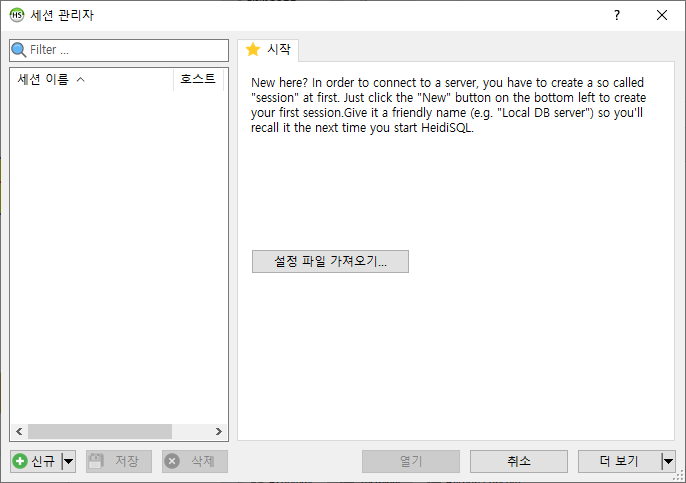
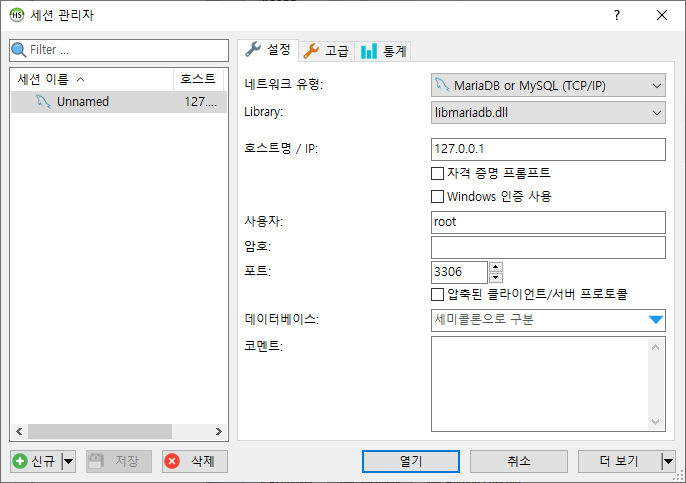
[ HeidiSQL ]

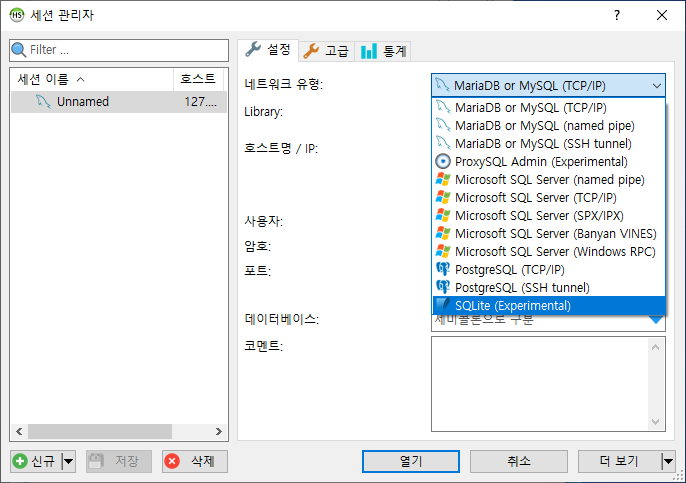
HeidiSQL은 이전에는 ‘MySQL Front’로 알려졌던 제품으로서 MySQL, MariaDB 그리고 SQLite 등 DBMS를 직접 접속하여 사용하려는 경우에 선택할 수 있는 DBMS 클라이언트 무료 프로그램이다.



신규 버튼을 클릭하면 다음 화면의 서브 윈도우가 출력된다.

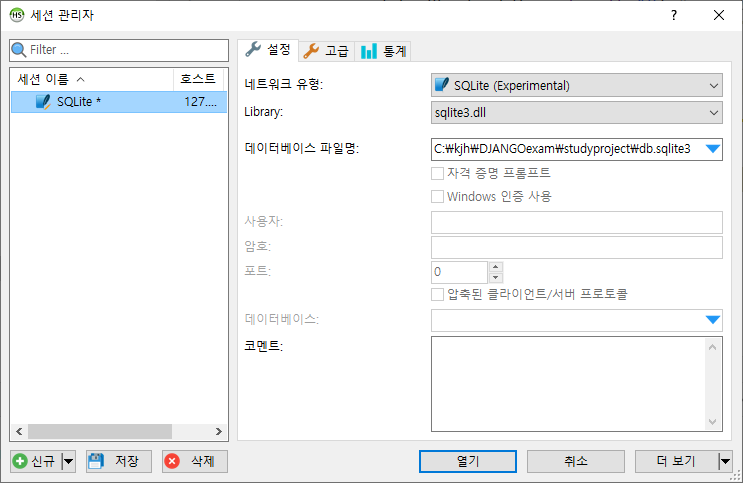


네트워크 유형에서 SQLite 항목을 선택한다.

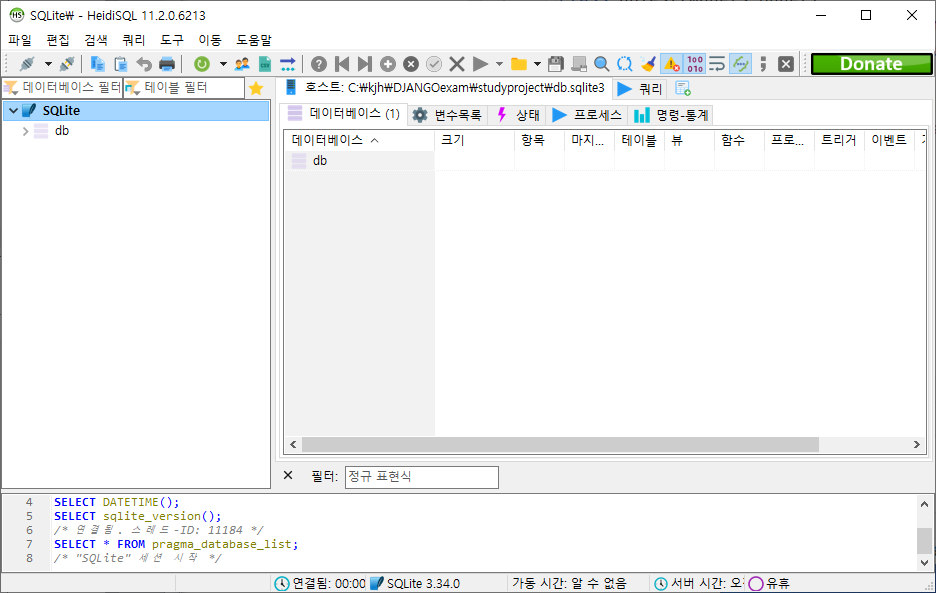


네트워크 유형에서 SQLite 항목을 선택하면 Library 에 sqlite3.dll 이 자동 설정되는 것을 볼 수 있다.

데이터베이스 파일명으로 studyproject 의 db.sqlite3 를 찾아서 설정한다.



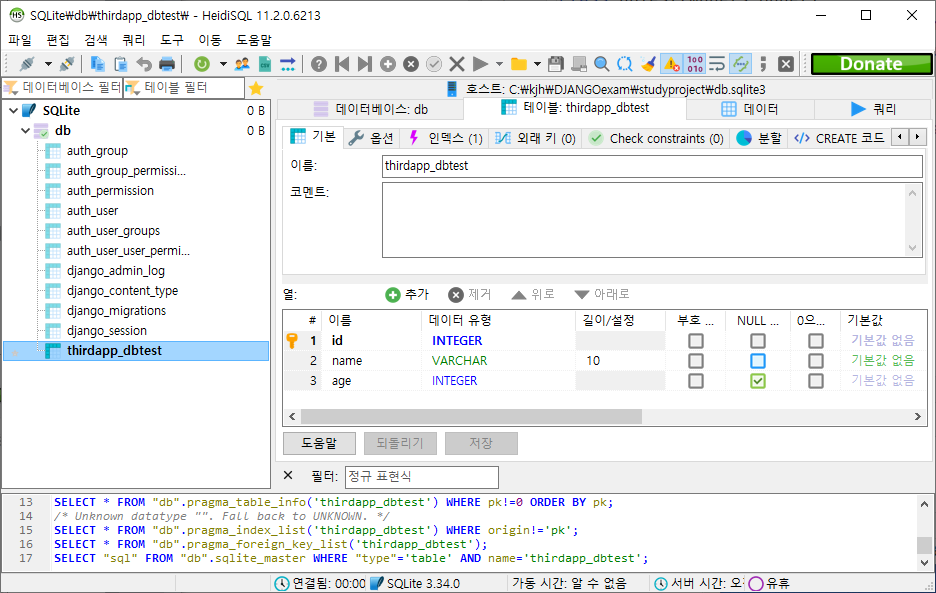
오픈된 studyproject 의 db.sqlite3 에는 db 라는 이름의 데이터베이스가 생성되어 있는 것을 볼 수 있다.



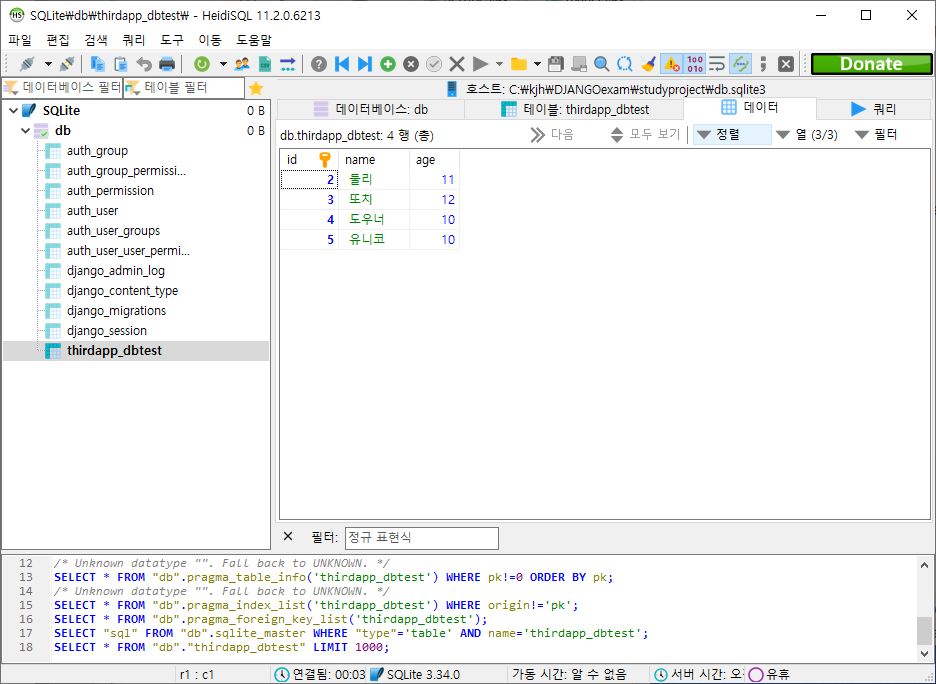
db 라는 이름의 데이터베이스에는 python manage.py migrate 명령의 실행에 의해서 생성된 테이블들이 여러 개 존재하며

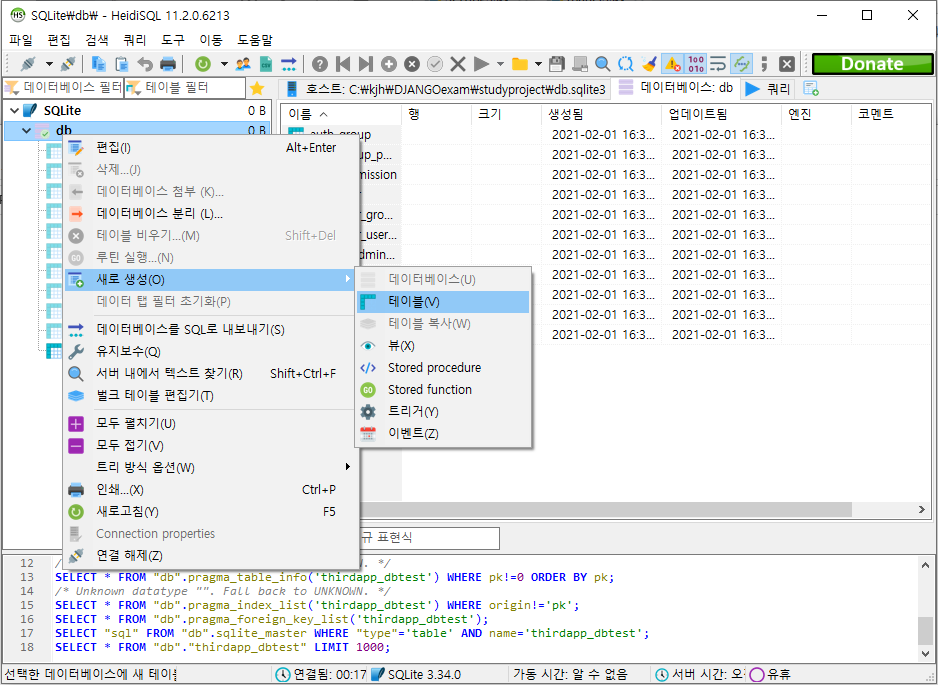
그 중에서 제일 아래에 있는 thirdapp\_dbtest 가 DBTest 라는 모델 클래스에 의해서 생성된 테이블이다.

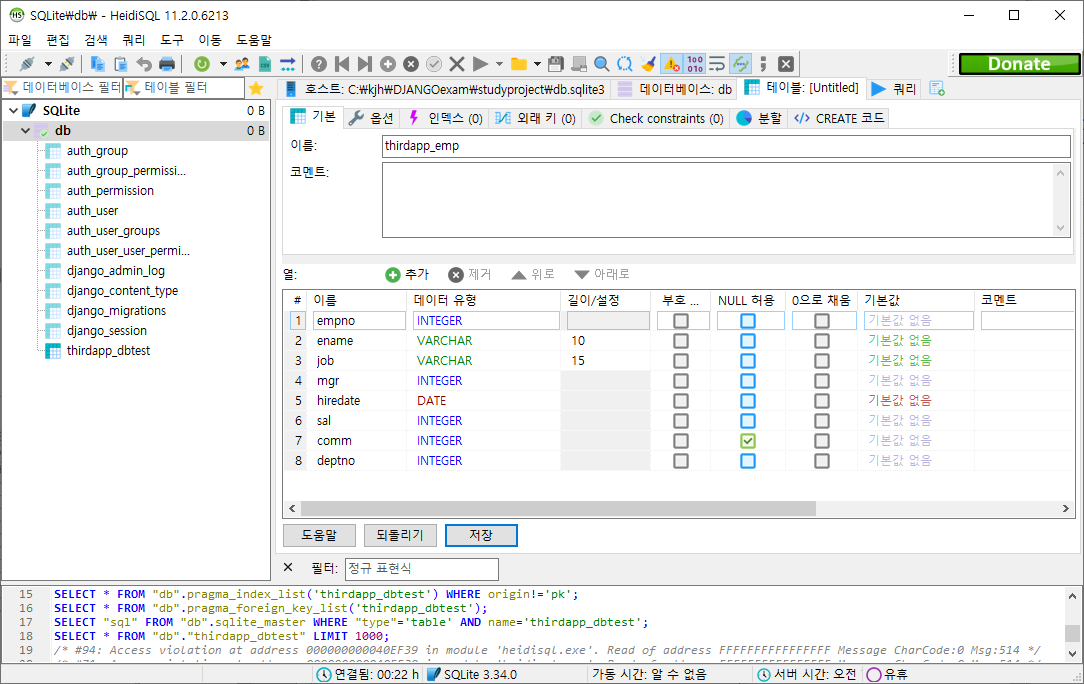
오른쪽에서 이 테이즐의 컬럼 사양을 채크해 볼 수 있다.



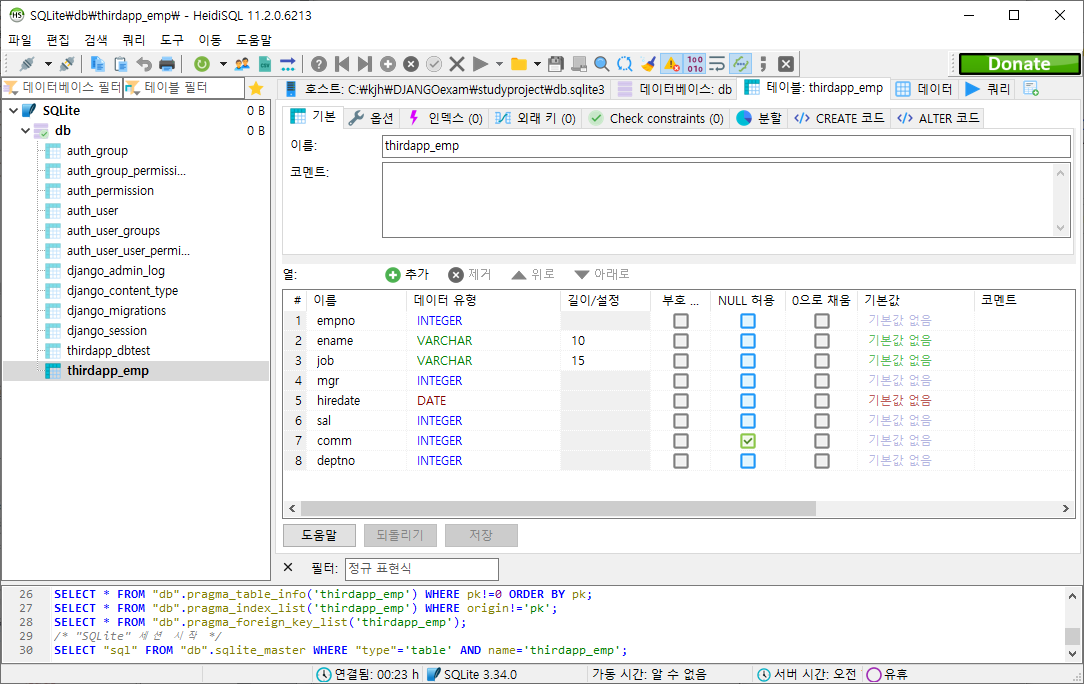
데이터라는 버튼을 클릭하면 thirdapp\_dbtest 테이블에 저장된 데이터들을 직접 확인할 수 있다.



다음 메뉴들을 클릭해서 새로운 테이블을 여기서 직접 생성해 본다. 

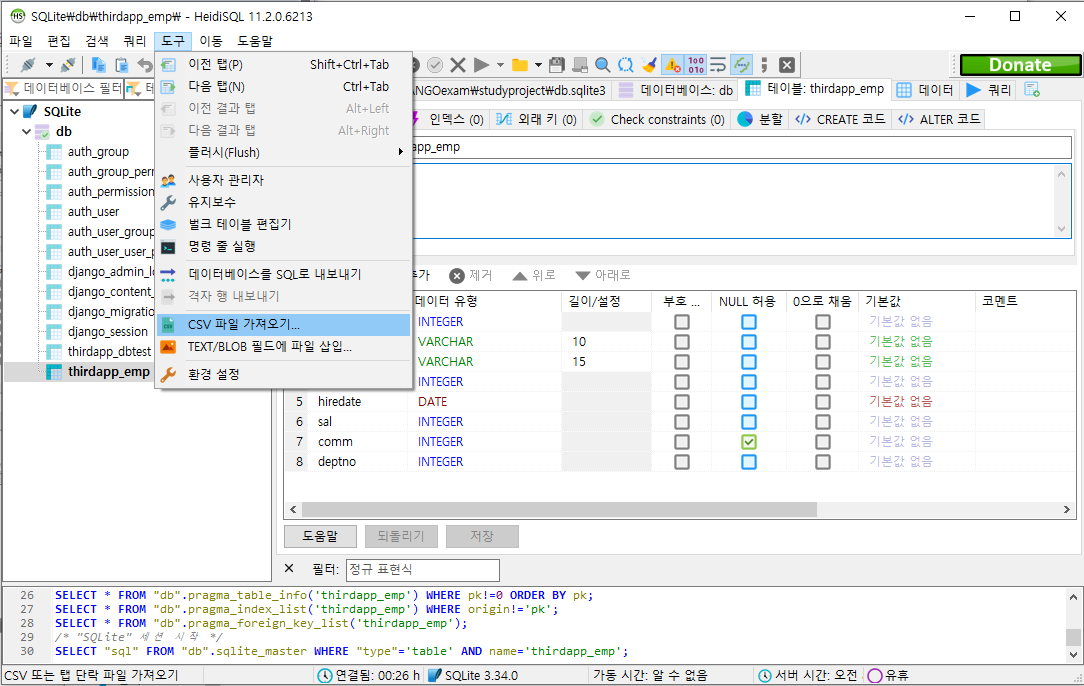


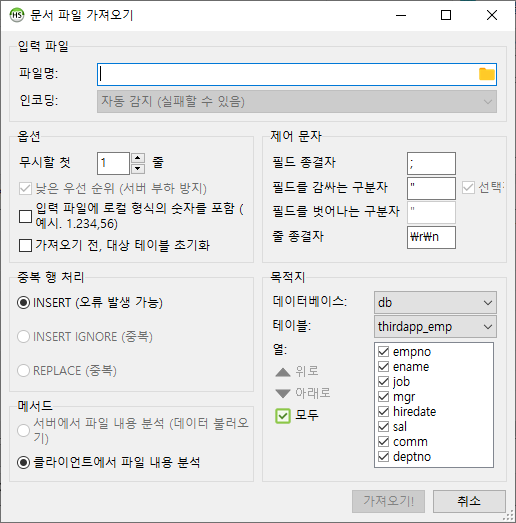
다음과 같이 thirdapp\_emp 라는 테이블이 생성된 것을 볼 수 있다.

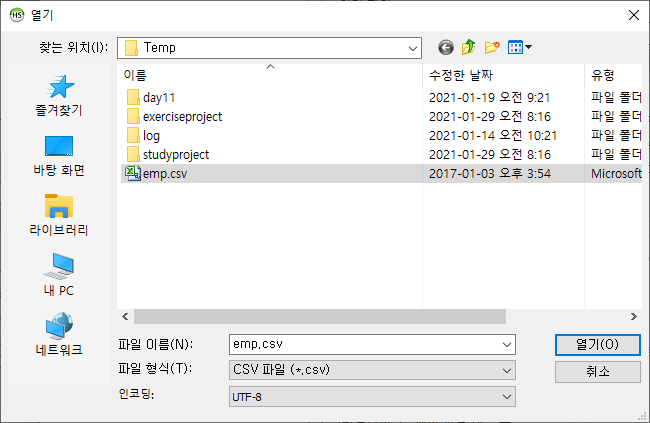


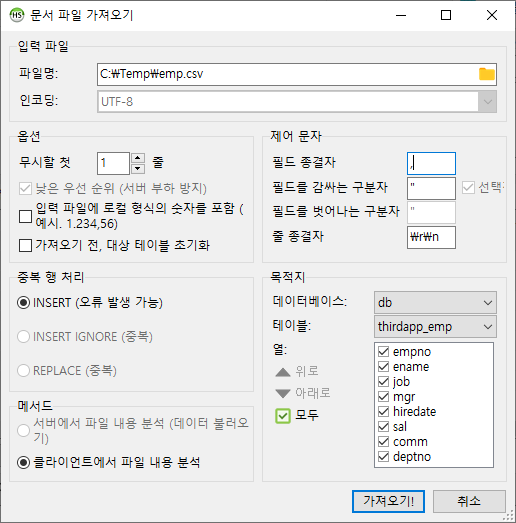
생성된 테이블에 csv 파일의 내용을 저장해 보자.

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



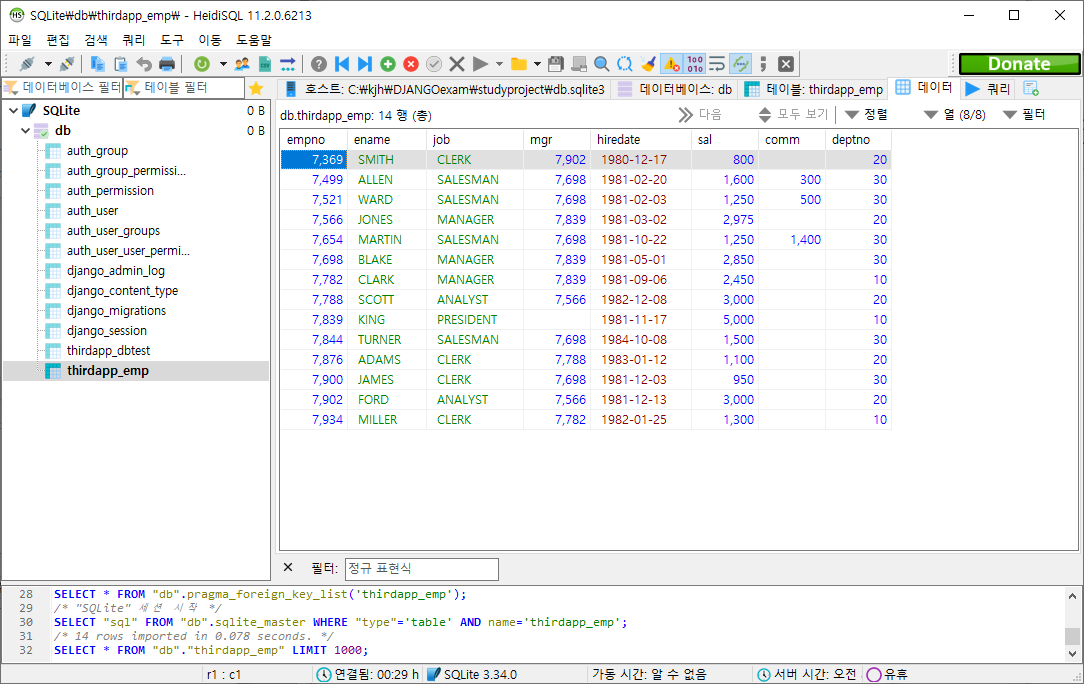






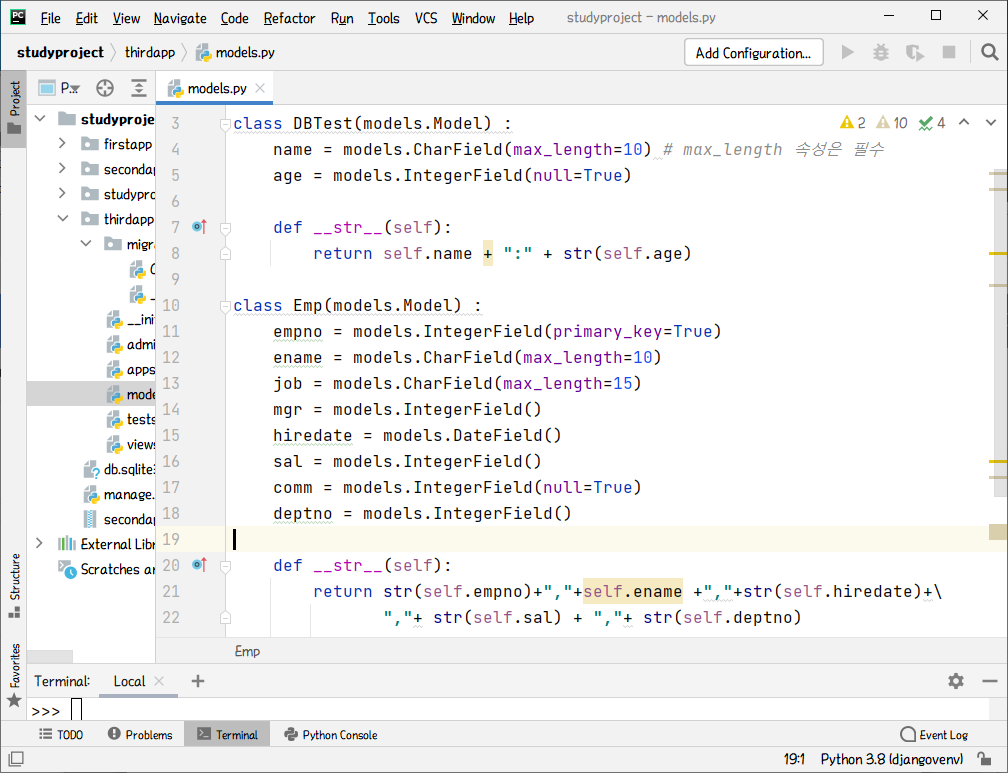
emp.csv의 경우 데이터 구분자가 , 이므로 ; 을 , 로 변경한다.

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



thirdapp\_emp 데이블의 내용을 파이썬 프로그램에서 다루기 위해 thirdapp 의 models.py 파일에 다음과 같이 Emp

라는 클래스를 생성한다.

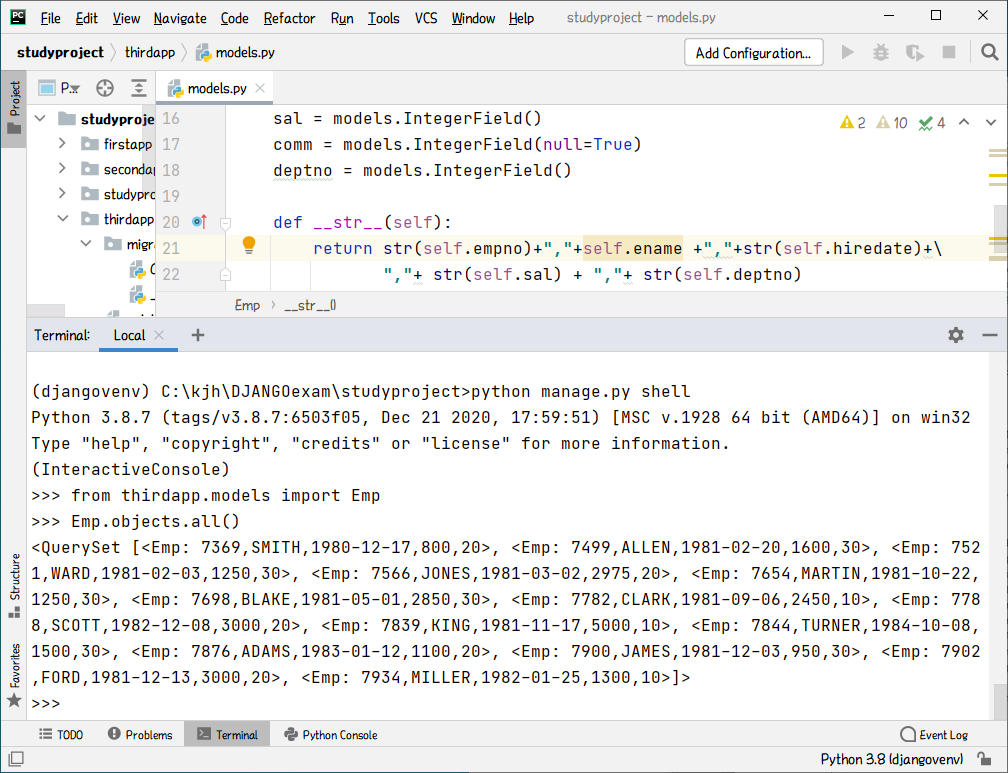


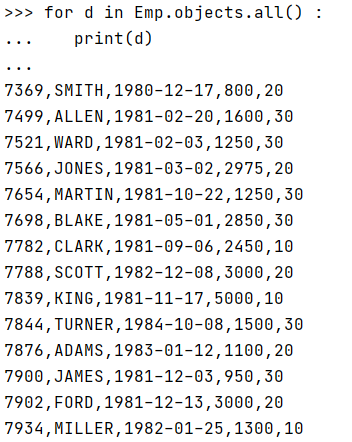
djangovenv 가상환경 기반의 터미널에서 python manage.py shell 명령을 실행시키고 인터랙티브 실행모드를 기동시킨

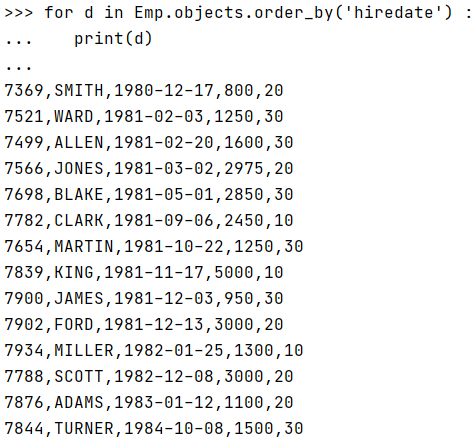
후에 다음 명령들을 실행시켜 본다.

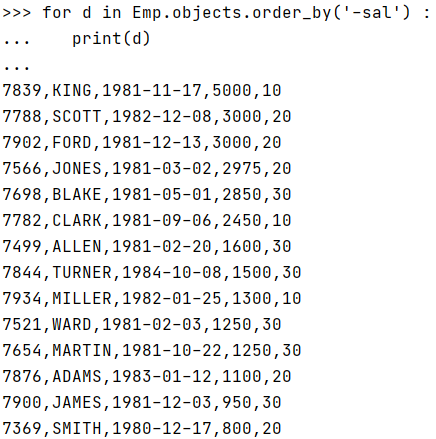
from thirdapp.models import Emp

Emp.objects.all()

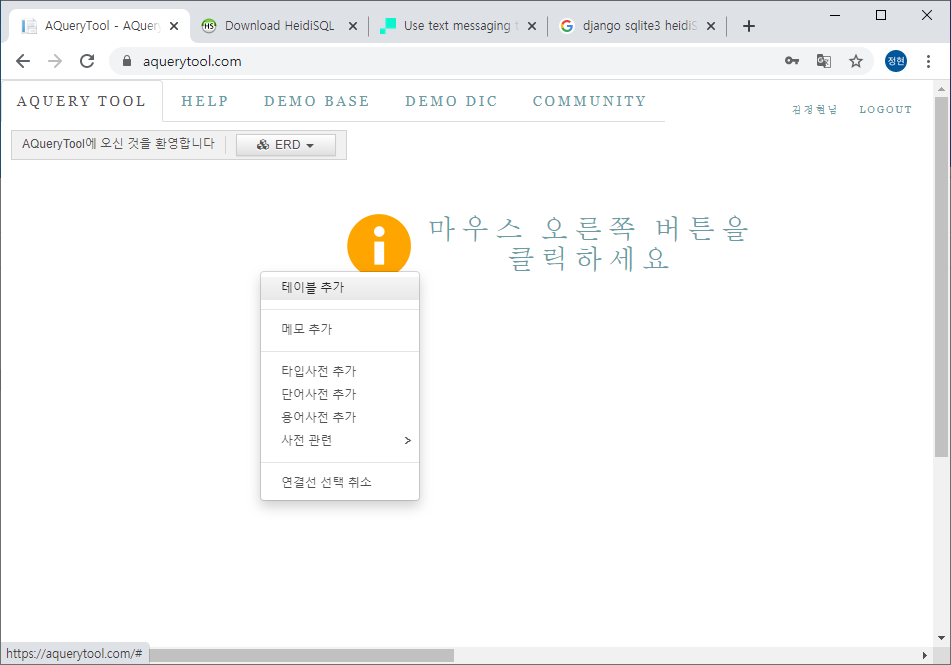


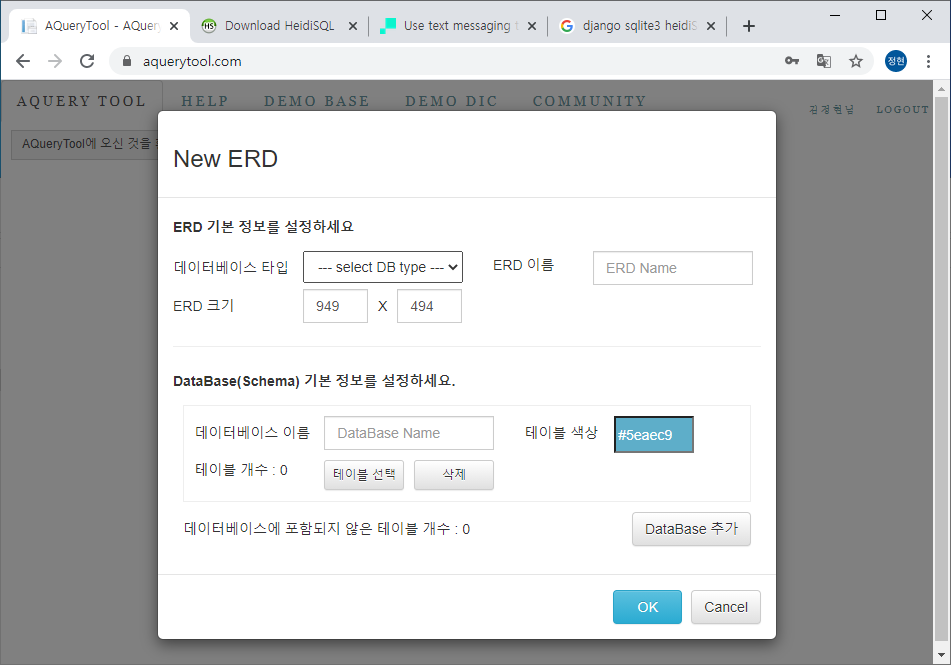


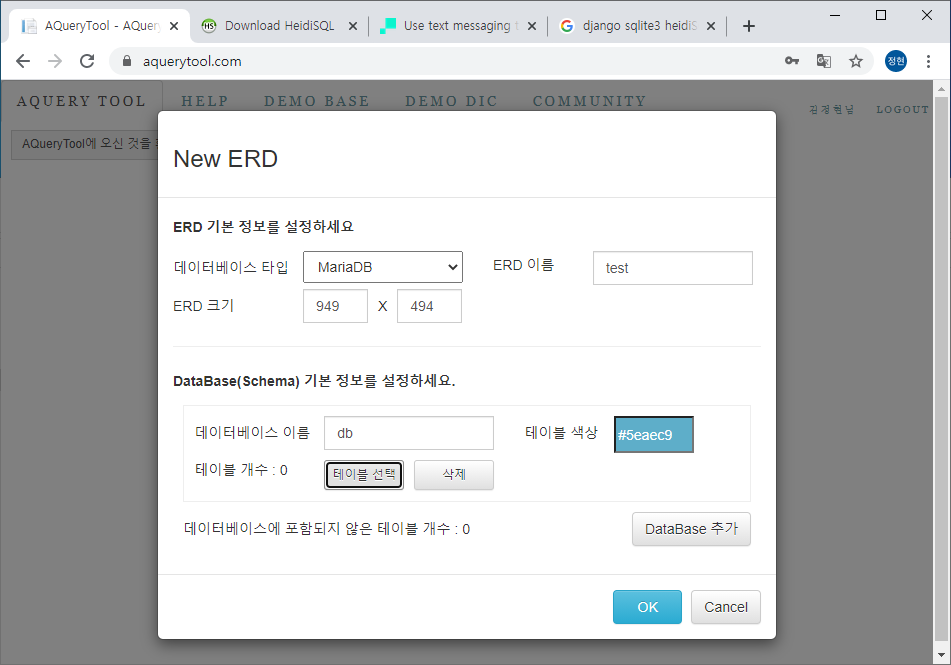


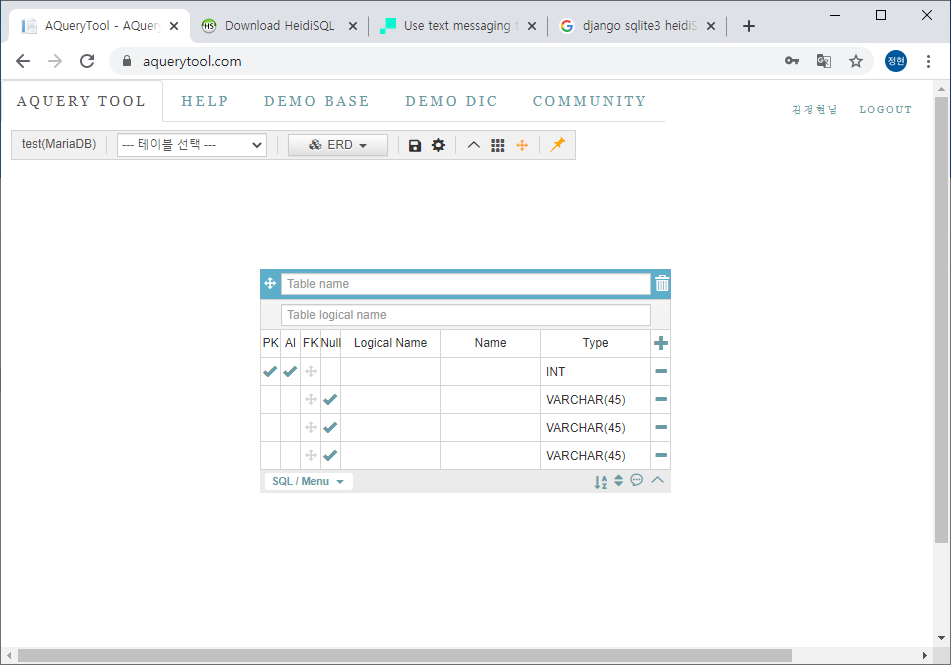


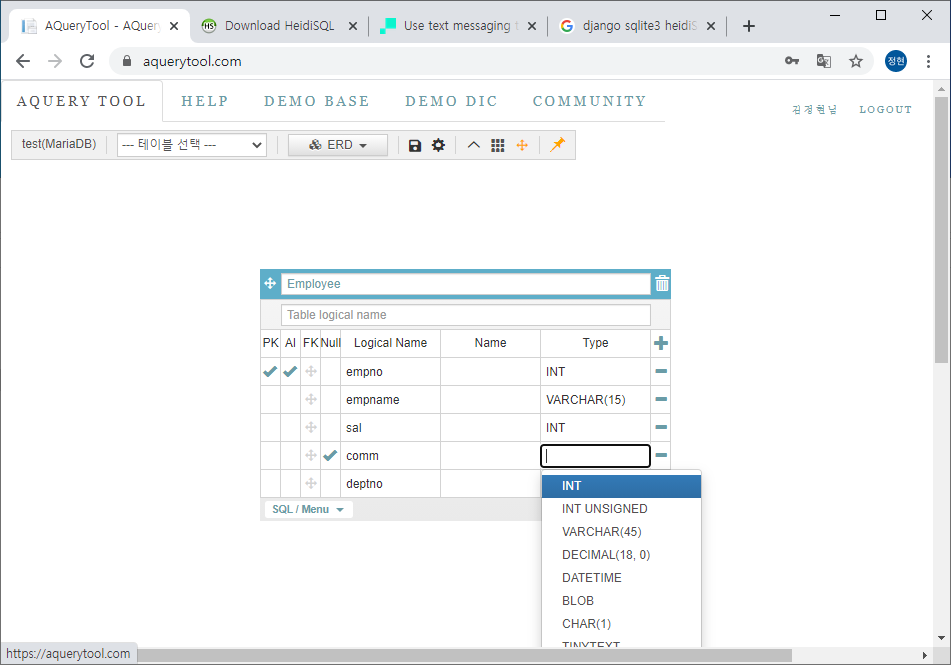
[ AqueryTool로 ERD 그리기 ]

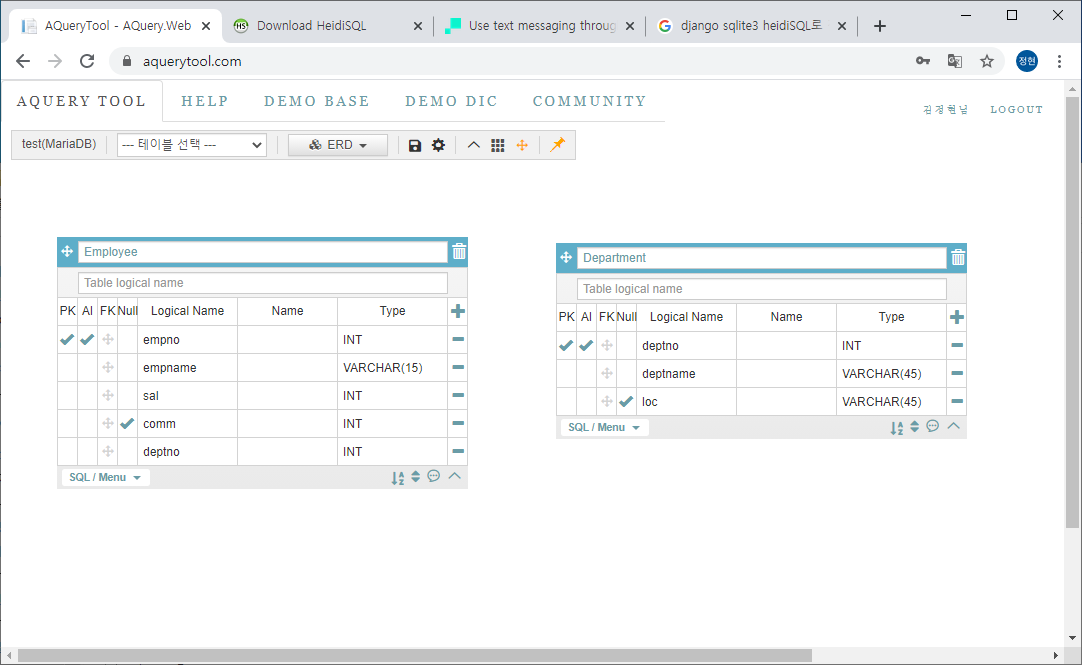


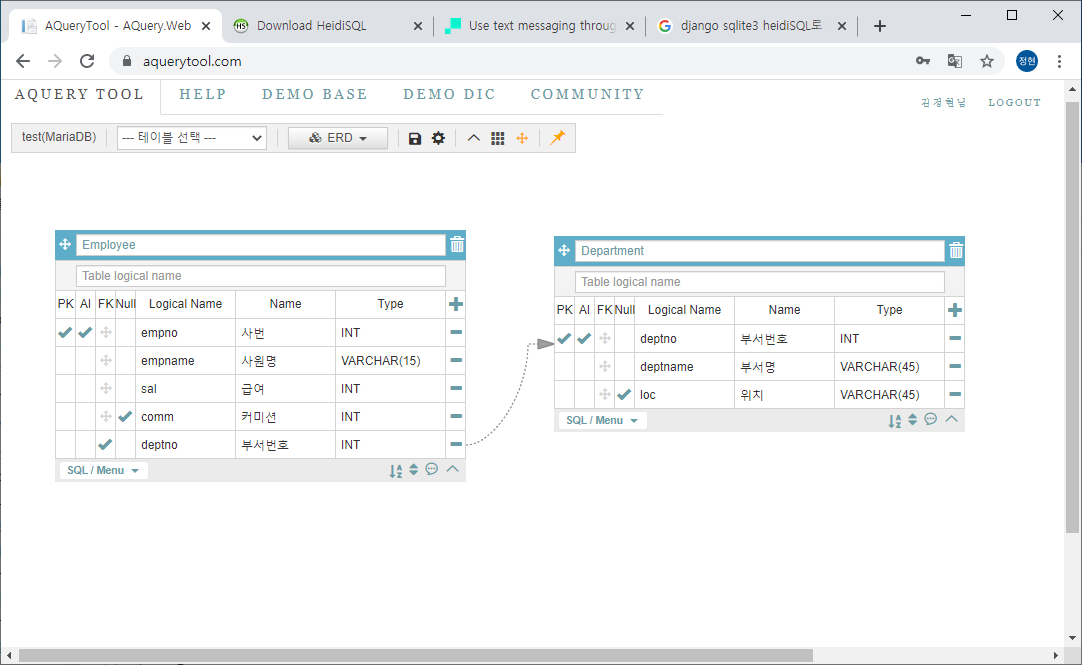


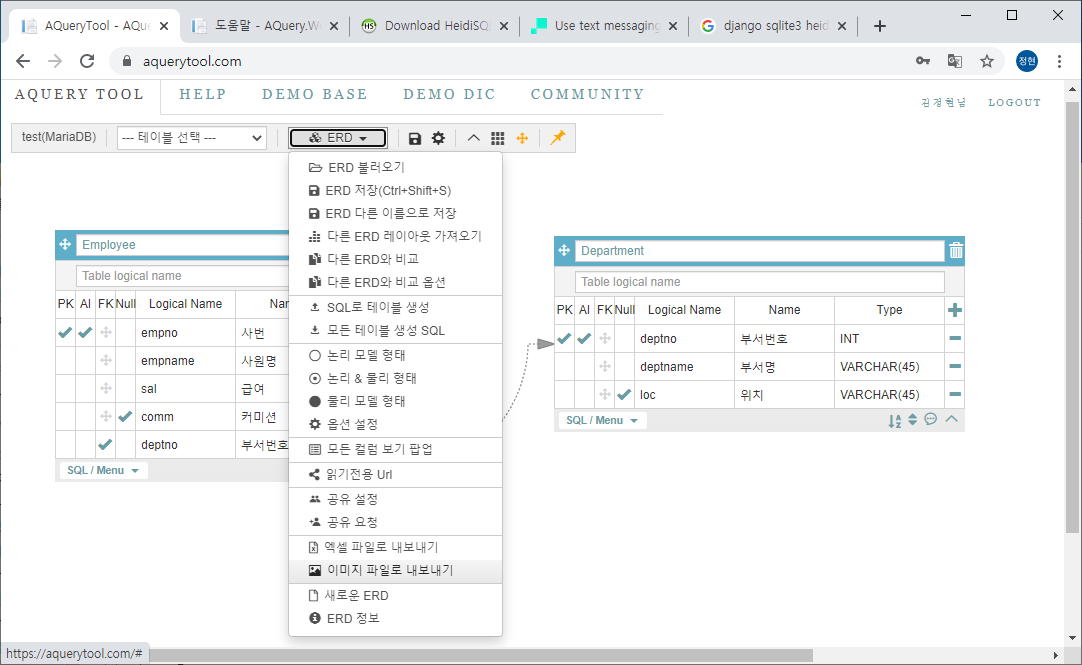












[ 제공된 forthapp 과 visitorapp 그리고 accountapp 을 등록하고 migrate 하기 ]

python manage.py makemigrations forthapp

python manage.py makemigrations visitorapp

