SQL Practice, NL2SQL

VLDB Lab.

Professor Sang-Won Lee

SQL Practice

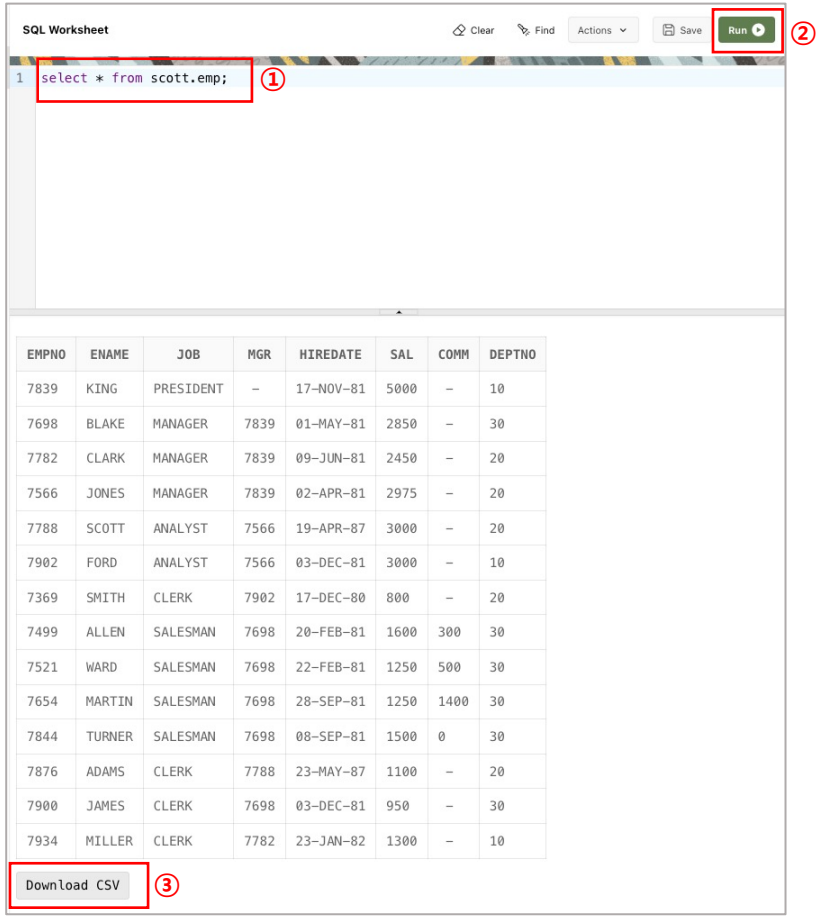
Scott Schema

EMP	
empno	Number(4)
ename	Varchar2(10)
job	Varchar2(9)
mgr	Number(4)
hiredate	date
sal	Number(7,2)
comm	Number(7,2)
deptno	Number(2)

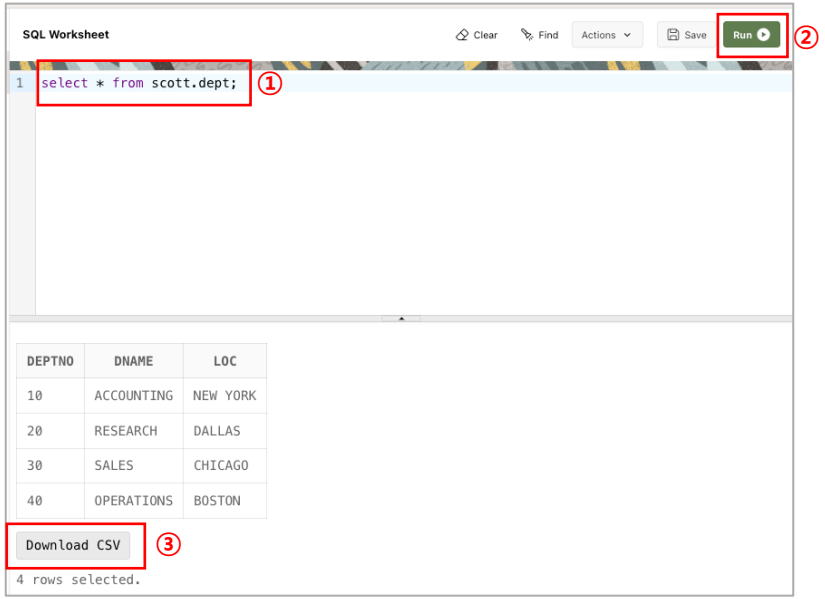
DEPT	
deptno	Number(2)
dname	Varchar2(14)
loc	Varchar2(13)

Export CSV File

1. LiveSQL 로그인 (https://livesql.oracle.com/)
2. Emp 테이블과 dept 테이블로부터 데이터를 선택 후 'Download CSV' 버튼을 클릭



EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	30
7782	CLARK	MANAGER	7839	09-JUN-81	2450	-	20
7566	JONES	MANAGER	7839	02-APR-81	2975	-	20
7788	SCOTT	ANALYST	7566	19-APR-87	3000	-	20
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	10
7369	SMITH	CLERK	7902	17-DEC-80	800	-	20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	23-MAY-87	1100	-	20
7900	JAMES	CLERK	7698	03-DEC-81	950	-	30
7934	MILLER	CLERK	7782	23-JAN-82	1300	-	10

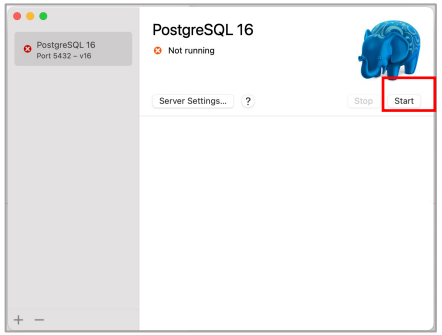


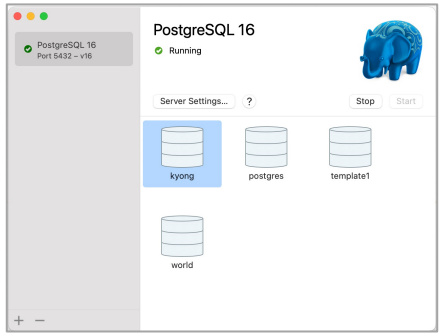
DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Start Postgres

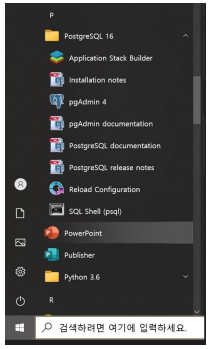
1. Postgres 실행

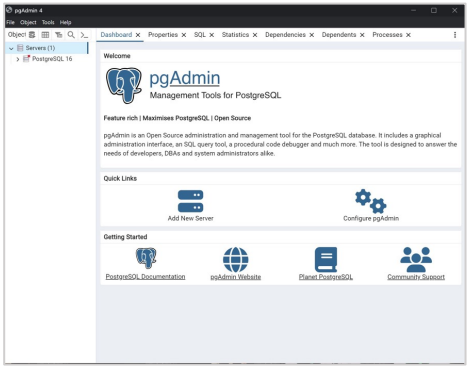
[Mac]



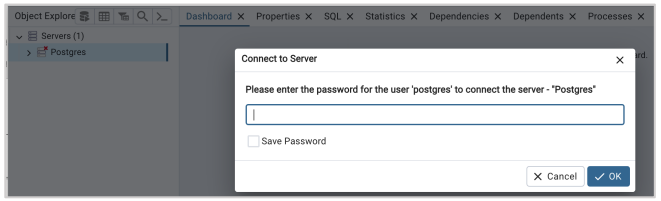


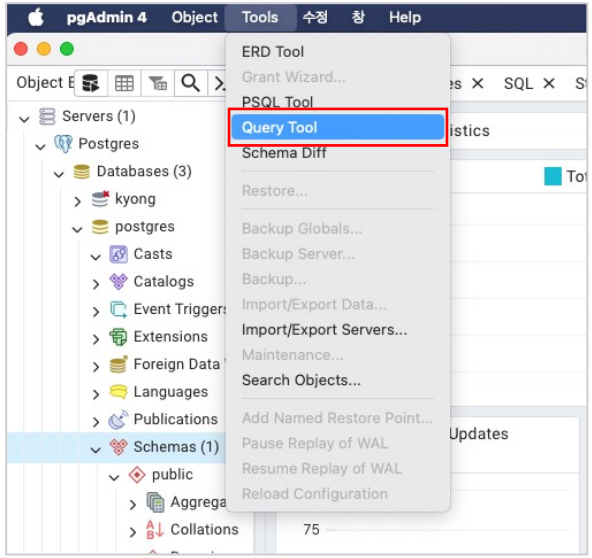
[Windows]

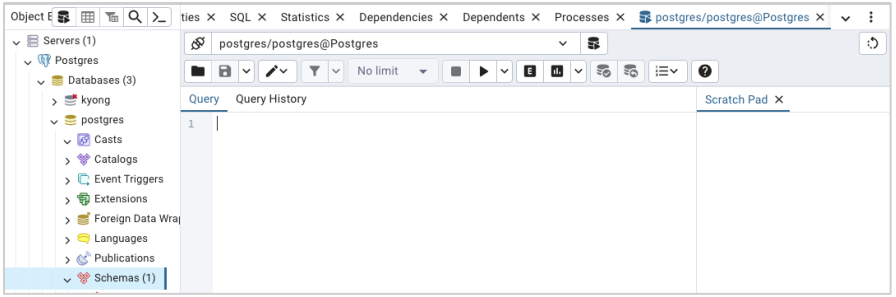




2. pgAdmin 실행 후 Query Tool 클릭



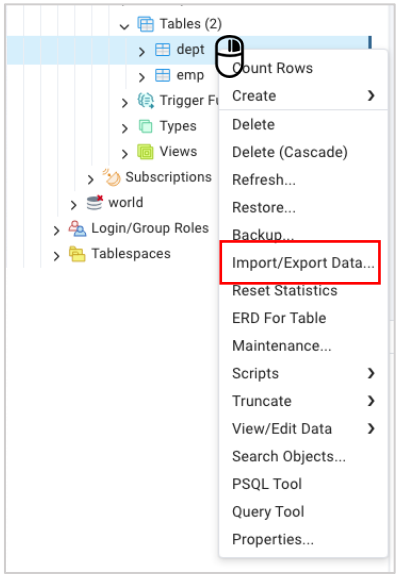


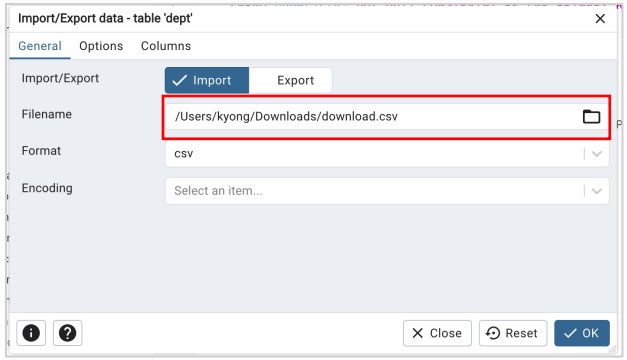


Now you can write sql query!

Import CSV File

1. 우클릭 후 'Import/Export Data…' 클릭





2. 1번 프로세스를 emp table에 대해서도 반복 수행

CREATE TABLE

1. Dept, emp 테이블 생성

postgres/postgres@Postgres

Query Query History

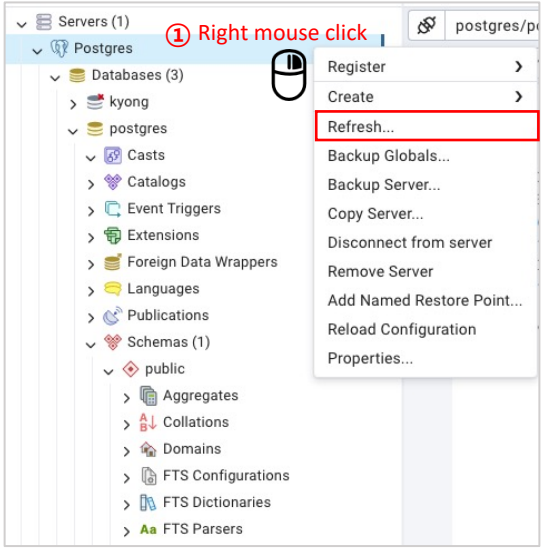
```
CREATE TABLE if not exists DEPT
(
    DEPTNO numeric(2) not null CONSTRAINT PK_DEPT PRIMARY KEY,
    DNAME VARCHAR(14) ,
    LOC VARCHAR(13)
)
```

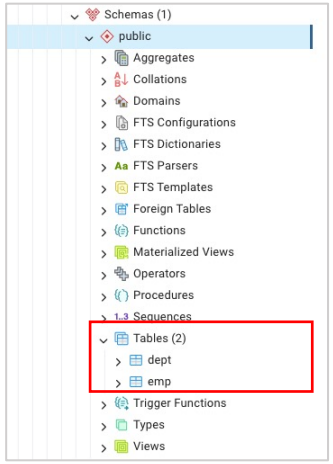
postgres/postgres@Postgres

Query Query History

```
CREATE TABLE if not exists EMP
(
    EMPNO numeric(4) not null CONSTRAINT PK_EMP PRIMARY KEY,
    ENAME VARCHAR(10),
    JOB VARCHAR(9),
    MGR numeric(4),
    HIREDATE DATE,
    SAL numeric(7,2),
    COMM numeric(7,2),
    DEPTNO numeric(2) CONSTRAINT FK_DEPTNO REFERENCES DEPT
);
```

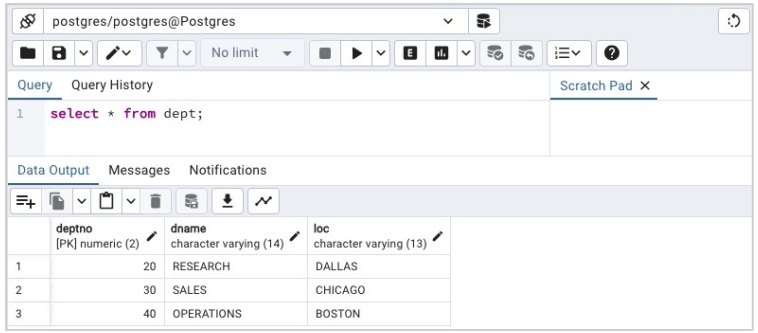
If you refresh postgres server, you can now see two tables.



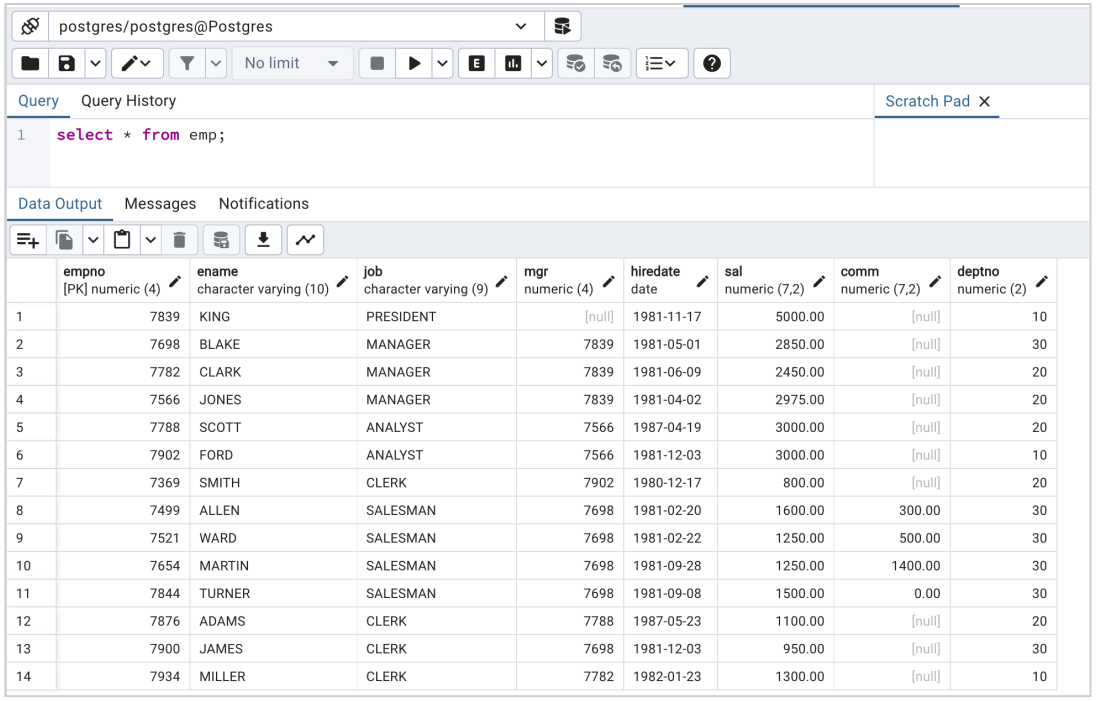


SELECT statement

1. 테이블 생성을 완료하면 다음과 같이 SELECT문으로 테이블 내 데이터 검색이 가능함.



	deptno [PK] numeric (2)	dname character varying (14)	loc character varying (13)
1	20	RESEARCH	DALLAS
2	30	SALES	CHICAGO
3	40	OPERATIONS	BOSTON



	empno [PK] numeric (4)	ename character varying (10)	job character varying (9)	mgr numeric (4)	hiredate date	sal numeric (7,2)	comm numeric (7,2)	deptno numeric (2)
1	7839	KING	PRESIDENT	[null]	1981-11-17	5000.00	[null]	10
2	7698	BLAKE	MANAGER	7839	1981-05-01	2850.00	[null]	30
3	7782	CLARK	MANAGER	7839	1981-06-09	2450.00	[null]	20
4	7566	JONES	MANAGER	7839	1981-04-02	2975.00	[null]	20
5	7788	SCOTT	ANALYST	7566	1987-04-19	3000.00	[null]	20
6	7902	FORD	ANALYST	7566	1981-12-03	3000.00	[null]	10
7	7369	SMITH	CLERK	7902	1980-12-17	800.00	[null]	20
8	7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30
9	7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30
10	7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30
11	7844	TURNER	SALESMAN	7698	1981-09-08	1500.00	0.00	30
12	7876	ADAMS	CLERK	7788	1987-05-23	1100.00	[null]	20
13	7900	JAMES	CLERK	7698	1981-12-03	950.00	[null]	30
14	7934	MILLER	CLERK	7782	1982-01-23	1300.00	[null]	10

SELECT statement

2. 기타 고급 쿼리들도 모두 실행 가능함.

Query Query History

```
select * from emp where job not in('PRESIDENT','MANAGER');
```

Query Query History

```
select max(sal) from emp;
```

Query Query History

```
select * from emp where deptno in (
    select deptno from emp where ename='JAMES'
    )
```

Query Query History

```
select emp.ename, emp.deptno, dept.loc from emp
    join dept on emp.deptno = dept.deptno
    where emp.ename = 'KING'
```

INSERT statement

- **Emp 테이블에 새로운 튜플 삽입**

[Syntax]

```
INSERT INTO table_name(column1, column2, …)
VALUES (value1, value2, …);
```

[Example]

postgres/postgres@Postgres

No limit

Query | Query History

```
INSERT INTO emp VALUES
(1234, 'BKMS1', 'MANAGER', null, '2024-03-26', 3000.00, null, 10);
```

postgres/postgres@Postgres

No limit

Query | Query History

```
select * from emp;
```

Data Output | Messages | Notifications

	empno [PK] numeric (4)	ename character varying (10)	job character varying (9)	mgr numeric (4)	hiredate date	sal numeric (7,2)	comm numeric (7,2)	deptno numeric (2)
1	7839	KING	PRESIDENT	[null]	1981-11-17	5000.00	[null]	10
2	7698	BLAKE	MANAGER	7839	1981-05-01	2850.00	[null]	30
3	7782	CLARK	MANAGER	7839	1981-06-09	2450.00	[null]	20
4	7566	JONES	MANAGER	7839	1981-04-02	2975.00	[null]	20
5	7788	SCOTT	ANALYST	7566	1987-04-19	3000.00	[null]	20
6	7902	FORD	ANALYST	7566	1981-12-03	3000.00	[null]	10
7	7369	SMITH	CLERK	7902	1980-12-17	800.00	[null]	20
8	7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30
9	7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30
10	7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30
11	7844	TURNER	SALESMAN	7698	1981-09-08	1500.00	0.00	30
12	7876	ADAMS	CLERK	7788	1987-05-23	1100.00	[null]	20
13	7900	JAMES	CLERK	7698	1981-12-03	950.00	[null]	30
14	7934	MILLER	CLERK	7782	1982-01-23	1300.00	[null]	10
15	1234	BKMS1	MANAGER	[null]	2024-03-26	3000.00	[null]	10

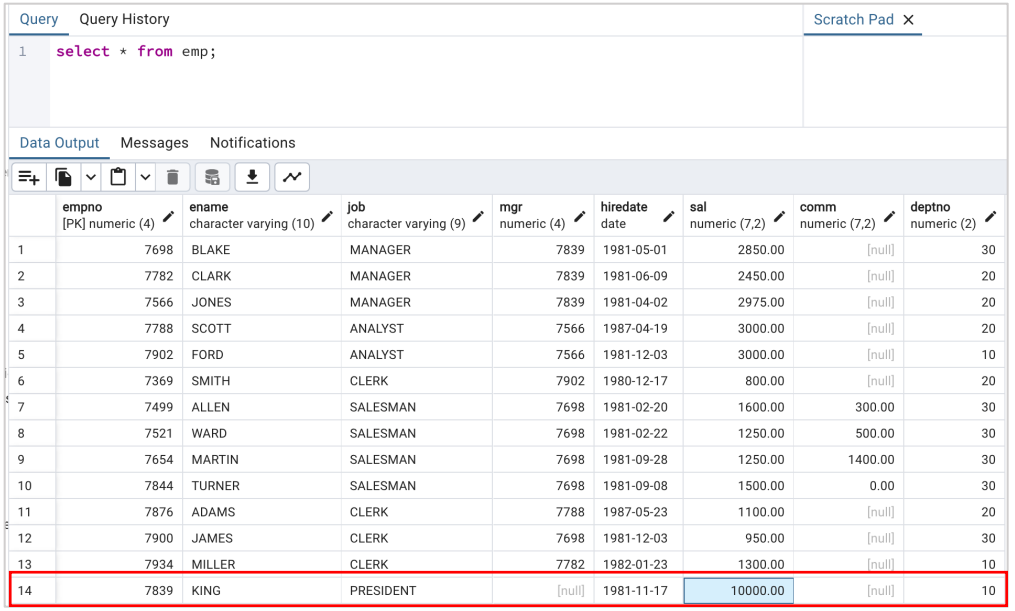
Update statement

- **Emp 테이블 튜플 업데이트**

[Syntax]

```
UPDATE table
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

[Example]



	empno [PK] numeric (4)	ename character varying (10)	job character varying (9)	mgr numeric (4)	hiredate date	sal numeric (7,2)	comm numeric (7,2)	deptno numeric (2)
1	7698	BLAKE	MANAGER	7839	1981-05-01	2850.00	[null]	30
2	7782	CLARK	MANAGER	7839	1981-06-09	2450.00	[null]	20
3	7566	JONES	MANAGER	7839	1981-04-02	2975.00	[null]	20
4	7788	SCOTT	ANALYST	7566	1987-04-19	3000.00	[null]	20
5	7902	FORD	ANALYST	7566	1981-12-03	3000.00	[null]	10
6	7369	SMITH	CLERK	7902	1980-12-17	800.00	[null]	20
7	7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30
8	7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30
9	7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30
10	7844	TURNER	SALESMAN	7698	1981-09-08	1500.00	0.00	30
11	7876	ADAMS	CLERK	7788	1987-05-23	1100.00	[null]	20
12	7900	JAMES	CLERK	7698	1981-12-03	950.00	[null]	30
13	7934	MILLER	CLERK	7782	1982-01-23	1300.00	[null]	10
14	7839	KING	PRESIDENT	[null]	1981-11-17	10000.00	[null]	10

DELETE statement

- **Emp 테이블 데이터 삭제**

[Syntax]

```
DELETE FROM
table_name
WHERE condition ;
```

[Example]

postgres/postgres@Postgres

No limit

Query | Query History

```
1 delete from emp where ename = 'BKMS1';
```

postgres/postgres@Postgres

No limit

Query | Query History

```
1 select * from emp;
```

Data Output | Messages | Notifications

	empno [PK] numeric (4)	ename character varying (10)	job character varying (9)	mgr numeric (4)	hiredate date	sal numeric (7,2)	comm numeric (7,2)	deptno numeric (2)
1	7839	KING	PRESIDENT	[null]	1981-11-17	5000.00	[null]	10
2	7698	BLAKE	MANAGER	7839	1981-05-01	2850.00	[null]	30
3	7782	CLARK	MANAGER	7839	1981-06-09	2450.00	[null]	20
4	7566	JONES	MANAGER	7839	1981-04-02	2975.00	[null]	20
5	7788	SCOTT	ANALYST	7566	1987-04-19	3000.00	[null]	20
6	7902	FORD	ANALYST	7566	1981-12-03	3000.00	[null]	10
7	7369	SMITH	CLERK	7902	1980-12-17	800.00	[null]	20
8	7499	ALLEN	SALESMAN	7698	1981-02-20	1600.00	300.00	30
9	7521	WARD	SALESMAN	7698	1981-02-22	1250.00	500.00	30
10	7654	MARTIN	SALESMAN	7698	1981-09-28	1250.00	1400.00	30
11	7844	TURNER	SALESMAN	7698	1981-09-08	1500.00	0.00	30
12	7876	ADAMS	CLERK	7788	1987-05-23	1100.00	[null]	20
13	7900	JAMES	CLERK	7698	1981-12-03	950.00	[null]	30
14	7934	MILLER	CLERK	7782	1982-01-23	1300.00	[null]	10

No ‘BKMS1’

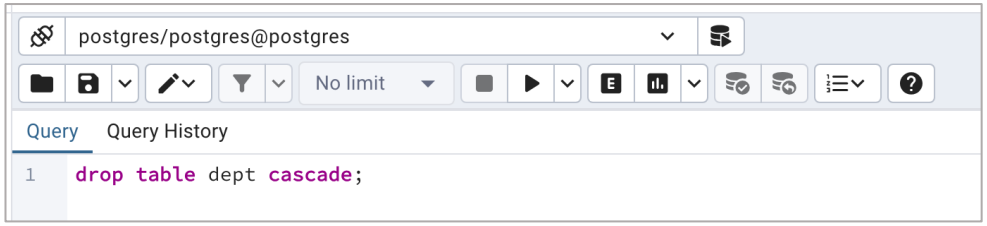
DROP Table

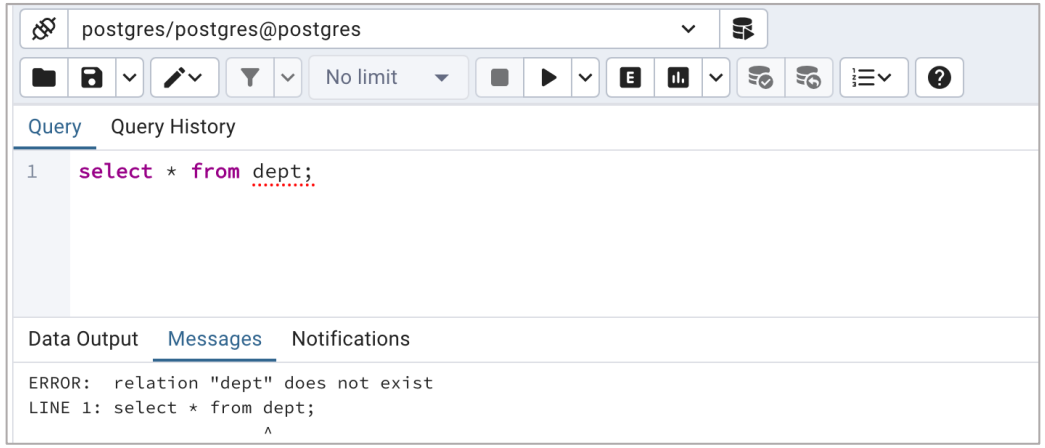
- **Drop table**

[Syntax]

```
DROP TABLE [IF EXISTS]
 table_name [CASCADE | RESTRICT];
```

[Example]





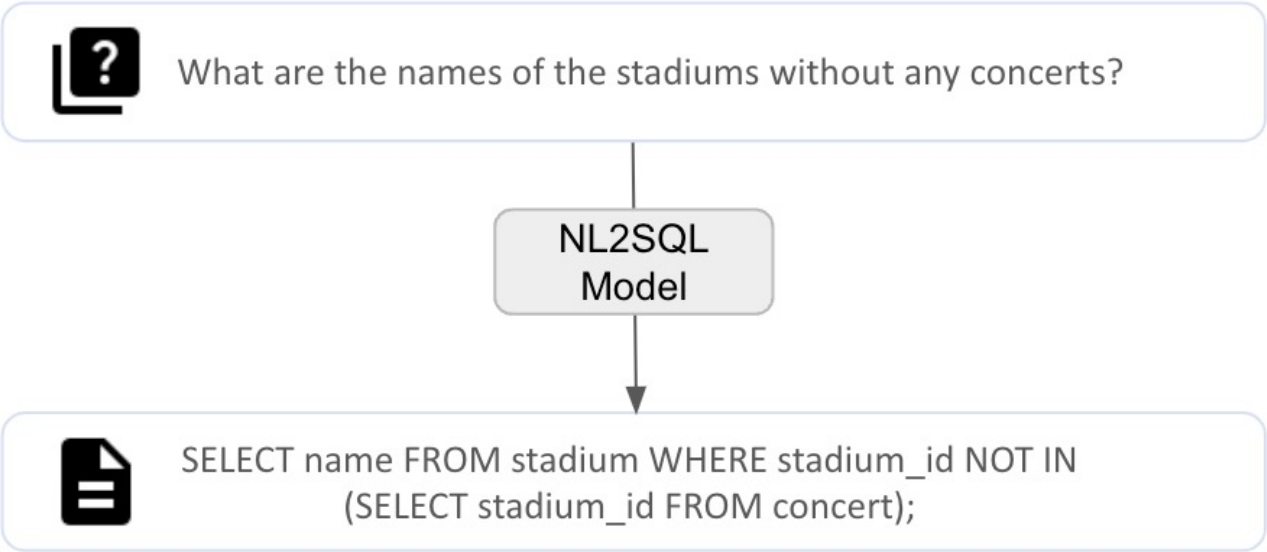
Think : Why does DROP table without cascade condition doesn't work in dept table?

Think : What is the difference between DROP table and TRUNCATE table?

NL2SQL

What is NL2SQL?

- 자연어 질문을 SQL 쿼리로 변환하는 작업
- 비전문가들이 SQL 쿼리를 쉽게 생성하고 데이터를 분석하기 쉽게 하는 것이 목표



NL2SQL Platform

- 최근에는 NL2SQL을 지원하는 많은 플랫폼들이 생겨남.
  - https://www.eversql.com/text-to-sql/
  - https://www.text2sql.ai
  - https://www.sqlai.ai/text-to-sql-ai

- 우리는 ChatGPT를 사용해서 간단하게 NL2SQL을 체험하는 것이 목적임.
  - https://chat.openai.com/

NL2SQL Prompt

- **ChatGPT 사용시, ChatGPT가 대답을 잘 생성할 수 있도록 적당한 프롬프트를 넣어주어야 함.**

- **프롬프트란?** 누군가(생성형 AI)의 특정한 작업 수행을 도우려 전달하는 메시지.

- 프롬프트를 어떻게 구성하느냐에 따라 다른 결과가 출력될 수 있음.

- NL2SQL에서는 **사용자 질문**과 **데이터베이스 스키마**를 사용해서 프롬프트를 구성해야함.

NL2SQL Practice (NLQ)

- **자연어 질문 예시**

  1. Display all employee whose location is DALLAS?

  2. Display all the departments where department has 3 employees?

  3. Delete all employees those who are reporting to BLAKE?

  4. Display average salary for job SALESMAN

  5. Display all ename, empno, dname, loc from emp, dept table without joining two tables?

NL2SQL Practice (Schema Information)

- 다음과 같은 스키마가 존재한다고 가정할 때, 스키마 정보에 대한 프롬프팅 방법이 여러가지 있을 수 있음.

1.

```
CREATE TABLE DEPT (DEPTNO NUMBER(2) CONSTRAINT PK_DEPT PRIMARY KEY,
DNAME VARCHAR2(14) , LOC VARCHAR2(13) ) ;

CREATE TABLE EMP (EMPNO NUMBER(4) CONSTRAINT PK_EMP PRIMARY KEY, ENAME
VARCHAR2(10), JOB VARCHAR2(9), MGR NUMBER(4), HIREDATE DATE, SAL
NUMBER(7,2), COMM NUMBER(7,2), DEPTNO NUMBER(2) CONSTRAINT FK_DEPTNO
REFERENCES DEPT);
```

2.

```
DEPT (DEPTNO, DNAME, LOC)
EMP (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)
```

NL2SQL Practice (Final Prompt)

- **최종 프롬프트는 다양하게 생성 가능.**

1.

```
### Create the Postgres SQL Query using the following schema:
# CREATE TABLE DEPT (DEPTNO NUMBER(2) CONSTRAINT PK_DEPT PRIMARY KEY,
DNAME VARCHAR2(14) , LOC VARCHAR2(13) ) ;
# CREATE TABLE EMP (EMPNO NUMBER(4) CONSTRAINT PK_EMP PRIMARY KEY,
ENAME VARCHAR2(10), JOB VARCHAR2(9), MGR NUMBER(4), HIREDATE DATE, SAL
NUMBER(7,2), COMM NUMBER(7,2), DEPTNO NUMBER(2) CONSTRAINT FK_DEPTNO
REFERENCES DEPT);
### Display all employee whose location is DALLAS?
```

2.

```
### Create the Postgres SQL Query using the following schema:
### [Schema]
# DEPT (DEPTNO, DNAME, LOC)
# EMP (EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)
# EMP.DEPTNO references DEPT.DEPTNO
### [NLQ]
###Display all employee whose location is DALLAS?
```