

Getting Started with PgSQL

wisoft



차례

About PostgreSQL

Installation Guide

Deployment Types

Installing PostgreSQL on Windows

PostgreSQL Package Security

Configuration Guide

PostgreSQL Configuration File

PostgreSQL Start & Stop

User Management

User privileges

Creating / Removing Users

Granting Permissions

Changing Passwords

Using PostgreSQL

Using JDBC Connector



About PostgreSQL

Database

Anytime, Anywhere

- All Website, Market, Bank, Hospital, Gov. Agency, Theater, Online Game, and ETC.
- However, the database is not visible

PostgreSQL

Michael Stonebraker

(http://wikipedia.org/wiki/Michael_Stonebraker)

Ingres, Postgres, Illustra, Informix, Sybase SQL Server, MS SQL Server

Open Source S/W

PostgreSQL License (<https://www.postgresql.org/about/licence/>)

PostgreSQL 특징

Interactive Graphics Retrieval System(Ingres)에서 출발

개발 단계부터 완벽한 ACID와 MVCC를 지원하는 아키텍처로 설계












대용량의 복잡한 트랜잭션 처리를 위한 RDBMS

RDBMS의 화이트페이퍼를 기반으로 Oracle, DB2 그리고 PostgreSQL이 구현

MySQL이나 MariaDB와 같은 오픈소스 제품과 달리 PostgreSQL의 경우 상용 RDBMS 급의 기능을 제공

Database Engine Ranking

2018.03.

Rank			DBMS	Database Model	Score		
Mar 2019	Feb 2019	Mar 2018			Mar 2019	Feb 2019	Mar 2018
1.	1.	1.	Oracle +	Relational, Multi-model 	1279.14	+15.12	-10.47
2.	2.	2.	MySQL +	Relational, Multi-model 	1198.25	+30.96	-30.62
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model 	1047.85	+7.79	-56.94
4.	4.	4.	PostgreSQL +	Relational, Multi-model 	469.81	-3.75	+70.46
5.	5.	5.	MongoDB +	Document	401.34	+6.24	+60.82
6.	6.	6.	IBM Db2 +	Relational, Multi-model 	177.20	-2.23	-9.47
7.	 9.	7.	Microsoft Access	Relational	146.20	+2.18	+14.26
8.	 7.	8.	Redis +	Key-value, Multi-model 	146.12	-3.32	+14.90
9.	 8.	9.	Elasticsearch +	Search engine, Multi-model 	142.79	-2.46	+14.25
10.	10.	 11.	SQLite +	Relational	124.87	-1.29	+10.06

ref. <https://db-engines.com/en/ranking>



Installation Guide

Overview

- Supported Multi Platform
 - Installer, Binary File, Source Code
 - Windows, OS X, Linux
- <https://www.postgresql.org/download/>
- BigSQL
 - <https://www.openscg.com/bigsql/postgresql/installers.jsp/>

Deployment Types

- <https://www.postgresql.org/support/versioning/>
- Stable
 - 일반적인 사용자가 사용하는 버전
- Development
 - 새로운 기능과 특징을 추가
 - Alpha, Beta, Release Candidate(RC)로 구분
- Maintenance
 - 정식 배포된 후, 5년 동안 버그 및 보안 관련 수정과 유지보수 지원

Installing PostgreSQL

MacOS

Step 01

- `brew install postgresql`
- Configuration
 - Service
`brew services start postgresql`
 - Temporal
`pg_ctl -D /usr/local/var/postgres start`

Step 02

- 기본 사용자 추가
 - `/usr/local/Cellar/postgresql/11.2/bin/createuser -s postgres`
- 환경 설정
 - `/usr/local/var/postgres/postgresql.conf`
`listen_addresses = '*'`
 - `/usr/local/var/postgres/pg_hba.conf`
`host all all 0.0.0.0/0 md5 # ipv4 area`

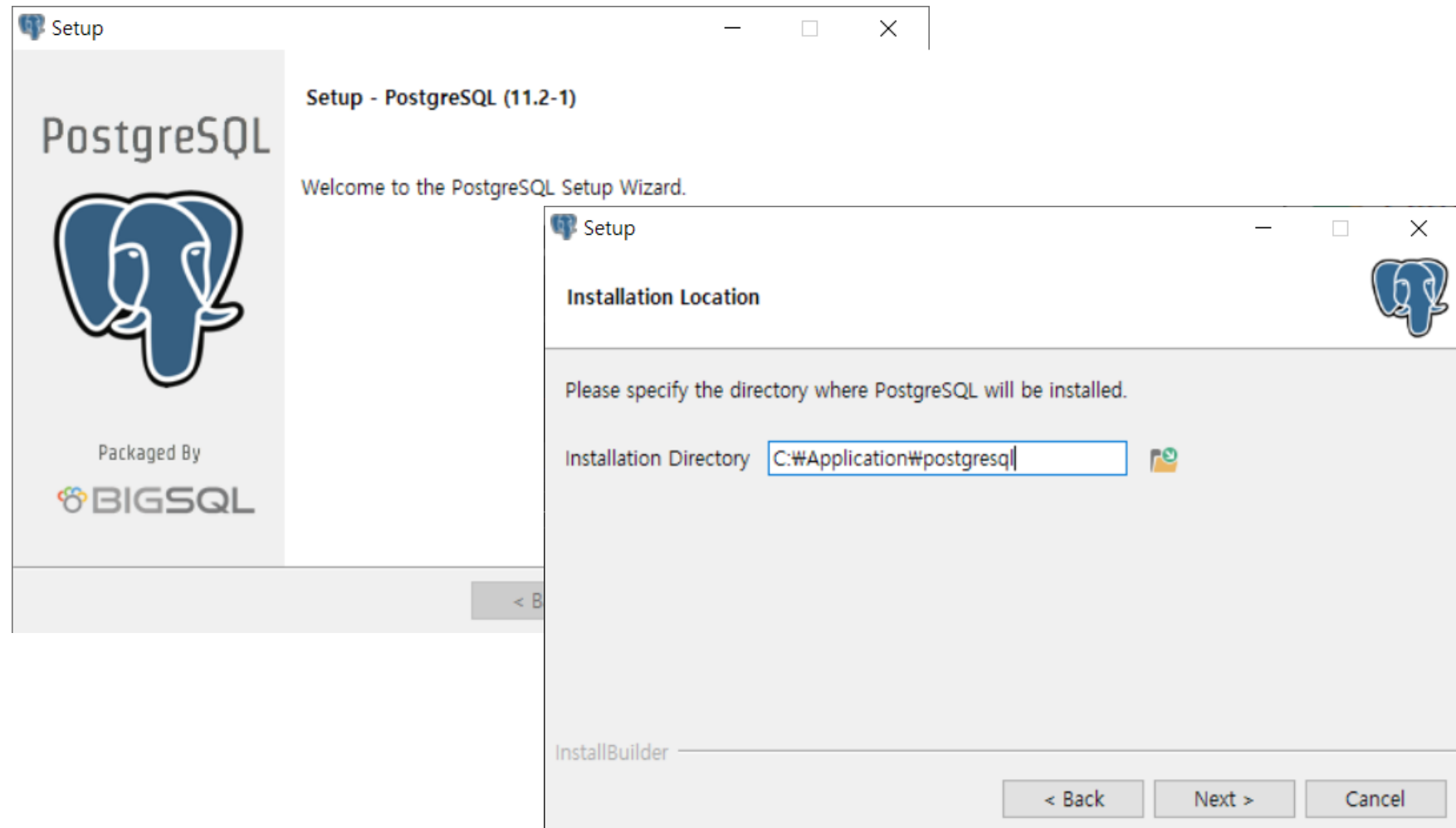
Microsoft Windows

Step 01

- Download PostgreSQL with BigSQL
 - <http://bit.ly/2ToMWYG> 또는 WiSoft NAS Software

Step 02

- Run Installer and Set Installation Directory



Step 03

- Select Component and Set Password

The image shows two overlapping windows from the PostgreSQL Setup application. The background window, titled 'Setup', is on the 'Select Components' screen. It instructs the user to 'Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.' It features a list box with two items: 'PostgreSQL Database Server' (checked) and 'pgAdmin3 LTS' (unchecked). The foreground window, also titled 'Setup', is on the 'Set Password' screen. It asks the user to 'Please provide a password for the superuser (postgres) database user'. It contains two password input fields: 'PostgreSQL Password' and 'Retype password', both filled with dots. Below these is a section for 'Advanced PostgreSQL Configuration Options' which is currently unchecked. This section includes a text field for 'PostgreSQL data directory' containing 'C:\Application\postgresql\data\pg11' and a text field for 'PostgreSQL Port' containing '5432'. Both windows have a blue elephant icon in the top right corner and 'InstallBuilder' branding at the bottom. The foreground window has '< Back', 'Next >', and 'Cancel' buttons at the bottom right.

Setup

Select Components

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

☒ PostgreSQL Database Server
☐ pgAdmin3 LTS

Click on description

InstallBuilder

Setup

Please provide a password for the superuser (postgres) database user

PostgreSQL Password:

Retype password:

☐ Advanced PostgreSQL Configuration Options

PostgreSQL data directory: C:\Application\postgresql\data\pg11

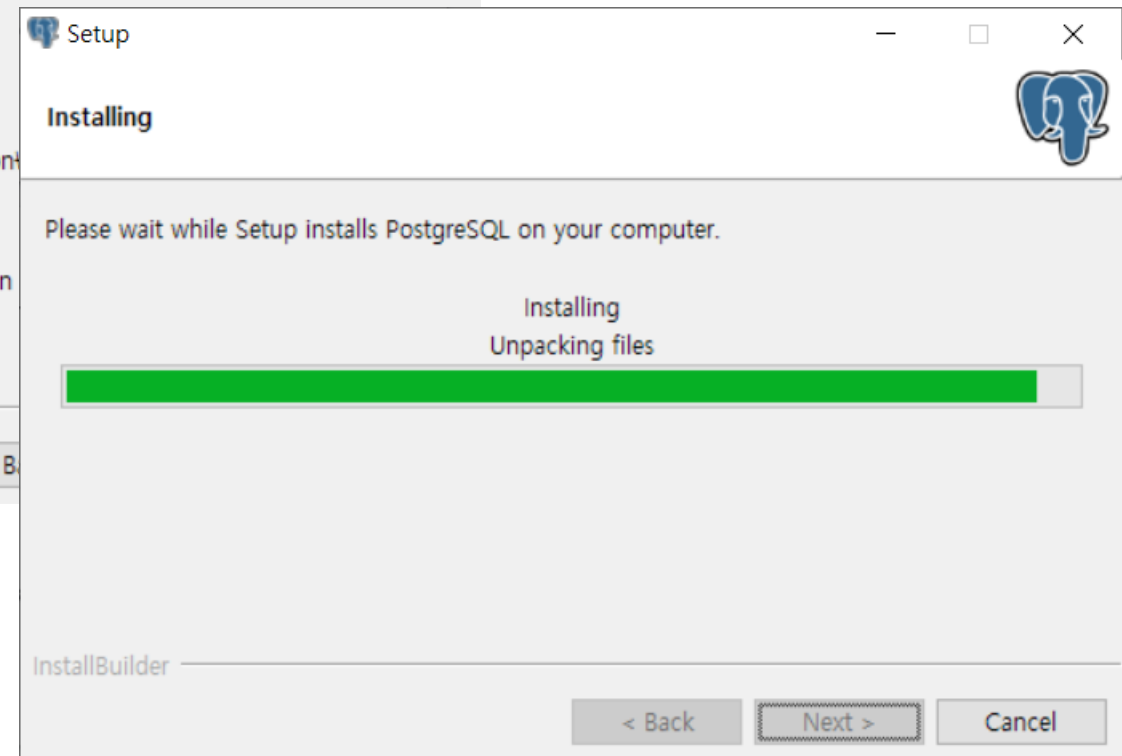
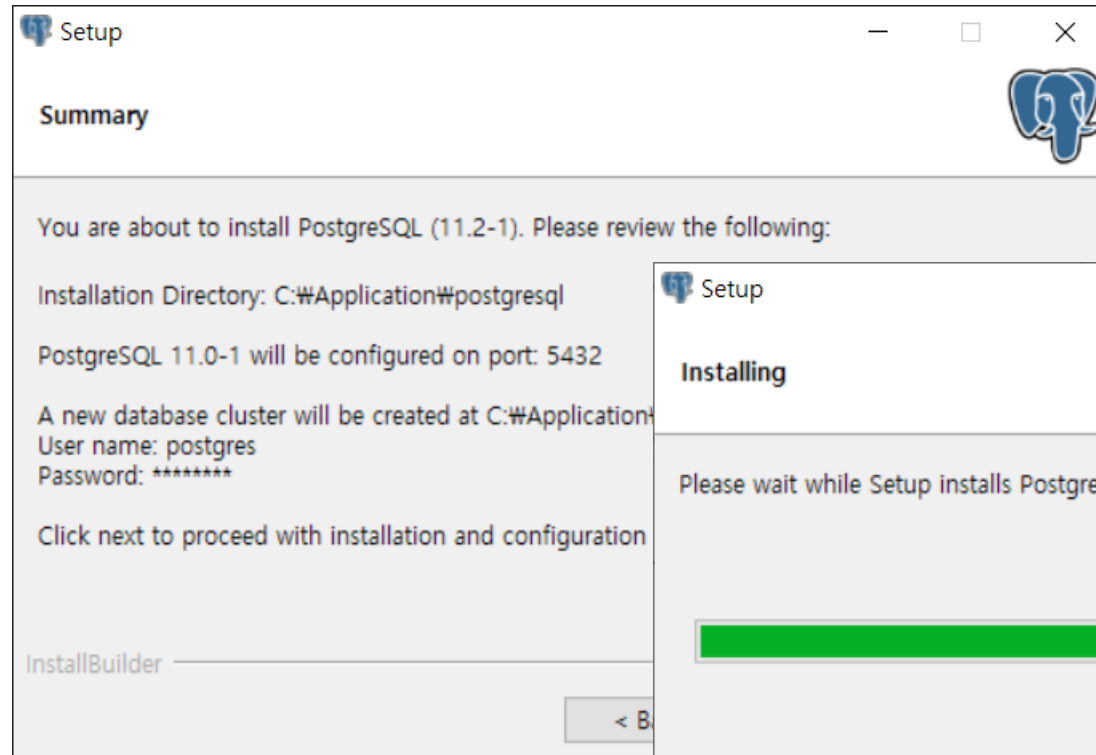
PostgreSQL Port: 5432

InstallBuilder

< Back Next > Cancel

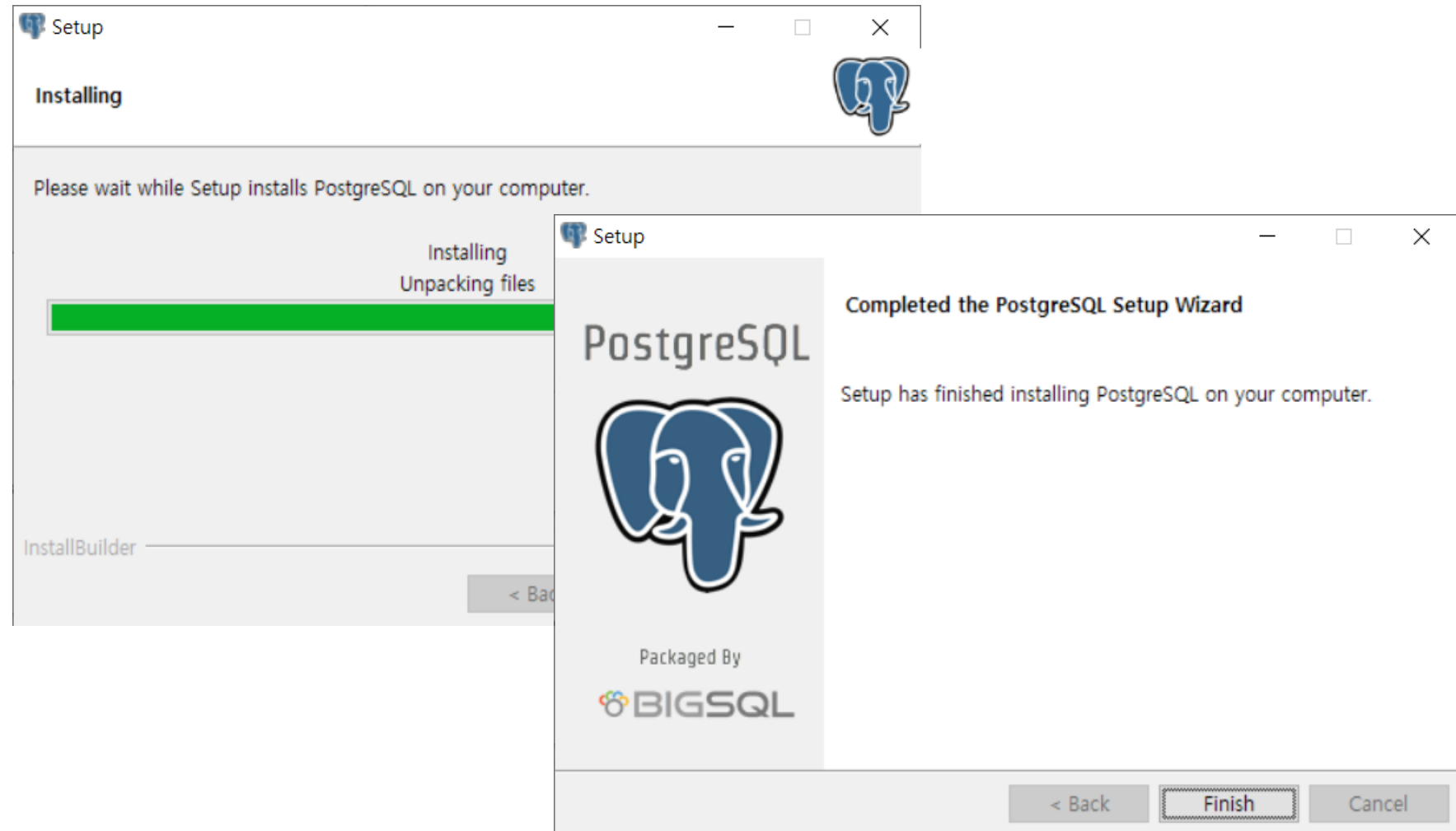
Step 04

- Summary and Installing



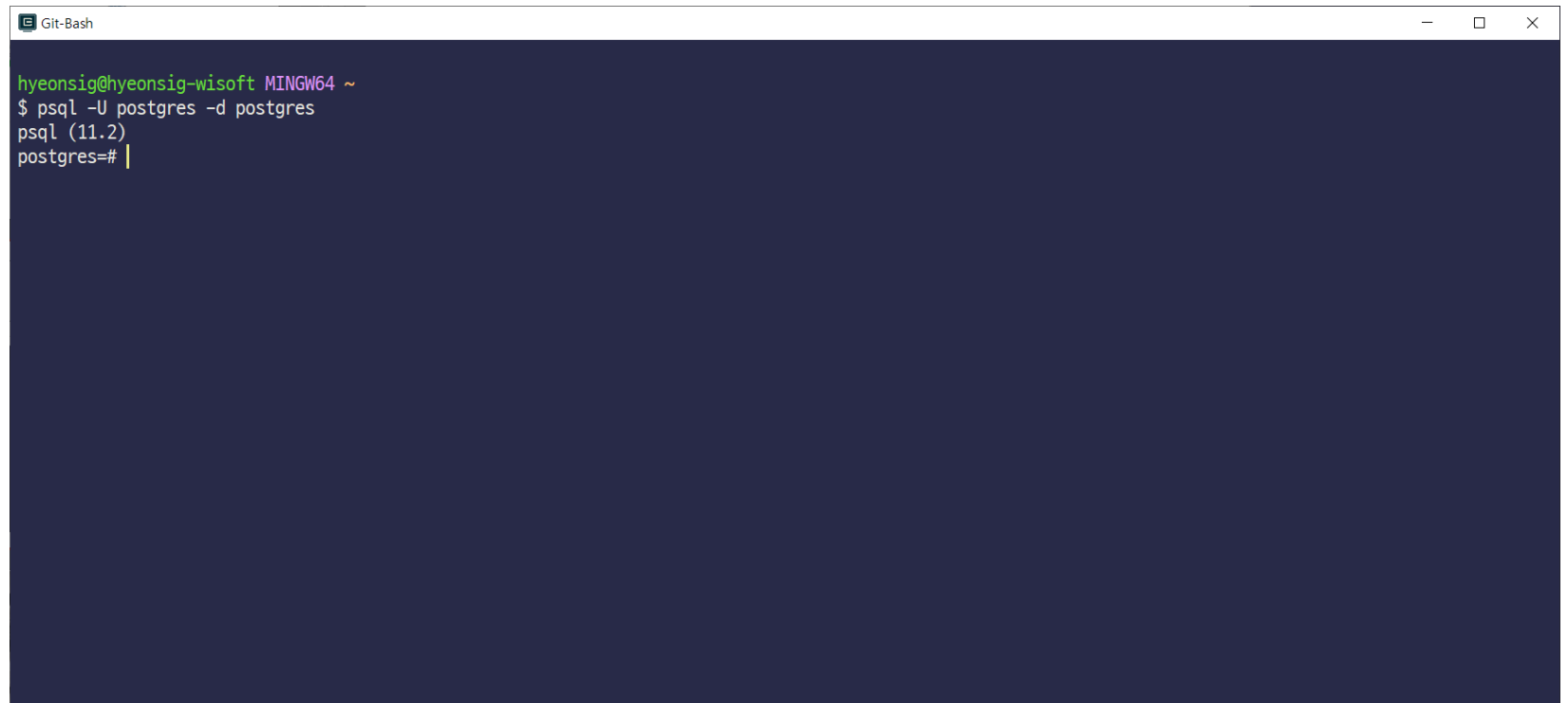
Step 05

- Ready to Install and Complete



접속 테스트

- Connection Test
 - `psql -U postgres -d postgres`



```
Git-Bash
hyeonsig@hyeonsig-wisoft MINGW64 ~
$ psql -U postgres -d postgres
psql (11.2)
postgres=#
```



User Management

User Privileges

- Classification of user privileges
 - Entire system
Administrator
CREATE USER, FILE, PROCESS, SHUTDOWN, AND ETC.
 - Database, Table, Column, and ETC.
CREATE, ALTER, DROP
SELECT, INSERT, UPDATE, DELETE
VIEW, TRIGGER, EXECUTE, AND ETC.
 - Additional privileges
- Whitepaper
 - <https://www.postgresql.org/docs/current/database-roles.html>

데이터베이스 생성과 삭제

데이터베이스 생성

- Syntax

```
CREATE DATABASE database_name ENCODING 'encoding_type';
```

- database_name

데이터베이스 식별자

- encoding_type

데이터베이스 인코딩 타입으로 'UTF-8'로 설정

- Whitepaper

<https://www.postgresql.org/docs/current/static/sql-createdatabase.html>

연습문제

- 'exercise'와 'test' 데이터베이스를 생성하시오.



데이터베이스 삭제

- Syntax

```
DROP DATABASE database_name;
```

- database_name

데이터베이스 식별자

- Whitepaper

<https://www.postgresql.org/docs/current/static/sql-dropdatabase.html>

연습문제

- 'test' 데이터베이스를 제거하시오.



사용자 생성과 삭제

Basic Step - Creating User

1st step

CREATE USER 명령을 사용하여 사용자를 생성

2nd step

GRANT 명령을 사용하여 사용자에게 권한 부여

사용자 계정 생성

- Syntax

```
CREATE USER username WITH PASSWORD 'password';
```

- username

사용자 계정(아이디)

- password

반드시 어려운 비밀번호로 입력하며, 분실하지 않도록 유의

- Whitepaper

<https://www.postgresql.org/docs/current/static/sql-createuser.html>

연습문제

- 다음 요구사항을 만족하도록 사용자를 등록하시오.
 - 사용자 아이디는 'scott'이며, 이 사용자의 비밀번호는 'tiger'이다.
 - 다른 사용자 아이디는 'test'이며, 이 사용자의 비밀번호는 'test1004'이다.



사용자 계정 삭제

- Syntax

```
DROP USER username;
```

- username

사용자 계정(아이디)

- Whitepaper

<https://www.postgresql.org/docs/current/static/sql-dropuser.html>

연습문제

- 사용자 아이디 'test'를 제거하시오.

사용자 계정 권한 부여

- Syntax

```
GRANT <permission> ON DATABASE <database_name> TO <username>  
WITH GRANT OPTION;
```

- permission

사용자에게 부여할 권한 지정

- database_name

데이터베이스 식별자

- username

사용자 계정

- Whitepaper

<https://www.postgresql.org/docs/current/static/sql-grant.html>

연습문제

- 'scott' 계정에 'exercise' 데이터베이스의 모든 권한을 부여하시오.

데이터베이스 접속

- Syntax

```
$ psql -U username -d database_name
```

- username

사용자 계정(아이디)

- database_name

데이터베이스 식별자

- Whitepaper

<https://www.postgresql.org/docs/current/static/app-psql.html>

연습문제

- 'scott' 계정으로 'exercise' 데이터베이스에 접속하시오.

사용자 계정 권한 회수

- Syntax

```
REVOKE <permission> ON DATABASE <database_name> FROM <username>;
```

- permission

사용자에게 부여할 권한 지정

- database_name

데이터베이스 식별자

- username

사용자 계정

- Whitepaper

<https://www.postgresql.org/docs/current/static/sql-revoke.html>

연습문제

- 'scott' 계정에 'exercise' 데이터베이스의 모든 권한을 회수하라.



Break Time





https://pbs.twimg.com/profile_background_images/492727362123886592/eLwJ4qKy.jpeg

Command Line Interface

데이터베이스 보기

- 시스템에 등록된 데이터베이스 보기
 - PSQL Console

```
SQL> \list
```

- Others

```
SELECT pg_database.datname, pg_user.username  
FROM pg_database, pg_user  
WHERE pg_database.datdba = pg_user.usesysid;
```

- pg_database, pg_user

PgSQL의 시스템 카탈로그

데이터베이스 접속

- psql 모드에서 데이터베이스에 접속하기

```
SQL> \c database_name
```

- database_name

데이터베이스 식별자

테이블 (생성, 수정, 삭제)

테이블이란?

- About Table?
 - 테이블은 관계형 데이터베이스의 기본적인 데이터 저장 단위
 - 사용자가 접근 가능한 모든 데이터를 보유하며, 행(Row)과 열(Column)로 구성

선수번호	성명	생년월일	등번호	포지션
1	하주석	1994/02/25	10	내야수
2	김민우	1992/05/15	15	투수
3	김태균	1982/05/29	52	내야수
4	정근우	1982/10/02	8	내야수
5열(Column)	이용규	1985/08/26	1	외야수

행(ROW)

테이블(TABLE)

테이블 생성하기 (1/2)

- Syntax

```
CREATE TABLE [IF NOT EXISTS] table_name (<column-definition>);
```

- table_name

테이블 식별자

- IF NOT EXISTS

기존에 동일한 이름의 테이블 이름이 있으면 생성하지 않음

- column-definition

테이블을 구성하는 컬럼 이름과 컬럼 도메인 정보 등을 지정

- Whitepaper

<https://www.postgresql.org/docs/current/static/sql-createtable.html>

테이블 생성하기 (2/2)

- Syntax <column-definition>

```
<column_name> <datatype>  
    [NOT NULL | NULL]  
    [CHECK <expression>]  
    [DEFAULT <default_value>]  
    [UNIQUE [KEY] | [PRIMARY] KEY]  
...
```

- NOT NULL | NULL - NULL 값 허용 지정
- CHECK - 컬럼의 제약조건 지정
- DEFAULT - 컬럼의 기본값 지정
- UNIQUE KEY | PRIMARY KEY - 유일 값, 기본 키 지정

PostgreSQL

데이터 타입

- 숫자형
 - INT(Small, Big), REAL, Double Precision, Numeric, Decimal, BIT, AND ETC
- 문자형
 - CHAR, VARCHAR, TEXT, AND ETC
- 날짜형
 - DATE, TIME, TIMESTAMP, AND ETC.
- 기타형
 - BOOLEAN, SERIAL(Small, Big), JSON, JSONB, AND ETC.
- Whitepaper - <https://www.postgresql.org/docs/current/static/datatype.html>

연습문제

- 'exercise' 데이터베이스에 'student' 테이블을 생성하라.

우리 조직의 학생은 학번(no), 성명(kname, 문자 4자리), 생년월일(birthday, 날짜)를 입력하고, 기본 키는 학번으로 지정한다. 단, 학번과 성명은 반드시 입력해야 한다.

테이블 목록 보기

- 시스템에 등록된 테이블의 목록 보기

- PSQL Console

```
SQL> \dt
```

- Others

```
SELECT *  
  FROM pg_tables  
 WHERE tableowner = 'scott';
```

- username

사용자 계정 식별자

- pg_tables

데이터베이스의 시스템 카탈로그

테이블 구조 보기

- 시스템에 등록된 테이블의 구조 보기

- PSQL Console

```
SQL> \d table_name
```

- Others

```
SELECT column_name, data_type, is_nullable, column_default  
FROM information_schema.columns  
WHERE table_name = 'table_name';
```

- column_name, table_name

컬럼과 테이블 식별자

- information_schema

데이터베이스의 시스템 카탈로그

테이블 수정하기 (1/2)

- Syntax

```
ALTER TABLE [IF EXISTS] table_name (<column-definition>);
```

- table_name

변경할 테이블 이름을 입력

- IF EXISTS

기존에 동일한 이름의 테이블 이름이 없으면 변경하지 않음

- column-definition

테이블을 구성하는 컬럼 이름과 컬럼 도메인 정보 등을 지정

- Whitepaper

<https://www.postgresql.org/docs/current/static/sql-altertable.html>

테이블 컬럼 추가하기

- Syntax

```
ADD COLUMN <column_name> <column_definition> [column_constraint]
```

- column_name
추가할 컬럼 이름 지정
- column_definition
컬럼의 도메인 등을 지정
- column_constraint
컬럼의 제약조건 등을 지정

테이블 컬럼 수정하기

(1/2)

- Syntax

```
RENAME COLUMN <old_name> TO <new_name>;
```

- old_name

변경해야 할 컬럼 이름 지정

- new_name

새롭게 변경하는 컬럼 이름 지정

테이블 컬럼 수정하기

(2/2)

- Syntax

```
ALTER COLUMN <column_name>  
    PGSQL_COMMAND {<column_definition>, [column_constraint]}
```

- column_name
수정할 컬럼 이름 지정
- column_definition
컬럼의 도메인 등을 지정
- column_constraint
컬럼의 제약조건 등을 지정

테이블 컬럼 삭제하기

- Syntax

```
DROP COLUMN <column_name>
```

- column_name

삭제할 컬럼 이름 지정

연습문제

- 'student' 테이블에 성별(gender, varchar(1))을 추가하라.

- 'student' 테이블의 'kname'을 'name'으로 변경하고,
성별 속성을 varchar(2)로 변경하라.

- 'student' 테이블에서 성별을 제거하라.

데이터 (등록, 조회, 수정, 삭제)

데이터 삽입

- Syntax

```
INSERT INTO <table_name> [(<column_name>[, column_name, ...])]  
VALUES ({expression | DEFAULT}, ...) [, (...), ...];
```

- table_name

데이터를 삽입할 테이블을 지정

- VALUES

삽입할 데이터 값

- expression

컬럼에 입력할 값을 나열하며, 기본 값을 사용하고 싶을 때에는 DEFAULT 키워드 사용

- Whitepaper

<https://www.postgresql.org/docs/9.6/static/sql-insert.html>

연습문제

- ‘exercise’ 데이터베이스의 ‘student’ 테이블에 다음 정보를 입력하시오.
 - 20110101, 홍길동, 1990-03-01
 - 20110201, 이지매, 1991-02-28
 - 20120301, 황진이, 1991-02-28



데이터 수정

- Syntax

```
UPDATE <table_name> SET column_name={expression | DEFAULT}  
                        [, columnmn_name={expression | DEFAULT }] ...  
[WHERE <query_condition>]
```

- table_name

갱신을 수행할 대상 테이블을 지정

- column_name={expression | DEFAULT}

해당 컬럼의 값을 표현식으로 변경하거나 기본 값으로 변경

- query_condition

변경하고자 하는 테이블 내의 대상을 지정하기 위해 사용하므로 매우 중요

- Whitepaper

<https://www.postgresql.org/docs/9.6/static/sql-update.html>

연습문제

- 황진이의 학번을 '20110301'로 변경하시오.



- 일지매의 생일을 '1991-02-28'에서 '1991-01-31'로 변경하시오.



데이터 삭제

- Syntax

```
DELETE FROM <table_name> [WHERE <query_condition>];
```

- table_name

테이블 내용을 삭제할 대상 테이블 이름을 입력

- query_condition

이 조건을 생략하는 테이블의 전체 내용을 제거

삭제하고자 하는 테이블 내의 대상을 지정하기 위해 사용하므로 매우 중요

- Whitepaper

<https://www.postgresql.org/docs/9.6/static/sql-delete.html>

데이터 조회

- Syntax

```
SELECT <target_column> FROM <table_name>  
[WHERE <query_condition>]  
[ORDER BY <column_name>];
```

- target_column

조회할 컬럼 리스트

- table_name

조회할 대상 테이블

- query_condition

조회하고자 하는 테이블 내의 대상을 지정

- Whitepaper

<https://www.postgresql.org/docs/9.6/static/sql-select.html>

연습문제

- 'student' 테이블에 등록된 모든 학생을 조회하라.

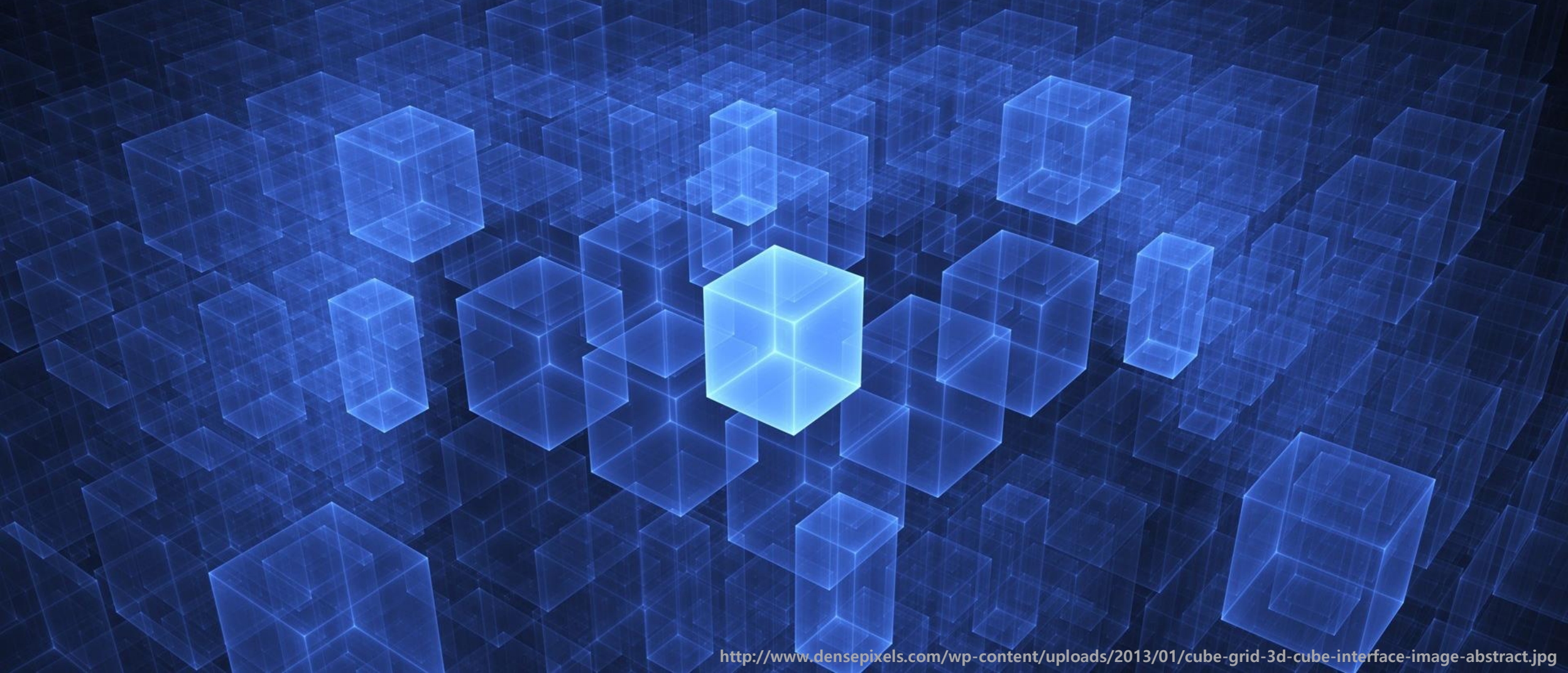
A dark gray rectangular box intended for writing the SQL query to retrieve all students from the 'student' table.

- 생일이 1990년 04월 이후에 출생한 학생을 조회하라.

A dark gray rectangular box intended for writing the SQL query to retrieve students born after April 1990.

- 'student' 테이블에 등록된 모든 학생을 조회하라.
단, 이름순으로 정렬하여 조회하라.

A dark gray rectangular box intended for writing the SQL query to retrieve all students sorted by name.



<http://www.densepixels.com/wp-content/uploads/2013/01/cube-grid-3d-cube-interface-image-abstract.jpg>

Using JDBC Connector

Download

JDBC Connector

- PostgreSQL JDBC Driver
 - <https://jdbc.postgresql.org/download.html>

- This is the current version of the driver.

using JDK 1.8 then you should use the JDBC42 version

(<https://jdbc.postgresql.org/download/postgresql-42.2.5.jar>)

using JDK 1.7 then you should use the JDBC41 version

(<https://jdbc.postgresql.org/download/postgresql-42.2.5.jre7.jar>)

using JDK 1.6 then you should use the JDBC4 version

(<https://jdbc.postgresql.org/download/postgresql-42.2.5.jre6.jar>)

Simple Client Program with Java

Sample Code

Attention!!

이 코드는 기억에서 잊으세요.

인터넷에서 검색하면, 가장 많이 노출되는
코드 패턴이지만 문제(?)가 많은 코드입니다.

```
package io.wisoft.seminar;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class BadJdbcExample {
    public static void main(final String... args) {
        try {
            Connection conn = DriverManager.getConnection
                ("jdbc:postgresql://localhost:5432/exercise", "scott", "tiger");

            Statement stmt;
            stmt = conn.createStatement();

            ResultSet rs;
            rs = stmt.executeQuery("SELECT * FROM STUDENT");

            if (stmt.execute("SELECT * FROM STUDENT")) {
                rs = stmt.getResultSet();
            }

            while (rs.next()) {
                System.out.print("[학번] " + rs.getString(1) + " || ");
                System.out.print("[이름] " + rs.getString(2) + " || ");
                System.out.println("[생일] " + rs.getString(3));
            }

            rs.close();
            stmt.close();
            conn.close();
        } catch (SQLException sqex) {
            System.out.println("SQLException: " + sqex.getMessage());
            System.out.println("SQLState: " + sqex.getSQLState());
        }
    }
}
```

QnA

hyeonsig@hanbat.ac.kr

<https://blog.ngelmaum.org>

Thank You