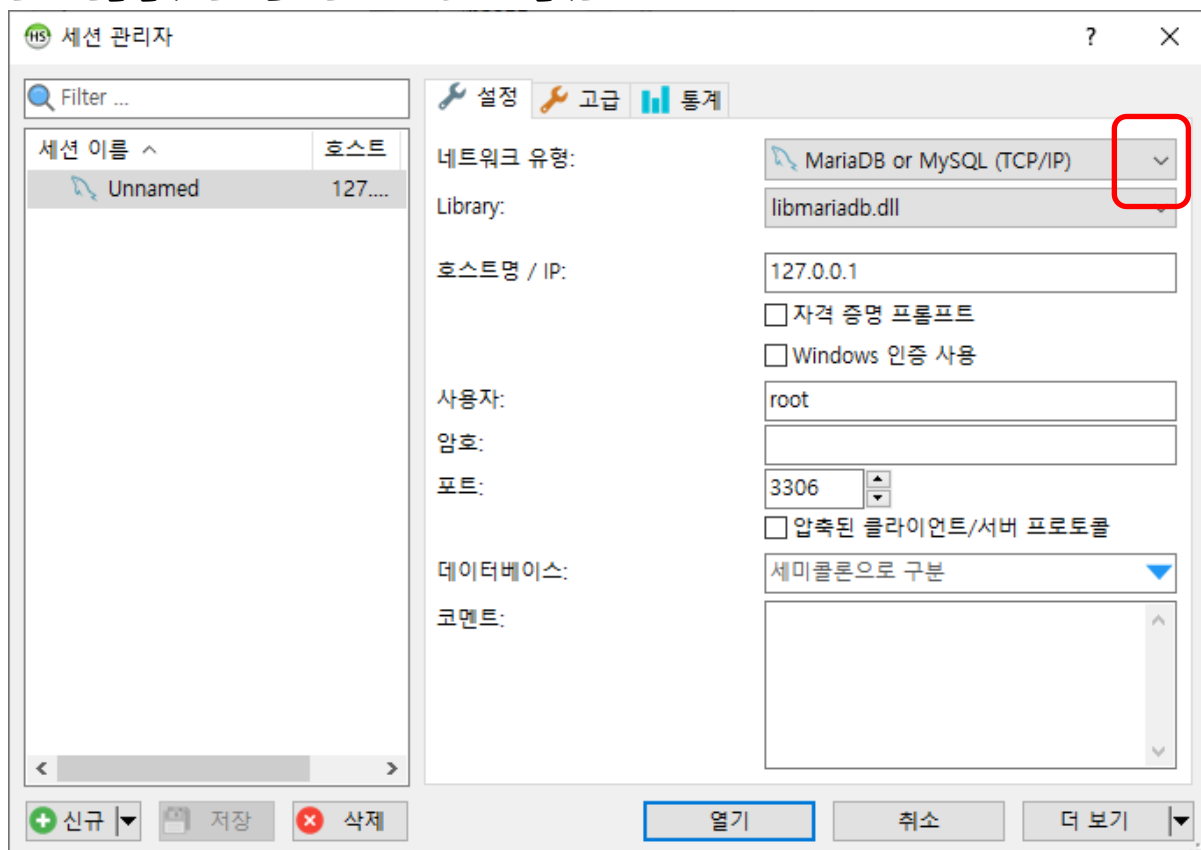


## [ HeidiSQL ]

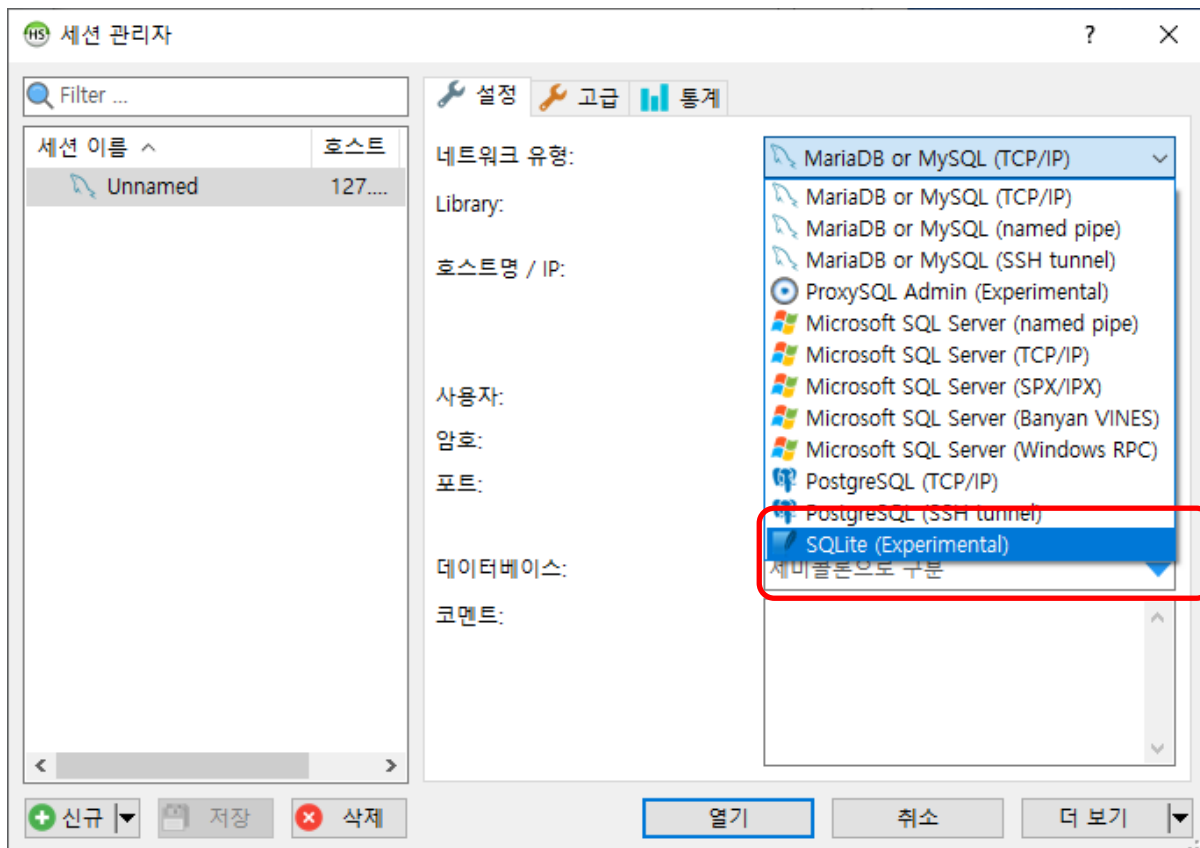
HeidiSQL은 이전에는 'MySQL Front'로 알려졌던 제품으로서 MySQL, MariaDB 그리고 SQLite 등 DBMS를 직접 접속하여 사용하려는 경우에 선택할 수 있는 DBMS 클라이언트 무료 프로그램이다.



신규 버튼을 클릭하면 다음 화면의 서브 윈도우가 출력된다.

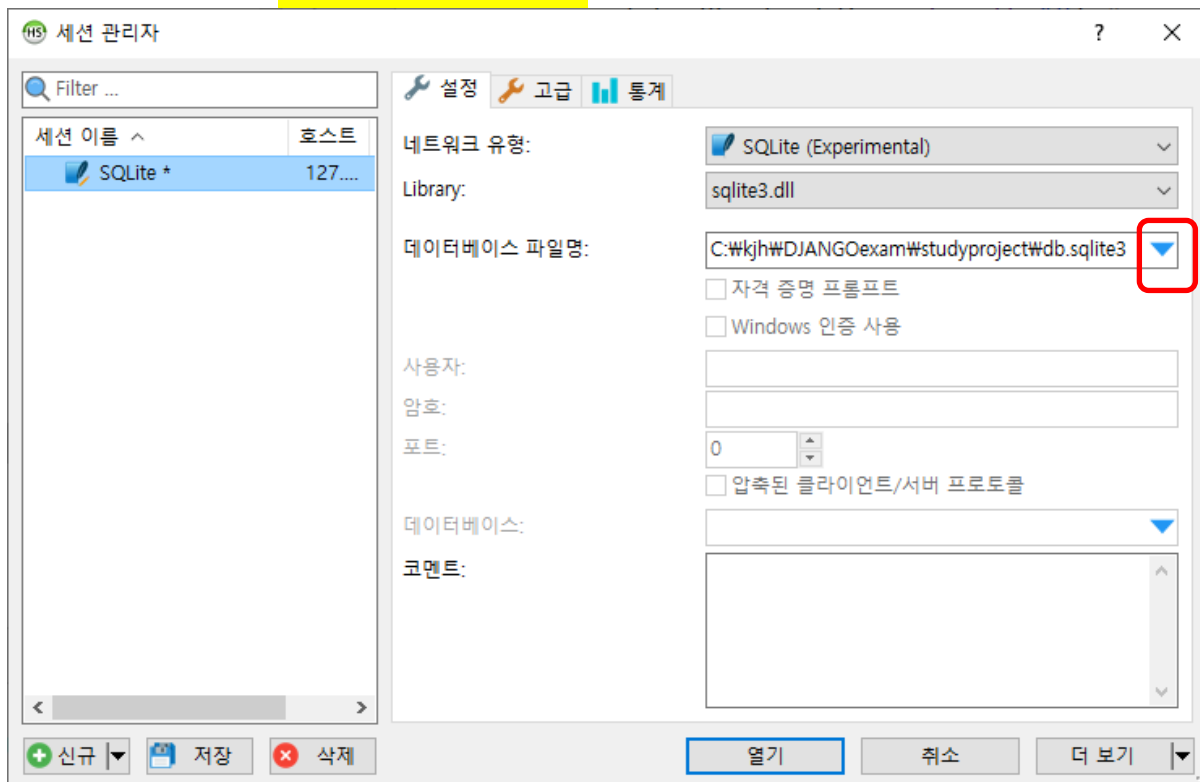


네트워크 유형에서 SQLite 항목을 선택한다.

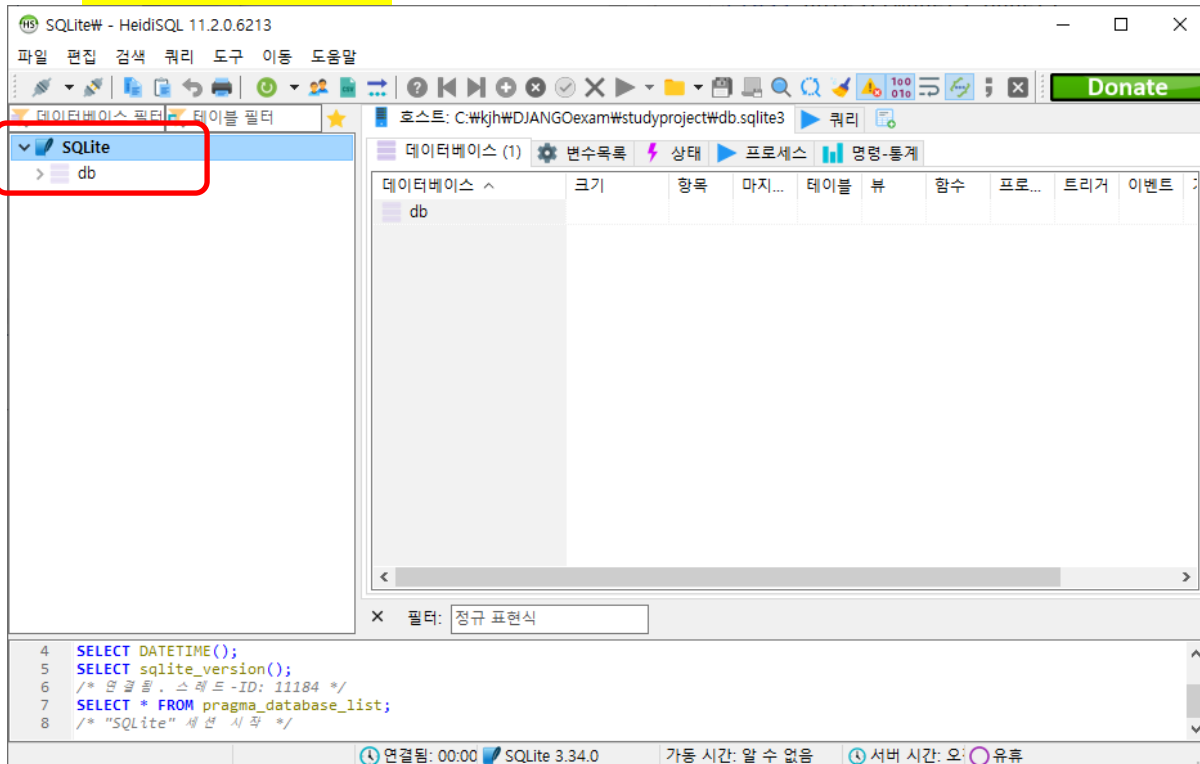


네트워크 유형에서 SQLite 항목을 선택하면 Library 에 sqlite3.dll 이 자동 설정되는 것을 볼 수 있다.

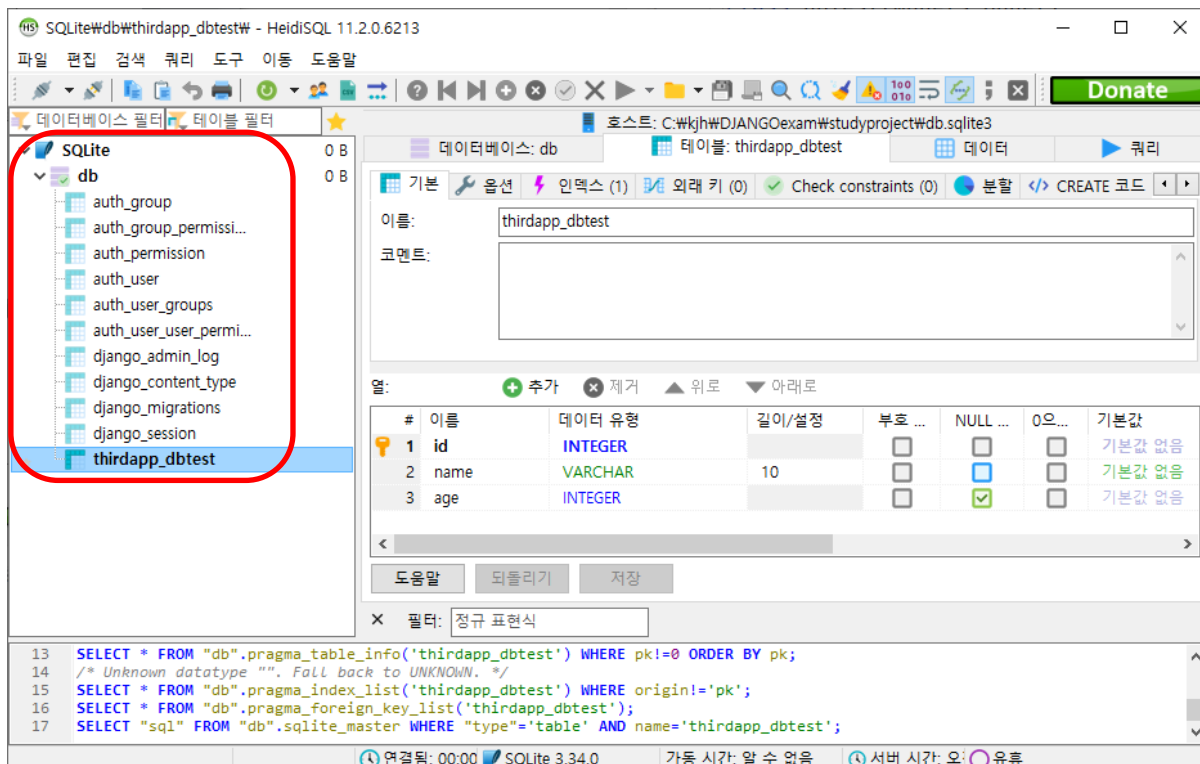
데이터베이스 파일명으로 studyproject 의 db.sqlite3 를 찾아서 설정한다.



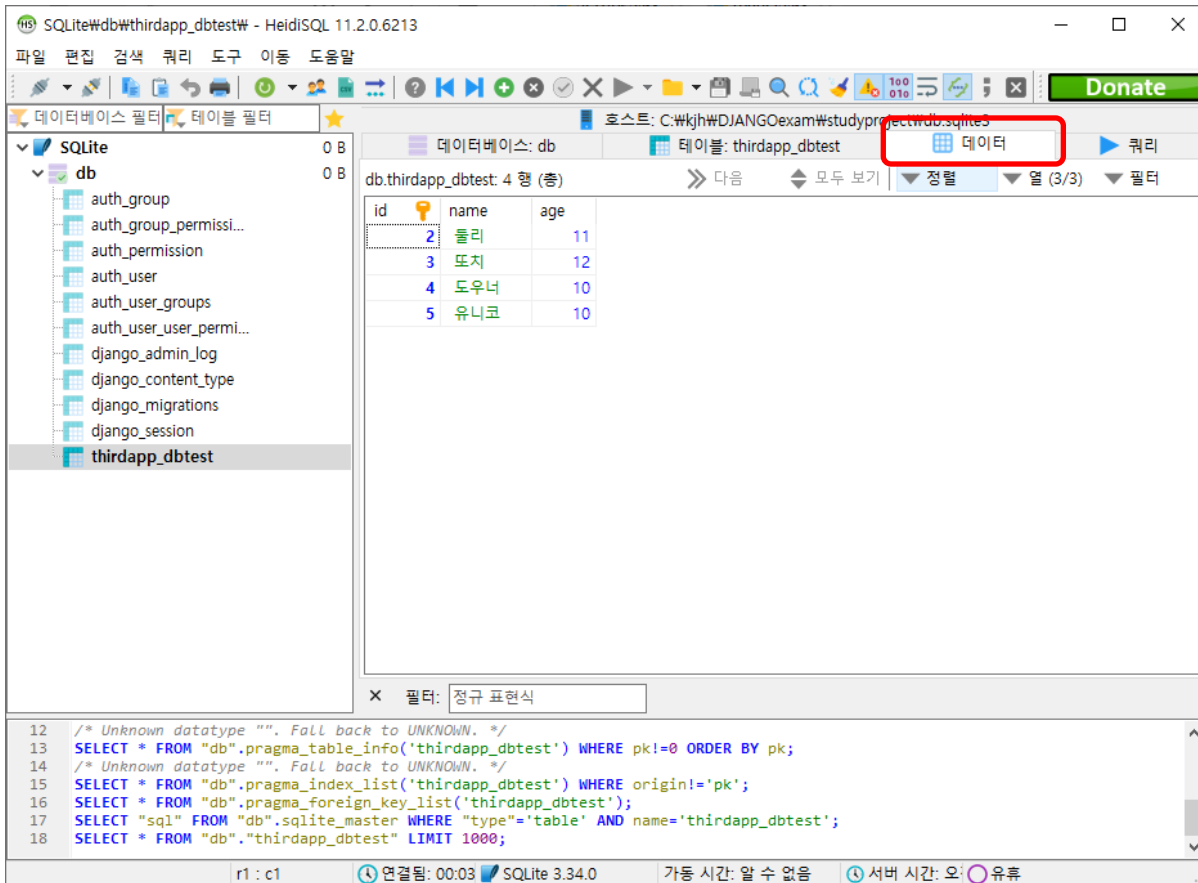
오픈된 studyproject 의 db.sqlite3 에는 db 라는 이름의 데이터베이스가 생성되어 있는 것을 볼 수 있다.



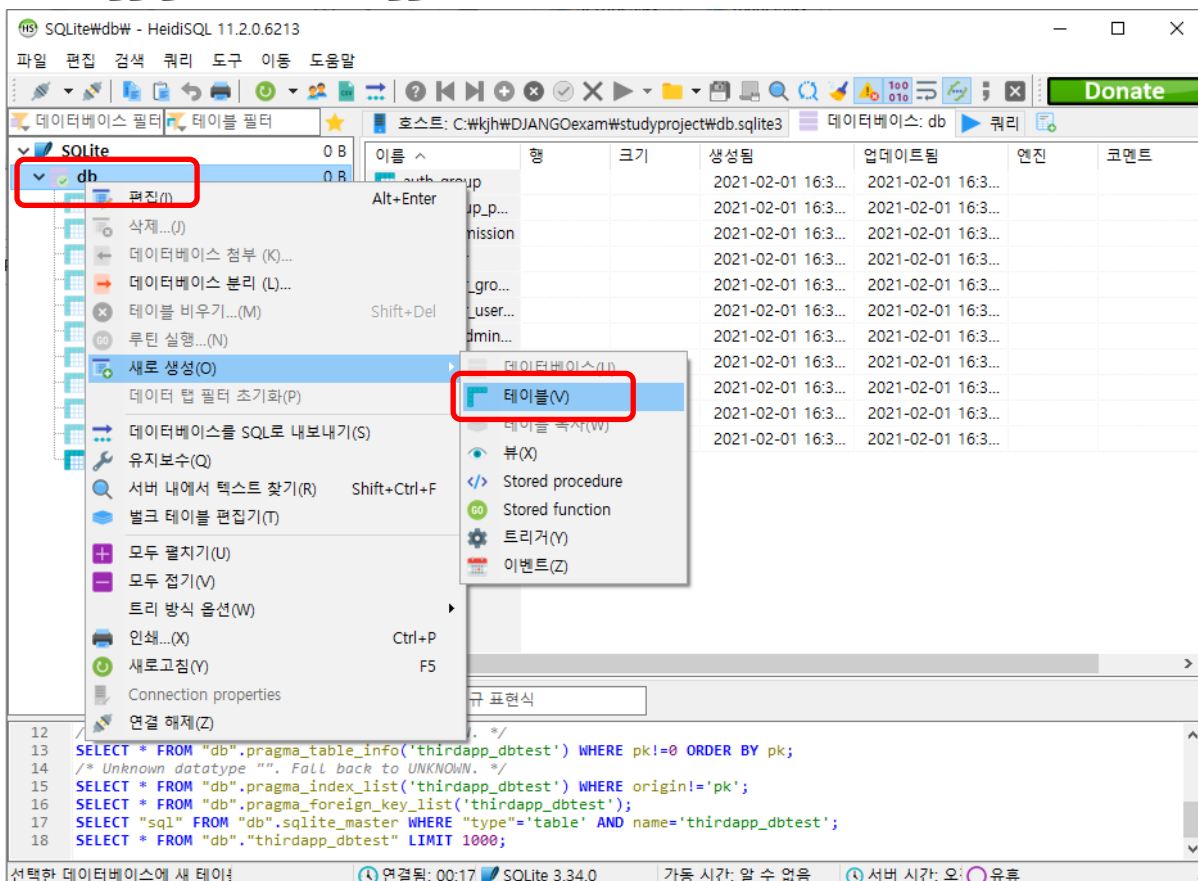
db 라는 이름의 데이터베이스에는 python manage.py migrate 명령의 실행에 의해서 생성된 테이블들이 여러 개 존재하며 그 중에서 제일 아래에 있는 thirdapp\_dbtest 가 DBTest 라는 모델 클래스에 의해서 생성된 테이블이다. 오른쪽에서 이 테이블의 컬럼 사양을 체크해 볼 수 있다.



데이터라는 버튼을 클릭하면 thirdapp\_dbtest 테이블에 저장된 데이터들을 직접 확인할 수 있다.



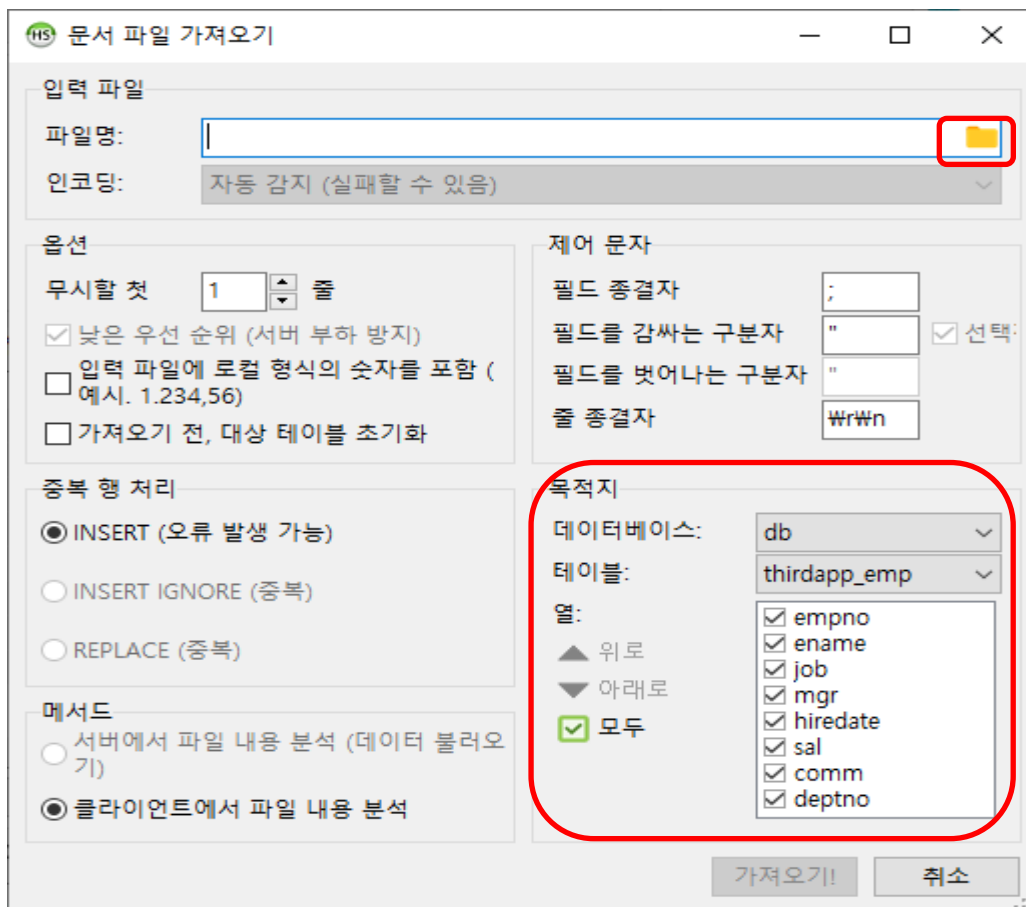
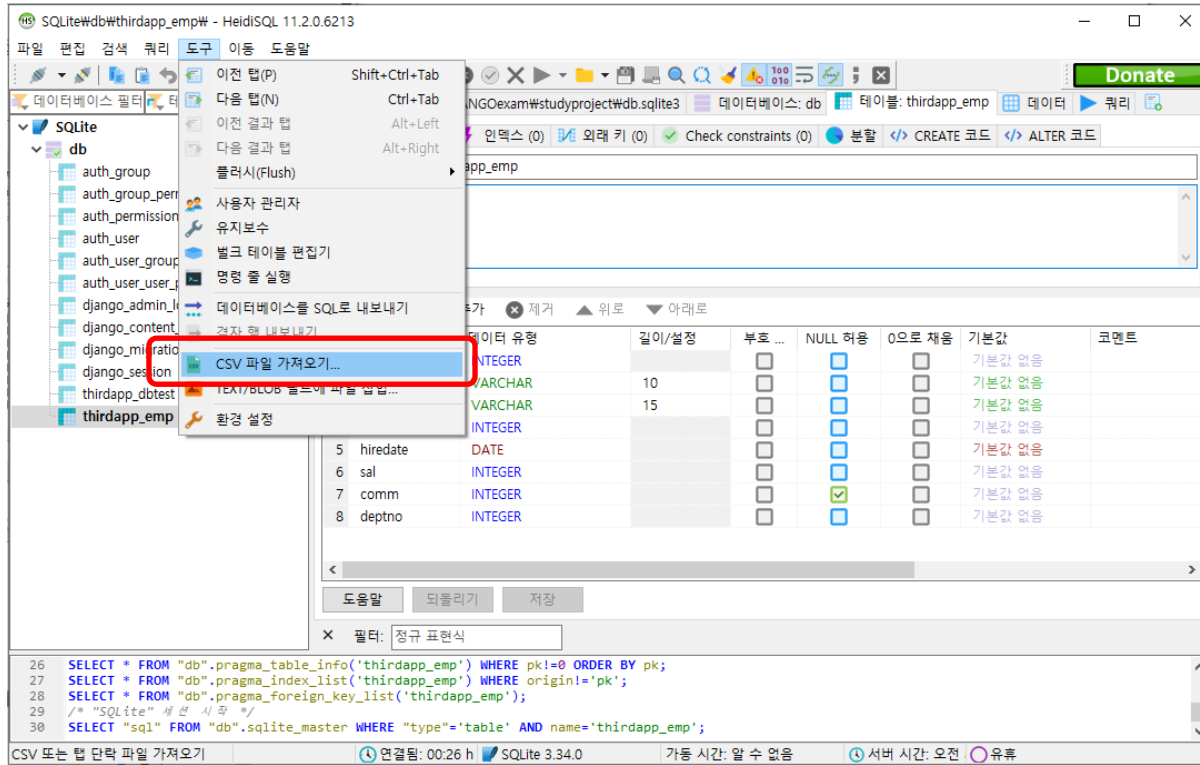
다음 메뉴들을 클릭해서 새로운 테이블을 여기서 직접 생성해 본다.

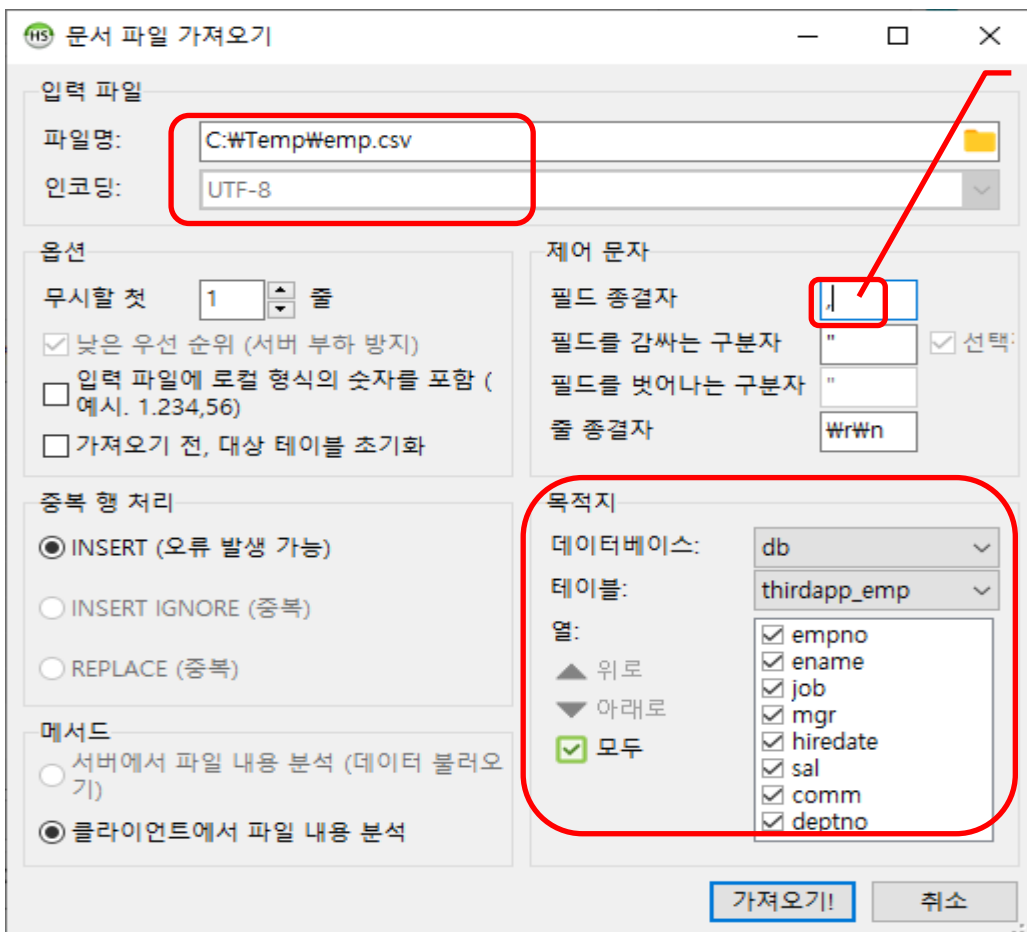
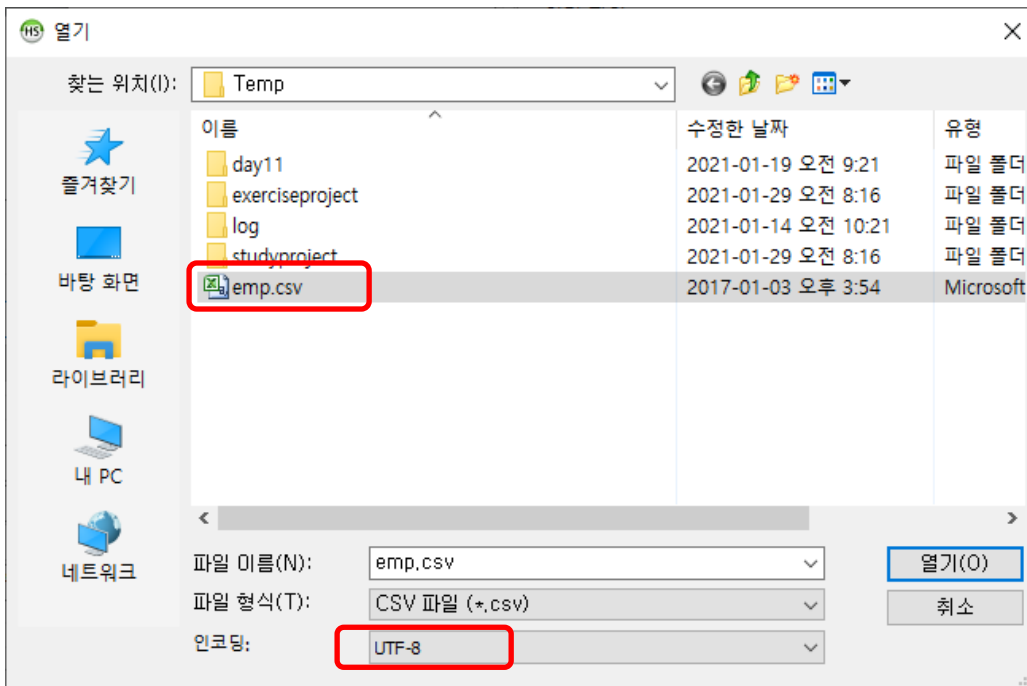




생성된 테이블에 csv 파일의 내용을 저장해 보자.

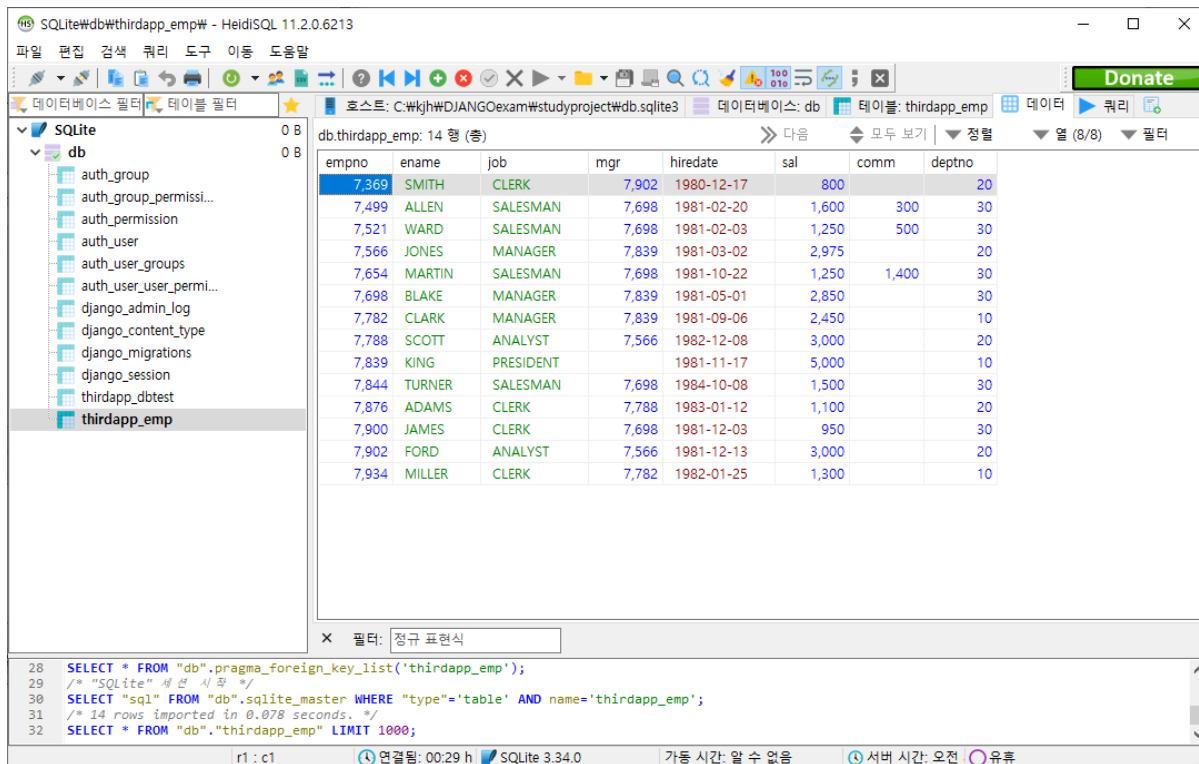
제공된 emp.csv 를 적당한 폴더에 저장한 다음에 다음 과정을 수행한다.(c:\Temp)





emp.csv의 경우 데이터  
구분자가 , 이므로 ; 을 ,  
로 변경한다.

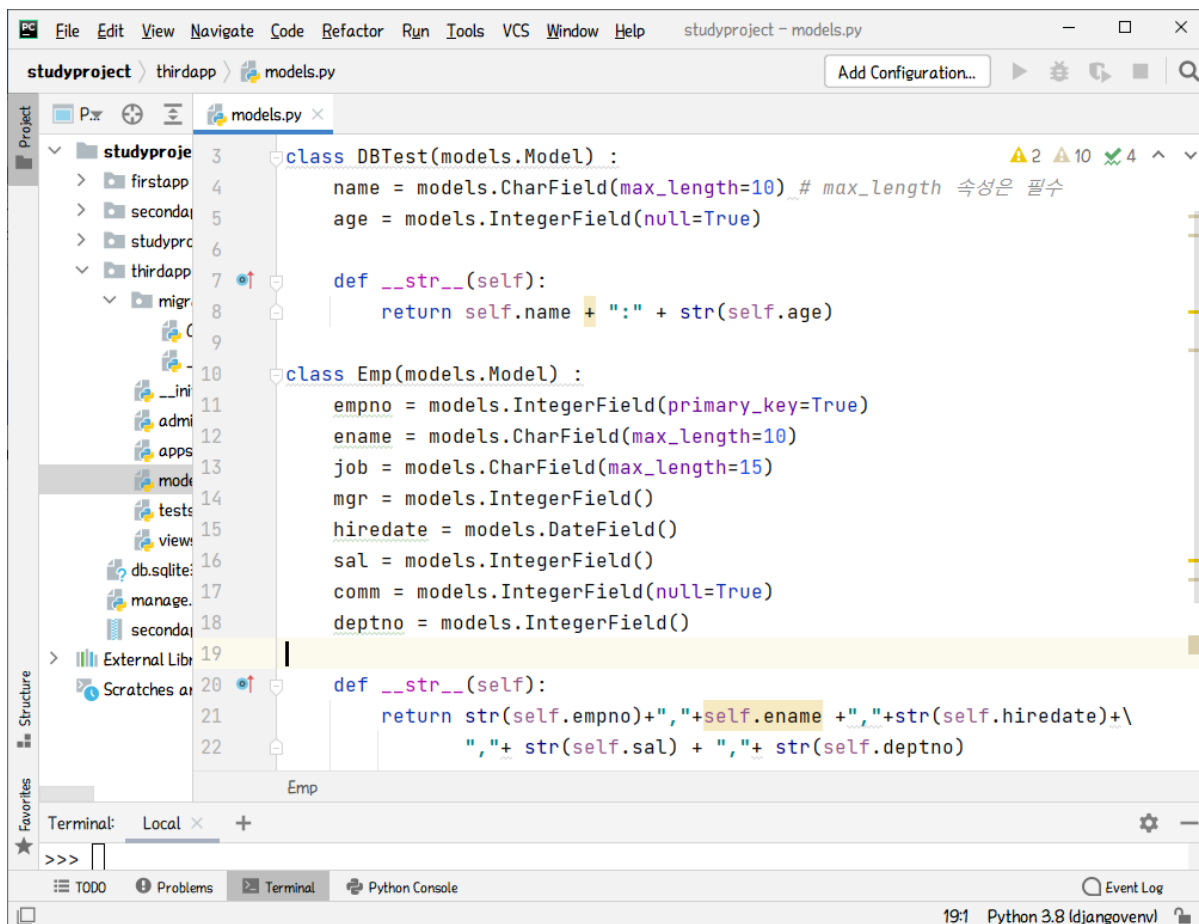
emp.csv 파일의 내용이 thirdapp\_emp 테이블에 저장된 것을 볼 수 있다.



The screenshot shows the HeidiSQL interface with the 'thirdapp\_emp' table selected. The table contains 14 rows of employee data. The columns are empno, ename, job, mgr, hiredate, sal, comm, and deptno. The data is as follows:

empno	ename	job	mgr	hiredate	sal	comm	deptno
7,369	SMITH	CLERK	7,902	1980-12-17	800		20
7,499	ALLEN	SALESMAN	7,698	1981-02-20	1,600	300	30
7,521	WARD	SALESMAN	7,698	1981-02-03	1,250	500	30
7,566	JONES	MANAGER	7,839	1981-03-02	2,975		20
7,654	MARTIN	SALESMAN	7,698	1981-10-22	1,250	1,400	30
7,698	BLAKE	MANAGER	7,839	1981-05-01	2,850		30
7,782	CLARK	MANAGER	7,839	1981-09-06	2,450		10
7,788	SCOTT	ANALYST	7,566	1982-12-08	3,000		20
7,839	KING	PRESIDENT		1981-11-17	5,000		10
7,844	TURNER	SALESMAN	7,698	1984-10-08	1,500		30
7,876	ADAMS	CLERK	7,788	1983-01-12	1,100		20
7,900	JAMES	CLERK	7,698	1981-12-03	950		30
7,902	FORD	ANALYST	7,566	1981-12-13	3,000		20
7,934	MILLER	CLERK	7,782	1982-01-25	1,300		10

thirdapp\_emp 테이블의 내용을 파이썬 프로그램에서 다루기 위해 thirdapp 의 models.py 파일에 다음과 같이 Emp 라는 클래스를 생성한다.



```
class DBTest(models.Model):
    name = models.CharField(max_length=10) # max_length 속성은 필수
    age = models.IntegerField(null=True)

    def __str__(self):
        return self.name + ":" + str(self.age)

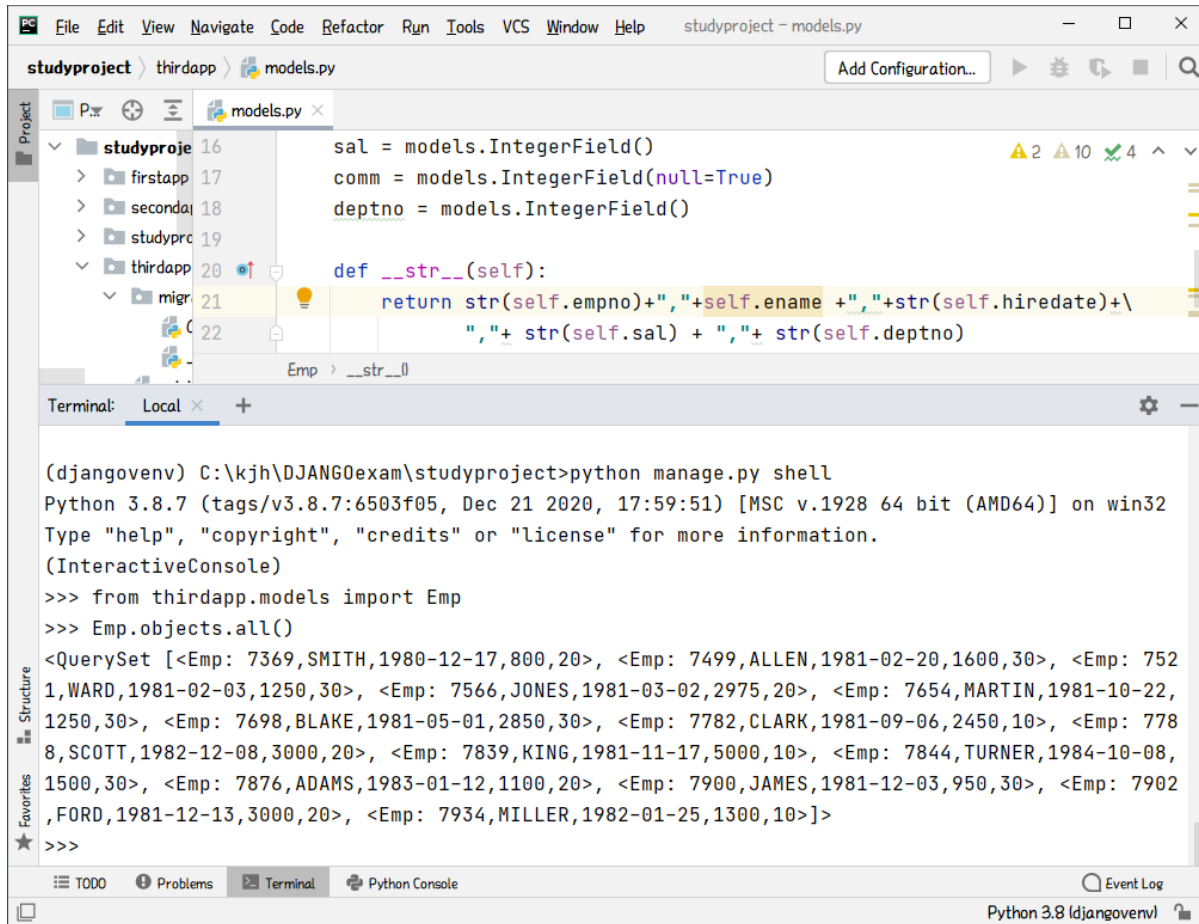
class Emp(models.Model):
    empno = models.IntegerField(primary_key=True)
    ename = models.CharField(max_length=10)
    job = models.CharField(max_length=15)
    mgr = models.IntegerField()
    hiredate = models.DateField()
    sal = models.IntegerField()
    comm = models.IntegerField(null=True)
    deptno = models.IntegerField()

    def __str__(self):
        return str(self.empno)+","+self.ename+","+str(self.hiredate)+\
            ","+ str(self.sal) + ","+ str(self.deptno)
```



djangoenv 가상환경 기반의 터미널에서 `python manage.py shell` 명령을 실행시키고 인터랙티브 실행모드를 기동시킨 후에 다음 명령들을 실행시켜 본다.

```
from thirdapp.models import Emp
Emp.objects.all()
```



The screenshot shows an IDE window titled 'studyproject - models.py'. The editor displays the following code in `models.py`:

```
16 sal = models.IntegerField()
17 comm = models.IntegerField(null=True)
18 deptno = models.IntegerField()
19
20 def __str__(self):
21     return str(self.empno)+" "+self.ename + " "+str(self.hiredate)+"\
22         ", "+ str(self.sal) + ", "+ str(self.deptno)
```

The terminal window at the bottom shows the following output:

```
(djangoenv) C:\kjh\DJANGOexam\studyproject>python manage.py shell
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from thirdapp.models import Emp
>>> Emp.objects.all()
<QuerySet [<Emp: 7369,SMITH,1980-12-17,800,20>, <Emp: 7499,ALLEN,1981-02-20,1600,30>, <Emp: 7521,WARD,1981-02-03,1250,30>, <Emp: 7566,JONES,1981-03-02,2975,20>, <Emp: 7654,MARTIN,1981-10-22,1250,30>, <Emp: 7698,BLAKE,1981-05-01,2850,30>, <Emp: 7782,CLARK,1981-09-06,2450,10>, <Emp: 7788,SCOTT,1982-12-08,3000,20>, <Emp: 7839,KING,1981-11-17,5000,10>, <Emp: 7844,TURNER,1984-10-08,1500,30>, <Emp: 7876,ADAMS,1983-01-12,1100,20>, <Emp: 7900,JAMES,1981-12-03,950,30>, <Emp: 7902,FORD,1981-12-13,3000,20>, <Emp: 7934,MILLER,1982-01-25,1300,10>]>
>>>
```

```
>>> for d in Emp.objects.all() :
...     print(d)
```

```
7369,SMITH,1980-12-17,800,20
7499,ALLEN,1981-02-20,1600,30
7521,WARD,1981-02-03,1250,30
7566,JONES,1981-03-02,2975,20
7654,MARTIN,1981-10-22,1250,30
7698,BLAKE,1981-05-01,2850,30
7782,CLARK,1981-09-06,2450,10
7788,SCOTT,1982-12-08,3000,20
7839,KING,1981-11-17,5000,10
7844,TURNER,1984-10-08,1500,30
7876,ADAMS,1983-01-12,1100,20
7900,JAMES,1981-12-03,950,30
7902,FORD,1981-12-13,3000,20
7934,MILLER,1982-01-25,1300,10
```

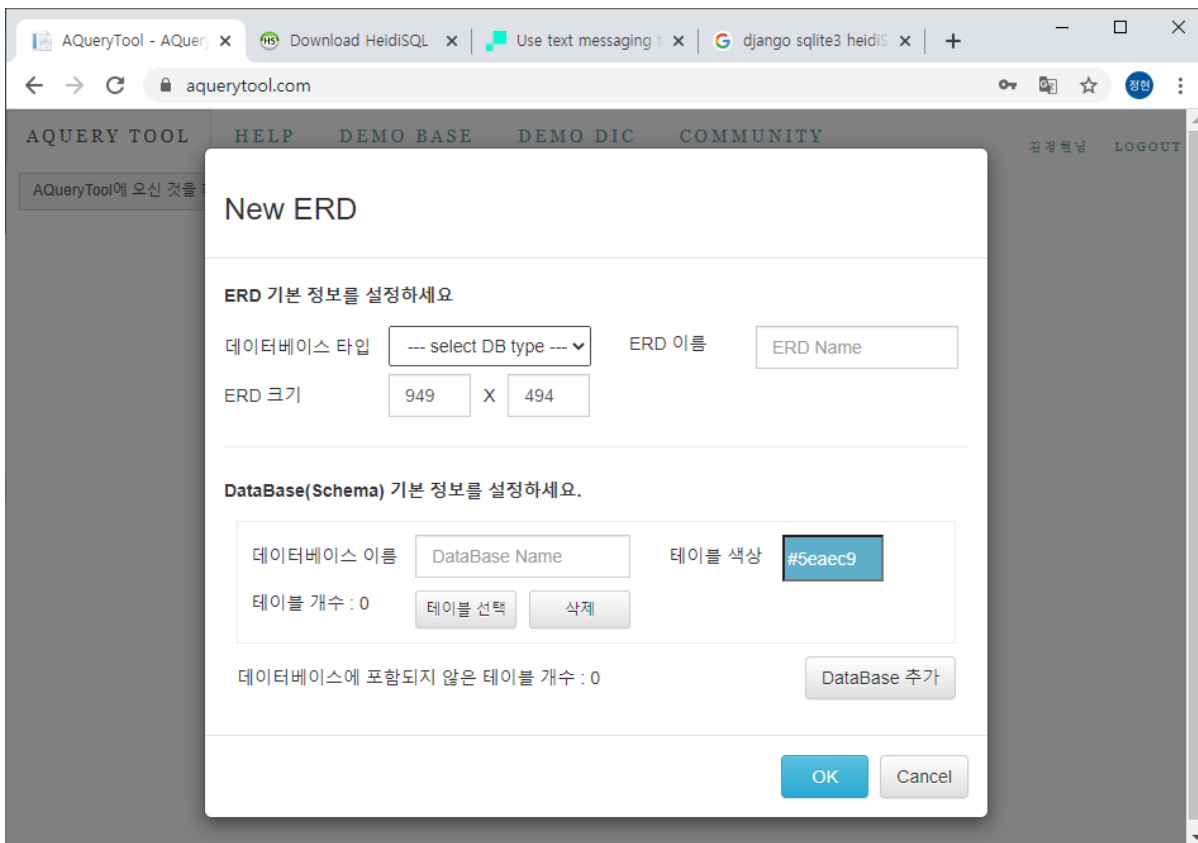
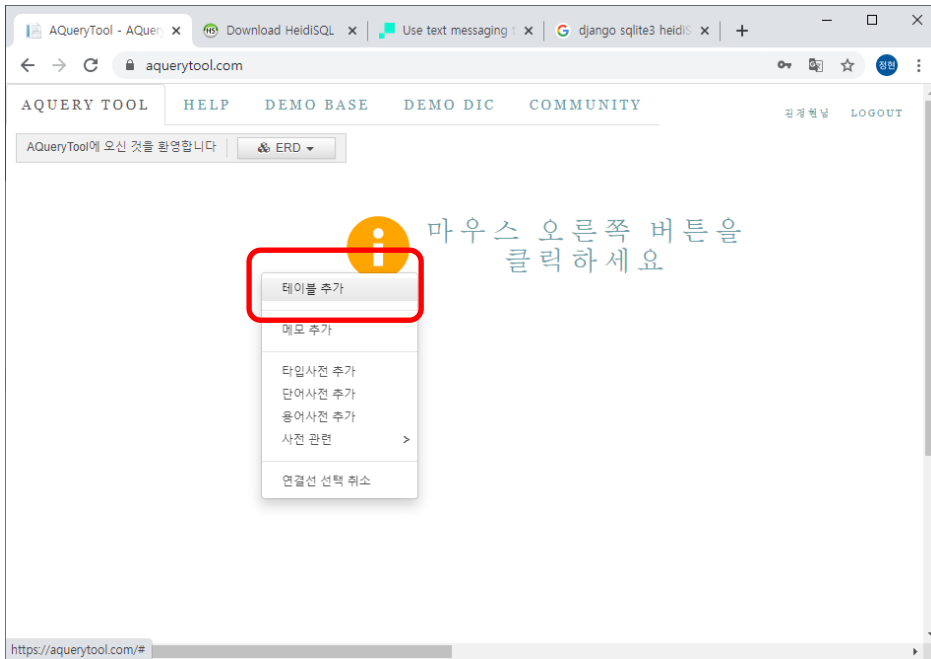
```
>>> for d in Emp.objects.order_by('hiredate') :  
...     print(d)
```

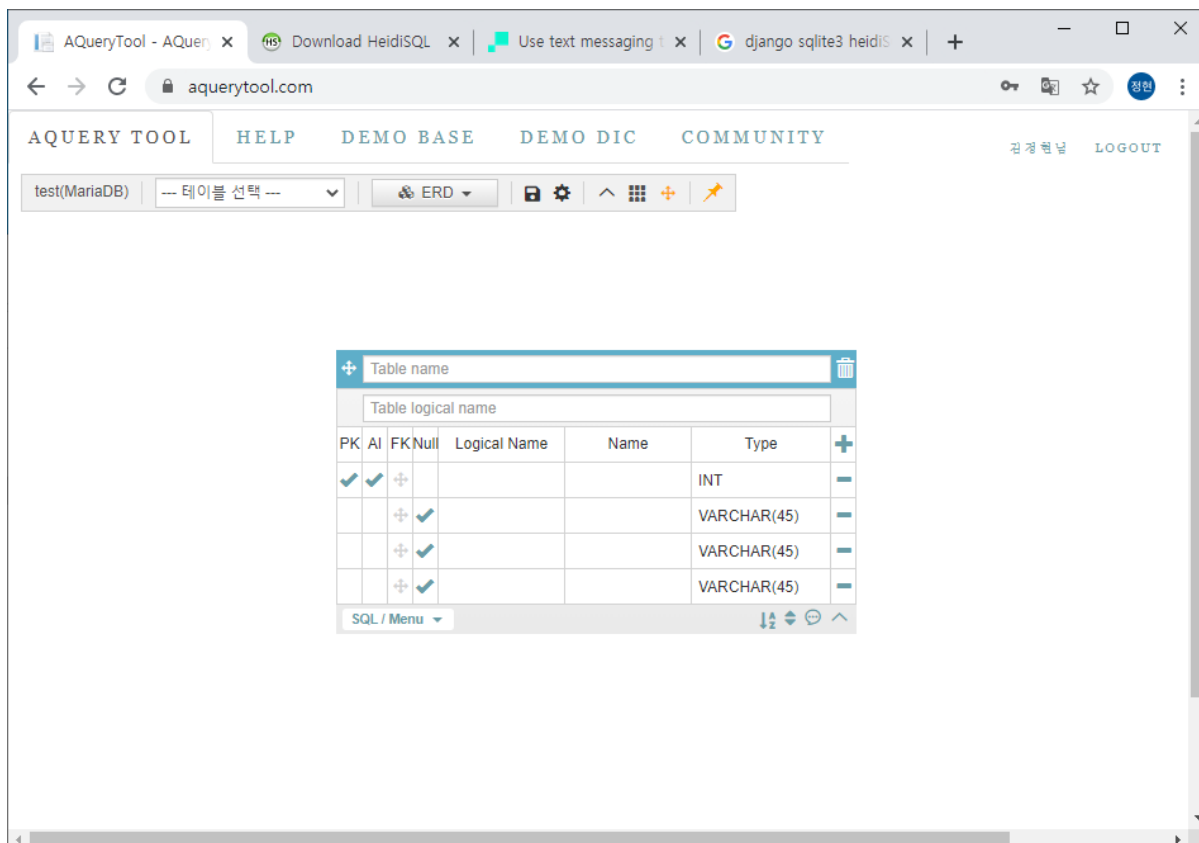
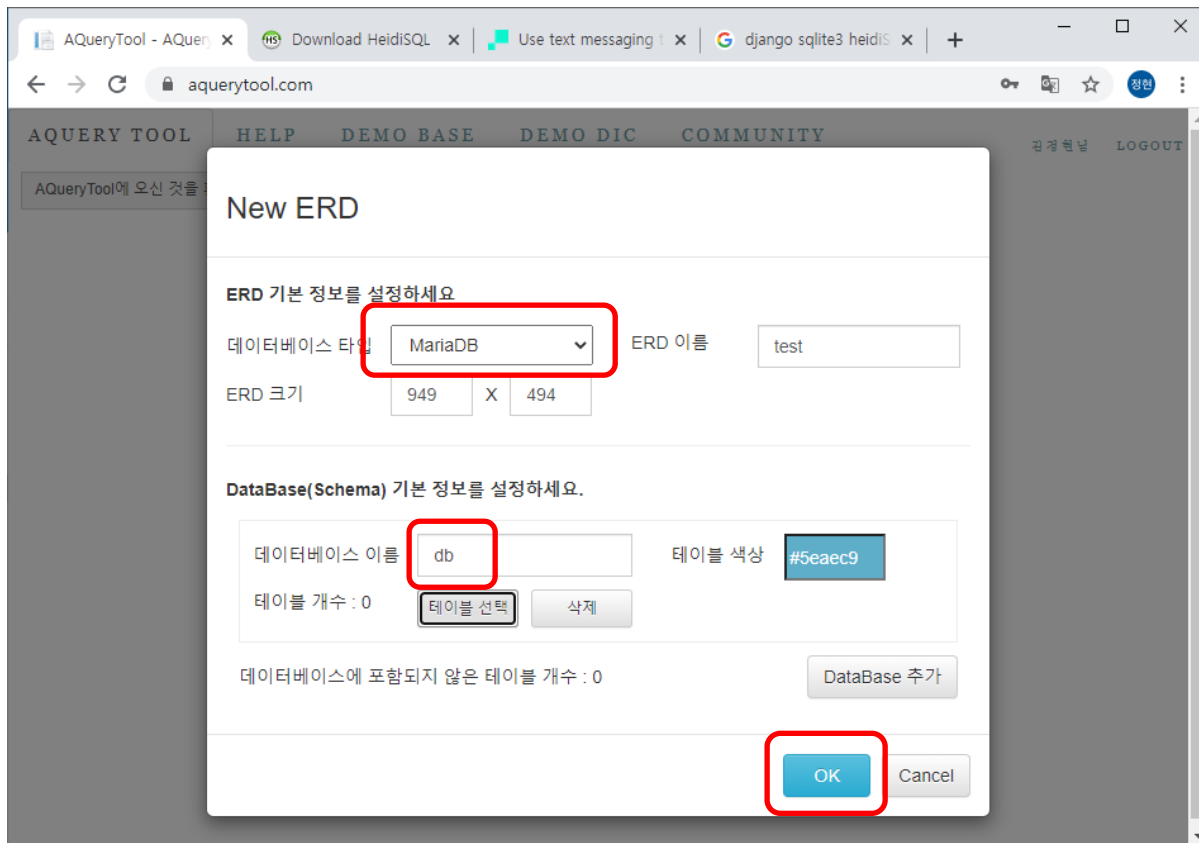
```
...  
7369, SMITH, 1980-12-17, 800, 20  
7521, WARD, 1981-02-03, 1250, 30  
7499, ALLEN, 1981-02-20, 1600, 30  
7566, JONES, 1981-03-02, 2975, 20  
7698, BLAKE, 1981-05-01, 2850, 30  
7782, CLARK, 1981-09-06, 2450, 10  
7654, MARTIN, 1981-10-22, 1250, 30  
7839, KING, 1981-11-17, 5000, 10  
7900, JAMES, 1981-12-03, 950, 30  
7902, FORD, 1981-12-13, 3000, 20  
7934, MILLER, 1982-01-25, 1300, 10  
7788, SCOTT, 1982-12-08, 3000, 20  
7876, ADAMS, 1983-01-12, 1100, 20  
7844, TURNER, 1984-10-08, 1500, 30
```

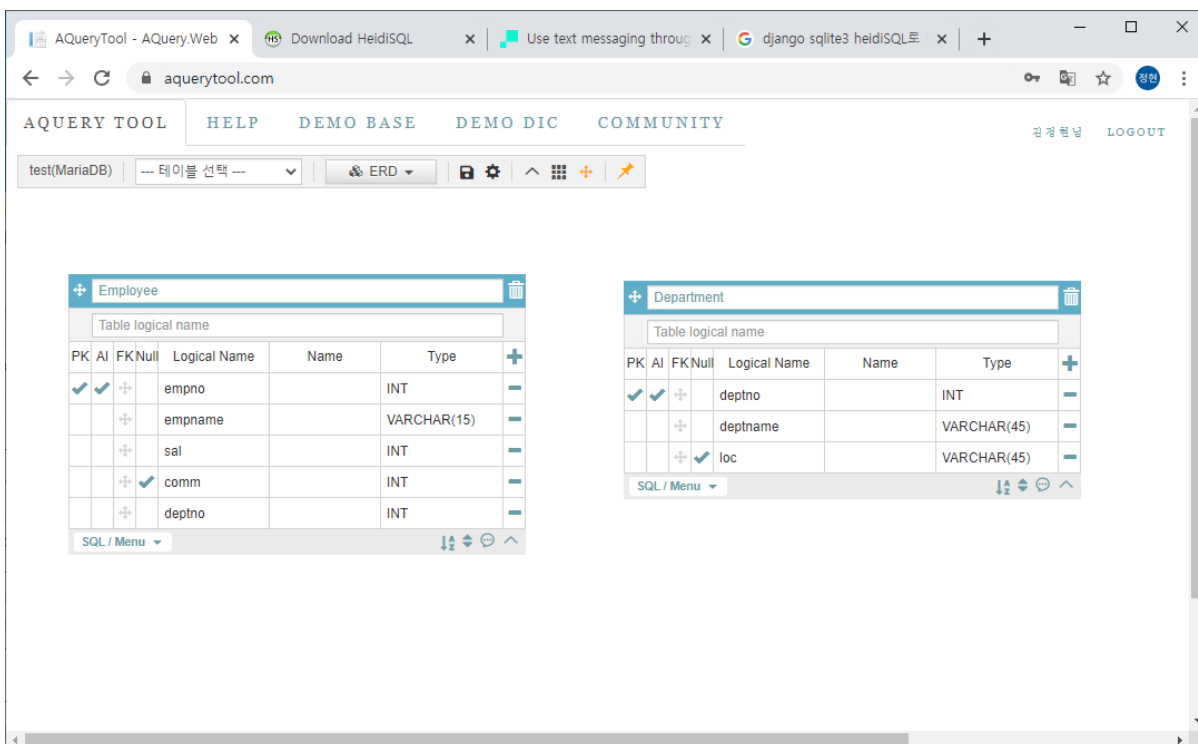
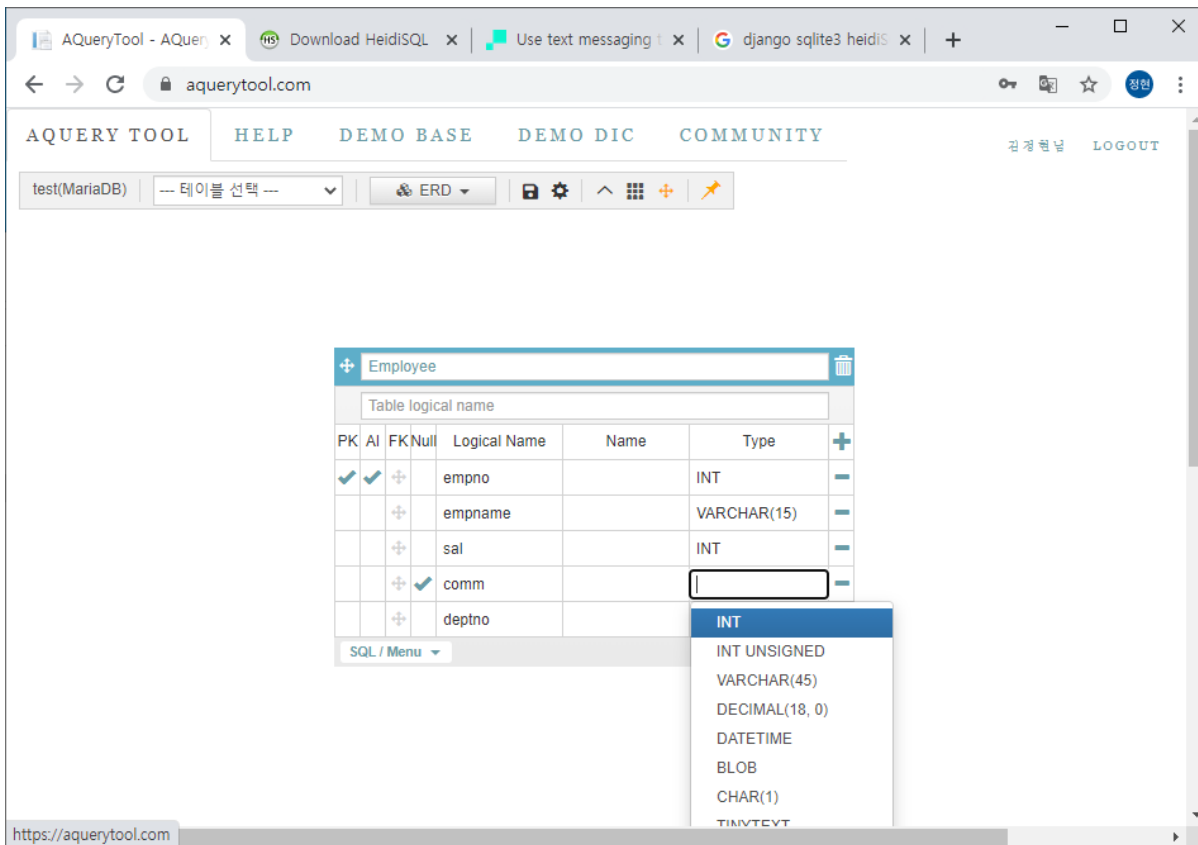
```
>>> for d in Emp.objects.order_by('-sal') :  
...     print(d)
```

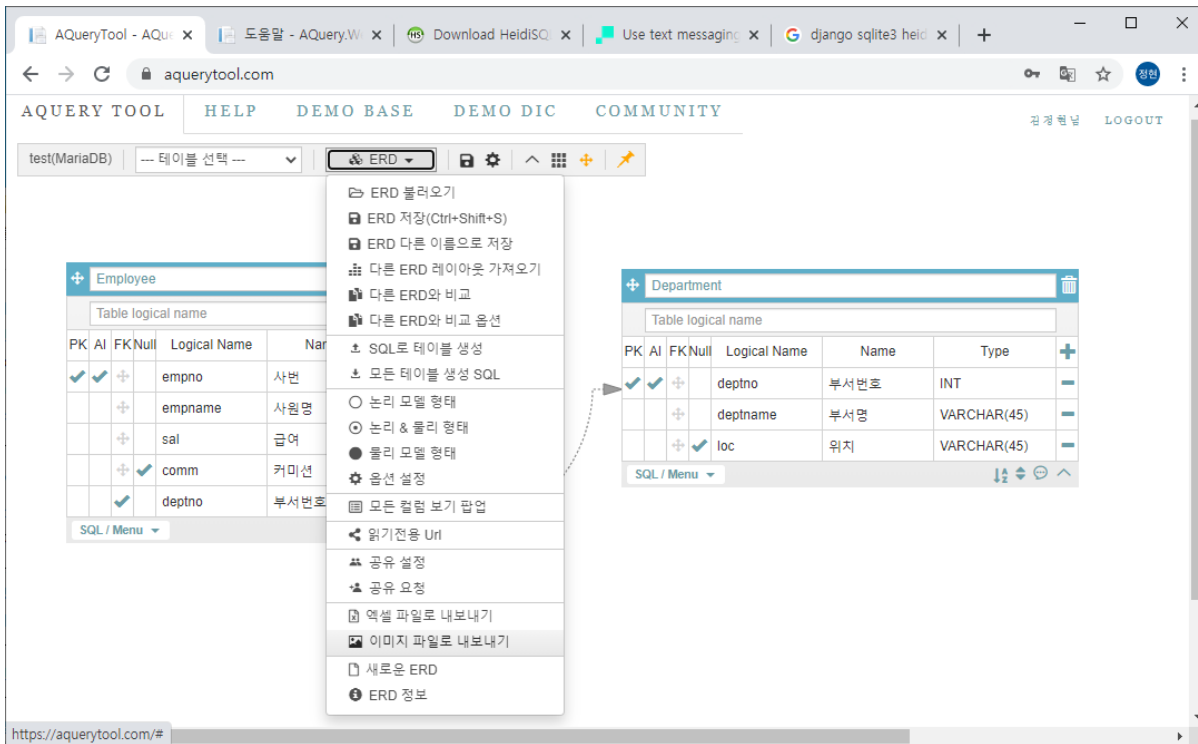
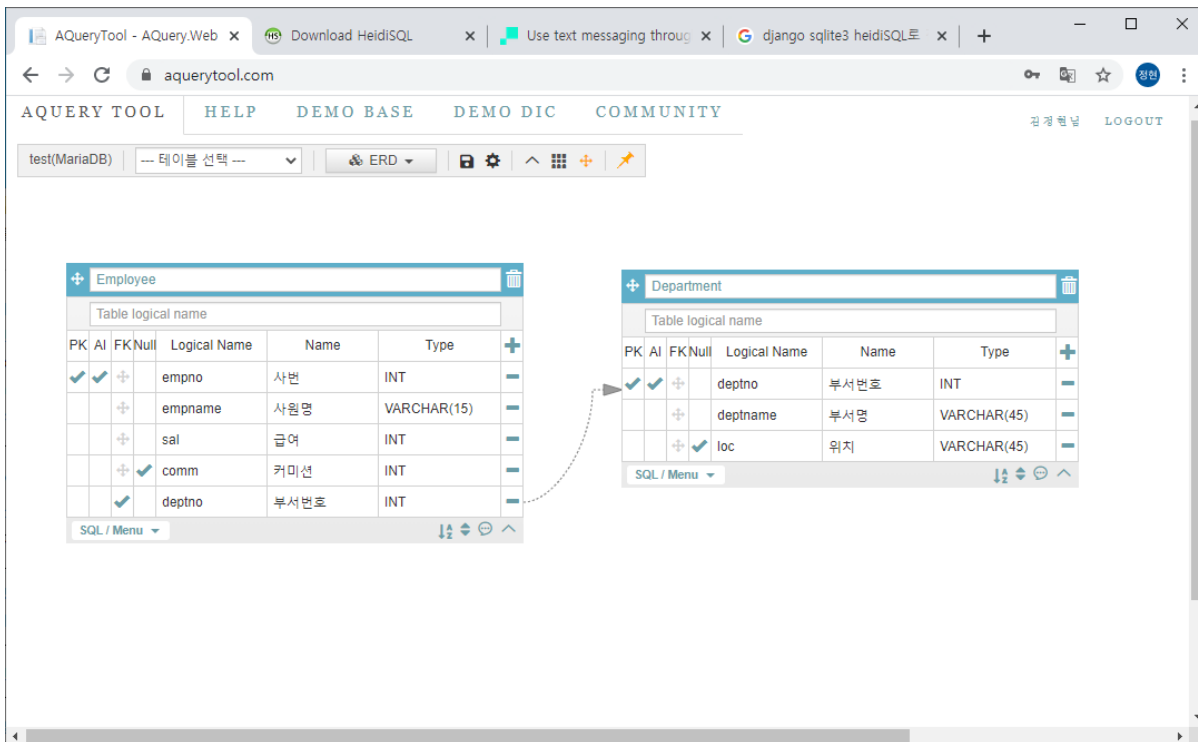
```
...  
7839, KING, 1981-11-17, 5000, 10  
7788, SCOTT, 1982-12-08, 3000, 20  
7902, FORD, 1981-12-13, 3000, 20  
7566, JONES, 1981-03-02, 2975, 20  
7698, BLAKE, 1981-05-01, 2850, 30  
7782, CLARK, 1981-09-06, 2450, 10  
7499, ALLEN, 1981-02-20, 1600, 30  
7844, TURNER, 1984-10-08, 1500, 30  
7934, MILLER, 1982-01-25, 1300, 10  
7521, WARD, 1981-02-03, 1250, 30  
7654, MARTIN, 1981-10-22, 1250, 30  
7876, ADAMS, 1983-01-12, 1100, 20  
7900, JAMES, 1981-12-03, 950, 30  
7369, SMITH, 1980-12-17, 800, 20
```

## [ AQueryTool로 ERD 그리기 ]









[ 제공된 forthapp 과 visitorapp 그리고 accountapp 을 등록하고 migrate 하기 ]

