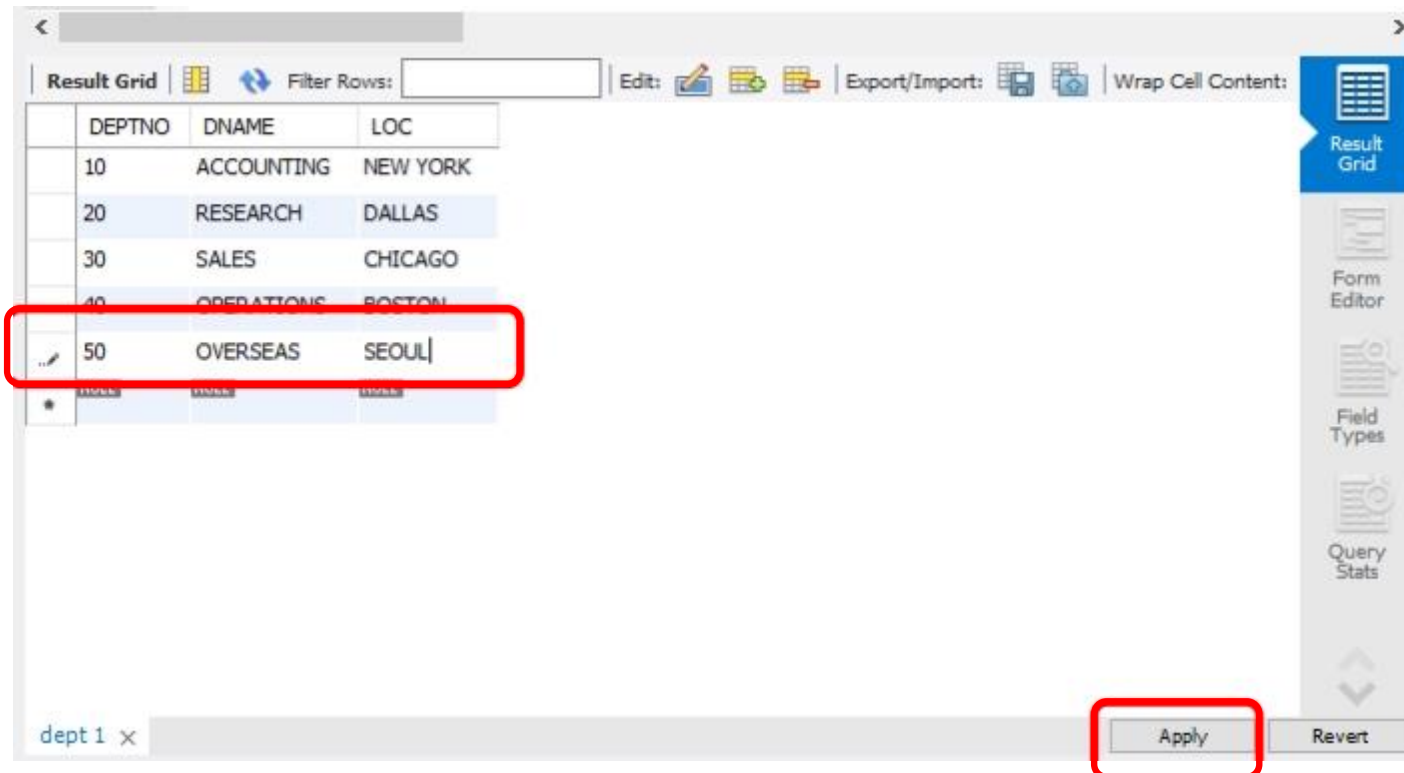


행추가버튼대신 **여기를**
직접클릭해도됨



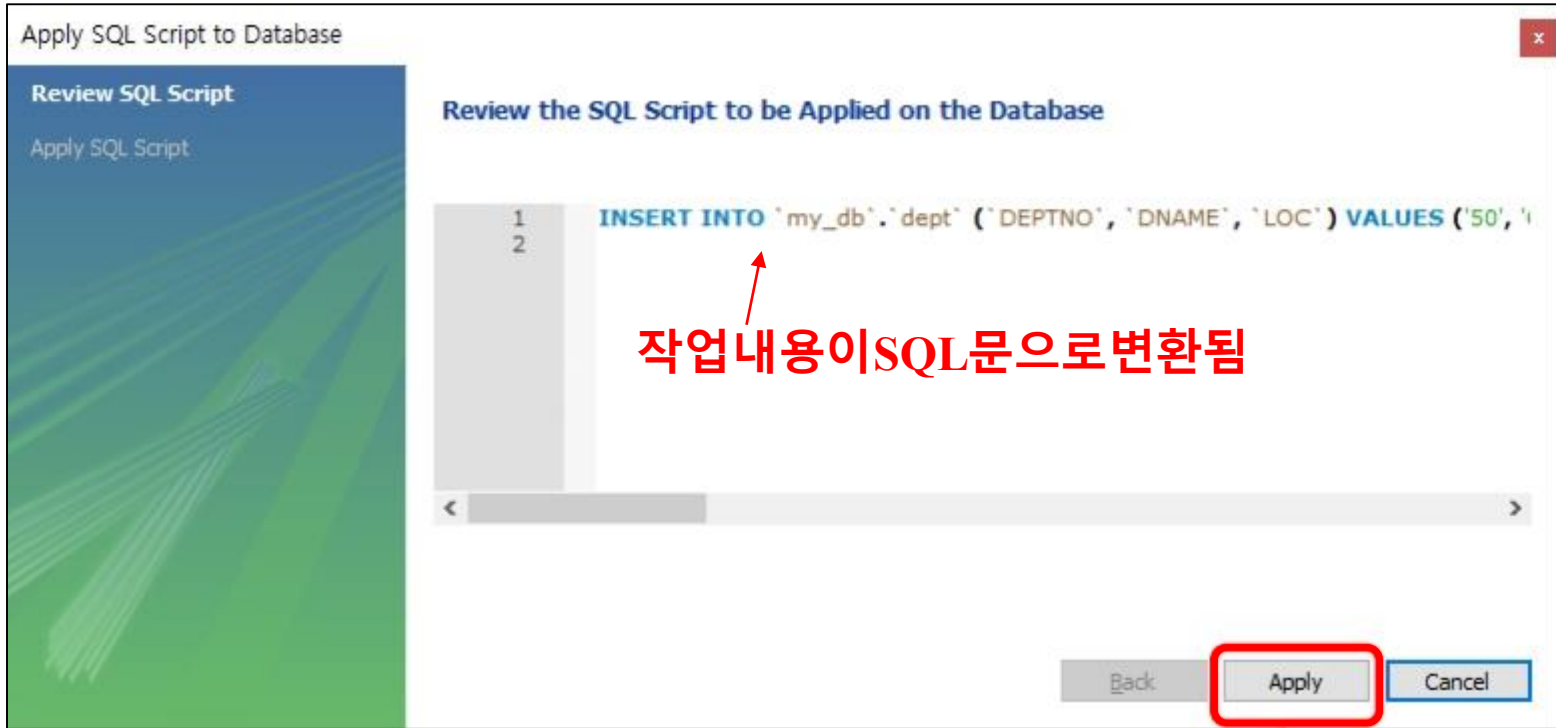
1. 테이블에 데이터 넣기

(3) 50, OVERSEAS, SEOUL 을 입력하고 저장한다



입력내용저장

1. 테이블에 데이터 넣기



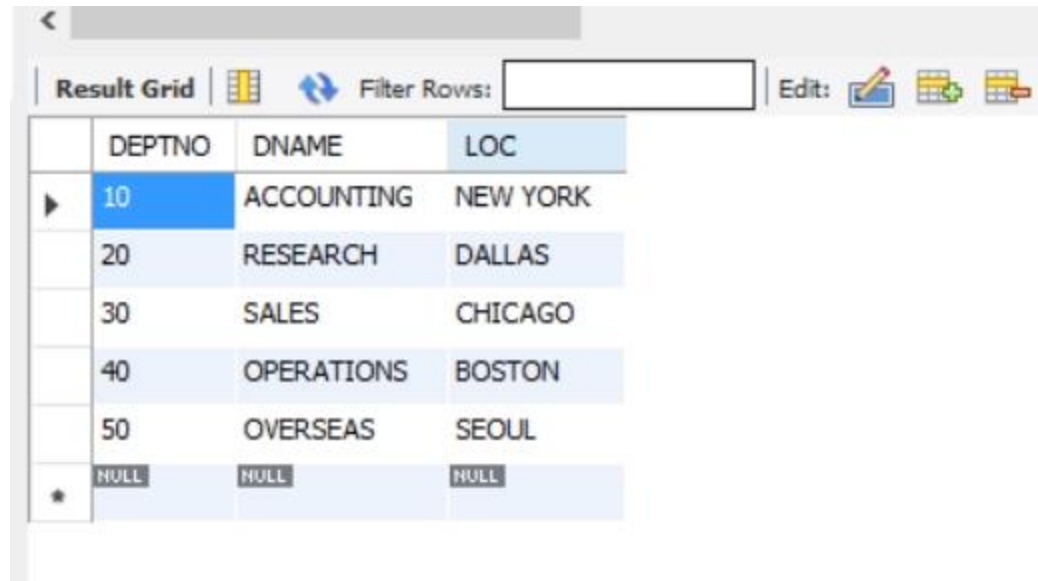
1. 테이블에 데이터 넣기



입력작업완료

1. 테이블에 데이터 넣기

데이터가 입력된 모습



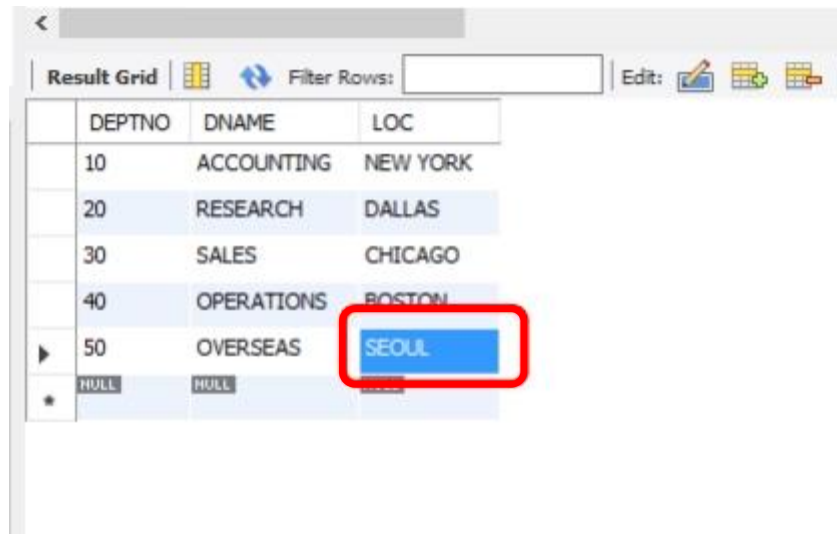
The screenshot shows the MySQL Workbench Result Grid interface. At the top, there is a toolbar with icons for 'Result Grid', 'Filter Rows' (a funnel icon), and 'Edit' (a pencil icon). Below the toolbar is a table with three columns: DEPTNO, DNAME, and LOC. The table contains five rows of data, with the first row (DEPTNO: 10, DNAME: ACCOUNTING, LOC: NEW YORK) highlighted in blue. The last row shows NULL values for all three columns. A small asterisk icon is visible in the first column of the last row.

| | DEPTNO | DNAME | LOC |
|---|--------|------------|----------|
| ▶ | 10 | ACCOUNTING | NEW YORK |
| | 20 | RESEARCH | DALLAS |
| | 30 | SALES | CHICAGO |
| | 40 | OPERATIONS | BOSTON |
| | 50 | OVERSEAS | SEOUL |
| * | NULL | NULL | NULL |

2. 데이터 수정하기

❑ dept 에서 SEOUL 을 BUSAN 으로 고쳐 보자

(1) SEOUL 을 선택한다

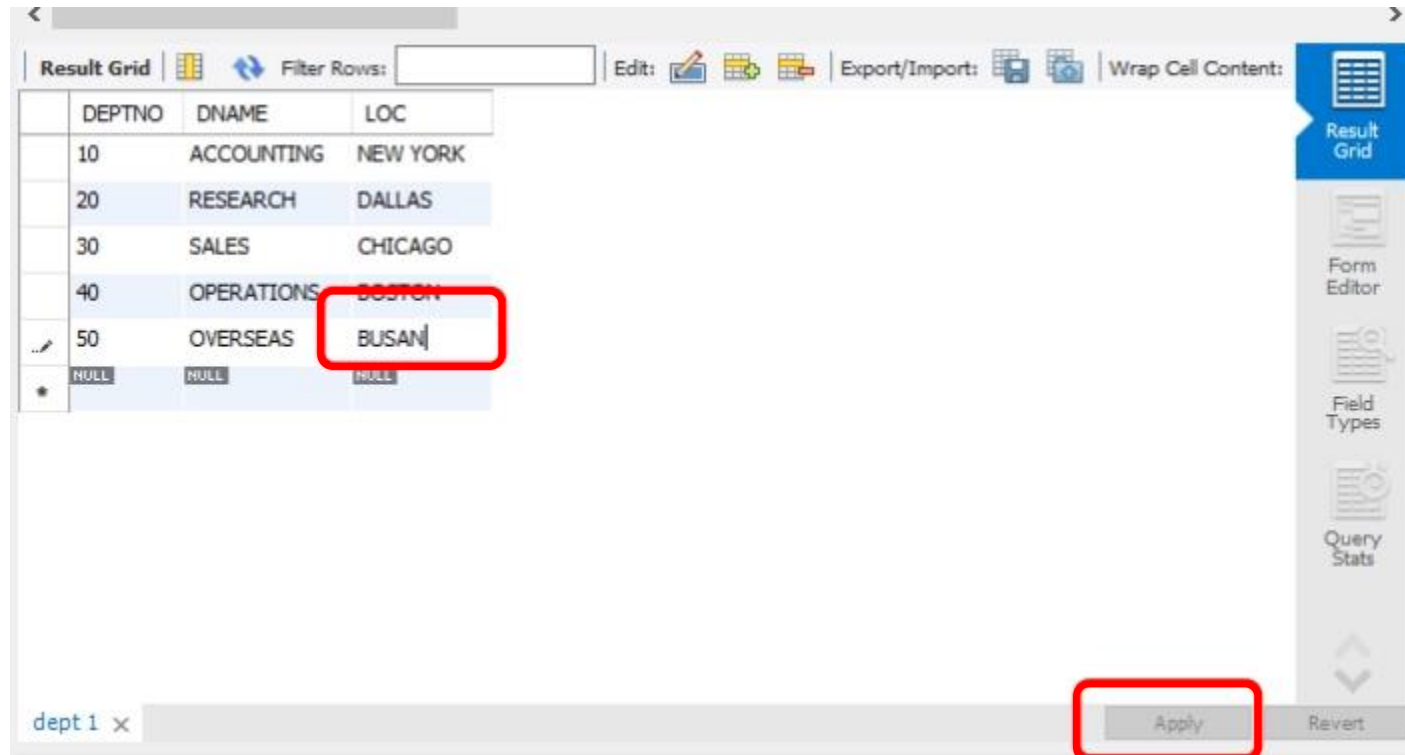


The screenshot shows the MySQL Workbench interface with the 'Result Grid' tab selected. The table 'dept' is displayed with columns DEPTNO, DNAME, and LOC. The row with DEPTNO 50, DNAME OVERSEAS, and LOC SEOUL is highlighted with a red rectangle, indicating it is the selected row for editing.

| DEPTNO | DNAME | LOC |
|--------|------------|----------|
| 10 | ACCOUNTING | NEW YORK |
| 20 | RESEARCH | DALLAS |
| 30 | SALES | CHICAGO |
| 40 | OPERATIONS | BOSTON |
| 50 | OVERSEAS | SEOUL |
| HULL | HULL | |

2. 데이터 수정하기

(2) 데이터를 수정한다

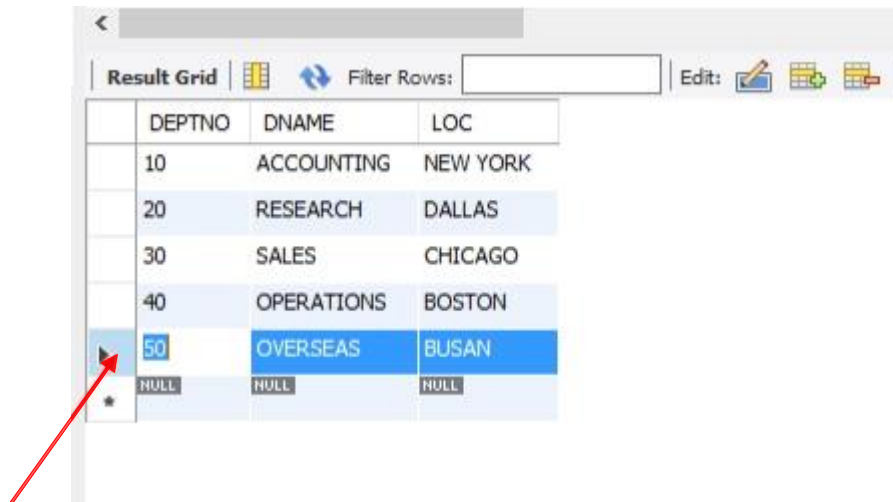


실행버튼클릭

3. 데이터 삭제하기

❑ dept 에서 50번 부서를 삭제한다

(1) 50번부서선택

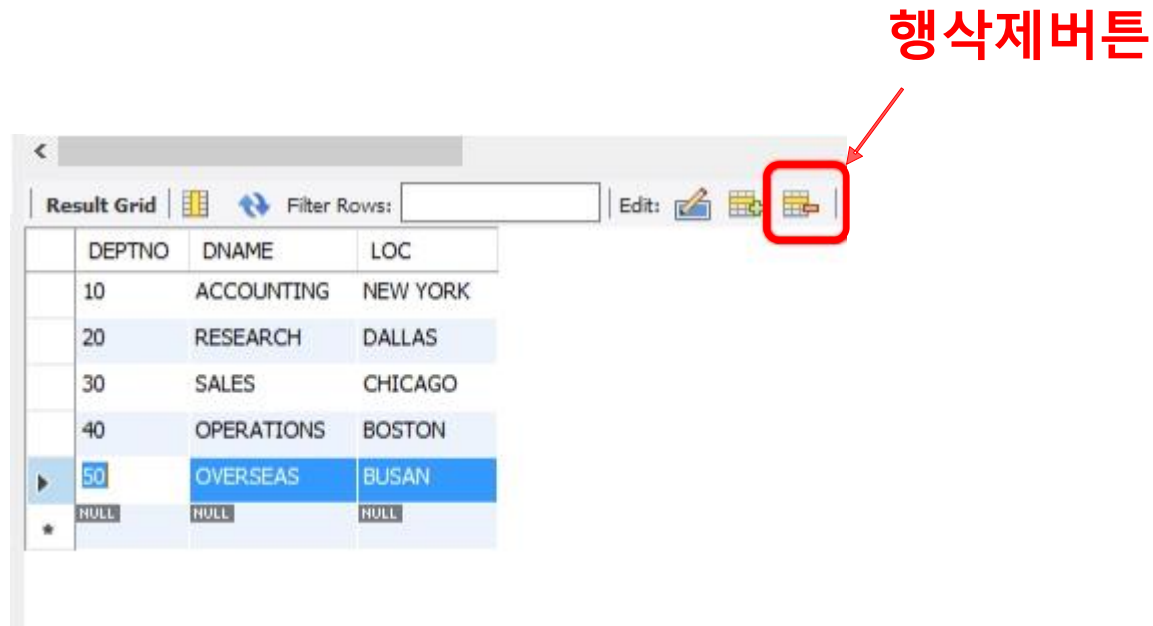


| | DEPTNO | DNAME | LOC |
|--|--------|------------|----------|
| | 10 | ACCOUNTING | NEW YORK |
| | 20 | RESEARCH | DALLAS |
| | 30 | SALES | CHICAGO |
| | 40 | OPERATIONS | BOSTON |
| | 50 | OVERSEAS | BUSAN |
| | NULL | NULL | NULL |

여기클릭하면선택됨

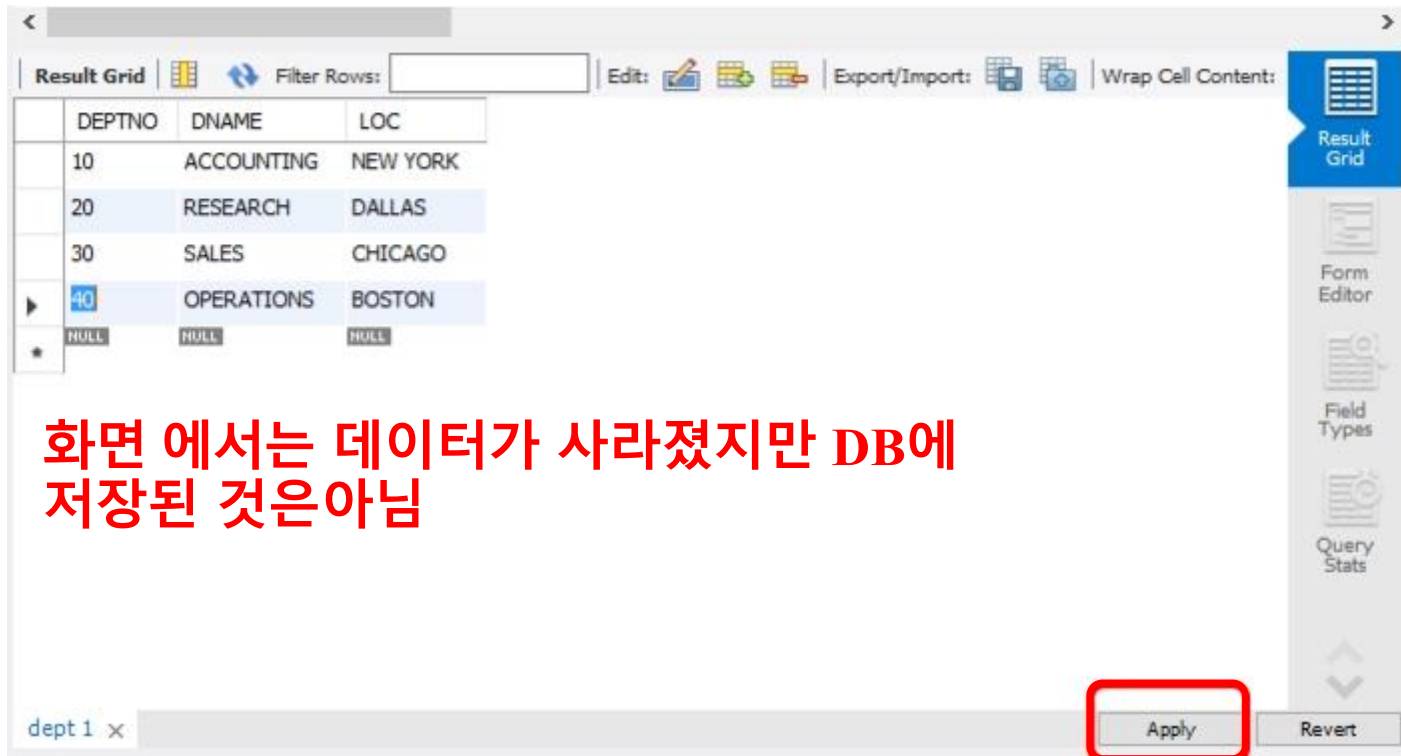
3. 데이터 삭제하기

(2) 행 삭제 버튼 클릭



3. 데이터 삭제하기

(3) [Apply] 버튼 클릭



The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with buttons for 'Result Grid', 'Filter Rows', 'Edit', 'Export/Import', and 'Wrap Cell Content'. Below the toolbar is a table with the following data:

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

Below the table, there's a text input field with 'dept 1' and a close button 'x'. To the right of the table, there's a vertical toolbar with buttons for 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'. At the bottom right, there are two buttons: 'Apply' and 'Revert'. The 'Apply' button is circled in red.

화면 에서는 데이터가 사라졌지만 DB에 저장된 것은아님

4. 정렬하기

❑ emp 테이블을 ename 을 기준으로 정렬해 보자

컬럼 이름을 한번 클릭하면 오름차순,
한번 더 클릭하면 내림차순으로 정렬

→ 저장되는 것은 X
(화면상에서만
보여줄)



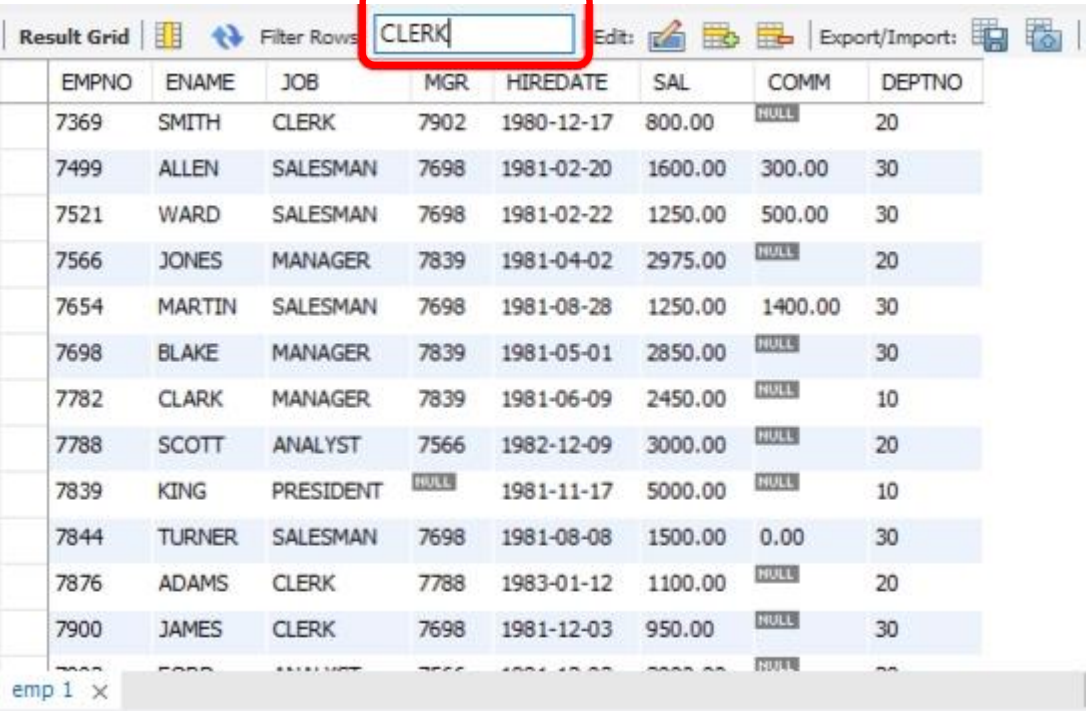
| | EMPNO | ENAME | JOB | MGR | HIREDATE | SAL | COMM | DEPTNO |
|---|-------|--------|-----------|------|------------|---------|---------|--------|
| ▶ | 7369 | SMITH | CLERK | 7902 | 1980-12-17 | 800.00 | HULL | 20 |
| | 7499 | ALLEN | SALESMAN | 7698 | 1981-02-20 | 1600.00 | 300.00 | 30 |
| | 7521 | WARD | SALESMAN | 7698 | 1981-02-22 | 1250.00 | 500.00 | 30 |
| | 7566 | JONES | MANAGER | 7839 | 1981-04-02 | 2975.00 | HULL | 20 |
| | 7654 | MARTIN | SALESMAN | 7698 | 1981-08-28 | 1250.00 | 1400.00 | 30 |
| | 7698 | BLAKE | MANAGER | 7839 | 1981-05-01 | 2850.00 | HULL | 30 |
| | 7782 | CLARK | MANAGER | 7839 | 1981-06-09 | 2450.00 | HULL | 10 |
| | 7788 | SCOTT | ANALYST | 7566 | 1982-12-09 | 3000.00 | HULL | 20 |
| | 7839 | KING | PRESIDENT | HULL | 1981-11-17 | 5000.00 | HULL | 10 |
| | 7844 | TURNER | SALESMAN | 7698 | 1981-08-08 | 1500.00 | 0.00 | 30 |
| | 7876 | ADAMS | CLERK | 7788 | 1983-01-12 | 1100.00 | HULL | 20 |
| | 7900 | JAMES | CLERK | 7698 | 1981-12-03 | 950.00 | HULL | 30 |
| | 7902 | WATSON | CLERK | 7698 | 1981-12-03 | 850.00 | HULL | 30 |

정렬된 결과가 DB에 저장되는 것은 아님

5. 키워드로 결과행(row) 필터링

❑ 담당업무가 CLERK 인 행만 보이시오

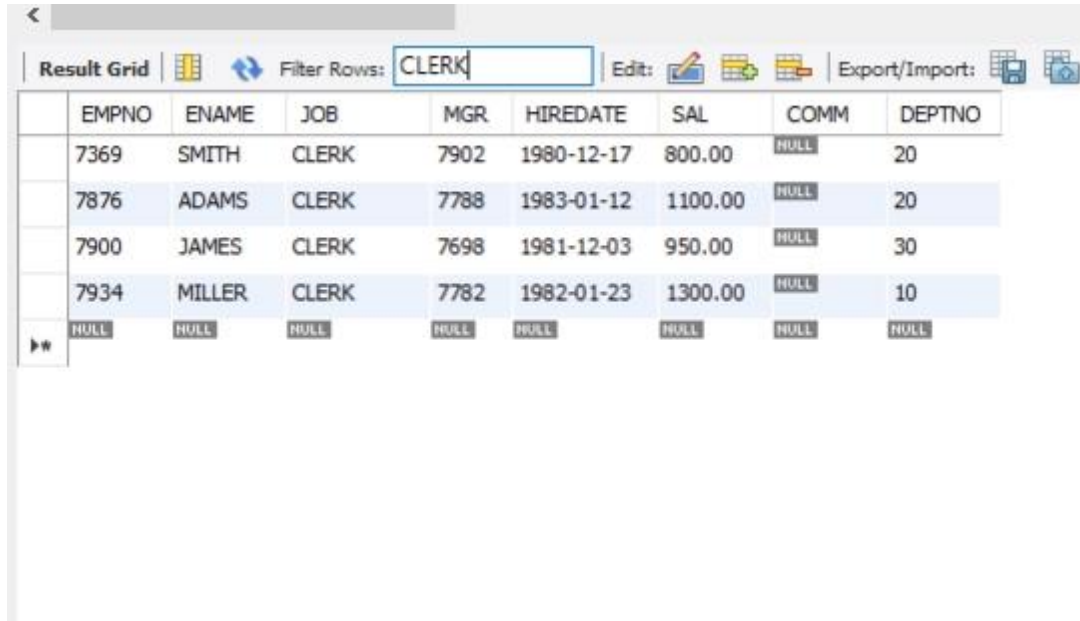
여기에 검색을 원하는 단어를 입력 후 <ENTER>



The screenshot shows the MySQL Workbench interface. At the top, the 'Result Grid' tab is active. A search bar labeled 'Filter Rows' contains the text 'CLERK', which is highlighted with a red rectangle. Below the search bar, a table of employee data is displayed. The table has columns: EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, and DEPTNO. The rows are filtered to show only those where the JOB is 'CLERK'. The rows shown are: 7369 SMITH CLERK 7902 1980-12-17 800.00 NULL 20, 7499 ALLEN SALESMAN 7698 1981-02-20 1600.00 300.00 30, 7521 WARD SALESMAN 7698 1981-02-22 1250.00 500.00 30, 7566 JONES MANAGER 7839 1981-04-02 2975.00 NULL 20, 7654 MARTIN SALESMAN 7698 1981-08-28 1250.00 1400.00 30, 7698 BLAKE MANAGER 7839 1981-05-01 2850.00 NULL 30, 7782 CLARK MANAGER 7839 1981-06-09 2450.00 NULL 10, 7788 SCOTT ANALYST 7566 1982-12-09 3000.00 NULL 20, 7839 KING PRESIDENT NULL 1981-11-17 5000.00 NULL 10, 7844 TURNER SALESMAN 7698 1981-08-08 1500.00 0.00 30, 7876 ADAMS CLERK 7788 1983-01-12 1100.00 NULL 20, 7900 JAMES CLERK 7698 1981-12-03 950.00 NULL 30. The bottom of the window shows a tab labeled 'emp 1'.

| EMPNO | ENAME | JOB | MGR | HIREDATE | SAL | COMM | DEPTNO |
|-------|--------|-----------|------|------------|---------|---------|--------|
| 7369 | SMITH | CLERK | 7902 | 1980-12-17 | 800.00 | NULL | 20 |
| 7499 | ALLEN | SALESMAN | 7698 | 1981-02-20 | 1600.00 | 300.00 | 30 |
| 7521 | WARD | SALESMAN | 7698 | 1981-02-22 | 1250.00 | 500.00 | 30 |
| 7566 | JONES | MANAGER | 7839 | 1981-04-02 | 2975.00 | NULL | 20 |
| 7654 | MARTIN | SALESMAN | 7698 | 1981-08-28 | 1250.00 | 1400.00 | 30 |
| 7698 | BLAKE | MANAGER | 7839 | 1981-05-01 | 2850.00 | NULL | 30 |
| 7782 | CLARK | MANAGER | 7839 | 1981-06-09 | 2450.00 | NULL | 10 |
| 7788 | SCOTT | ANALYST | 7566 | 1982-12-09 | 3000.00 | NULL | 20 |
| 7839 | KING | PRESIDENT | NULL | 1981-11-17 | 5000.00 | NULL | 10 |
| 7844 | TURNER | SALESMAN | 7698 | 1981-08-08 | 1500.00 | 0.00 | 30 |
| 7876 | ADAMS | CLERK | 7788 | 1983-01-12 | 1100.00 | NULL | 20 |
| 7900 | JAMES | CLERK | 7698 | 1981-12-03 | 950.00 | NULL | 30 |

5. 키워드로 결과행(row) 필터링



The screenshot shows the MySQL Workbench Result Grid interface. The 'Filter Rows' field is set to 'CLERK'. The table displays the following data:

| | EMPNO | ENAME | JOB | MGR | HIREDATE | SAL | COMM | DEPTNO |
|----|-------|--------|-------|------|------------|---------|------|--------|
| | 7369 | SMITH | CLERK | 7902 | 1980-12-17 | 800.00 | HULL | 20 |
| | 7876 | ADAMS | CLERK | 7788 | 1983-01-12 | 1100.00 | HULL | 20 |
| | 7900 | JAMES | CLERK | 7698 | 1981-12-03 | 950.00 | HULL | 30 |
| | 7934 | MILLER | CLERK | 7782 | 1982-01-23 | 1300.00 | HULL | 10 |
| ▶▶ | HULL | HULL | HULL | HULL | HULL | HULL | HULL | HULL |

전체데이터를 다시보려면 입력한 키워드를 지운후 <ENTER>
또는  클릭

6. 데이터 내보내기

- emp 테이블에서 급여가 1500 이상인 직원들의 데이터만 조회하여 csv 파일에 저장해 보자

(1) SQL 문을 실행하여 급여가 1500 이상인 행들만 선택한다

```
1 SELECT *
2 FROM emp
3 WHERE sal > 1500 ;
4
```

내보내기
— 불러오기

| Result Grid | | | | | | | | |
|-----------------------------------|-------|-------|-----------|------|------------|---------|--------|--------|
| Filter Rows: <input type="text"/> | | | | | | | | |
| Edit: | | | | | | | | |
| Export/Import: | | | | | | | | |
| | EMPNO | ENAME | JOB | MGR | HIREDATE | SAL | COMM | DEPTNO |
| ▶ | 7499 | ALLEN | SALESMAN | 7698 | 1981-02-20 | 1600.00 | 300.00 | 30 |
| | 7566 | JONES | MANAGER | 7839 | 1981-04-02 | 2975.00 | NULL | 20 |
| | 7698 | BLAKE | MANAGER | 7839 | 1981-05-01 | 2850.00 | NULL | 30 |
| | 7782 | CLARK | MANAGER | 7839 | 1981-06-09 | 2450.00 | NULL | 10 |
| | 7788 | SCOTT | ANALYST | 7566 | 1982-12-09 | 3000.00 | NULL | 20 |
| | 7839 | KING | PRESIDENT | NULL | 1981-11-17 | 5000.00 | NULL | 10 |
| | 7902 | FORD | ANALYST | 7566 | 1981-12-03 | 3000.00 | NULL | 20 |
| * | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

6. 데이터 내보내기

(2) 데이터 내보내기 아이콘 클릭

The screenshot shows the MySQL Workbench interface. The 'Result Grid' displays a table of employee data. The 'Export Resultset' dialog box is open, showing a file explorer view. The 'Save as type' is set to 'CSV (*.csv)'. The 'Save' button is highlighted with a red box.

| EMPNO | ENAME | JOB | MGR | HIREDATE | SAL | COMM | DEPTNO |
|-------|-------|-----------|------|------------|---------|--------|--------|
| 7499 | ALLEN | SALESMAN | 7698 | 1981-02-20 | 1600.00 | 300.00 | 30 |
| 7566 | JONES | MANAGER | 7839 | 1981-04-02 | 2975.00 | NULL | 20 |
| 7698 | BLAKE | MANAGER | 7839 | 1981-05-01 | 2850.00 | NULL | 30 |
| 7782 | CLARK | MANAGER | 7839 | 1981-06-09 | 2450.00 | NULL | 10 |
| 7788 | SCOTT | ANALYST | 7566 | 1982-12-09 | 3000.00 | NULL | 20 |
| 7839 | KING | PRESIDENT | NULL | 1981-11-17 | 5000.00 | NULL | 10 |
| 7902 | FORD | ANALYST | 7566 | 1981-12-03 | 3000.00 | NULL | 10 |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

Export Resultset

구분: 새 폴더

이름: Donikook ICT, main, fax, BNM, manda.jpg, netBeansProjects, eCam, Processing, R, Visual Studio 2015, 사용자 지정 Office 서식 파일, 스캔 문서, 카카오톡 받은 파일, Book1.csv, Book2.csv

수정된 날짜: 2016-11-15 오후..., 2016-09-22 오후..., 2016-03-04 오전..., 2016-03-16 오후..., 2016-12-05 오후..., 2015-11-25 오후..., 2015-08-27 오후..., 2015-08-25 오후..., 2017-03-02 오후..., 2016-07-21 오후..., 2017-03-08 오후..., 2017-03-25 오후..., 2016-05-19 오전..., 2016-05-19 오전...

파일 형식: CSV (*.csv)

저장(S) 취소

6. 데이터 내보내기

| | A | B | C | D | E | F | G | H | I |
|----|-------|-------|-----------|------|------------|------|------|--------|---|
| 1 | EMPNO | ENAME | JOB | MGR | HIREDATE | SAL | COMM | DEPTNO | |
| 2 | 7499 | ALLEN | SALESMAN | 7698 | 1981-02-20 | 1600 | 300 | 30 | |
| 3 | 7566 | JONES | MANAGER | 7839 | 1981-04-02 | 2975 | NULL | 20 | |
| 4 | 7698 | BLAKE | MANAGER | 7839 | 1981-05-01 | 2850 | NULL | 30 | |
| 5 | 7782 | CLARK | MANAGER | 7839 | 1981-06-09 | 2450 | NULL | 10 | |
| 6 | 7788 | SCOTT | ANALYST | 7566 | 1982-12-09 | 3000 | NULL | 20 | |
| 7 | 7839 | KING | PRESIDENT | NULL | 1981-11-17 | 5000 | NULL | 10 | |
| 8 | 7902 | FORD | ANALYST | 7566 | 1981-12-03 | 3000 | NULL | 20 | |
| 9 | | | | | | | | | |
| 10 | | | | | | | | | |

*.csv 의 형태로 파일이 저장된다

7. 데이터 불러오기

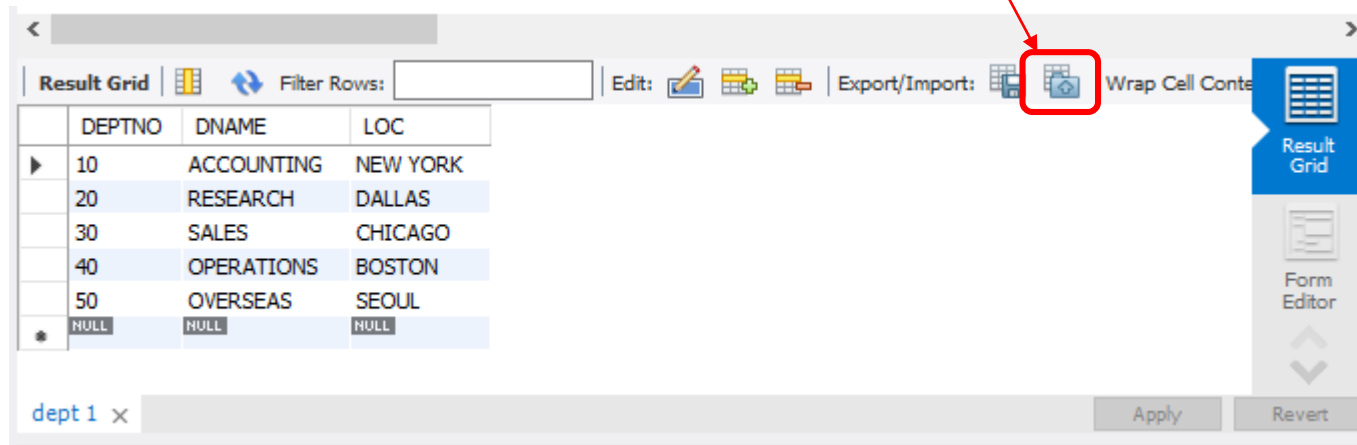
- ❑ .csv 파일의 데이터를 읽어 테이블에 저장한다
 - .csv 파일의 구조와 테이블의 구조가 일치해야 함

(1) 다음과 같은 내용의 new_dept.csv 파일을 작성한다

| | A | B | C | D |
|---|--------|-------------|----------|---|
| 1 | deptno | dname | location | |
| 2 | 80 | Develop | INCHON | |
| 3 | 85 | Manufacture | DAEGU | |
| 4 | 90 | Sales 2 | JEJU | |
| 5 | | | | |
| 6 | | | | |

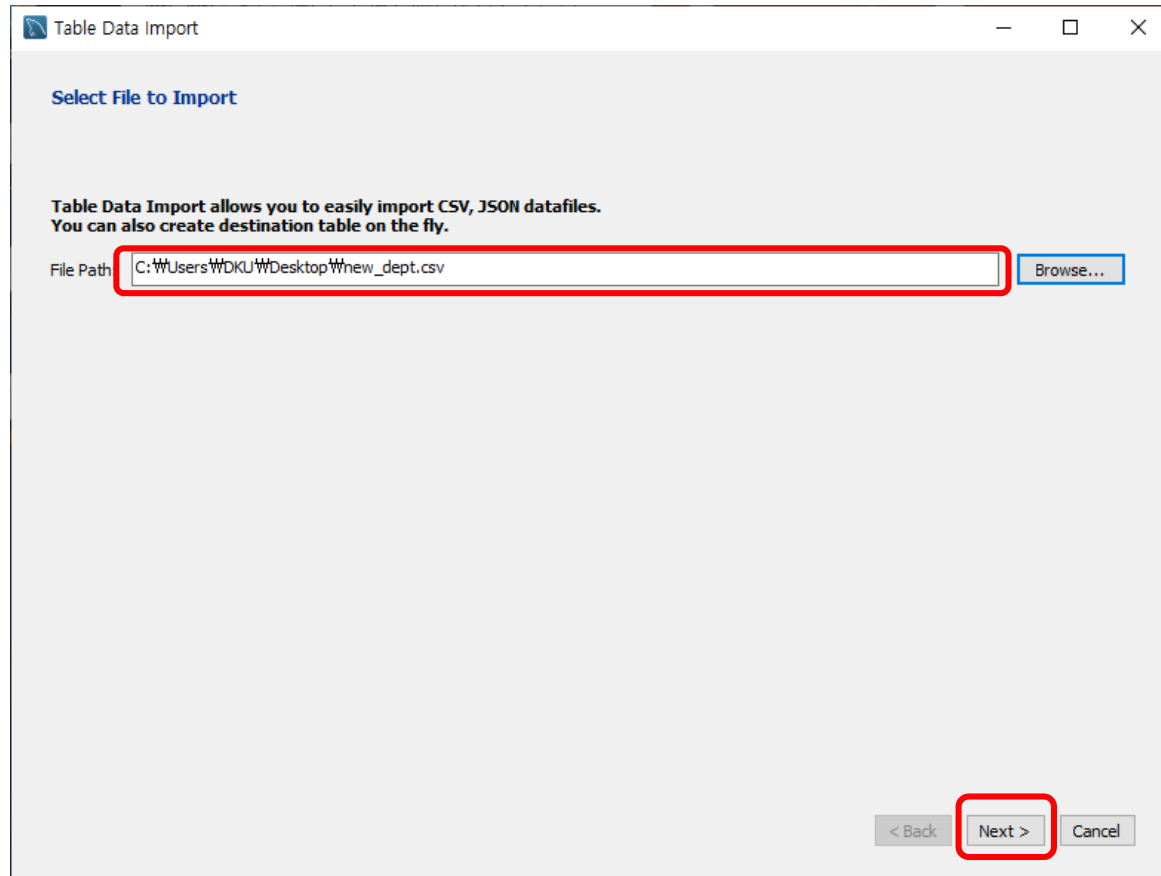
7. 데이터 불러오기

(2) dept 테이블 내용 화면에서 import 클릭



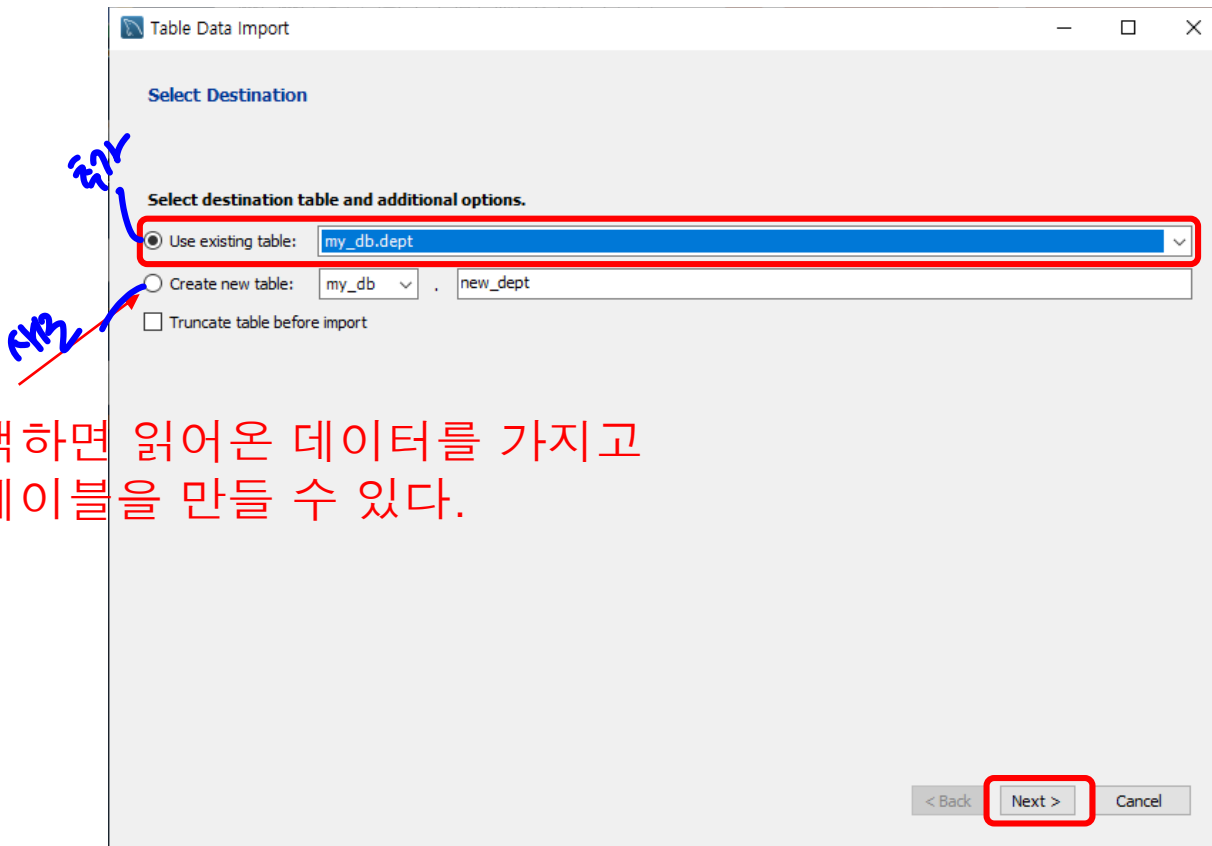
7. 데이터 불러오기

(3) new_dept.csv 파일을 불러온다



7. 데이터 불러오기

(4) 데이터를 저장할 dept 테이블을 지정



7. 데이터 불러오기

(5) .csv 파일의 컬럼과 dept 테이블의 컬럼을 일치시킨다.

Table Data Import

Configure Import Settings

Detected file format: csv

Encoding: utf-8

Columns:

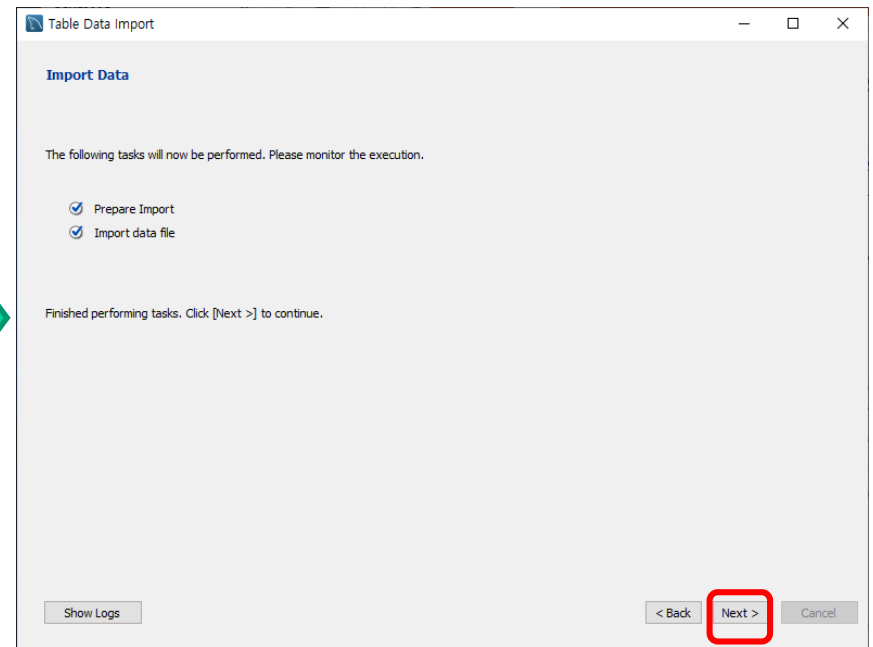
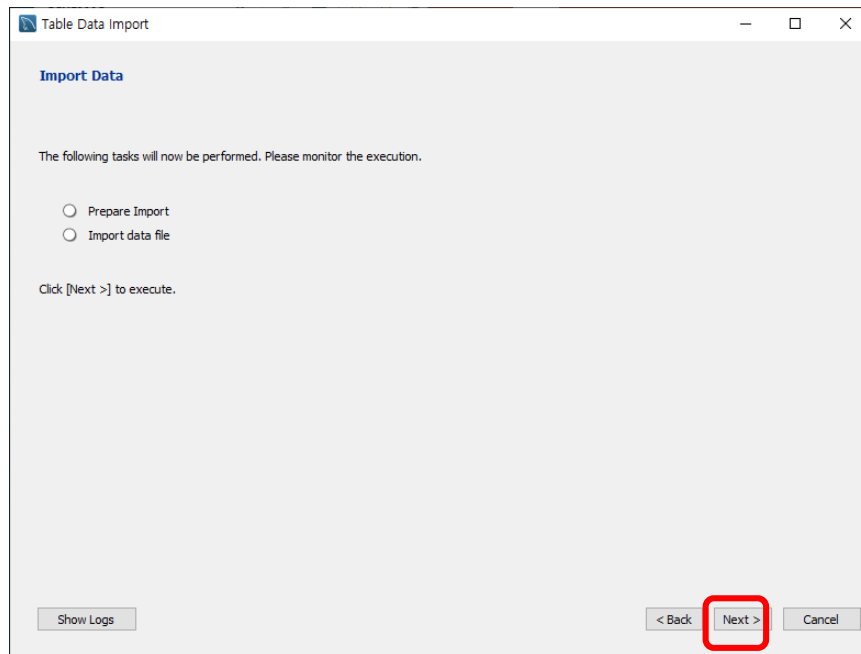
| Source Column | Dest Column |
|---------------|-------------|
| deptno | DEPTNO |
| dname | DNAME |
| location | LOC |

csv dept

| deptno | dname | location |
|--------|-------------|----------|
| 80 | Develop | INCHON |
| 85 | Manufacture | DAEGU |
| 90 | Sales 2 | JEJU |

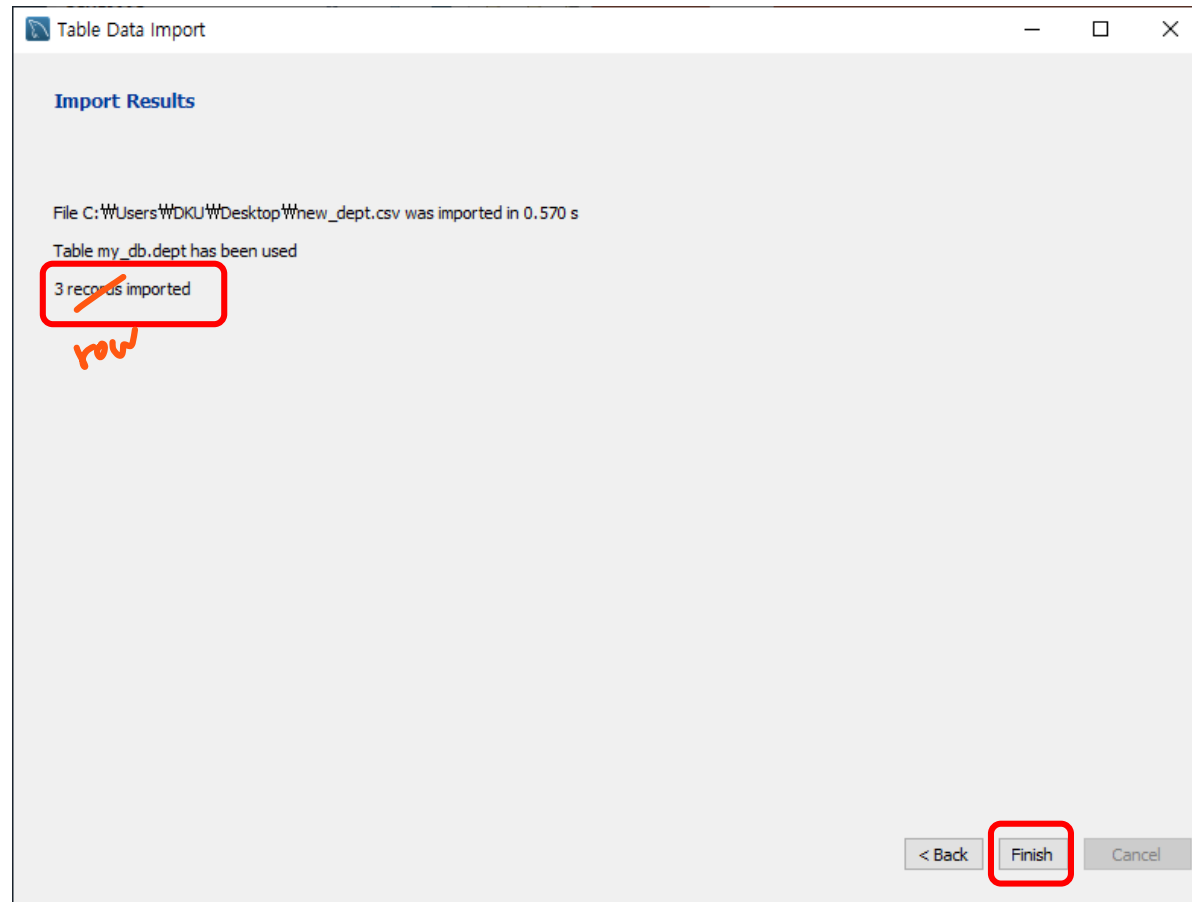
< Back Next > Cancel

7. 데이터 불러오기



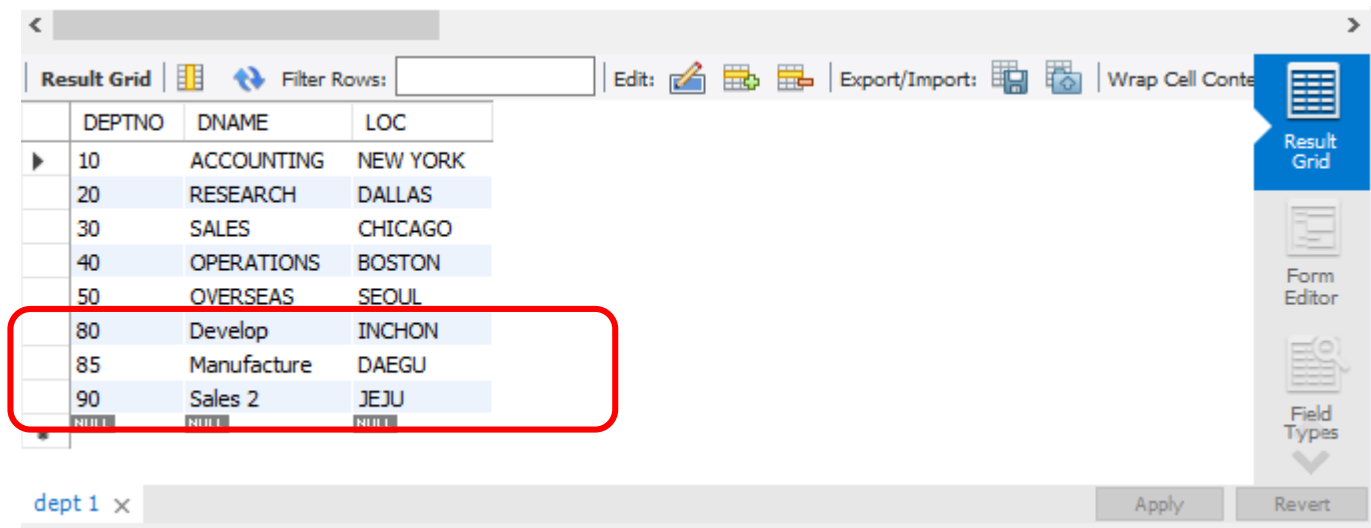
7. 데이터 불러오기

(6) 작업이 완료됨



7. 데이터 불러오기

- dept 테이블의 내용을 새로 불러오면 추가된 내용을 확인할 수 있음



The screenshot shows the MySQL Workbench interface with the 'dept' table selected. The table data is displayed in a grid view. The columns are DEPTNO, DNAME, and LOC. The data rows are as follows:

| DEPTNO | DNAME | LOC |
|--------|-------------|----------|
| 10 | ACCOUNTING | NEW YORK |
| 20 | RESEARCH | DALLAS |
| 30 | SALES | CHICAGO |
| 40 | OPERATIONS | BOSTON |
| 50 | OVERSEAS | SEOUL |
| 80 | Develop | INCHON |
| 85 | Manufacture | DAEGU |
| 90 | Sales 2 | JEJU |
| NULL | NULL | NULL |

The last three rows (80, 85, 90) are highlighted with a red box, indicating the newly added data. The interface also shows a toolbar with options like 'Result Grid', 'Filter Rows', 'Edit', 'Export/Import', and 'Wrap Cell Content'. The bottom status bar shows 'dept 1' and 'Apply'/'Revert' buttons.

[과제 1]

1. dept 테이블에 다음과 같이 2개의 새로운 부서를 입력하시오

| | | |
|----|--------------|--------|
| 60 | EUROPEBRANCH | LONDON |
| 70 | ASIABRANCH | SEOUL |

2. dept 테이블에서 SEOUL 을 INCHON 으로 바꾸시오

3. dept 테이블에서 60, 70번 부서를 삭제하시오

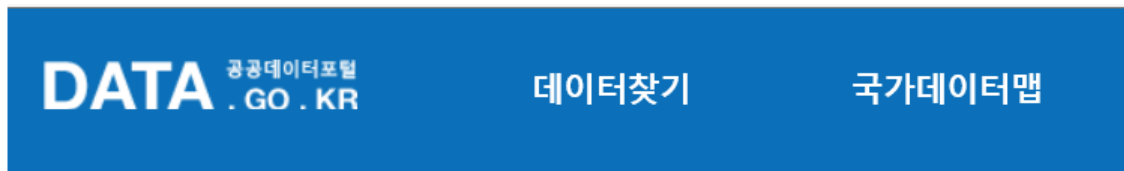
4. emp 테이블에서 부서번호가 10인 행들의 정보만 보이시오

5. emp 테이블에서 입사일자 역순으로 행들을 보이시오

6. world 데이터베이스의 city 테이블에서 contryCode 가 EGY 인 행들만 선택하여 egypt_city.csv 파일에 저장하시오

[과제 1]

- ❑ 7. 서울시 학교 현황 정보를 다운 받아 **중학교** 현황을 추출하여 my_db 에 middle 테이블로 저장하고 내용을 보이시오
- ❑ <https://www.data.go.kr/dataset/3070110/fileData.do>



파일데이터 (1건)

서울특별시교육청_서울시 학교별 현황 정보

← 학교별 기본 현황 자료(학급수, 학생수, 교원수) 제공

수정일 2020-07-09 조회 3209

여기 클릭

데이터셋



URL 복사

서울특별시교육청_서울시 학교별 현황 정보

학교별 기본 현황 자료(학급수, 학생수, 교원수) 제공



0



0

관심

바로가기

오류신고 및

담당자 문의

※ 다운로드 및 활용신청은 PC버전에서만 가능합니다.

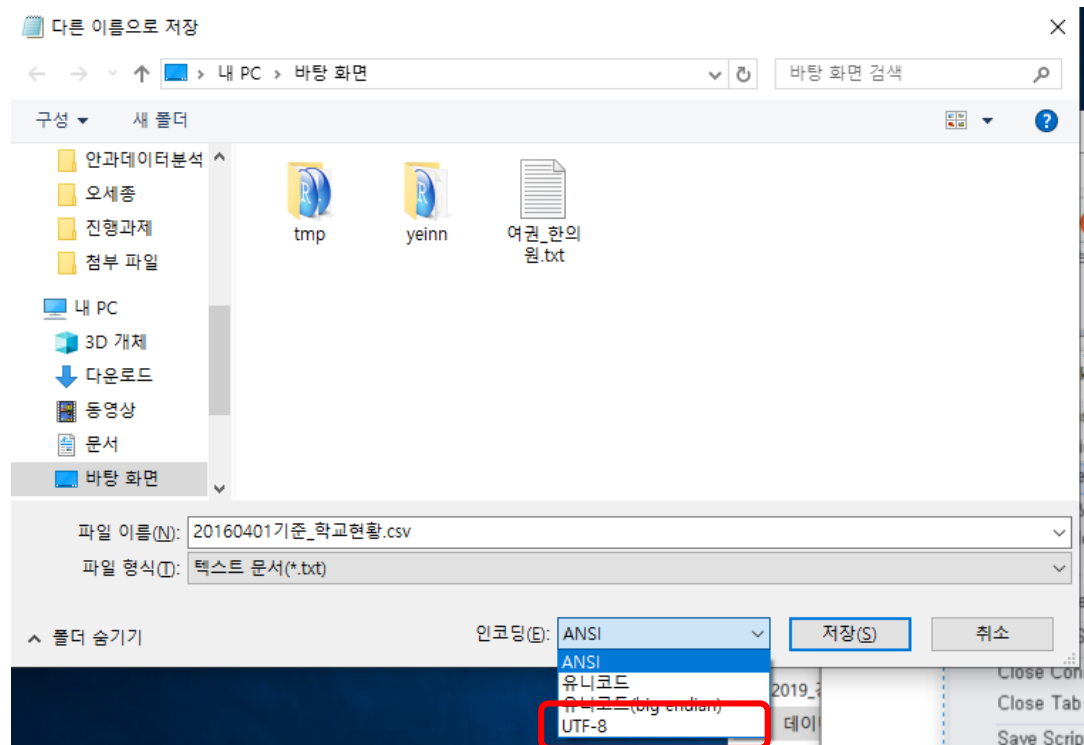
여기 클릭

한줄로 컬럼이름을 붙이되
영문으로 작성한다

위 두줄은 삭제

| 2016. 4. 1. 기준 중학교 현황 | | | | | | | | | | | | | | | |
|-----------------------|-----|------|----------------|------|-------|---|---------------|-----|-----|----------|---|-----|-----|-----|-----|
| 2016. 4. 1. 기준 중학교 현황 | | | | | | | | | | | | | | | |
| 교육청 | 소재구 | 학교명 | 남녀 | 우편번호 | 주소 | 전화번호 | 학급수 | | | 특수 학급 | 계 | 1학년 | | | |
| | | | | | | | 1학년 | 2학년 | 3학년 | | | 계 | 여 | 계 | |
| 6 | 동부 | 동대문구 | 동대문중학교 | 공학 | 02553 | 서울특별시 중대문구 사가정로 89, 동대문중학교 (전농동) | 02-3407-4103 | 8 | 8 | 9 | 2 | 27 | 230 | 120 | 246 |
| 7 | 동부 | 중랑구 | 동원중학교 | 공학 | 02062 | 서울특별시 중랑구 양천역로 32 (방우동, 동원중학교) | 02-493-8129 | 5 | 5 | 5 | - | 15 | 117 | 35 | 95 |
| 8 | 동부 | 중랑구 | 면목중학교 | 공학 | 02193 | 서울특별시 중랑구 용마산로70길 37, (면목동) (면목동) | 02-437-0278 | 9 | 7 | 9 | - | 25 | 236 | 97 | 205 |
| 9 | 동부 | 중랑구 | 봉화중학교 | 공학 | 02179 | 서울특별시 중랑구 방우로70길 89 (방우동, 봉화중학교) | 02-435-8348 | 6 | 5 | 5 | 2 | 18 | 145 | 47 | 102 |
| 10 | 동부 | 중랑구 | 상봉중학교 | 공학 | 02081 | 서울특별시 중랑구 상봉중앙로8나길 28 (상봉동) | 02-436-4452 | 9 | 9 | 10 | 1 | 29 | 258 | 127 | 268 |
| 11 | 동부 | 동대문구 | 성일중학교 | 공학 | 02575 | 서울특별시 동대문구 왕산로23길 55 (제기동) | 02-926-0211 | 5 | 5 | 5 | 1 | 16 | 114 | 27 | 109 |
| 12 | 동부 | 동대문구 | 송인중학교 | 공학 | 02592 | 서울특별시 중대문구 천호대로47길 34, 송인중학교 (당신리동) | 02-2215-0176 | 6 | 5 | 5 | 2 | 18 | 143 | 63 | 127 |
| 13 | 동부 | 중랑구 | 신현중학교 | 공학 | 02075 | 서울특별시 중랑구 봉화산로 162, 신현중학교 (상봉동) | 02-3421-0631 | 9 | 9 | 9 | 2 | 29 | 257 | 137 | 257 |
| 14 | 동부 | 중랑구 | 용마중학교 | 공학 | 02254 | 서울특별시 중랑구 연곡로28길 18, 용마중학교 (연곡동) | 02-439-4841 | 8 | 7 | 8 | 2 | 25 | 225 | 99 | 196 |
| 15 | 동부 | 중랑구 | 원목중학교 | 공학 | 02022 | 서울특별시 중랑구 신내로29길 57, 원목중학교 (목동) | 02-2094-9800 | 11 | 10 | 12 | - | 33 | 339 | 164 | 312 |
| 16 | 동부 | 중랑구 | 장안중학교 | 공학 | 02088 | 서울특별시 중랑구 동일로 780 (중화동, 장안중학교) | 02-6930-3300 | 7 | 7 | 7 | 3 | 24 | 163 | 56 | 205 |
| 17 | 동부 | 동대문구 | 장평중학교 | 공학 | 02640 | 서울특별시 동대문구 답십리로72길 158 (장안동) | 02-2243-8280 | 9 | 9 | 9 | 1 | 28 | 274 | 147 | 257 |
| 18 | 동부 | 동대문구 | 전농중학교 | 공학 | 02506 | 서울특별시 중대문구 사가정로18길 71, 전농중학교 (전농동) | 02-2242-6652 | 7 | 6 | 7 | 1 | 21 | 194 | 90 | 159 |
| 19 | 동부 | 동대문구 | 전동중학교 | 공학 | 02511 | 서울특별시 중대문구 갈라향로로 291, 전동중학교 (위장동) | 02-2212-9265 | 6 | 6 | 6 | 2 | 20 | 162 | 59 | 158 |
| 20 | 동부 | 동대문구 | 전일중학교 | 공학 | 02506 | 서울특별시 중대문구 사가정로18길 67 (전농동, 전일중학교) | 02-2245-2745 | 6 | 6 | 7 | 1 | 20 | 175 | 80 | 171 |
| 21 | 동부 | 중랑구 | 중랑중학교 | 공학 | 02047 | 서울특별시 중랑구 동일로140길 33 (중화동) | 02-437-5311 | 6 | 6 | 7 | 1 | 20 | 170 | 73 | 164 |
| 22 | 동부 | 중랑구 | 중화중학교 | 공학 | 02193 | 서울특별시 중랑구 용마산로 384, (면목3.8동 88번지) (면목동) | 02-439-5811 | 9 | 7 | 9 | 3 | 28 | 237 | 103 | 218 |
| 23 | 동부 | 동대문구 | 청량중학교 | 공학 | 02461 | 서울특별시 중대문구 왕산로 301, 청량중학교 (합정동) | 02-966-4284 | 7 | 7 | 7 | 2 | 23 | 177 | 90 | 170 |
| 24 | 동부 | 중랑구 | 태릉중학교 | 공학 | 02040 | 서울특별시 중랑구 공릉로2길 60-25, 태릉중학교 (목동) | 02-977-4028 | 6 | 6 | 6 | 1 | 19 | 161 | 68 | 132 |
| 25 | 동부 | 동대문구 | 회경중학교 | 공학 | 02498 | 서울특별시 중대문구 방우로184길 20 (위장동, 회경중학교) | 02-2244-1359 | 6 | 5 | 6 | 1 | 18 | 141 | 45 | 145 |
| 26 | 동부 | 동대문구 | 경희여자중학교 | 여자 | 02447 | 서울특별시 중대문구 갈라대로 26, 경희여자중학교 (회기동) | 02-8295-5232 | 6 | 6 | 7 | - | 19 | 173 | 173 | 183 |
| 27 | 동부 | 동대문구 | 경희중학교 | 남자 | 02447 | 서울특별시 중대문구 경희대로 26, 경희중학교 (회기동) | 02-966-6402 | 7 | 7 | 7 | - | 21 | 187 | - | 179 |
| 28 | 동부 | 동대문구 | 대방중학교 | 공학 | 02581 | 서울특별시 동대문구 안암로 6 (신설동) 대방중학교 | 02-940-2257 | 6 | 7 | 7 | - | 20 | 123 | 46 | 178 |
| 29 | 동부 | 동대문구 | 동국대학교사범대학부속중학교 | 남자 | 02520 | 서울특별시 동대문구 장안벽꽃로 201 (장안동) | 02-6716-1701 | 7 | 7 | 8 | - | 22 | 149 | - | 168 |
| 30 | 동부 | 중랑구 | 송곡여자중학교 | 여자 | 02057 | 서울특별시 중랑구 양원역로 67 (망우본동) | 070-7124-3670 | 3 | 4 | 4 | - | 11 | 60 | 60 | 72 |
| 31 | 동부 | 중랑구 | 영란여자중학교 | 여자 | 02061 | 서울특별시 중랑구 망우로73길 56 (망우동) | 02-2209-0143 | 7 | 7 | 7 | - | 21 | 128 | 128 | 138 |
| 32 | 동부 | 동대문구 | 정희여자중학교 | 여자 | 02480 | 서울특별시 동대문구 홍릉로15길 50 (제기동) | 02-967-0178 | 4 | 4 | 4 | - | 12 | 65 | 65 | 70 |
| 33 | 동부 | 중랑구 | 혜원여자중학교 | 여자 | 02142 | 서울특별시 중랑구 봉우재로 58길 39 (망우동) | 070-8688-2704 | 7 | 7 | 7 | - | 21 | 166 | 166 | 213 |
| 34 | 동부 | 동대문구 | 회경여자중학교 | 여자 | 02502 | 서울특별시 중대문구 천호대로47길 34, 회경여자중학교 (당신리동) | 02-2215-0176 | 7 | 7 | 7 | - | 21 | 166 | 166 | 176 |

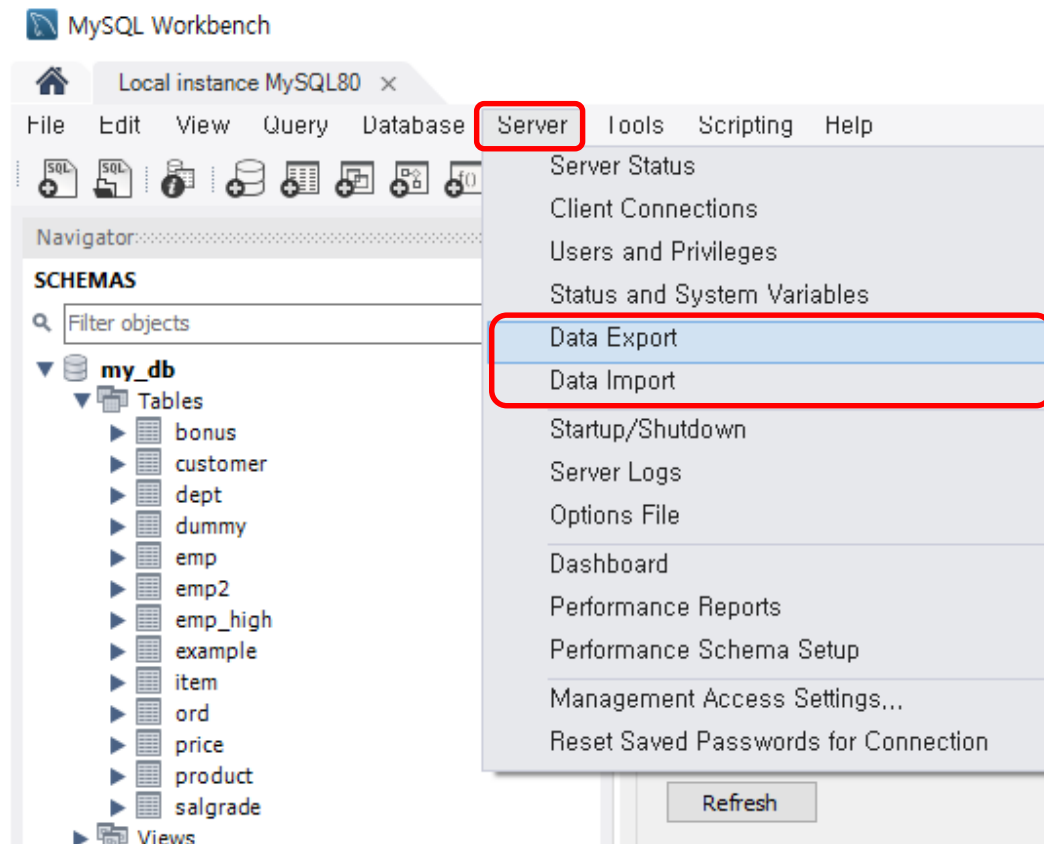
- ❑ 중학교정보만 csv 파일에 저장을 한다.
- ❑ 한글이 포함되어 있는데 mysql 과 한글코드가 다르므로 csv 파일을 메모장에서 읽은뒤 아래와 같이 인코딩을 utf-8 포맷으로 저장한다.



Backup & Recovery

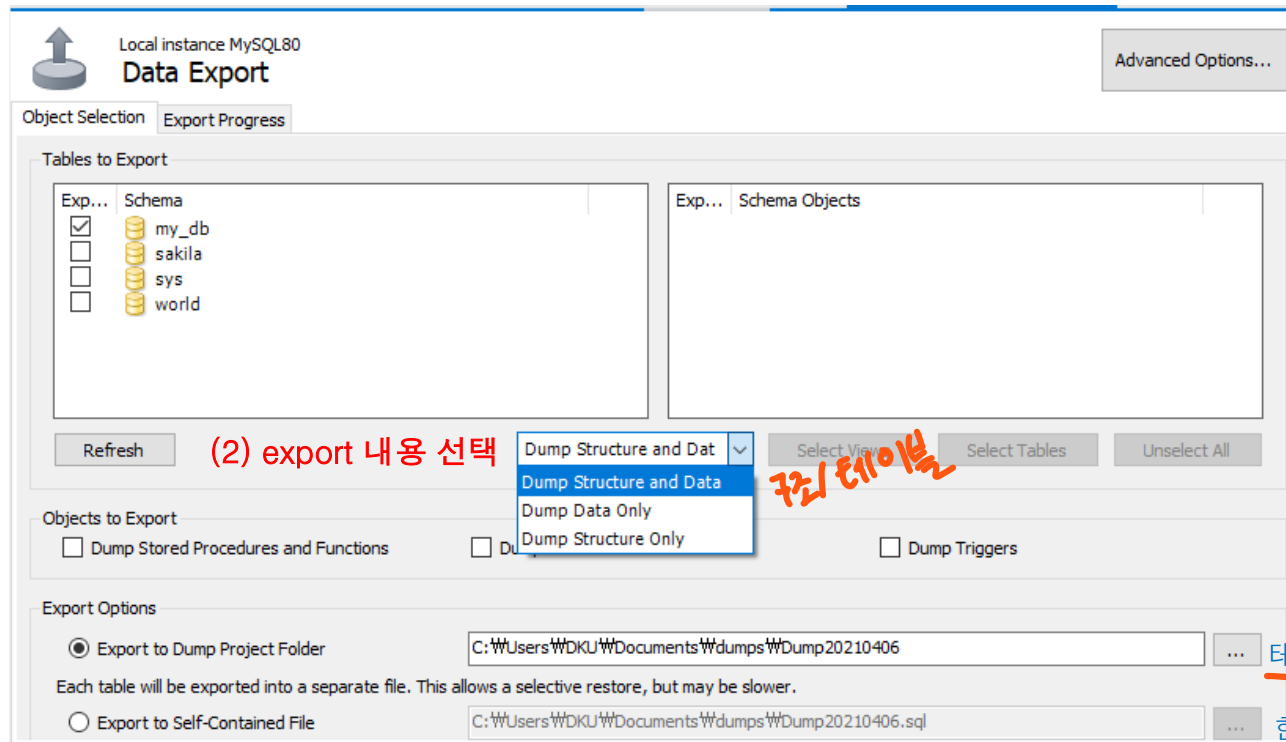
7.3 DB export/import

❑ Workbench 메뉴이용 export



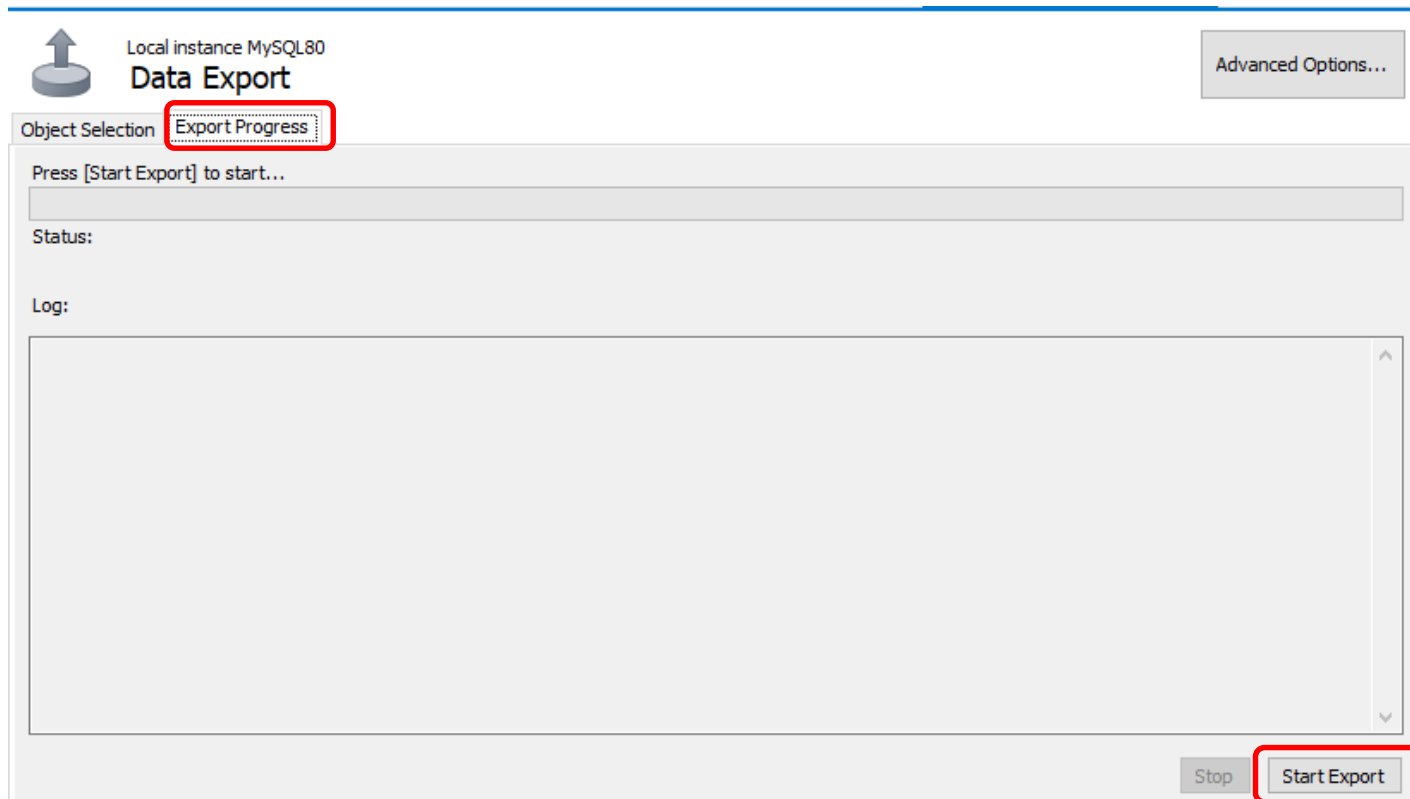
7.3 DB export/import

❑ Workbench 메뉴이용 export



7.3 DB export/import

❑ Workbench 메뉴이용 export

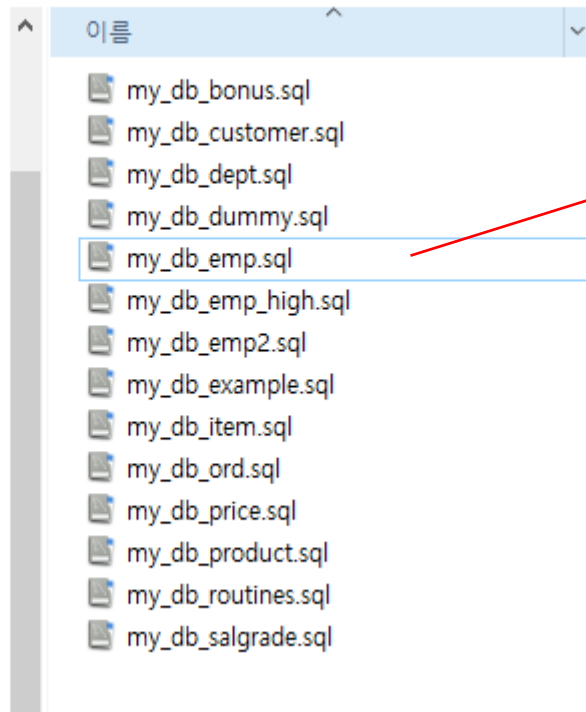


7.3 DB export/import

❑ Workbench 메뉴이용 export

Export 결과

내 PC > 문서 > dumps > Dump20210406



```
my_db_emp.sql - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
-- Table structure for table `emp`
--
DROP TABLE IF EXISTS `emp`;
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `emp` (
  `EMPNO` decimal(4,0) NOT NULL,
  `ENAME` varchar(10) DEFAULT NULL,
  `JOB` varchar(9) DEFAULT NULL,
  `MGR` decimal(4,0) DEFAULT NULL,
  `HIREDATE` date DEFAULT NULL,
  `SAL` decimal(7,2) DEFAULT NULL,
  `COMM` decimal(7,2) DEFAULT NULL,
  `DEPTNO` decimal(2,0) NOT NULL,
  PRIMARY KEY (`EMPNO`),
  KEY `EMP_SELF_KEY` (`MGR`),
  KEY `EMP_FOREIGN_KEY` (`DEPTNO`),
  CONSTRAINT `EMP_FOREIGN_KEY` FOREIGN KEY (`DEPTNO`) REFERENCES `dept` (`DEPTNO`),
  CONSTRAINT `EMP_SELF_KEY` FOREIGN KEY (`MGR`) REFERENCES `emp` (`EMPNO`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;
--
-- Dumping data for table `emp`
--
```

7.3 DB export/import

❑ Workbench 메뉴이용 import

Local instance MySQL80
Data Import

Import from Disk | Import Progress

Import Options

☒ Import from Dump Project Folder ...

Select the Dump Project Folder to import. You can do a selective restore.

☐ Import from Self-Contained File ...

Select the SQL/dump file to import. Please note that the whole file will be imported.

Default Schema to be Imported To

Default Target Schema: New...

*이걸 데이터베이스
넣으라 (이게 DB 하나 만들
or
mysql에 넣어야 OK!!)*

The default schema to import the dump into.
NOTE: this is only used if the dump file doesn't contain its schema, otherwise it is ignored.

Select Database Objects to Import (only available for Project Folders)

| Imp... | Schema |
|--------|--------|
|--------|--------|

| Imp... | Schema Objects |
|--------|----------------|
|--------|----------------|

[과제 2]

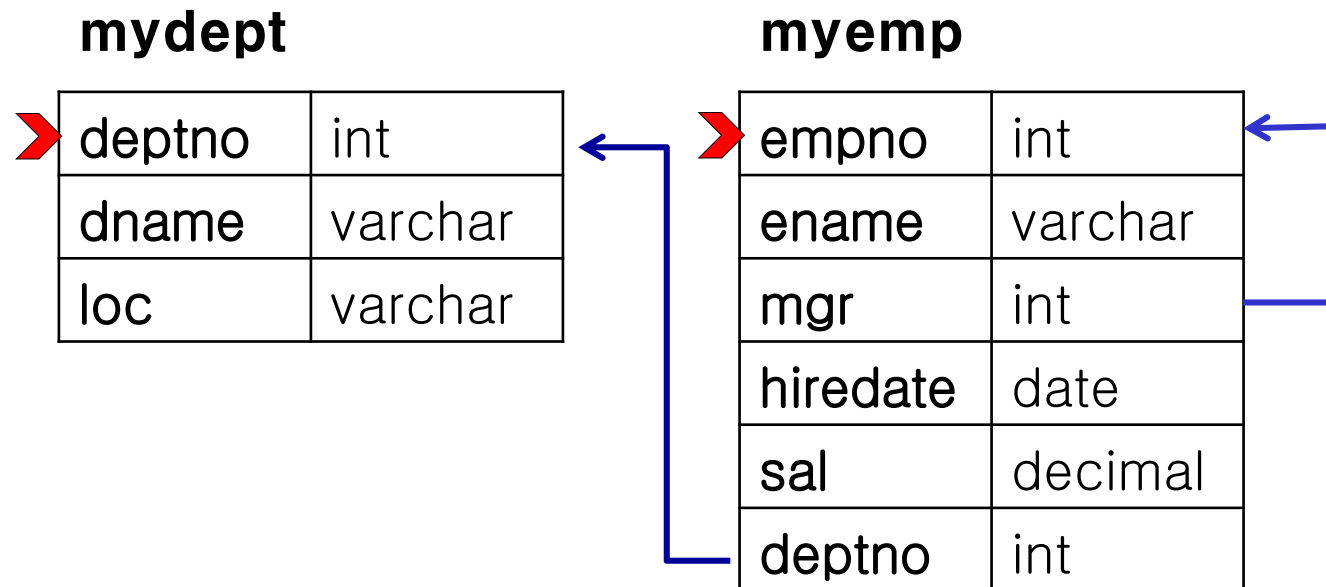
- ❑ 1. world 데이터베이스를 export 하되 구조(structure)와 데이터를 함께 export 하시오

Export 된 내용을 확인해 보시오 (화면캡처)

테이블 생성

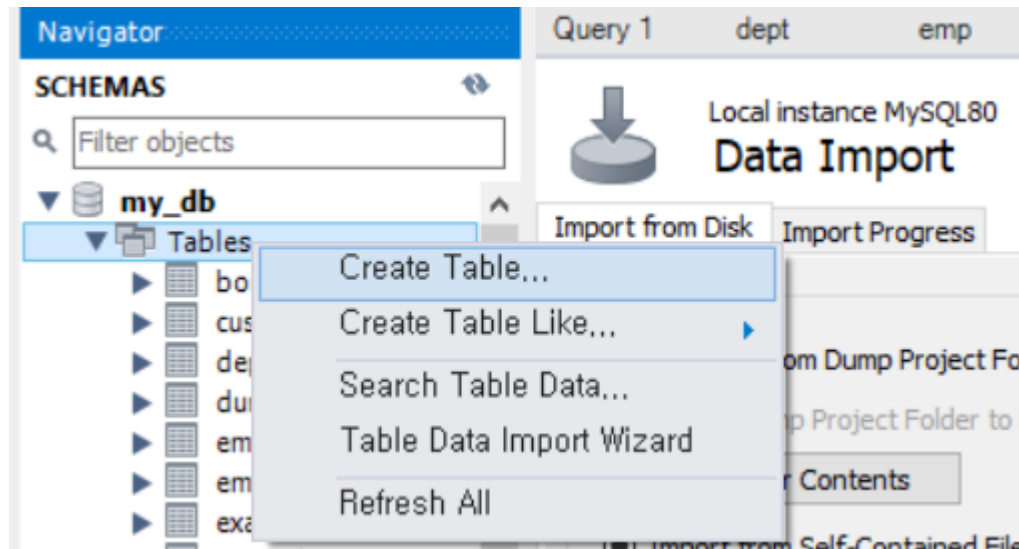
8. 테이블 생성

- ❑ Mysql worjbench 를 이용하여 mydept 와 myemp 테이블을 생성해보자



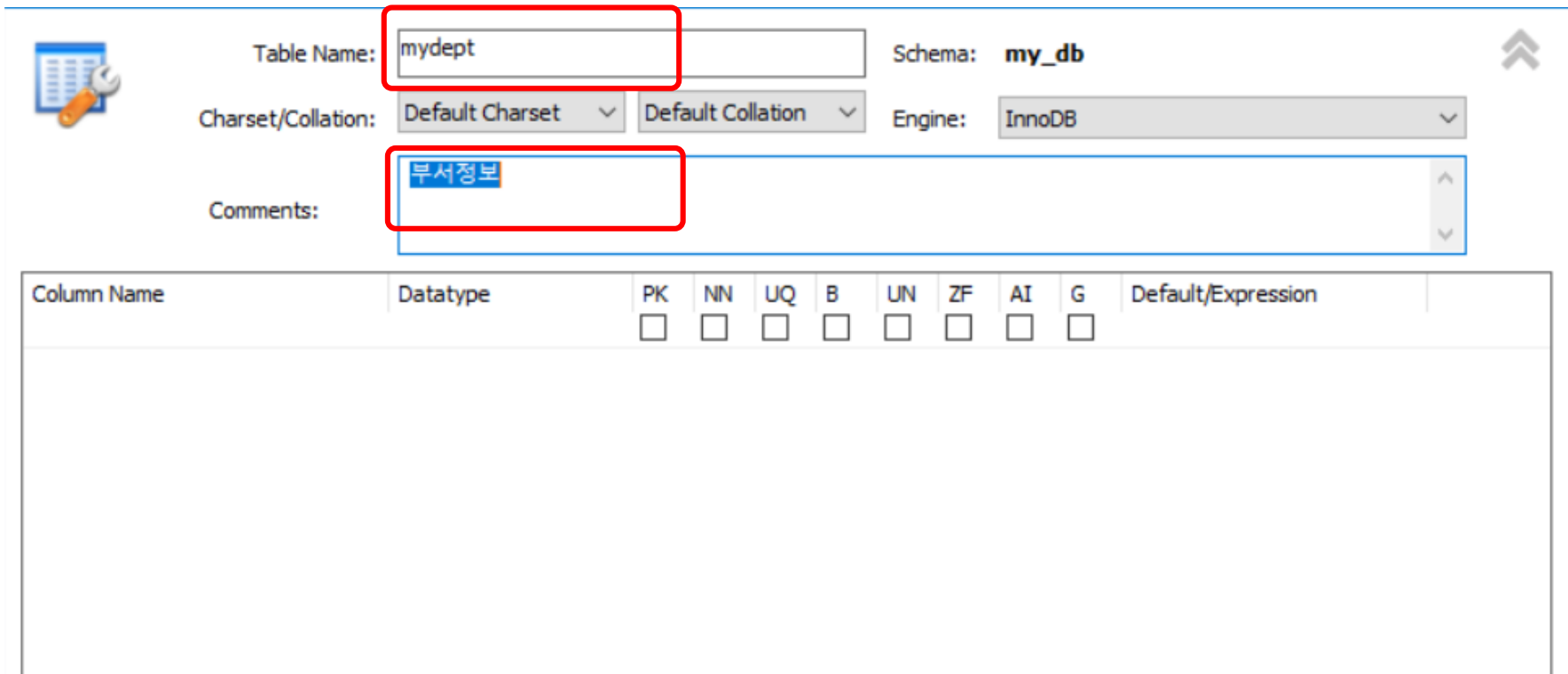
10. 테이블 생성

❑ (1) Create table 선택



8. 테이블 생성

❑ (2) 테이블 이름과 설명 입력

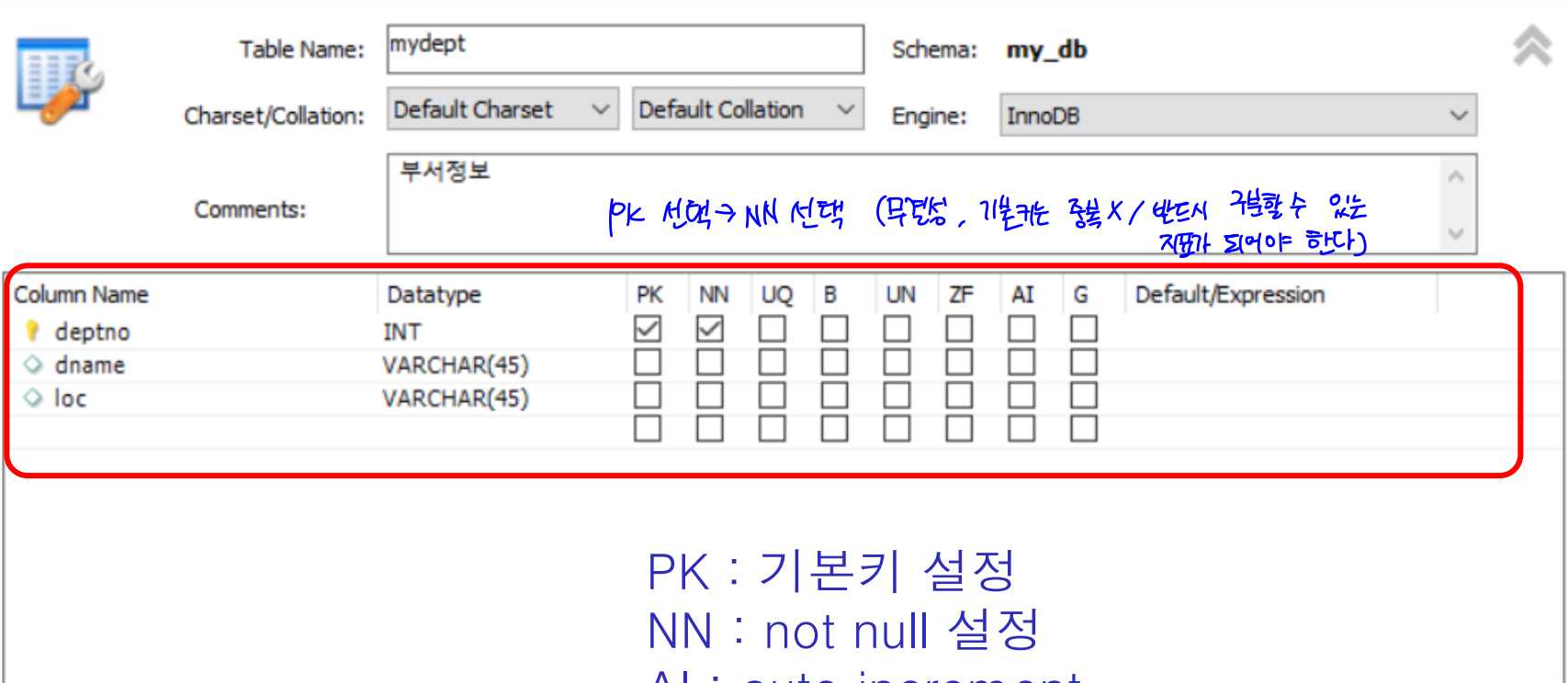


The image shows the MySQL Workbench Table Creation Wizard. The 'Table Name' field is set to 'mydept' and is highlighted with a red box. The 'Schema' is 'my_db'. The 'Charset/Collation' is set to 'Default Charset' and 'Default Collation'. The 'Engine' is 'InnoDB'. The 'Comments' field is set to '부서정보' and is also highlighted with a red box. Below the form is a table for defining columns.

| Column Name | Datatype | PK | NN | UQ | B | UN | ZF | AI | G | Default/Expression |
|-------------|----------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------|
| | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |

8. 테이블 생성

□ (3) 컬럼 정보 입력



The image shows the MySQL Workbench Table Creation Wizard. The 'Table Name' is 'mydept' and the 'Schema' is 'my_db'. The 'Engine' is 'InnoDB'. The 'Comments' field contains '부서정보'. A red box highlights the 'Column Name' table, which lists three columns: 'deptno' (INT), 'dname' (VARCHAR(45)), and 'loc' (VARCHAR(45)). The 'PK' column for 'deptno' is checked, and the 'NN' column for 'deptno' is also checked. The 'AI' column for 'deptno' is unchecked. The 'NN' column for 'dname' is unchecked, and the 'AI' column for 'dname' is unchecked. The 'NN' column for 'loc' is unchecked, and the 'AI' column for 'loc' is unchecked.

Table Name: mydept Schema: my_db

Charset/Collation: Default Charset Default Collation Engine: InnoDB

Comments: 부서정보

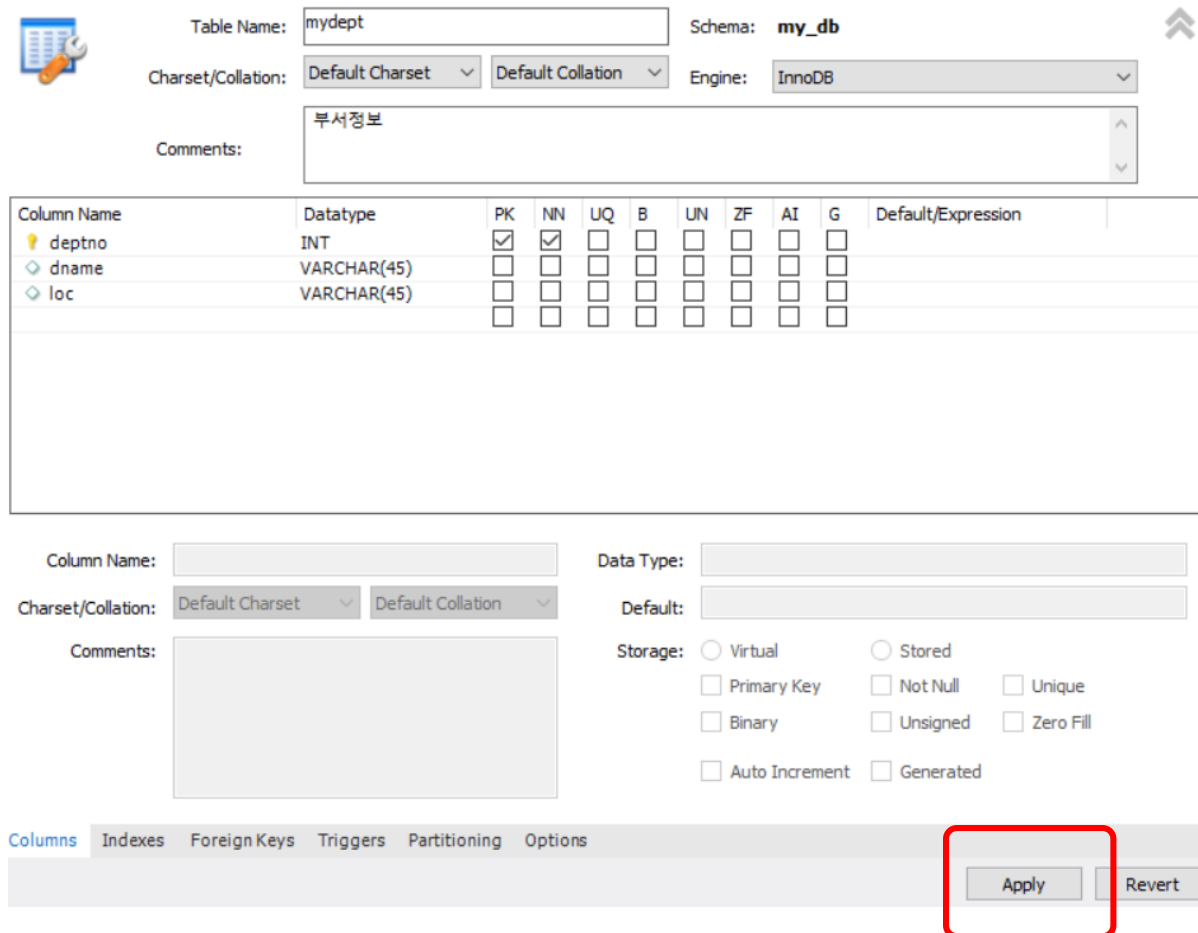
PK 선택 → NN 선택 (무결성, 기본키는 중복 X / 반드시 구분할 수 있는
지표가 되어야 한다)

| Column Name | Datatype | PK | NN | UQ | B | UN | ZF | AI | G | Default/Expression |
|-------------|-------------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------|
| deptno | INT | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| dname | VARCHAR(45) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| loc | VARCHAR(45) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |

PK : 기본키 설정
NN : not null 설정
AI : auto increment

8. 테이블 생성

❑ (4) [Apply] 버튼 클릭



The screenshot shows the MySQL Workbench Table Creation Wizard. The 'Table Name' is 'mydept' and the 'Schema' is 'my_db'. The 'Engine' is 'InnoDB'. The 'Comments' field contains '부서정보'. The 'Columns' tab is selected, showing a table with three columns: 'deptno' (INT, PK, NN), 'dname' (VARCHAR(45)), and 'loc' (VARCHAR(45)). The 'Apply' button is highlighted with a red box.

Table Name: mydept Schema: my_db

Charset/Collation: Default Charset Default Collation Engine: InnoDB

Comments: 부서정보

| Column Name | Datatype | PK | NN | UQ | B | UN | ZF | AI | G | Default/Expression |
|-------------|-------------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------|
| deptno | INT | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| dname | VARCHAR(45) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| loc | VARCHAR(45) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |

Column Name: Data Type: Charset/Collation: Default Collation: Default: Comments:

Storage: ☐ Virtual ☐ Stored

☐ Primary Key ☐ Not Null ☐ Unique

☐ Binary ☐ Unsigned ☐ Zero Fill

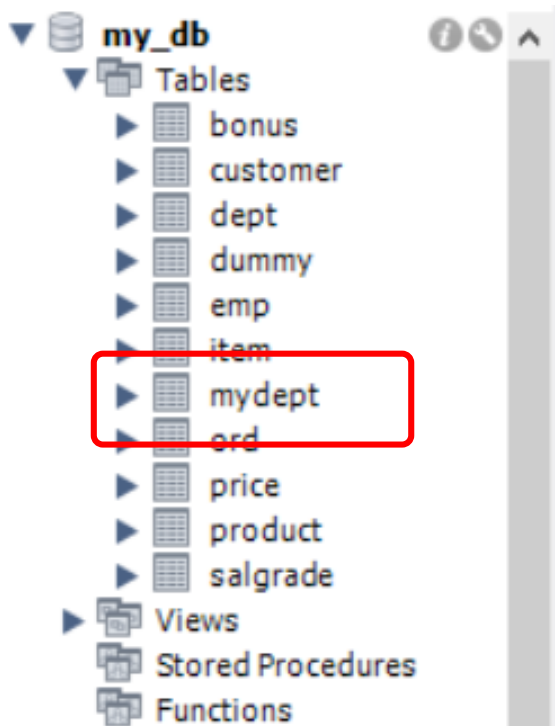
☐ Auto Increment ☐ Generated

Columns Indexes Foreign Keys Triggers Partitioning Options

Apply Revert

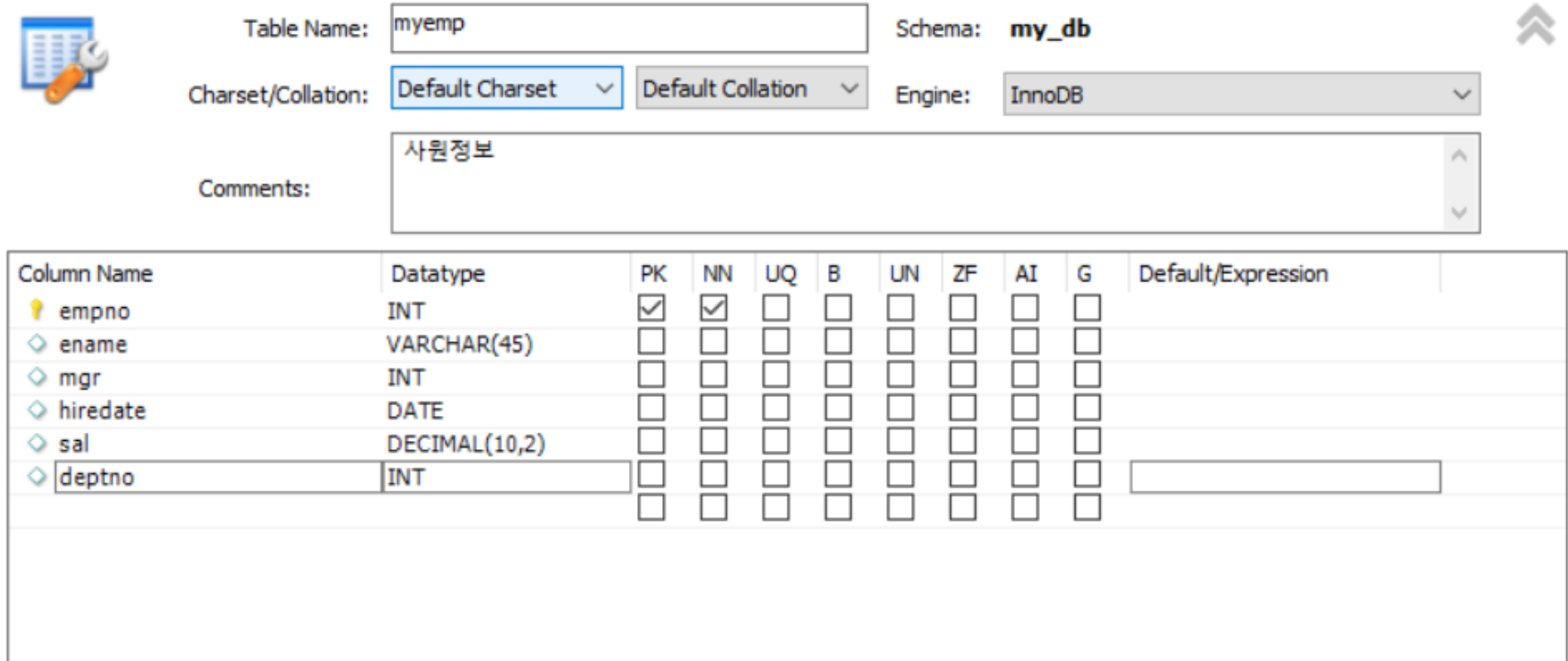
8. 테이블 생성

□ (5) 생성된 테이블 확인



8. 테이블 생성

- ❑ (6) myemp 도 동일한 방법으로 입력

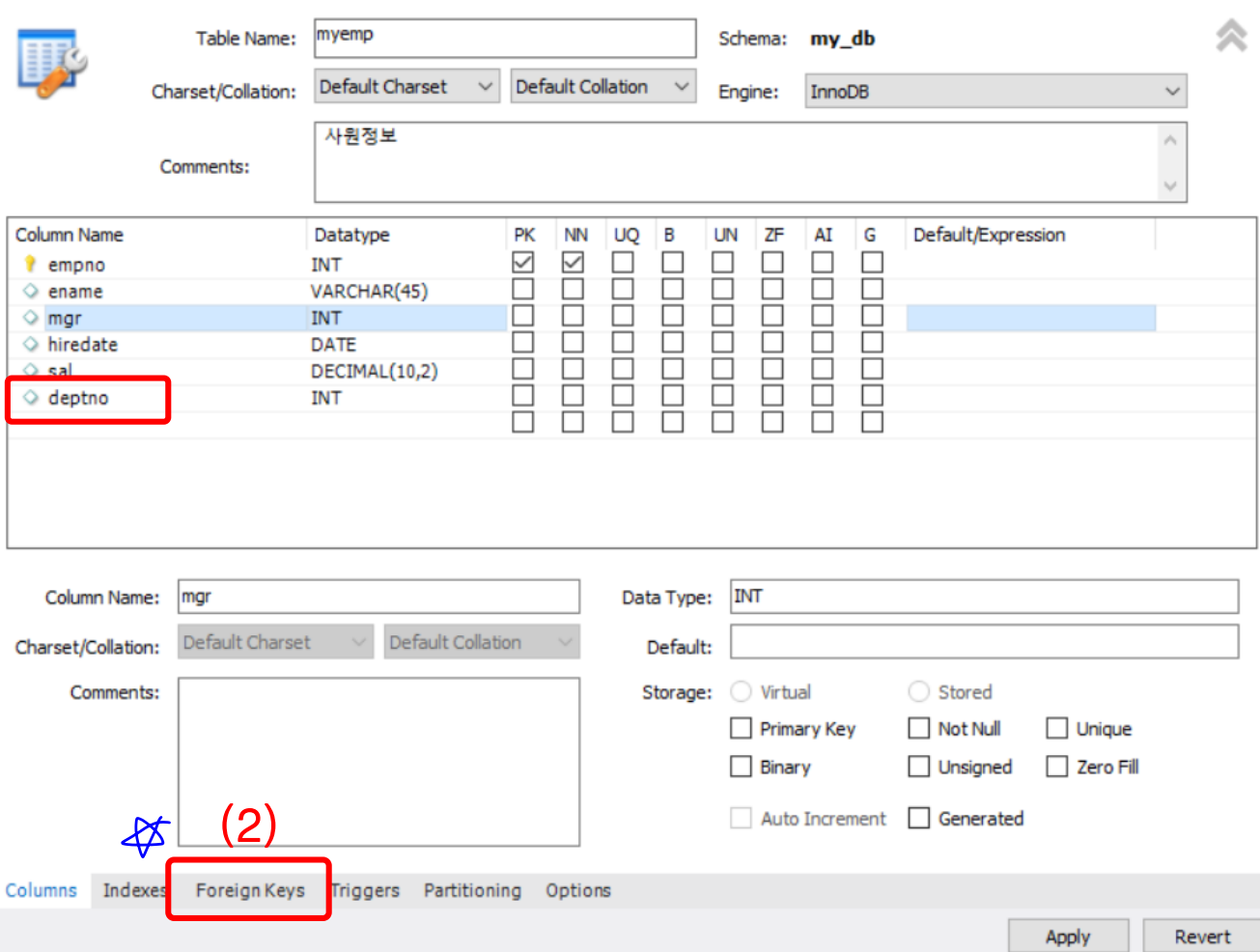


The image shows the MySQL Workbench Table Creation Wizard for a table named 'myemp'. The schema is 'my_db' and the engine is 'InnoDB'. The charset/collation is set to 'Default Charset' and 'Default Collation'. The comments field contains '사원정보'. The columns are defined as follows:

| Column Name | Datatype | PK | NN | UQ | B | UN | ZF | AI | G | Default/Expression |
|-------------|---------------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------|
| empno | INT | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| ename | VARCHAR(45) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| mgr | INT | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| hiredate | DATE | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| sal | DECIMAL(10,2) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| deptno | INT | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |

8. 테이블 생성

❑ (7) myemp -> mydept 외래키 설정



The image shows the MySQL Workbench Table Designer for a table named 'myemp' in the 'my_db' schema. The table is using the InnoDB engine. The columns are: empno (INT, PK, NN), ename (VARCHAR(45)), mgr (INT), hiredate (DATE), sal (DECIMAL(10,2)), and deptno (INT). The 'deptno' column is highlighted with a red box and labeled (1). The 'mgr' column is highlighted with a blue box. Below the column list, the 'mgr' column properties are shown, including its data type (INT) and storage options. The 'Storage' section has 'Virtual' selected. The 'Foreign Keys' tab is selected at the bottom, and the 'Foreign Keys' button is highlighted with a red box and labeled (2).

| Column Name | Datatype | PK | NN | UQ | B | UN | ZF | AI | G | Default/Expression |
|-------------|---------------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------|
| empno | INT | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| ename | VARCHAR(45) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| mgr | INT | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| hiredate | DATE | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| sal | DECIMAL(10,2) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| deptno | INT | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |

Column Name: mgr Data Type: INT

Charset/Collation: Default Charset Default Collation

Default:

Storage: ☐ Virtual ☐ Stored

☐ Primary Key ☐ Not Null ☐ Unique

☐ Binary ☐ Unsigned ☐ Zero Fill

☐ Auto Increment ☐ Generated

Columns Indexes **Foreign Keys** Triggers Partitioning Options

Apply Revert

8. 테이블 생성

The screenshot shows the 'Foreign Key' configuration window in MySQL Workbench. The table 'myemp' in schema 'my_db' is being configured with InnoDB engine. A foreign key named 'fk_myemp_deptno' is being created, linking the 'deptno' column of 'myemp' to the 'deptno' column of the referenced table 'mydept' in the same schema. The 'On Update' and 'On Delete' actions are both set to 'NO ACTION'. The 'Apply' button is highlighted with a red box.

Handwritten Annotations:

- 외래키 이름 (Foreign Key Name): Points to 'fk_myemp_deptno'.
- 부모테이블 (Parent Table): Points to 'mydept' in the 'Referenced Table' field.
- 자식테이블 컬럼 (Child Table Column): Points to 'deptno' in the 'Column' list.
- 부모테이블 컬럼 (참조 컬럼) (Parent Table Column (Referenced Column)): Points to 'deptno' in the 'Referenced Column' field.
- 참조 무결성 지켜!! (Maintain Referential Integrity!!): Points to the 'On Update' and 'On Delete' dropdowns.
- 부모 Table 변경 시 어케 할건지?! (What to do when Parent Table is changed?!): Points to the 'On Update' and 'On Delete' dropdowns.
- 부모테이블의 기본키 참조 (Refer to Parent Table's Primary Key): Points to the 'deptno' column in the 'Referenced Column' field.
- always: Points to the 'deptno' column in the 'Column' list.

Buttons: Apply (highlighted with a red box), Revert.

8. 테이블 생성

❑ (8) myemp → myemp 외래키 설정

Table Name: myemp Schema: my_db

Charset/Collation: utf8mb4 utf8mb4_0900_ai_ Engine: InnoDB

Comments: 직원정보

| Foreign Key Name | Referenced Table |
|------------------|-------------------|
| fk_myemp_deptno | 'my_db`.`mydept`' |
| fk_myemp_mgr | 'my_db`.`myemp`' |

| Column | Referenced Column |
|---|-------------------|
| <input type="checkbox"/> empno | |
| <input type="checkbox"/> ename | |
| <input checked="" type="checkbox"/> mgr | empno |
| <input type="checkbox"/> hiredate | |
| <input type="checkbox"/> sal | |
| <input type="checkbox"/> deptno | |

Foreign Key Options

On Update: RESTRICT

On Delete: RESTRICT

☐ Skip in SQL generation

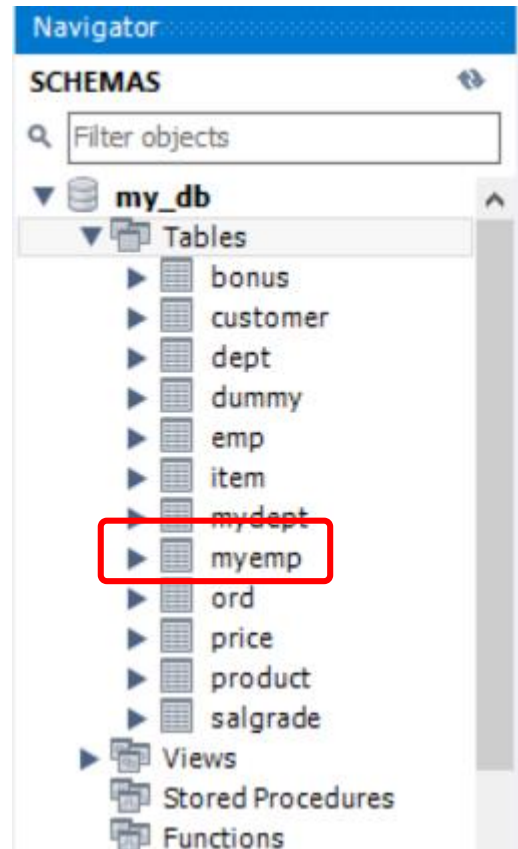
Foreign Key Comment

Columns Indexes **Foreign Keys** Triggers Partitioning Options

Apply Revert

8. 테이블 생성

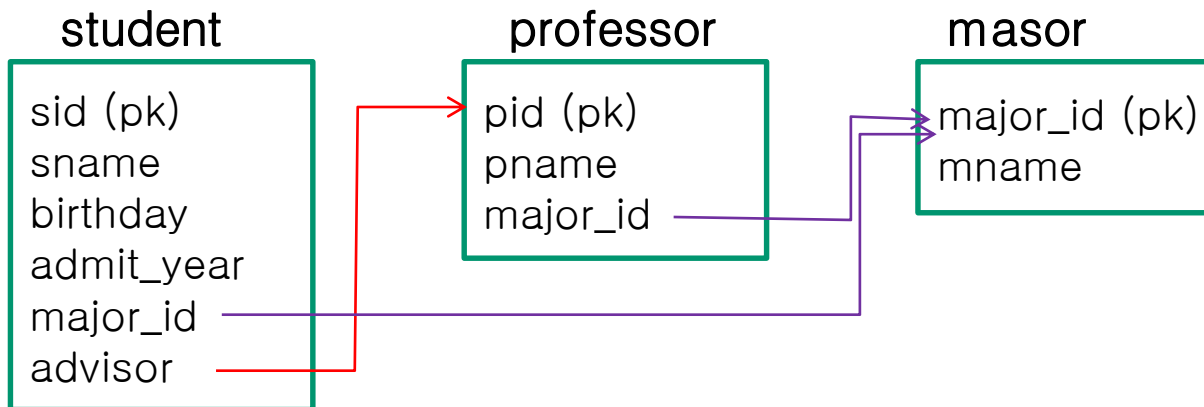
□ (9) 생성된 테이블 확인



[과제 3]

❑ Mysql workbench 의 기능을 이용하여 다음 작업을 수행하시오

1. 다음과 같이 student, professor, major 테이블을 만드시오



컬럼의
자료형은
각자 알아서
정한다

2. 각각의 테이블에 10 개 정도의 행(row) 을 입력하시오

3. 테이블에 입력된 데이터를 보이시오

4. 각 테이블의 입력된 데이터를 csv 포맷 파일로 저장(export)하시오

5. 각 테이블의 입력된 데이터를 모두 삭제 하시오

6. 저장된 csv 포맷 파일 데이터를 불러 오시오 (import)