

# 데이터사이언스를위한지식관리시스템

13주차: PostgreSQL

# Postgre? PostgreSQL?

- 발음: 포스트그레스큐엘
- 그러나 [포스트그레] 라고 칭하는 경우가 종종 있습니다
- 왜 그럴까요?
- Postgres + SQL = PostgresSQL => PostgreSQL => 접미사(?) 제거 => Postgre

# PostgreSQL이 역사

- <https://www.postgresql.org/docs/current/history.html> 를 읽어보겠습니다
- The Berkeley POSTGRES Project로 시작했습니다(1986)
- Postgres -> Postgres95 -> PostgreSQL(Postgre)
- MySQL/MariaDB보다 오히려 긴 역사를 가지고 있습니다

# Psql? Pgsql?

- Psql은 SQL 쿼리를 주고받는 프로그램 입니다
- 클라이언트인 것입니다
- Pgsql은 PosgreSQL의 비공식 줄임말중 하나 이기도 합니다
- 그런데 <https://ko.wikipedia.org/wiki/PL/pgSQL> 를 참고하면 또다른 의미를 갖기도 합니다
- PL/Python 이라는 것을 이용해 프로시저를 사용할 수 있습니다
- 프로시저: SQL 내부에서 사용되는 언어로 이해하시면 됩니다

# ANSI SQL 규격

- 표준 SQL 규격이라고 보시면 됩니다
- 이 표준 규격을 준수하면 다른 SQL간의 상호 연동 또는 이관이 유연하게 이루어질 수 있습니다
- 가장 최근의 규격은 <https://en.wikipedia.org/wiki/SQL:2023> 입니다
- PostgreSQL의 ANSI 2023 준수 관련한 문서는 <https://www.postgresql.org/docs/current/features.html> 를 참고하겠습니다
- MySQL의 ANSI규격 준수 관련한 문서는 <https://dev.mysql.com/doc/refman/9.4/en/compatibility.html> 를 참고하겠습니다
- MySQL의 이 문서에는 ANSI 2023에 관한 직접적인 언급은 없습니다
- 단, ANSI 모드라는 것이 존재합니다

# PostgreSQL을 Ubuntu에 설치하기

- 아래와 같이 Ubuntu에서는 패키지가 준비되어 있습니다

```
sudo apt update  
sudo apt install postgresql
```

- 2025년 11월 기준으로 위 명령으로 설치시에는 postgresql-16이 설치될 것입니다
- Ubuntu의 기본 저장소가 postgresql-16을 가지고 있는 것입니다

# PostgreSQL을 Ubuntu에 설치하기

- <https://www.postgresql.org/download/linux/ubuntu/> 를 좀더 따라가보겠습니다
- If the version included in your version of Ubuntu is not the one you want, you can use the PostgreSQL Apt Repository.
- 2025년 11월 기준으로 PostgreSQL의 최신 버전은 18버전입니다
- 이를 설치하기 위해서는 아래 내용을 따릅니다

```
sudo apt install -y postgresql-common  
sudo /usr/share/postgresql-common/pgdg/apt.postgresql.org.sh
```

- 이후에는 아래와 같이 버전을 선택하여 설치할 수 있습니다

```
sudo apt install postgresql-18
```

# PostgreSQL을 Ubuntu에 설치하기

- 설치가 완료되었다면 아래와 같이 TCP 5432 포트를 수신대기 하는지 확인합니다

```
$ sudo netstat -nap | grep 5432
tcp        0      0 127.0.0.1:5432      0.0.0.0:*           LISTEN      61677/postgres
```

- 아래 명령으로 접속되는지 확인합니다
- Psql 클라이언트를 이용해 postgresql 프롬프트에 들어온 것입니다

```
$ sudo -u postgres psql
psql (18.1 (Ubuntu 18.1-1.pgdg24.04+2))
Type "help" for help.

postgres=#
```



# Psql을 통해 postgresql 살펴보기

- psql 프롬프트에서 \du 라고 입력하면 사용자의 목록이 나옵니다
- mariadb와 달리 postgres라는 사용자가 보입니다

```
postgres-# \du
                                List of roles
Role name |                               Attributes
-----+-----
postgres | Superuser, Create role, Create DB, Replication, Bypass RLS

postgres-#
```

- \l 을 입력하면 DB의 목록이 보입니다
- <https://www.postgresql.org/docs/current/manage-ag-templatedbs.html> 를 참고합니다

```
postgres-# \l
                                List of databases
Name      | Owner   | Encoding | Locale Provider | Collate | Ctype  | Locale | ICU Rules | Access privileges
-----+-----+-----+-----+-----+-----+-----+-----+-----
postgres  | postgres | UTF8     | libc            | C.UTF-8 | C.UTF-8 |         |           |
template0 | postgres | UTF8     | libc            | C.UTF-8 | C.UTF-8 |         | =c/postgres  +
          |          |          |                 |         |         |         | postgres=CTc/postgres
template1 | postgres | UTF8     | libc            | C.UTF-8 | C.UTF-8 |         | =c/postgres  +
          |          |          |                 |         |         |         | postgres=CTc/postgres
(3 rows)

postgres-#
```

# Posgresql의 외부 접속 허용하기

- mariadb와 동일하게 최초에는 localhost접속만 허용됩니다
- WSL 기반의 ubuntu에서 port forwarding을 시도할 경우 0.0.0.0/0 허용이 필요합니다
- /etc/postgresql/18/main/postgresql.conf 의 아래 내용 수정이 필요합니다

```
# - Connection Settings -  
listen_addresses = '*'  
  
#listen_addresses = 'localhost'
```

- /etc/postgresql/18/main/pg\_hba.conf 의 아래 내용 수정이 필요합니다

```
# IPv4 local connections:  
#host      all             all             127.0.0.1/32     scram-sha-256  
host      all             all             0.0.0.0/0        md5
```

- 수정 후에는 postgresql 서버를 재시작합니다

```
$ sudo systemctl restart postgresql  
$ sudo netstat -nap | grep 5432  
tcp        0      0 0.0.0.0:5432          0.0.0.0:*        LISTEN     71203/postgres
```

# GUI 클라이언트 사용하기

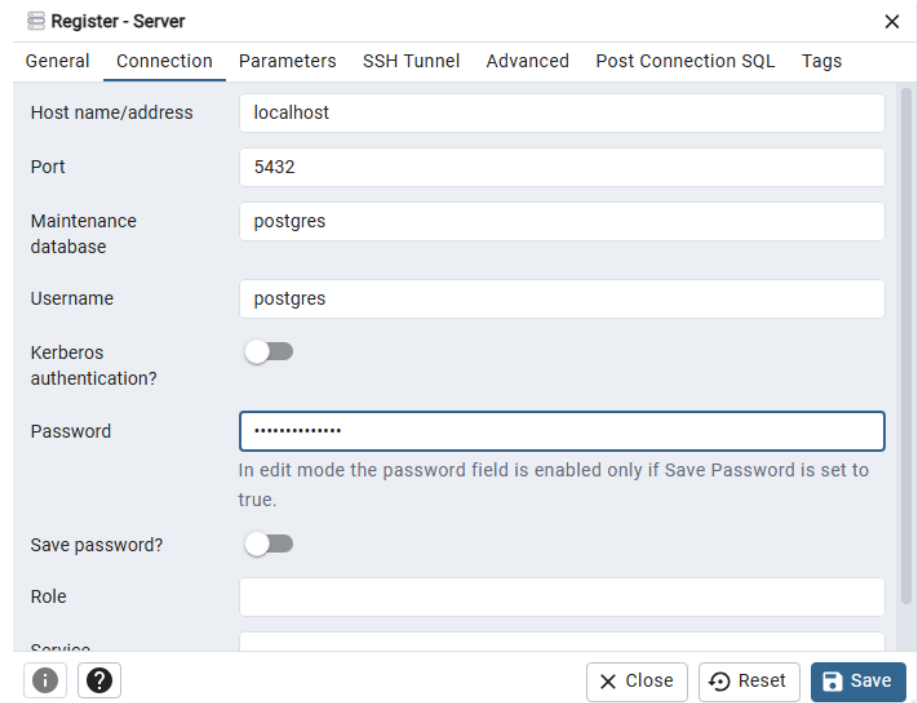
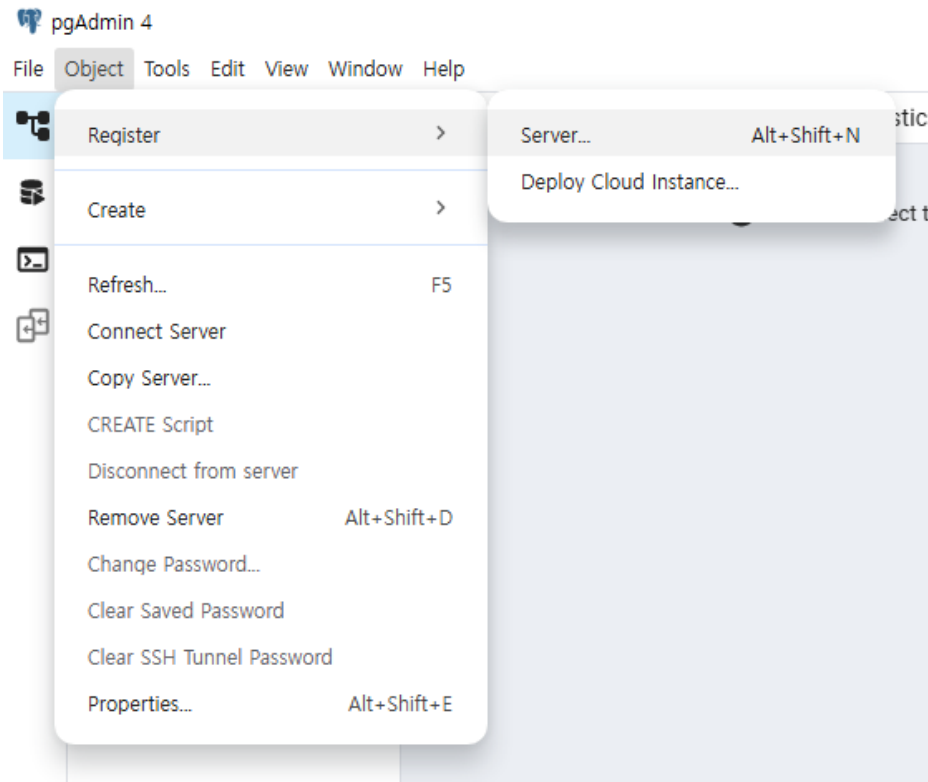
- PostgreSQL의 공식 GUI 클라이언트는 pgAdmin 입니다
- <https://www.pgadmin.org/download/> 에서 클라이언트 운영체제에 맞게 선택합니다
- 예를 들어 Windows 사용자이신 경우 <https://www.pgadmin.org/download/pgadmin-4-windows/> 에서 다운로드 받습니다
- 서버에서는 패스워드를 설정해줍니다
- mariadb에서도 진행했던 부분입니다

```
postgres=# ALTER USER postgres WITH PASSWORD 'your_password';  
ALTER ROLE  
postgres=# \q
```

```
postgres=# CREATE USER myuser WITH PASSWORD 'my_password';  
CREATE ROLE  
postgres=#
```

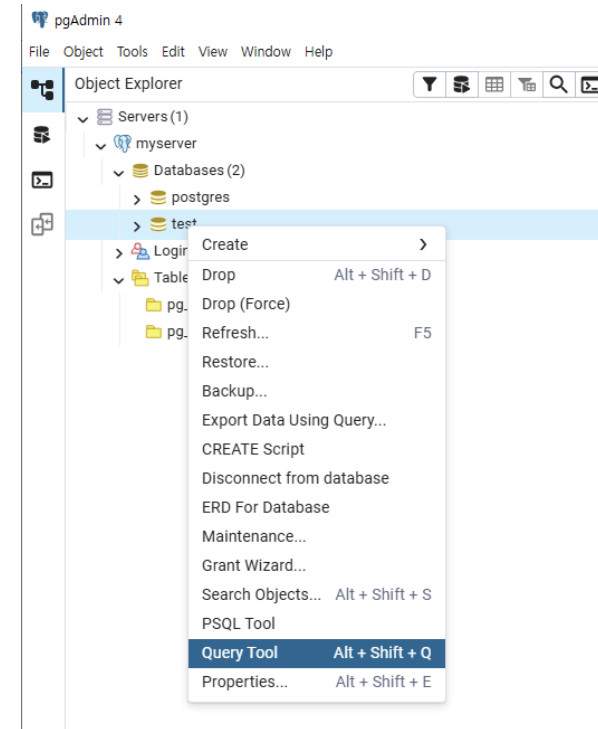
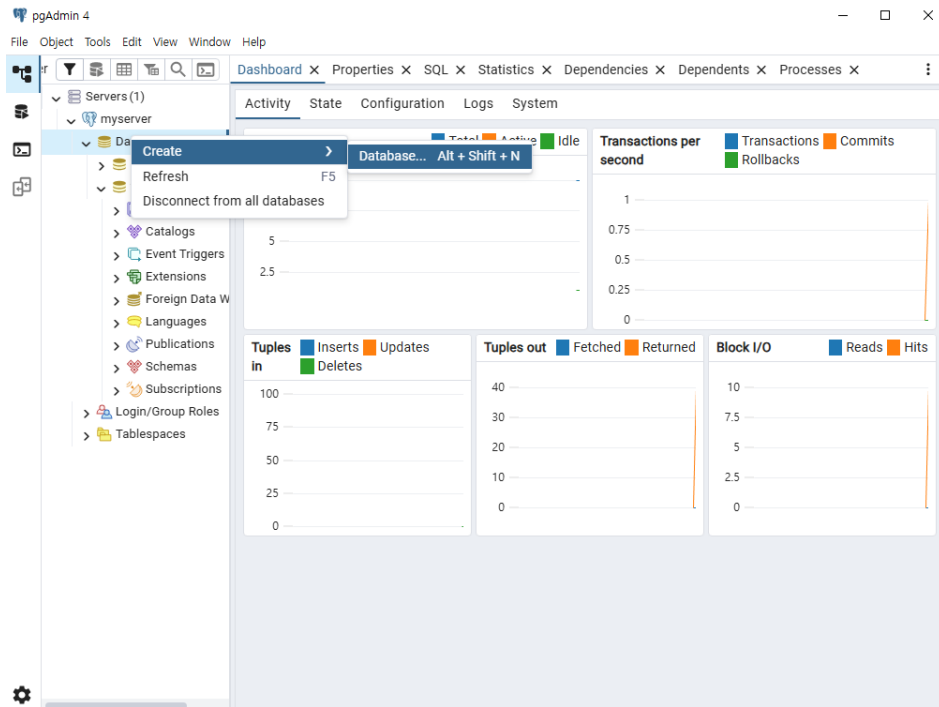
# GUI 클라이언트 사용하기

- 아래와 같이 서버 접속을 생성합니다
- Object > Register > Server > General > 이름 정하기 > Connection > Host name에 localhost > Password 지정 > Save



# GUI 클라이언트 사용하기

- 서버에 연결되면 Workspace가 열립니다
- 서버이름 > Databases > Create > Database > 이름 지정 > save
- 생성한 Database에서 Query Tool을 아래와 같이 실행합니다



# MariaDB Vs PostgreSQL(ANSI 준수율)

- PostgreSQL의 장점 중 하나는 ANSI SQL의 규격을 준수하는 ANSI 준수율이 높은 것입니다
- 온라인강의실의 실습 파일을 이용해 MariaDB에서는 지원하지 않지만 PostgreSQL에서는 가능한 ANSI SQL 쿼리를 확인하고 비교해봅니다

# MariaDB Vs PostgreSQL(단순트랜잭션)

- PostgreSQL의 다른 장점 중 하나는 대용량 트랜잭션에 대해 강한 것입니다
- 그러나 단순쿼리 기반의 트랜잭션은 MariaDB가 더 빠릅니다
- 온라인강의실의 실습 파일을 이용해 단순쿼리 기반의 트랜잭션을 두 데이터베이스로부터 비교해봅니다

# MariaDB Vs PostgreSQL

## (복잡한쿼리 기반의 트랜잭션)

- 복잡한 쿼리 기반의 트랜잭션은 PostgreSQL이 더 빠릅니다
- 온라인강의실의 실습 파일을 이용해 복잡한쿼리 기반의 트랜잭션을 두 데이터베이스로부터 비교해 봅니다



# PostGIS

- GIS에 특화된 함수를 제공하는 extension 입니다
- 아래 패키지의 설치가 필요합니다

```
sudo apt install postgis
```

- 온라인강의실의 실습 파일을 이용해 PostGIS에서 제공하는 함수를 활용해보입니다