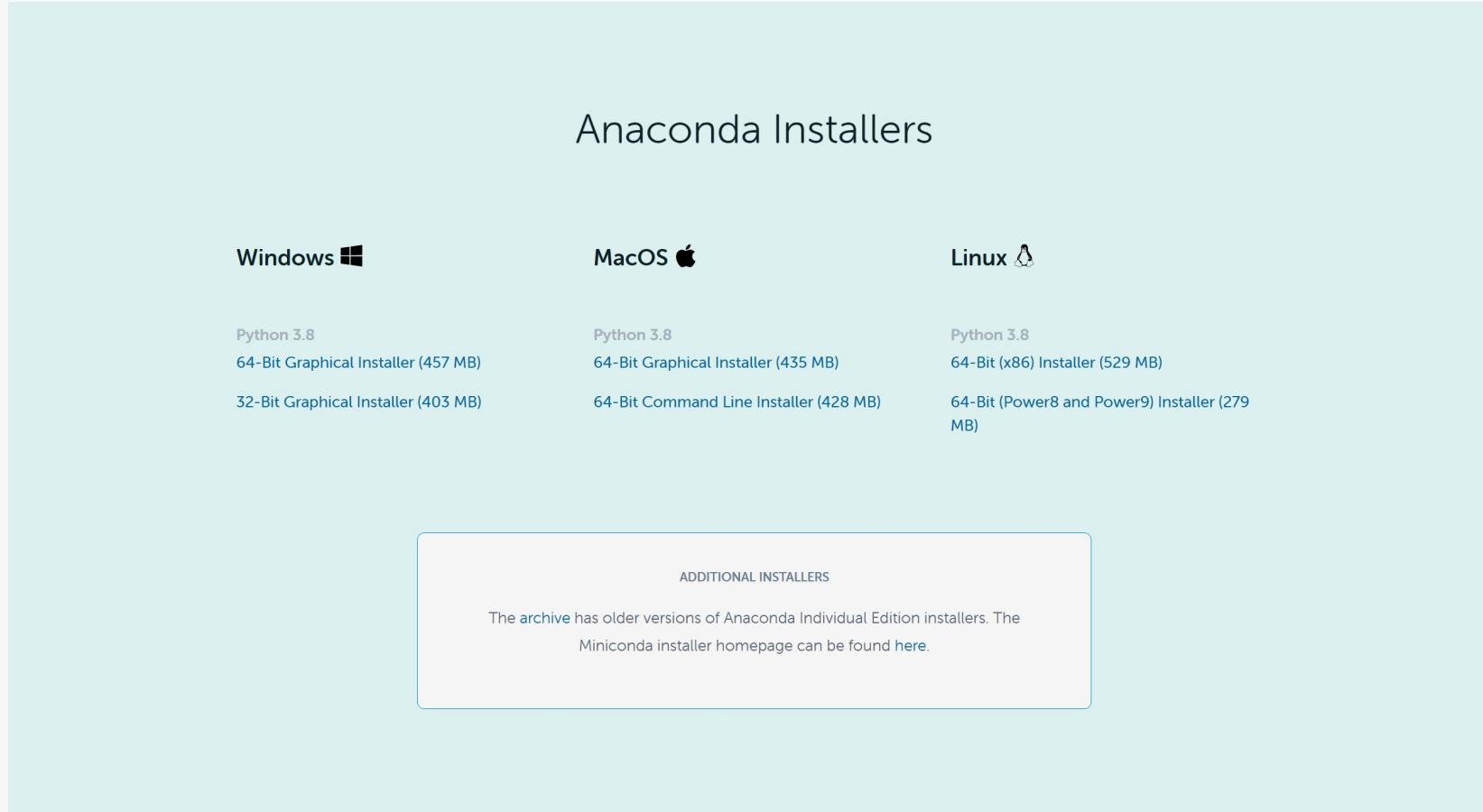


개발환경설정

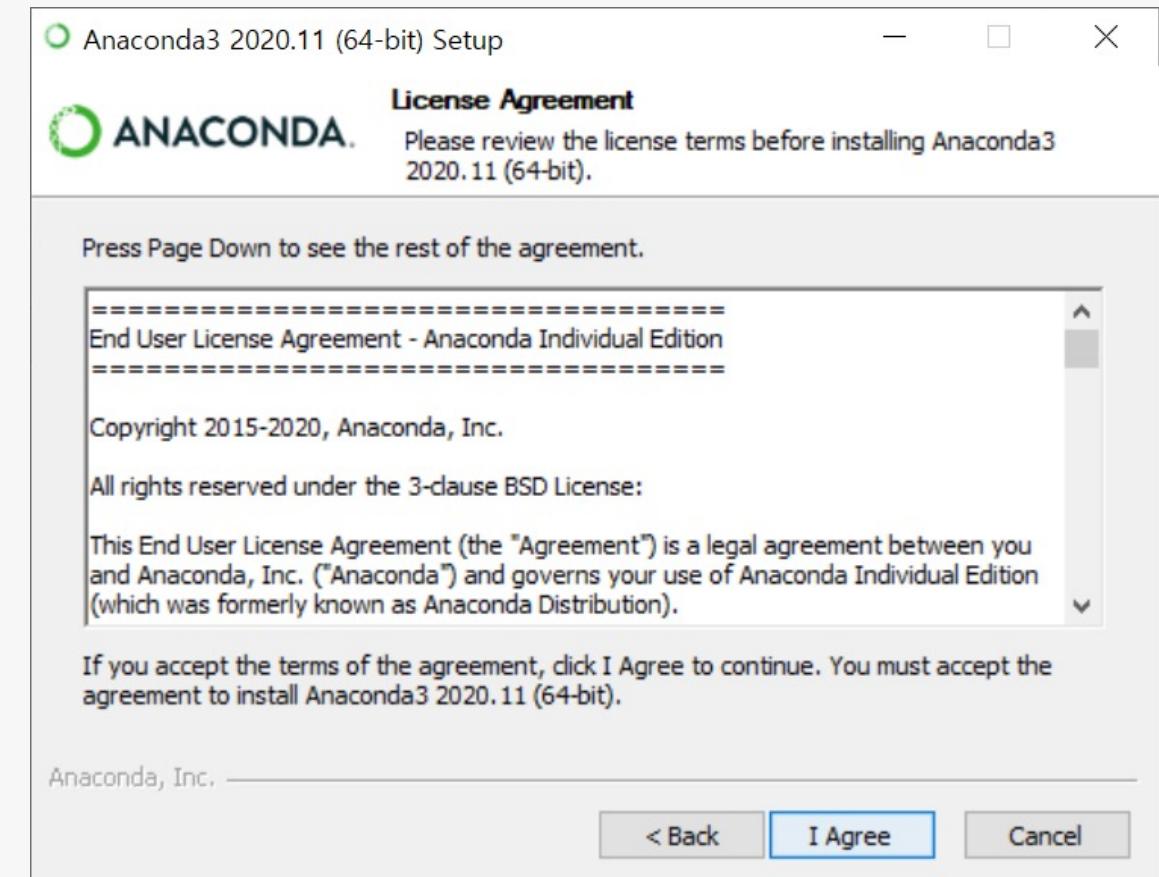
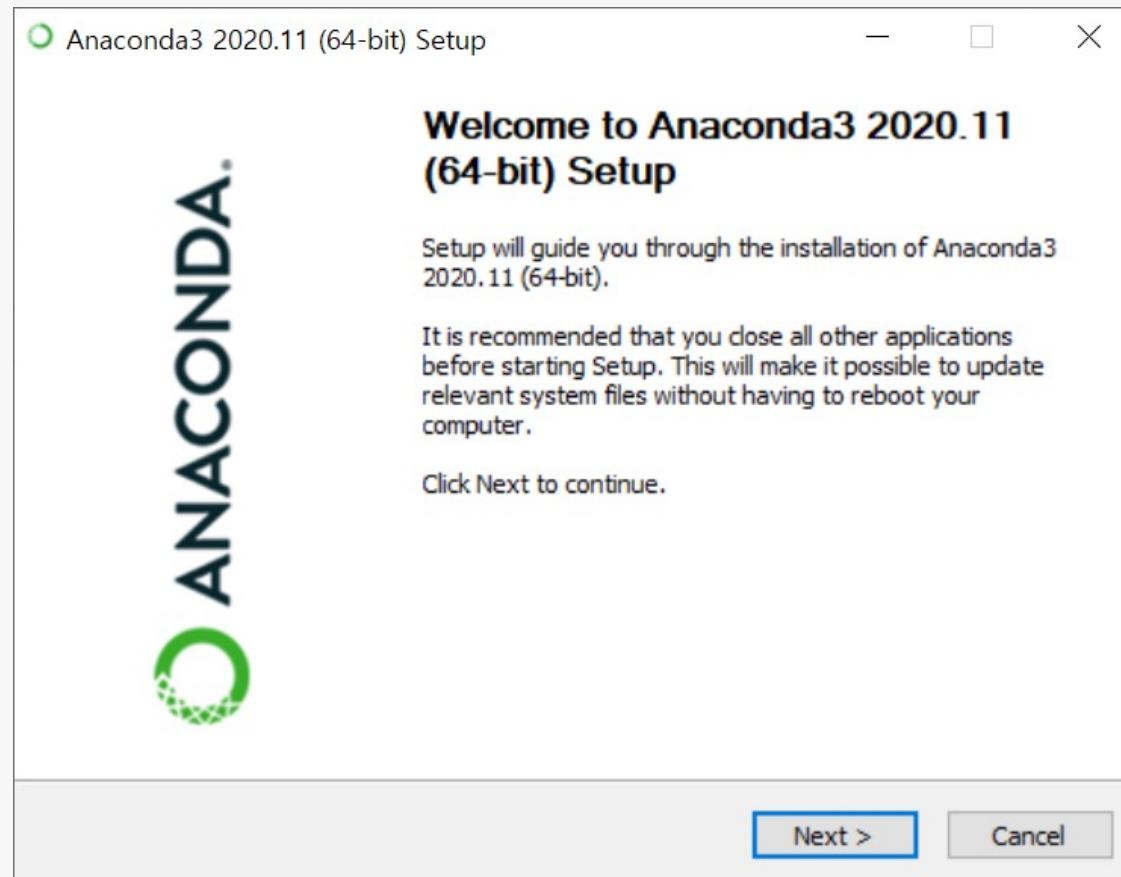
with  python™

OS별 머신러닝 & 딥러닝을 구현할 수 있도록 도와주는 패키지
(웹사이트 주소: <https://www.anaconda.com/products/individual>)



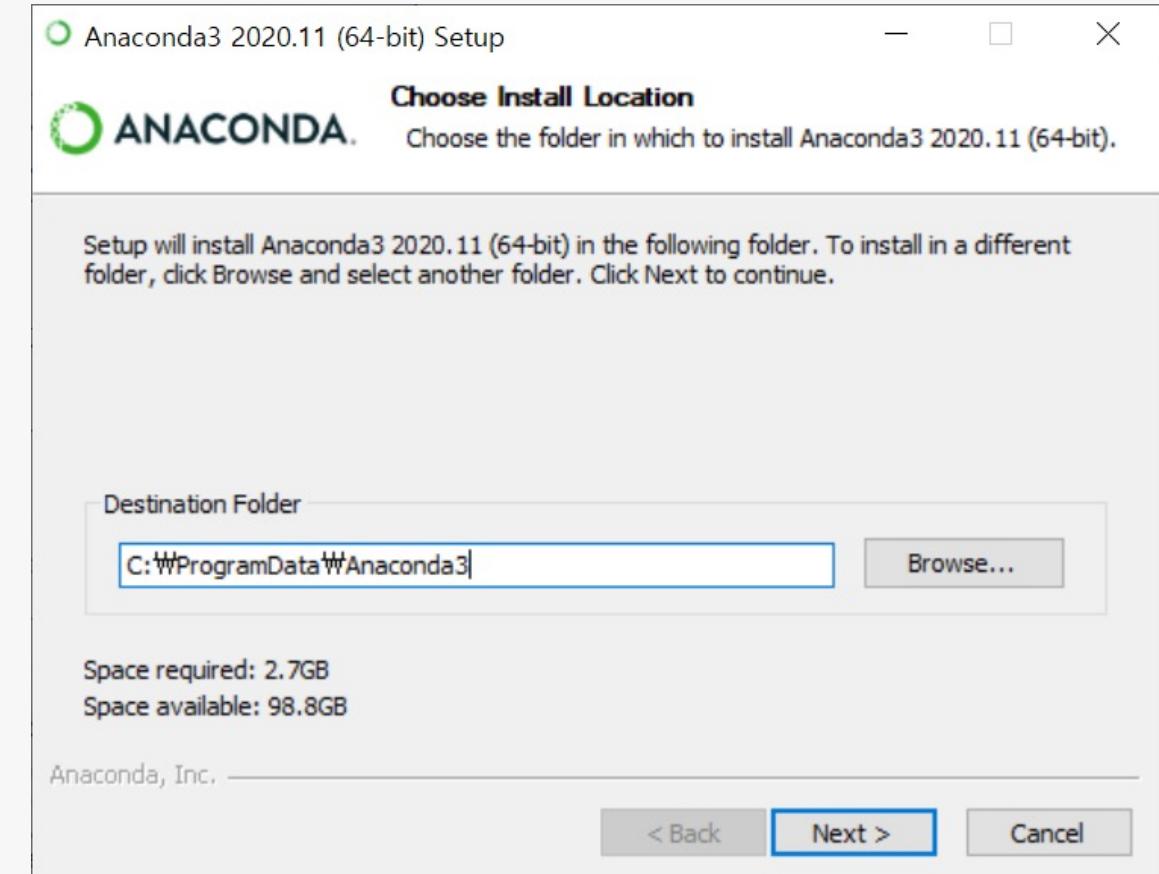
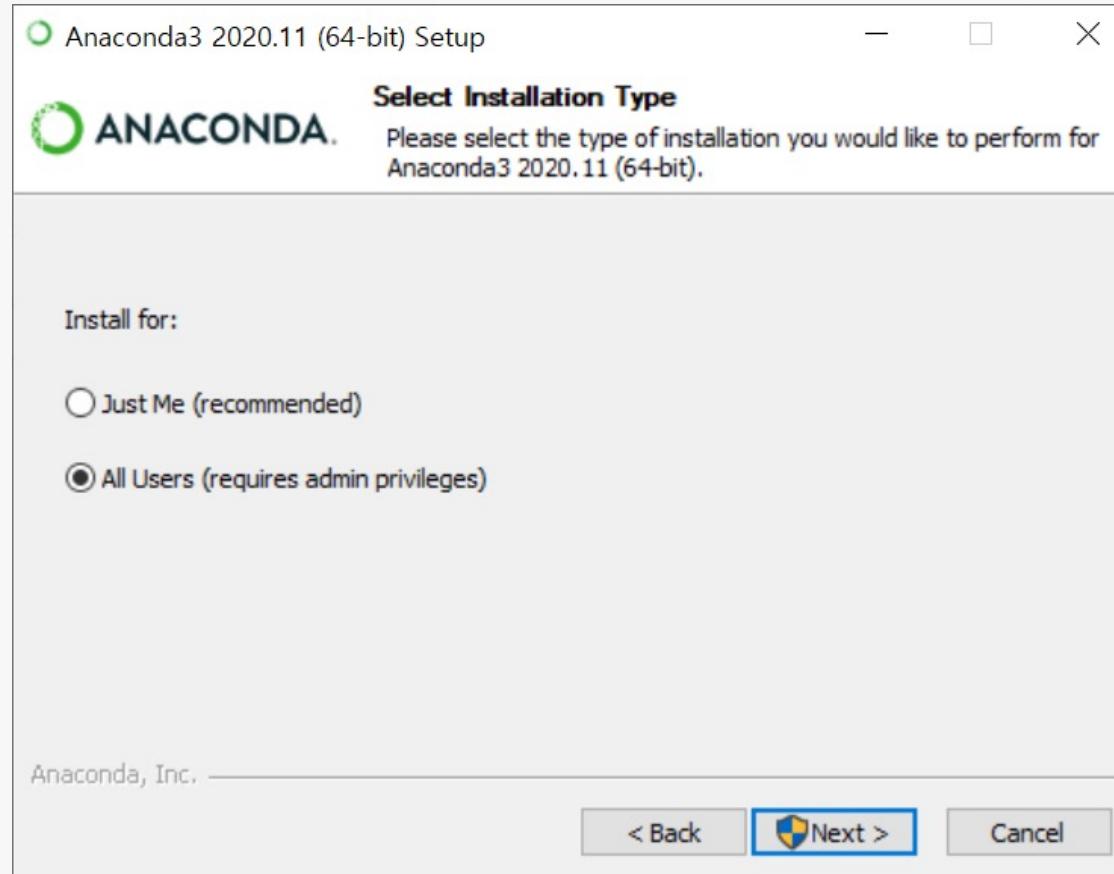
00. 아나콘다 – Windows 10

(2020년 12월 기준)



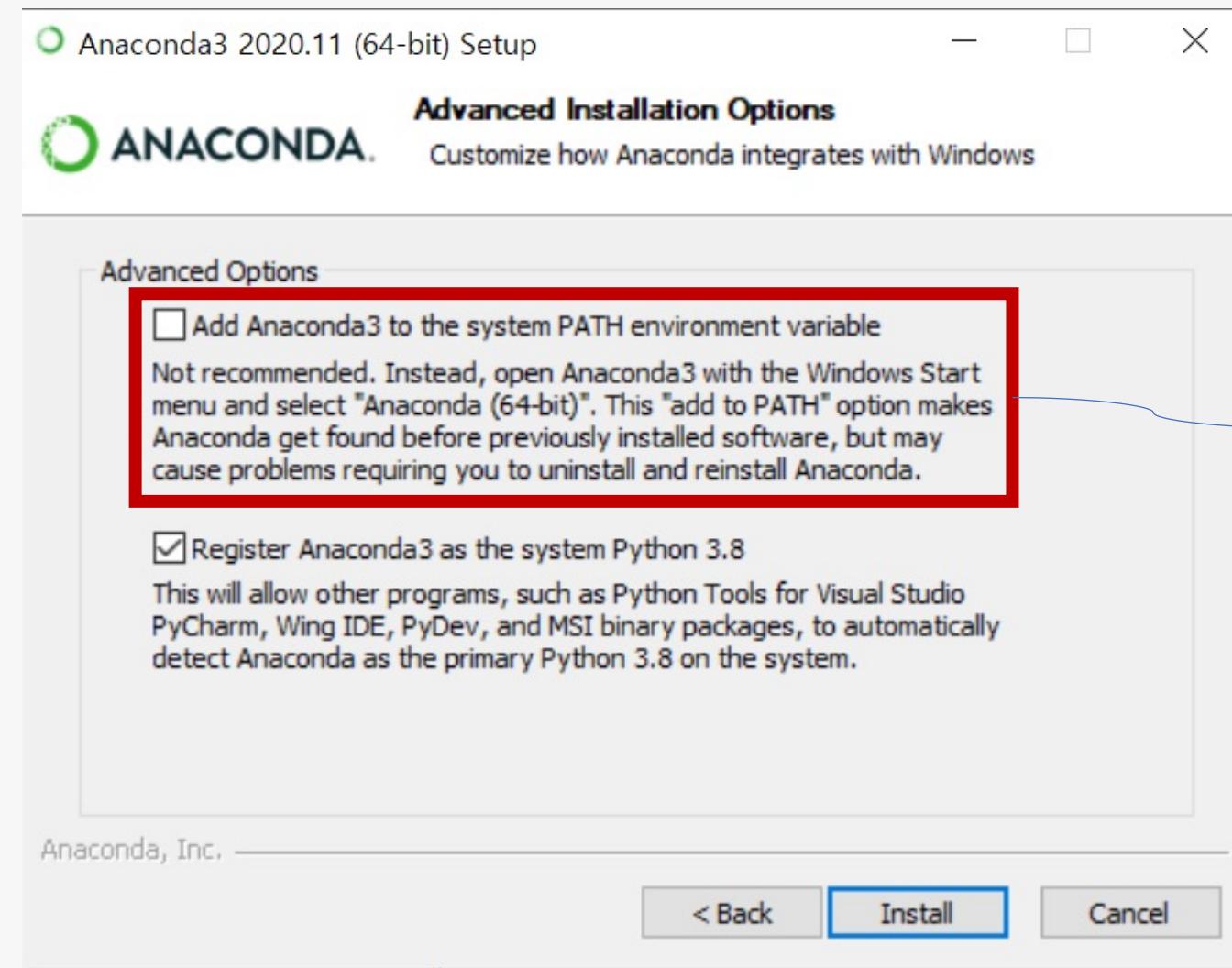
00. 아나콘다 – Windows 10

(2020년 12월 기준)



00. 아나콘다 – Windows 10

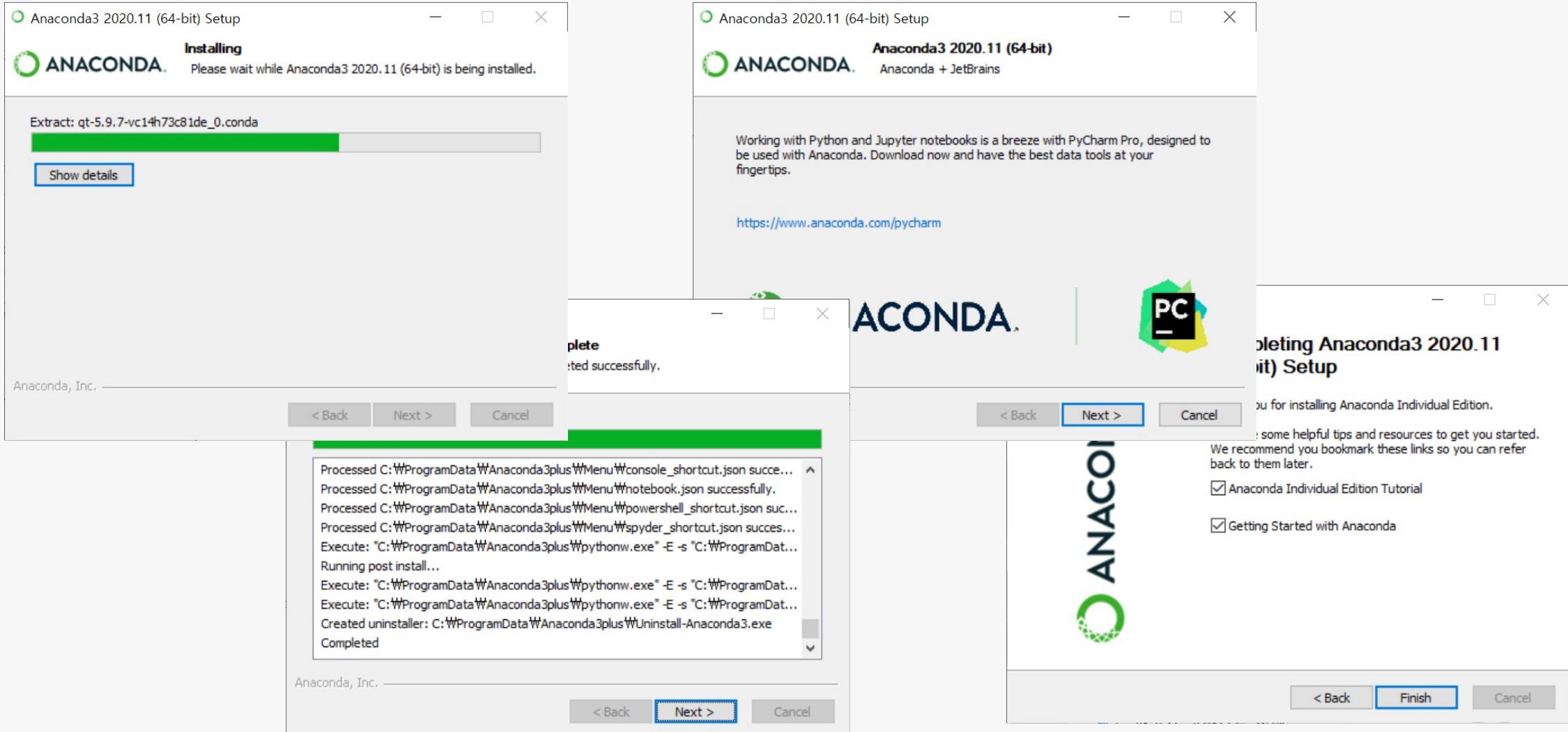
(2020년 12월 기준)



- 환경 변수의 개념을 잘 모른다면 추가하지 않는다.
- 환경 변수는 언제든지 추가가 가능하다.

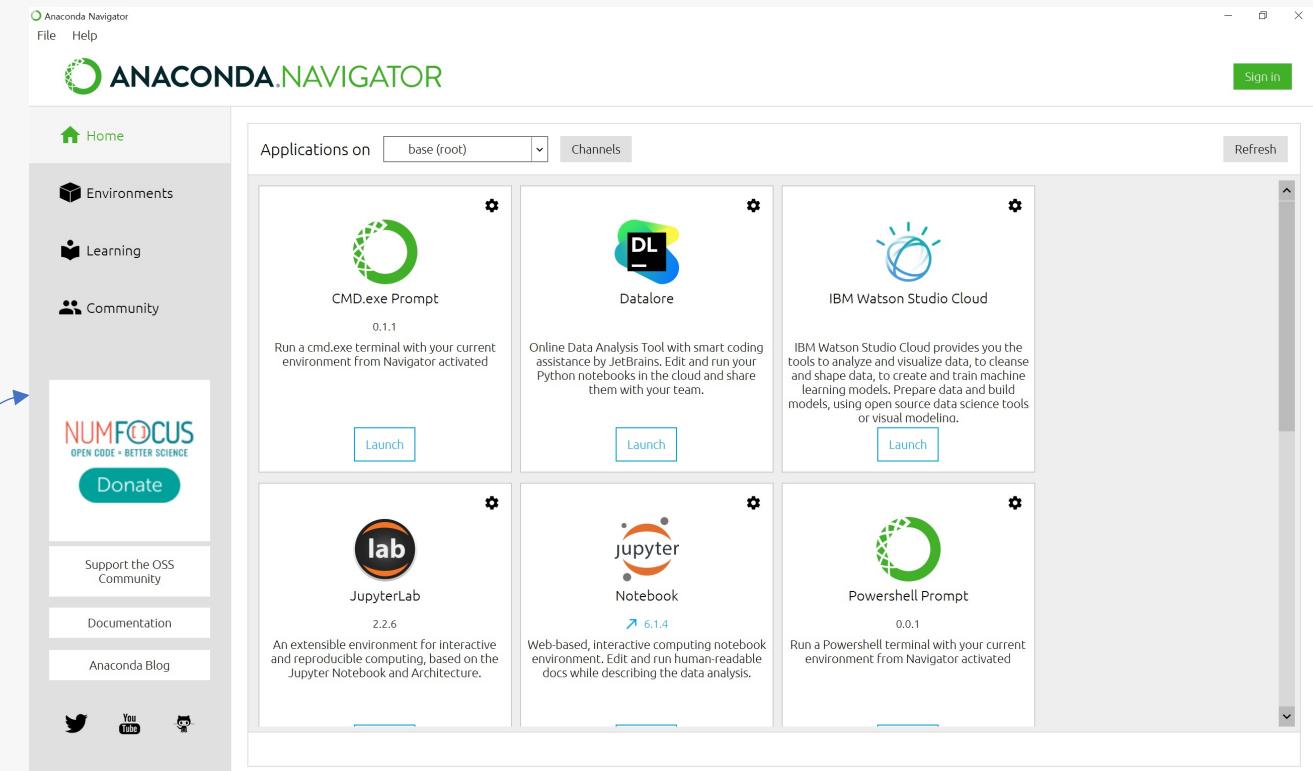
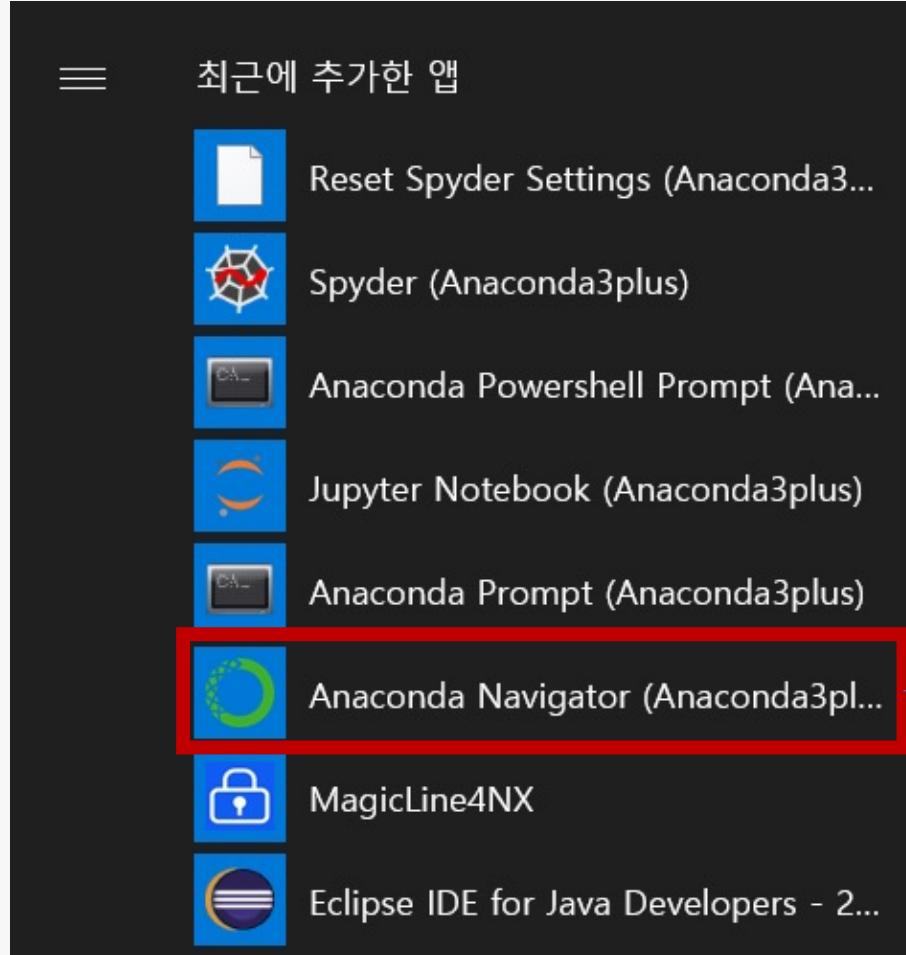
00. 아나콘다 – Windows 10

(2020년 12월 기준)

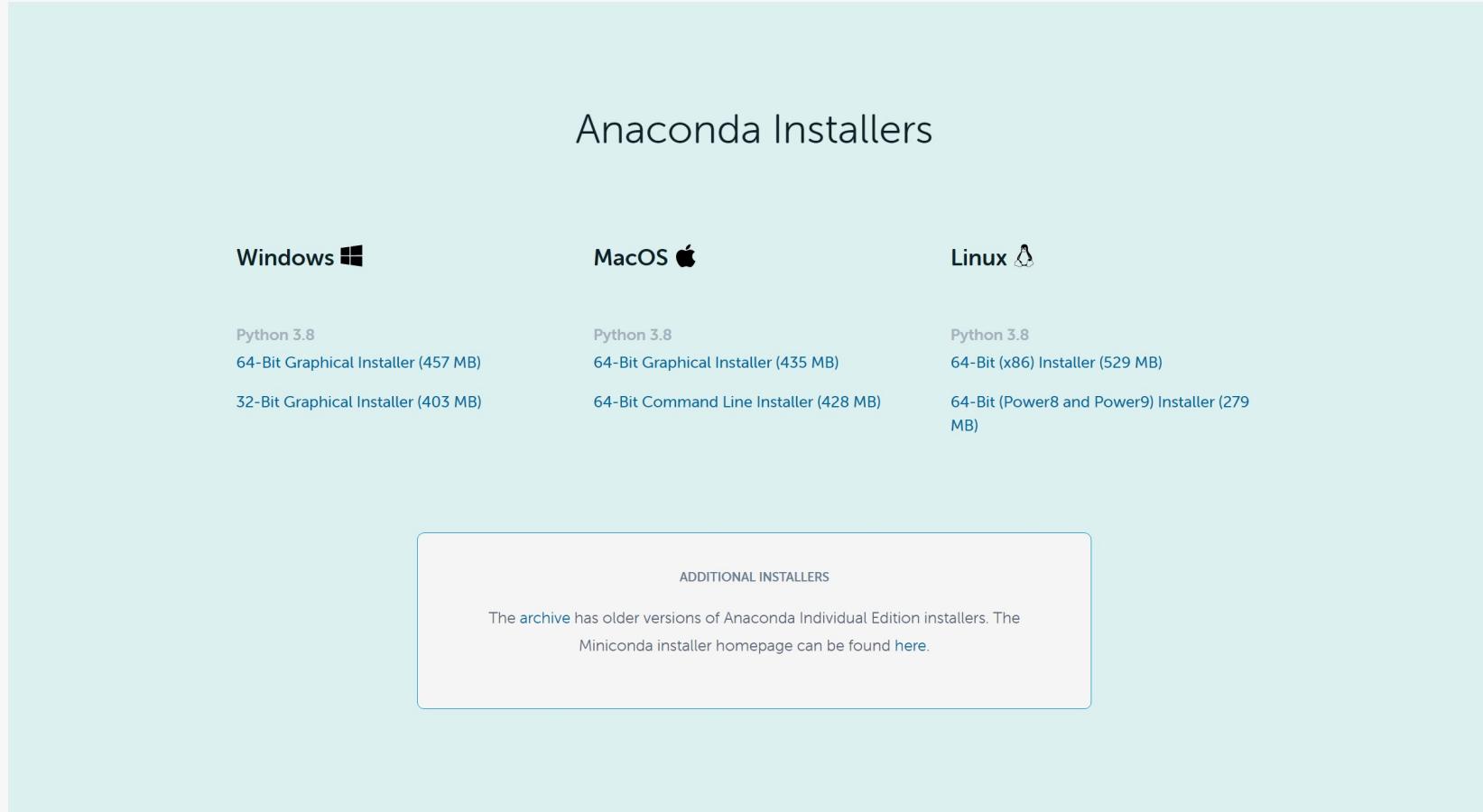


00. 아나콘다 – Windows 10

(2020년 12월 기준)

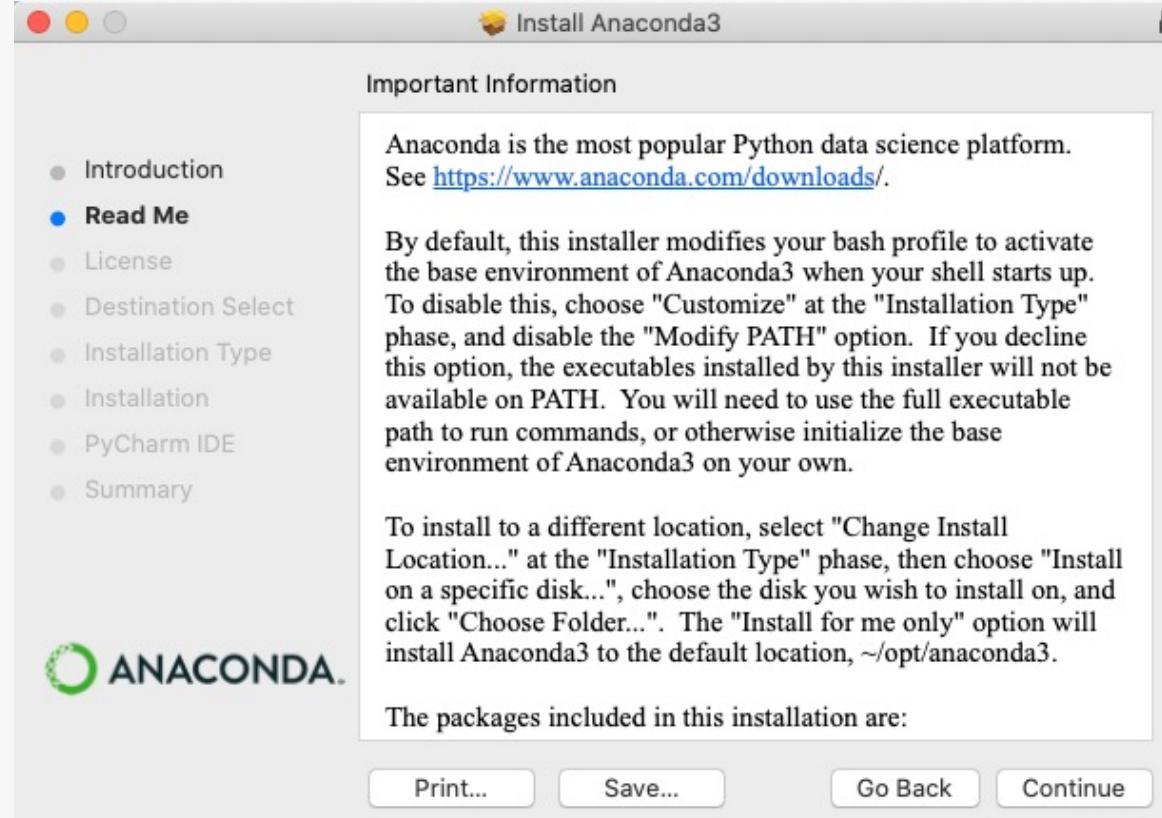
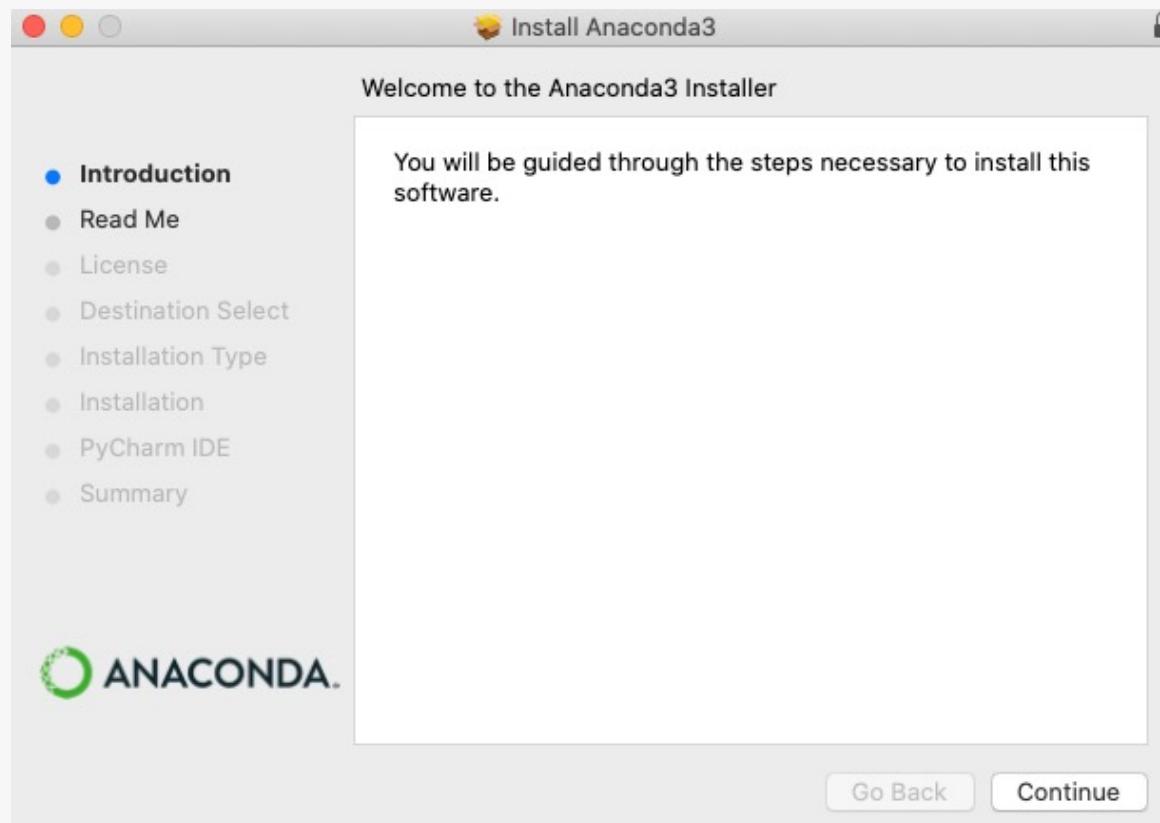


Shell Script 실행 방법 모른다면, 64-Bit Graphical Installer 선택
Shell Script 실행 방법 알고 있다면, 64-Bit Command Line Installer 선택



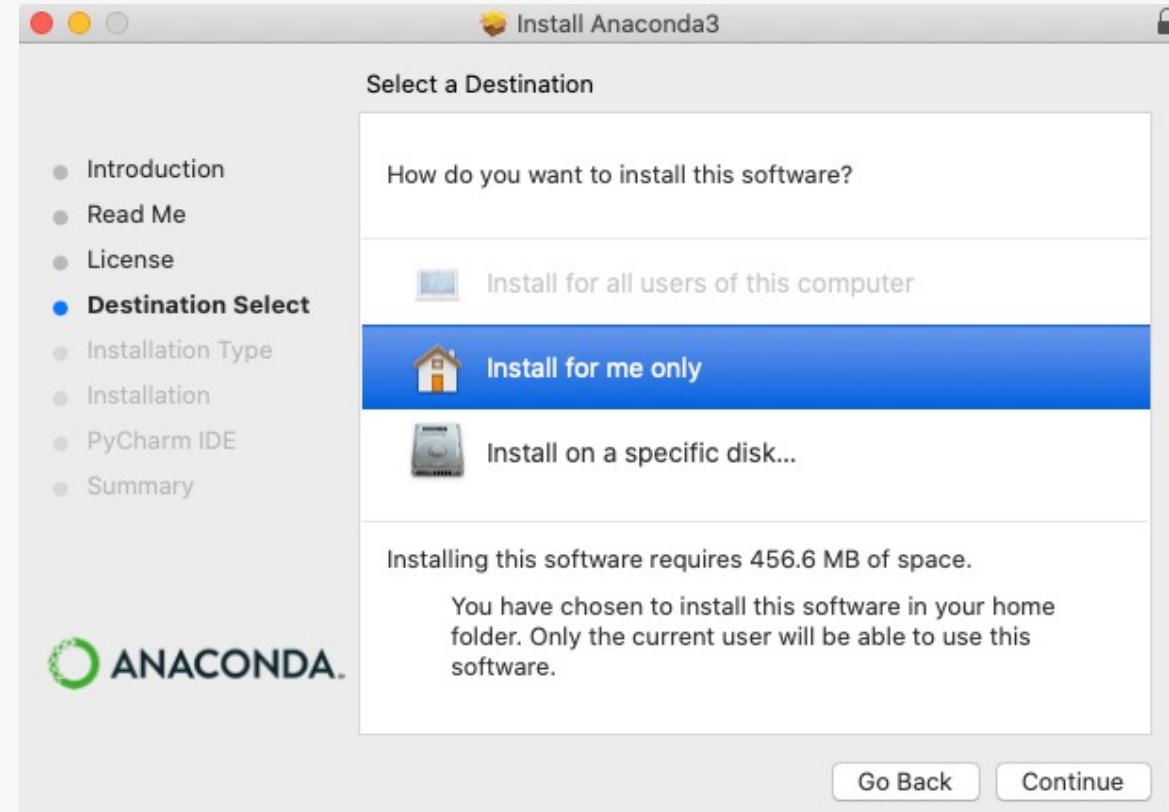
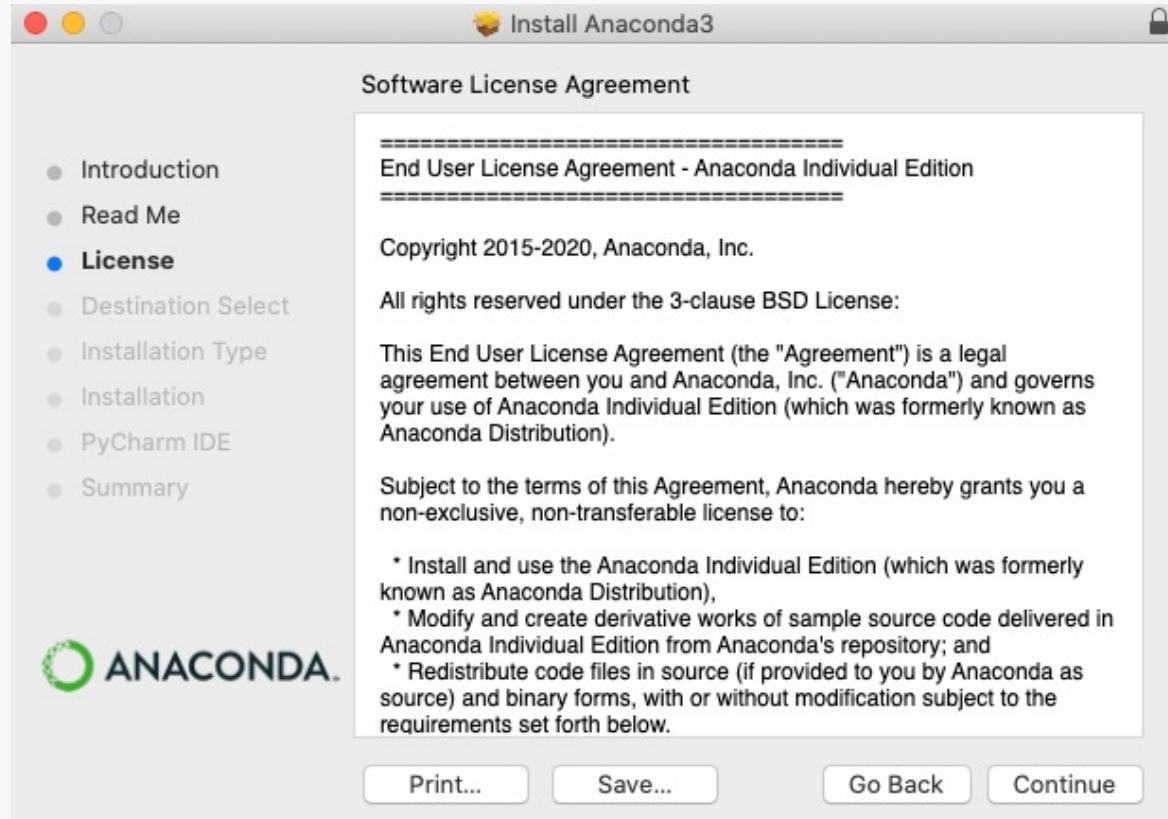
01. 아나콘다 – MacOS

(2020년 12월 기준)



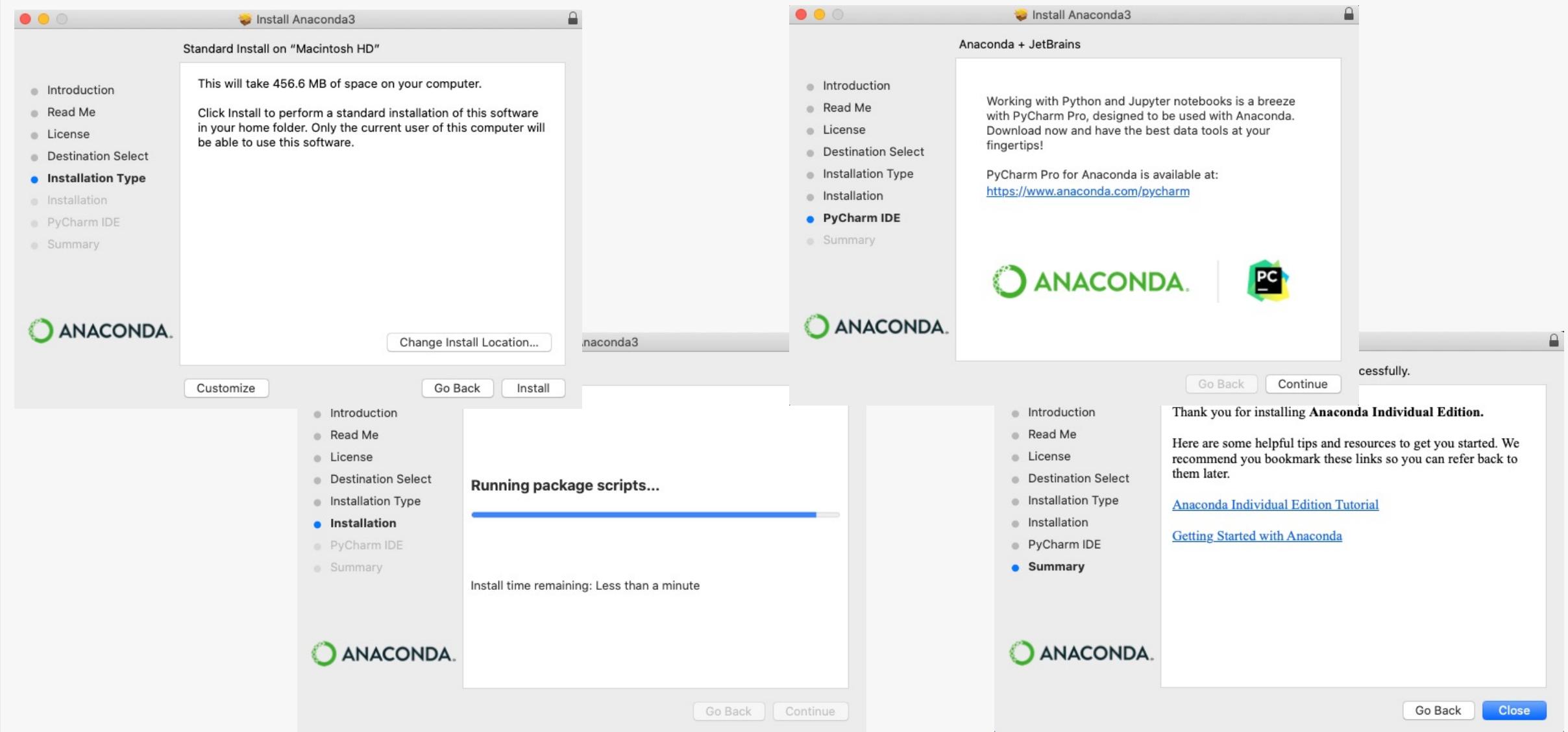
01. 아나콘다 – MacOS

(2020년 12월 기준)



01. 아나콘다 – MacOS

(2020년 12월 기준)



01. 아나콘다 – MacOS

(2020년 12월 기준)

The screenshot shows the Anaconda Navigator application window. The left sidebar has links for Home, Environments, Learning, and Community, along with a NUMFOCUS donation button and social media links. The main area displays a grid of applications:

Application	Description	Version	Action
Datalore	Online Data Analysis Tool with smart coding assistance by JetBrains.	2.2.6	Launch
IBM Watson Studio Cloud	IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.	2.2.6	Launch
JupyterLab	An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.	2.2.6	Launch
jupyter Notebook	Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.	6.1.4	Launch
PyCharm Community	An IDE by JetBrains for pure Python development. Supports code completion, listing, and debugging.	2020.2.1	Launch
Qt Console	PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.	4.7.7	Launch
Spyder	Scientific PYthon Development EnvIRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features.	4.1.5	Launch
VS Code	Streamlined code editor with support for development operations like debugging, task running and version control.	1.52.1	Launch
Glueviz	Multidimensional data visualization across files. Explore relationships within and among related datasets.	1.0.0	Install
Orange 3	Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.	3.26.0	Install
PyCharm Professional	A full-fledged IDE by JetBrains for both Scientific and Web Python development. Supports HTML, JS, and SQL.	2020.2.1	Install
RStudio	A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.	1.1.456	Install

데이터 분석 강의 Intro

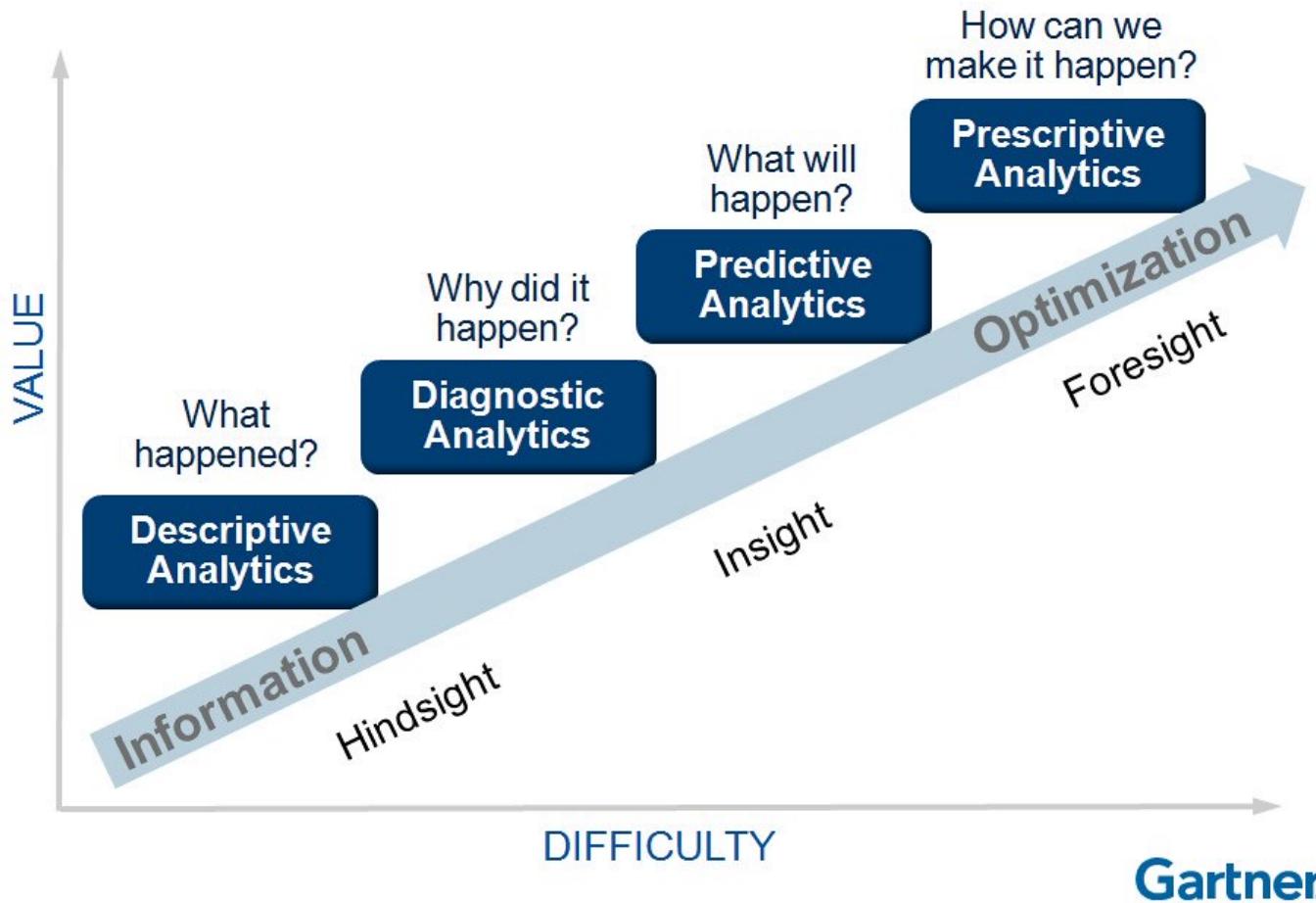
with



PostgreSQL

01. 데이터 분석 가치 4단계

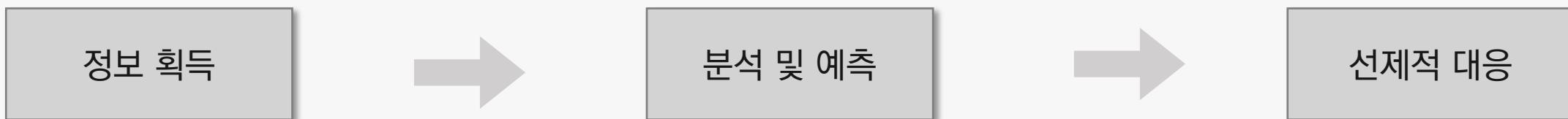
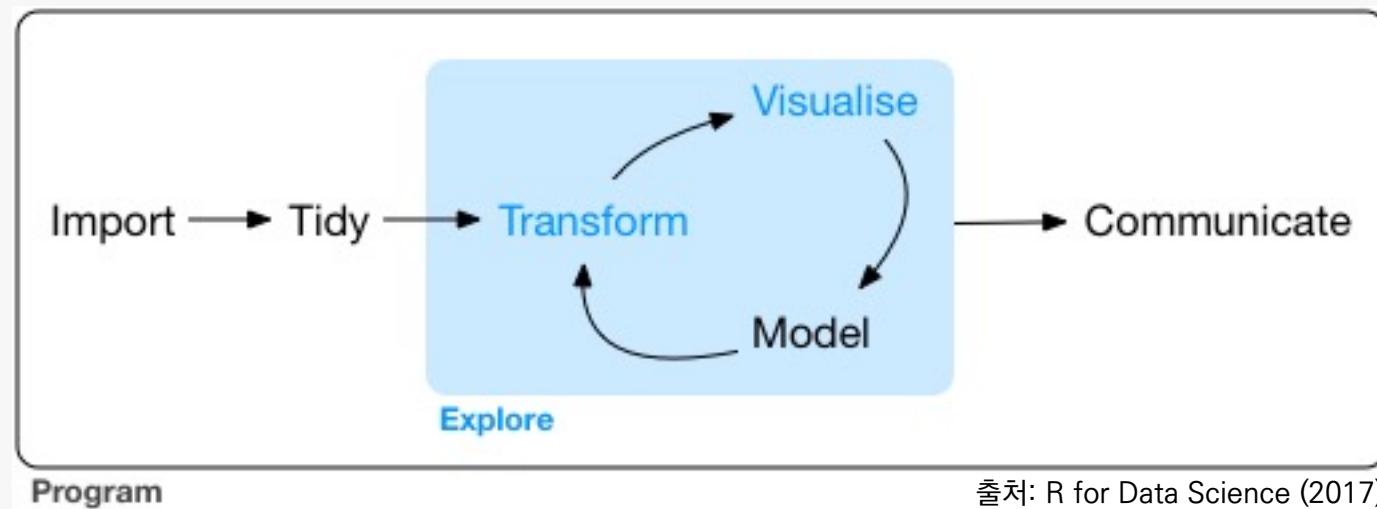
Analytic Value Escalator



출처: Gartner (2012)

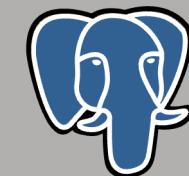


01. 데이터 분석 4단계



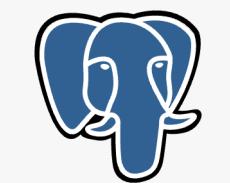
개발환경설정

with



PostgreSQL

01. PostgreSQL



PostgreSQL



DBMS
(Database Management System)

관계형 데이터베이스
(Relational Database)

PostgreSQL의 인기

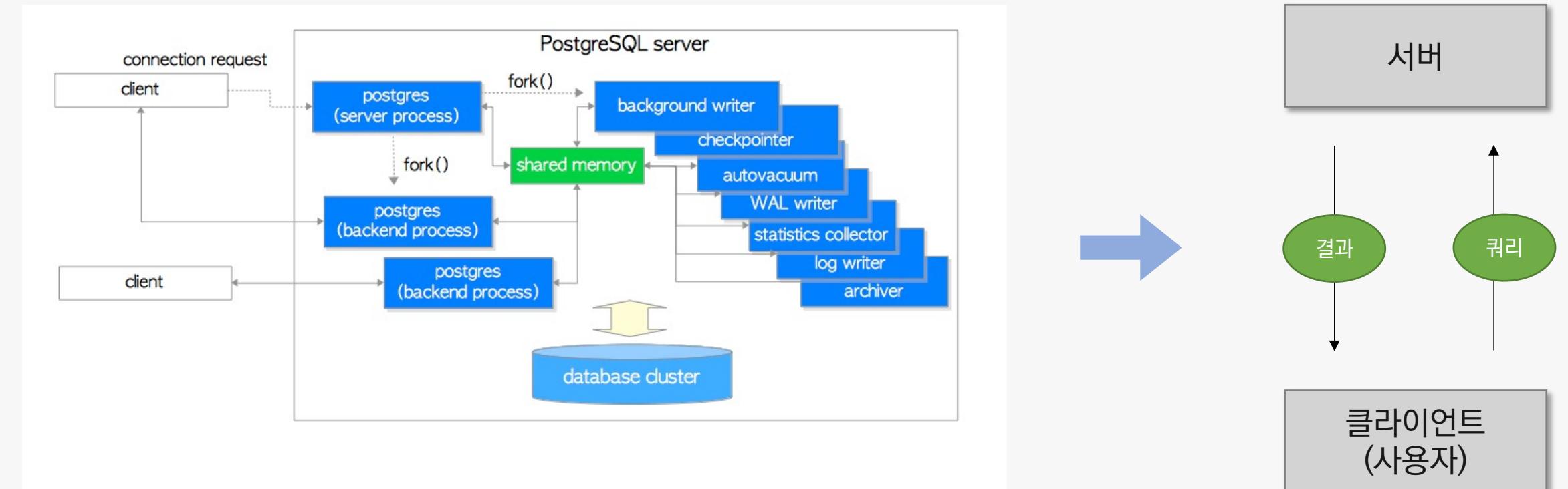
PostgreSQL의 장점

- 정의 (Definition) : 응용 프로그램이 요구하는 DB구조, 변경, 제거
 - 조작 (Manipulation): 삽입, 갱신, 삭제
 - 제어 (Control): DB 접근할 수 있는 사용자 제한 및 성능 관리
-
- 통상적으로 RDB라고 함
 - 행과 열로 구분하는 2차원 테이블 형태로 구성
 - 오라클 DBMS, MS SQL SERVER, MYSQL, PostgreSQL
-
- DB Engines Ranking 4위 (350개 DB) (2020 ~ 2021)¹
 - StackOverflow 2020 Survey 2위 (Most Loved & Wanted)²
-
- 무료 라이선스 (GPL) / 라이선스 비용 절감
 - 다양한 플랫폼 지원 (Windows, MacOS, Linux 등)

출처 1: <https://db-engines.com/en/ranking>

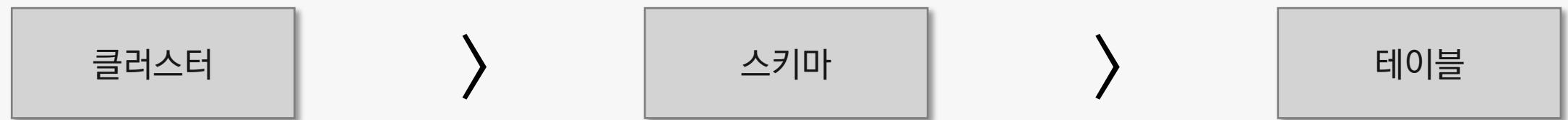
출처 2: <https://insights.stackoverflow.com/survey/2020#technology-most-loved-dreaded-and-wanted-databases-loved4>

02. PostgreSQL 프로세스 구조



출처 : <https://mangkyu.tistory.com/71>

03. 기본 용어 설명



- DB 최상위 개념
- 스키마(Schema) 개체들의 논리적인 집합

- 테이블, 뷰, 함수, 인덱스 등
- Public 스키마 (Default)

- 2차원의 표 (행과 열)
- Relation
- 개체 식별 위한 Key 존재

04. PostgreSQL 설치 (Windows 10)

(2021년 6월 기준)

PostgreSQL 다운로드

(웹사이트 주소: <https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>)

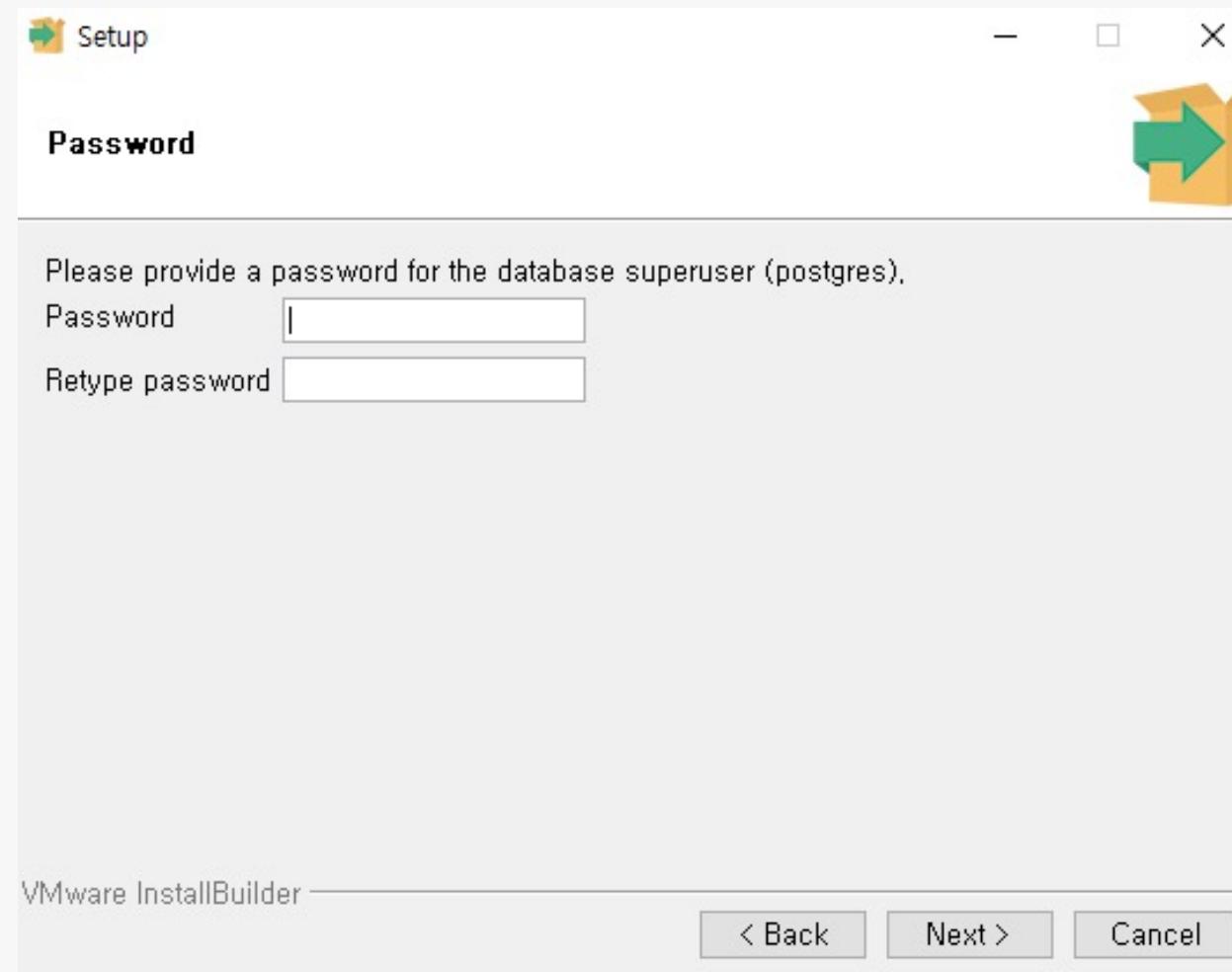
The screenshot shows the 'PostgreSQL Database Download' page. At the top, there's a navigation bar with links for Why EDB, Products, Services, Support, Resources, Plans, Contact, Sign In, and Downloads (which is highlighted with a red border). Below the navigation is a table listing PostgreSQL versions and their download links for various platforms.

Version	Linux x86-64	Linux x86-32	Mac OS X	Windows x86-64	Windows x86-32
13.3	N/A	N/A	Download	Download	N/A
12.7	N/A	N/A	Download	Download	N/A
11.12	N/A	N/A	Download	Download	N/A
10.17	Download				
9.6.22	Download				
9.5.25 (Not Supported)	Download				
9.4.26 (Not Supported)	Download				
9.3.25 (Not Supported)	Download				

04. PostgreSQL 설치 (Windows 10)

(2021년 6월 기준)

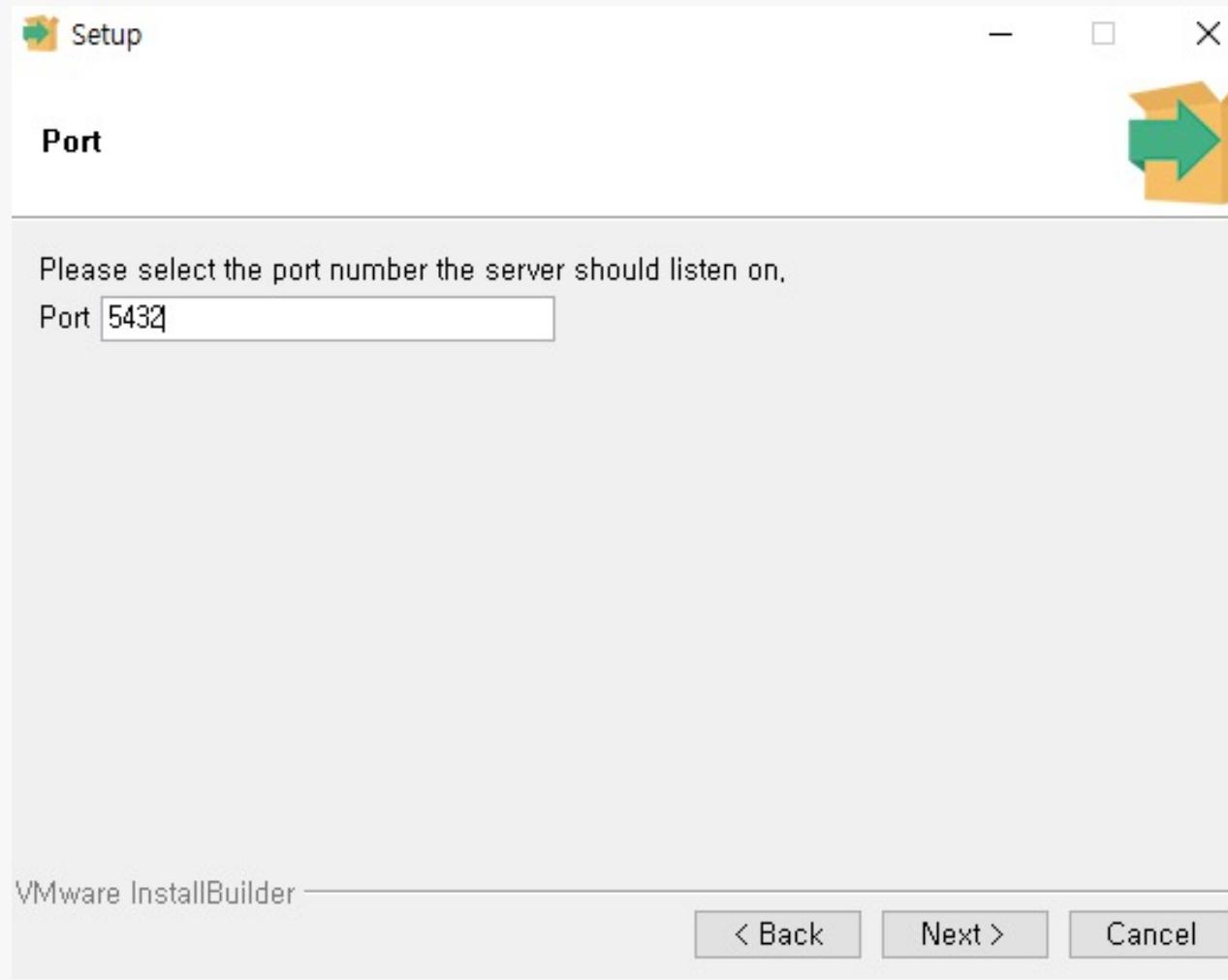
입력한 비밀번호를 기억하도록 합니다.



04. PostgreSQL 설치 (Windows 10)

(2021년 6월 기준)

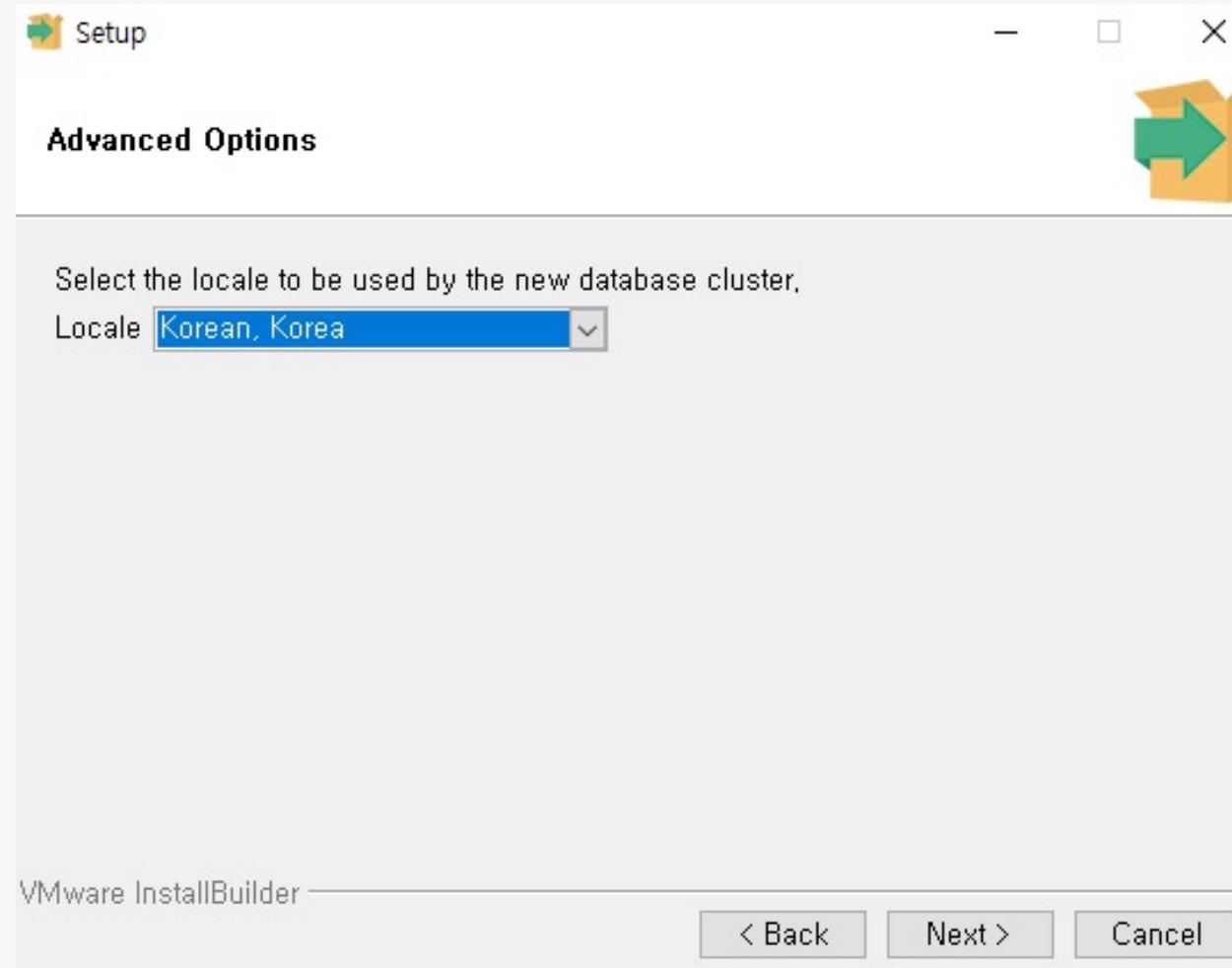
포트번호가 5432가 맞는지 확인합니다.



04. PostgreSQL 설치 (Windows 10)

(2021년 6월 기준)

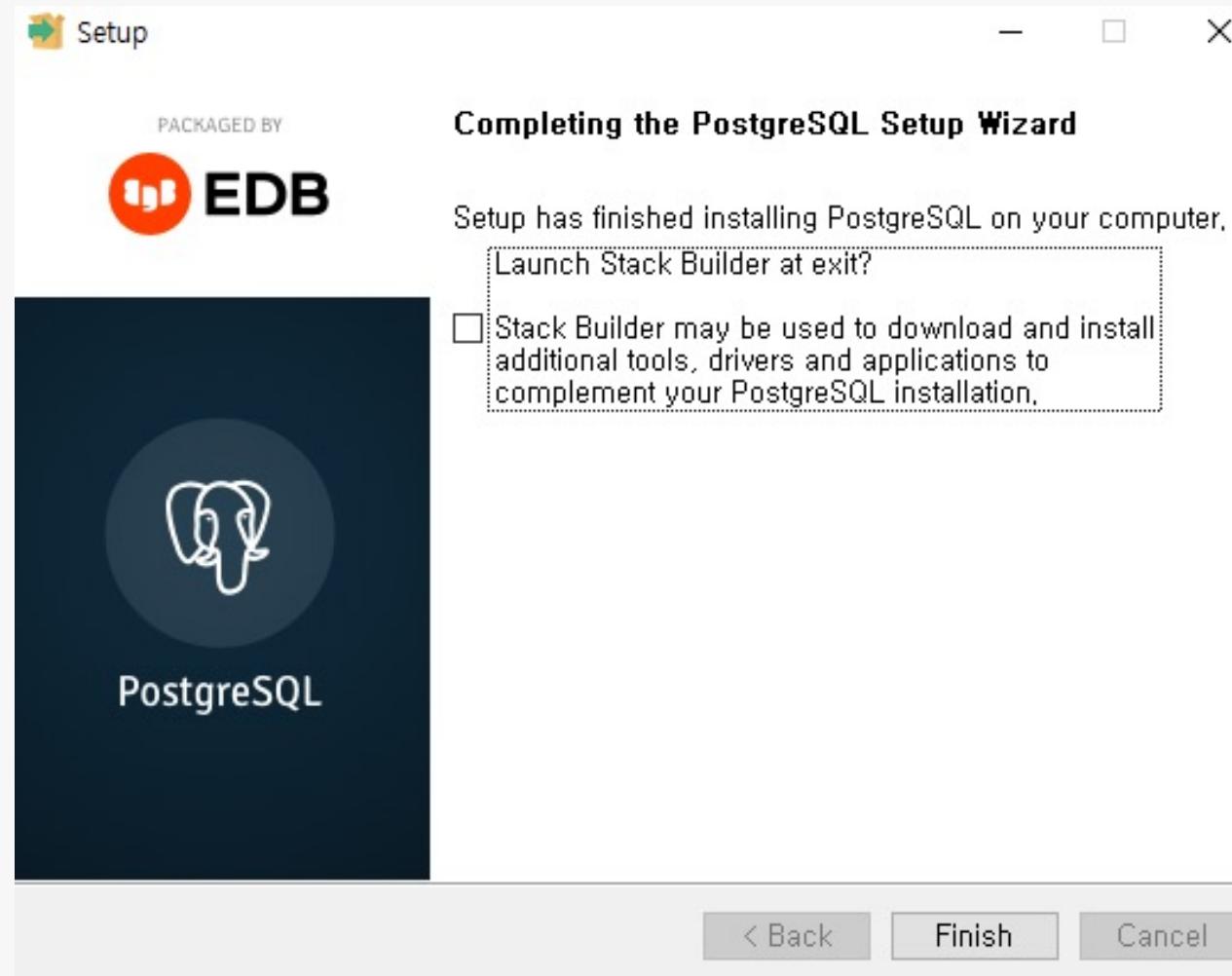
언어는 한국어 또는 영어로 선택하도록 합니다.



04. PostgreSQL 설치 (Windows 10)

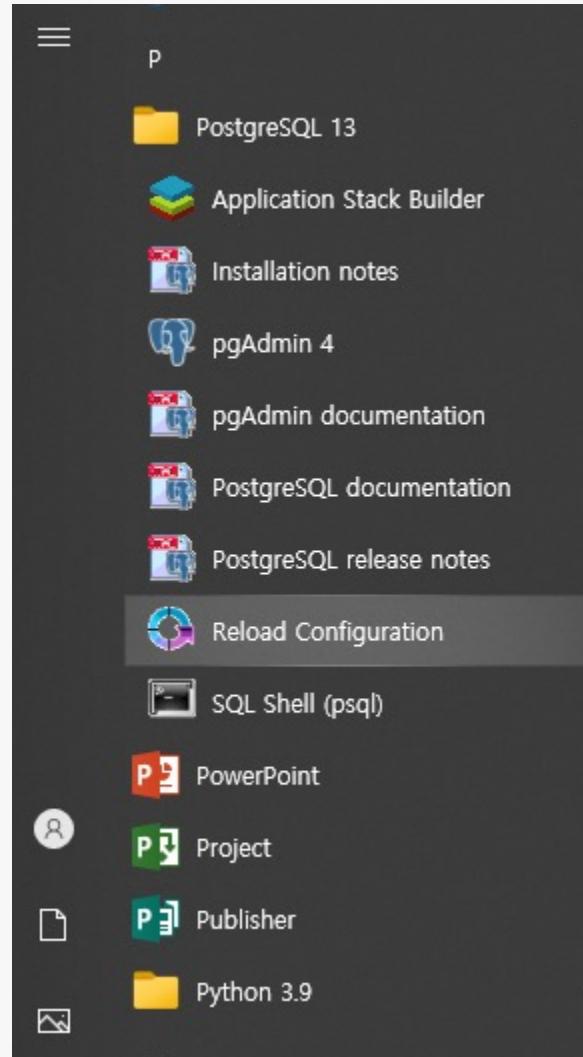
(2021년 6월 기준)

설치가 완료되면 Stack Builder 체크 박스는 해제 하도록 합니다.



04. PostgreSQL 설치 (Windows 10)

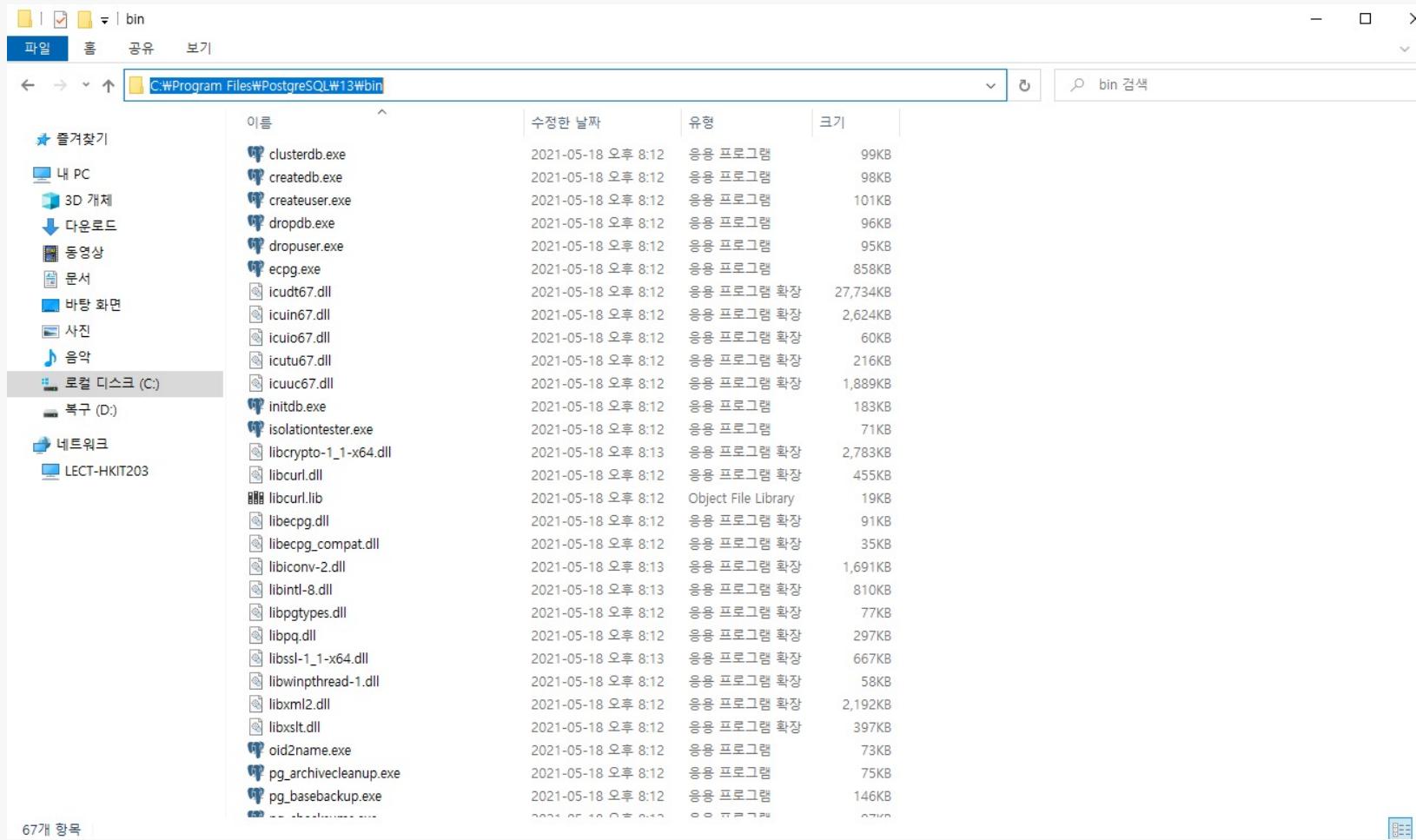
(2021년 6월 기준)



04. PostgreSQL 설치 (Windows 10)

(2021년 6월 기준)

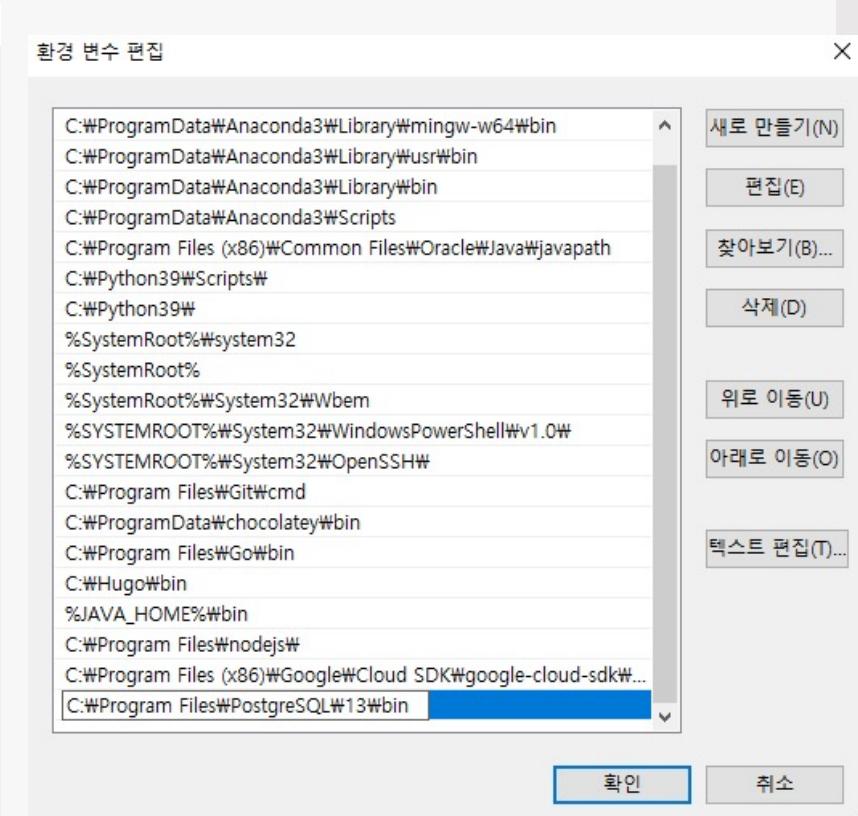
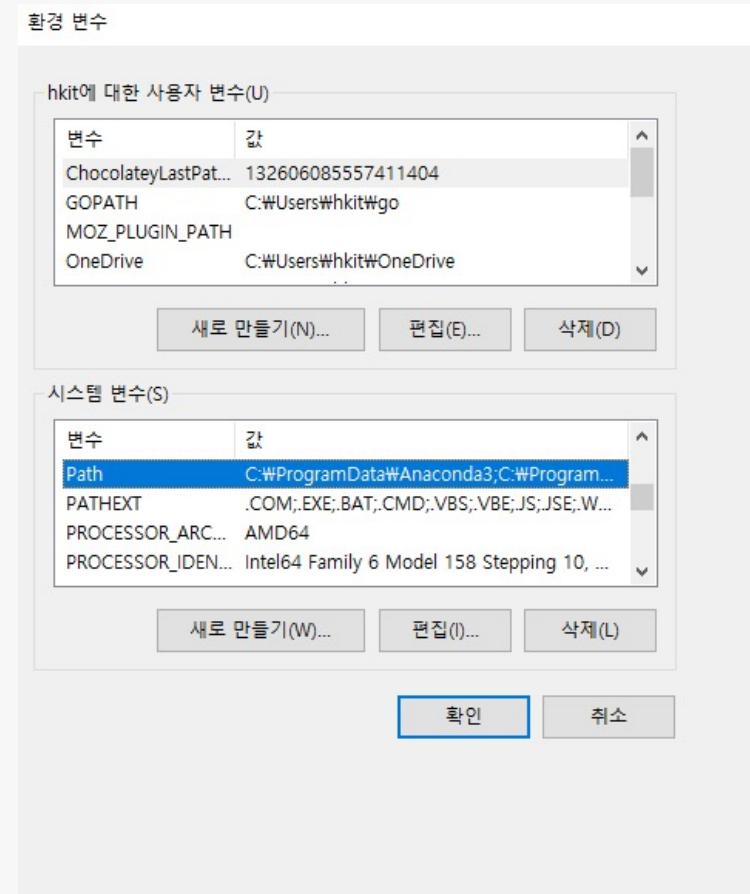
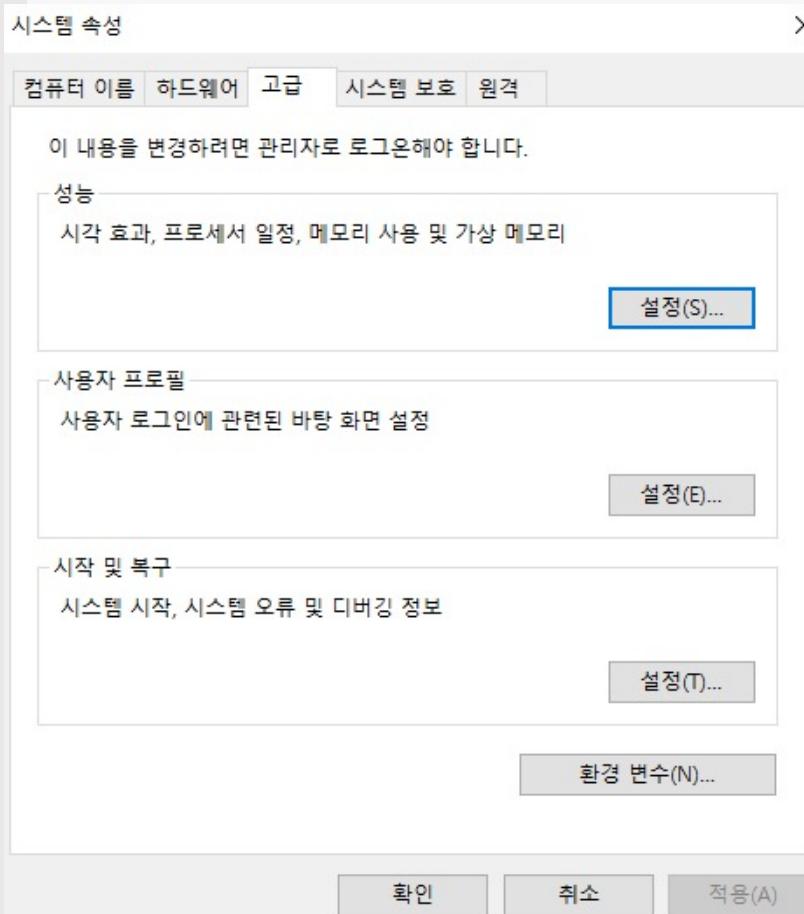
환경변수 추가를 위해 아래 경로를 복사합니다.
경로: C:\Program Files\PostgreSQL\13\bin



04. PostgreSQL 설치 (Windows 10)

(2021년 6월 기준)

시스템 속성 – 환경 변수 – 환경 변수 편집
주소: C:\Program Files\PostgreSQL\13\bin



04. PostgreSQL 설치 (Windows 10)

(2021년 6월 기준)

SQL Shell 스크립트 접속 예시

The image shows two separate Command Prompt windows side-by-side. Both windows have a title bar labeled '명령 프롬프트'.

The left window contains the following text:

```
C:\Users\Whkit>
C:\Users\Whkit>
C:\Users\Whkit>
C:\Users\Whkit>
```

The right window contains the following text:

```
C:\Users\Whkit>
C:\Users\Whkit>
C:\Users\Whkit>
C:\Users\Whkit>psql -U postgres
postgres 사용자의 암호:
psql (13.3)
도움말을 보려면 "help"를 입력하십시오.

postgres=#
```

05. PostgreSQL 설치 (MacOS)

(2021년 6월 기준)

PostgreSQL 다운로드

(웹사이트 주소: <https://postgresapp.com/downloads.html>)

The screenshot shows the Postgres.app website's download page. At the top left is a large blue elephant icon. To its right, the text "Postgres.app" is displayed in a large, bold font, followed by the subtitle "The easiest way to get started with PostgreSQL on the Mac". Below this are four navigation buttons: "Introduction" (disabled), "Downloads" (selected and highlighted in blue), "Documentation", and "GitHub". A progress bar indicates "← 5747 Stars!".

Latest Release

If you're new to Postgres, this is the file you should download. It includes everything you need to get started with PostgreSQL and PostGIS.

Postgres.app with PostgreSQL 13

Postgres.app v2.4.3 · Requires macOS 10.12 · Download Size 66MB

PostgreSQL 13.3 / PostGIS 3.1.1 / plv8 2.3.15

Download

05. PostgreSQL 설치 (MacOS)

(2021년 6월 기준)

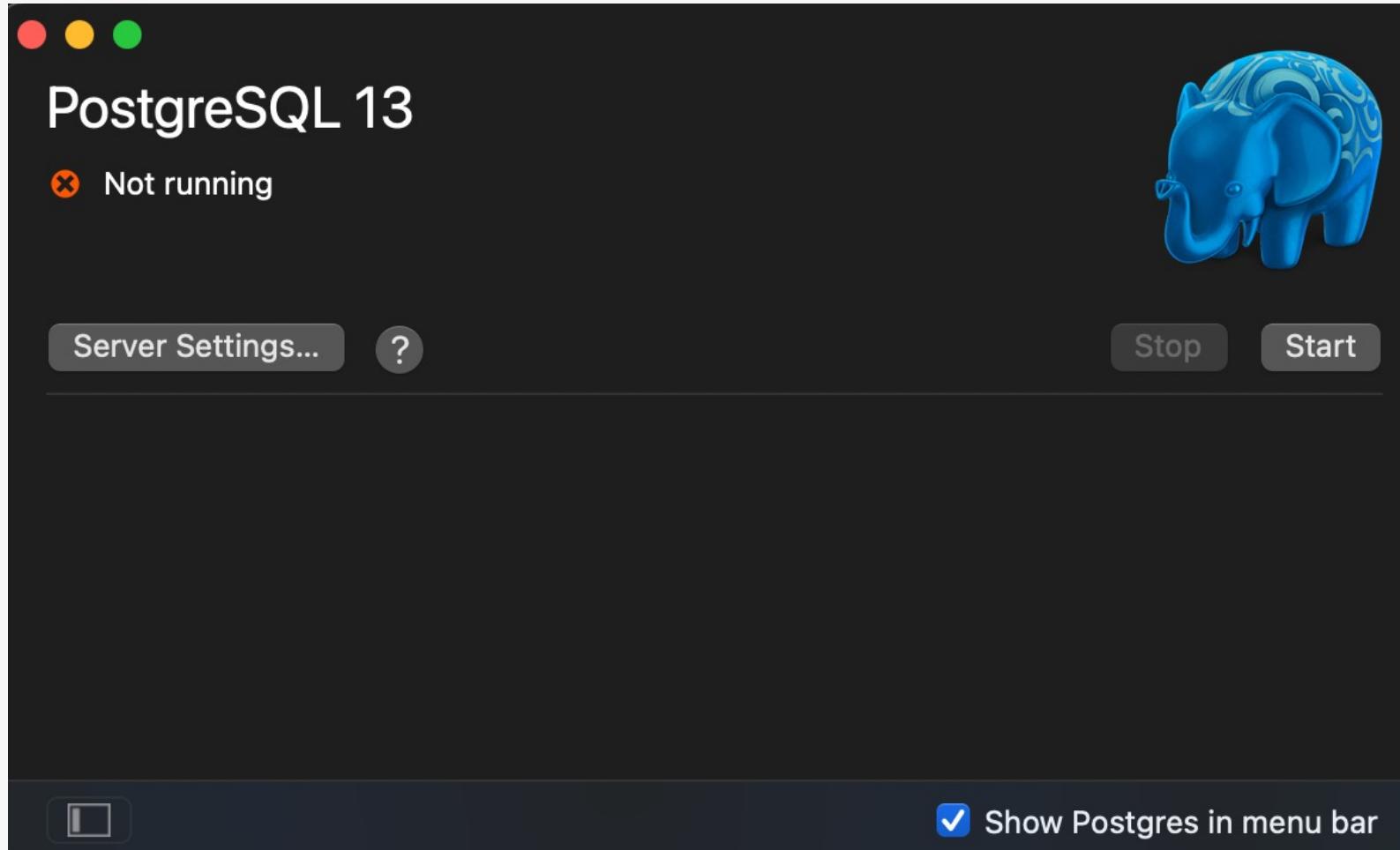
PostgreSQL 다운로드
(웹사이트 주소: <https://postgresapp.com/downloads.html>)



05. PostgreSQL 설치 (MacOS)

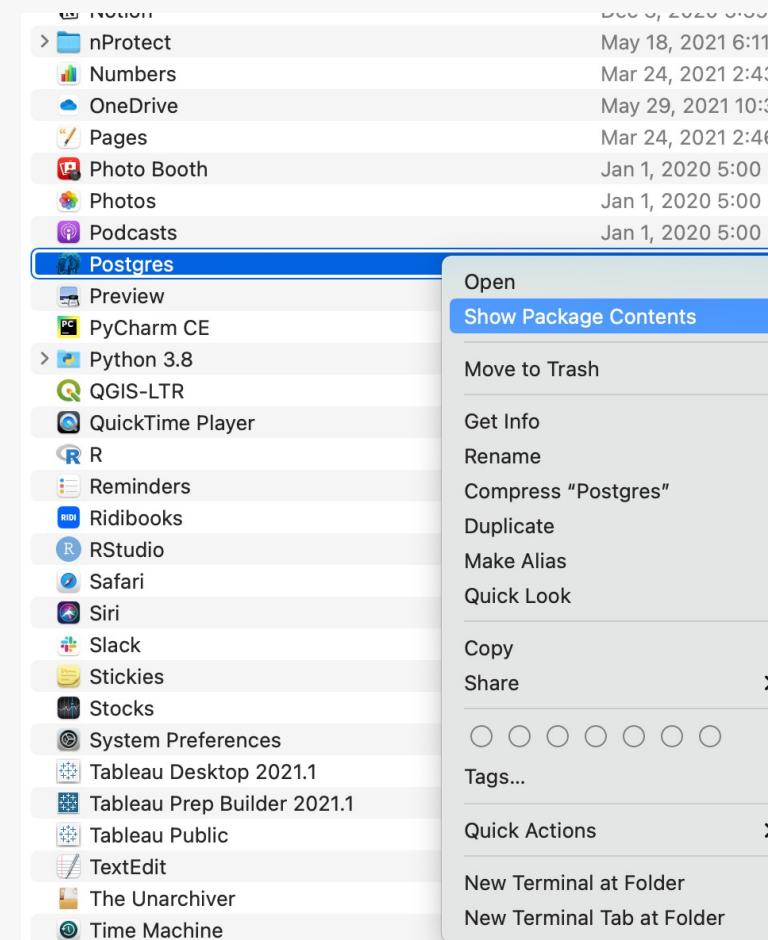
(2021년 6월 기준)

PostgreSQL 다운로드
(웹사이트 주소: <https://postgresapp.com/downloads.html>)



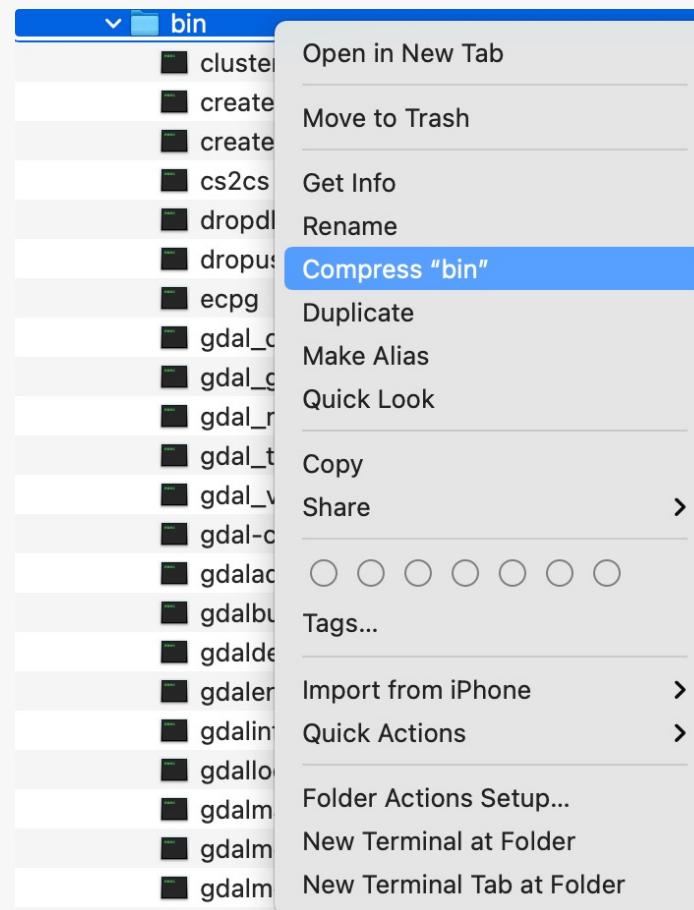
환경변수 추가

/Applications/Postgres.app/Contents/Versions/13/bin



환경변수 추가

중요: 맥 옵션(Options)키를 클릭하면 Copy “bin” as Pathname 바뀝니다.

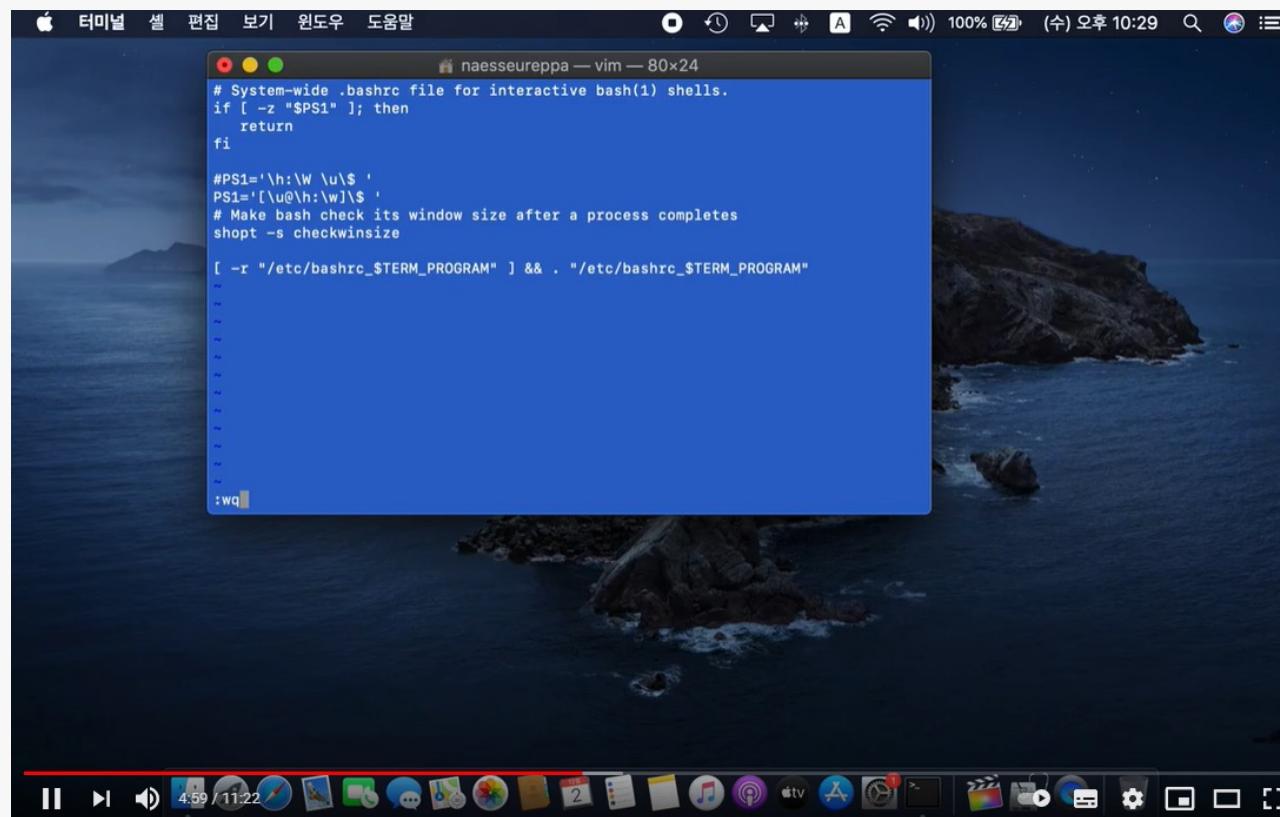


환경변수 추가

환경변수 추가를 위해서는 bash_profile 등을 수정해야 합니다.

처음 작업하시는 분들은 아래 영상을 보시기를 바랍니다.

영상주소: <https://youtu.be/mCFckNvrCQg>



05. PostgreSQL 설치 (MacOS)

(2021년 6월 기준)

환경변수 추가

```
export PATH=/opt/homebrew/bin:$PATH:/Applications/Postgres.app/Contents/Versions/13/bin
```



```
[REDACTED]-MacBook-Air-2:~ evan$ psql  
psql (13.3)  
Type "help" for help.
```

```
evan=#
```

06. PostgreSQL 활용을 위한 다양한 GUI (Tools)



Postico

A Modern PostgreSQL Client for the Mac



DBeaver Community

Free Universal Database Tool



Version 4
pgAdmin



TablePlus

06. PostgreSQL 활용을 위한 다양한 GUI (Tools)

참조: <https://americanopeople.tistory.com/316>

소프트웨어 이야기/데이터 저장소 + 시각화

(PostgreSQL) PostgreSQL Client Tool 비교하기

americano_people | 2019. 4. 13. 19:18

나는 PostgreSQL Client 유목민이다 🌟🌟

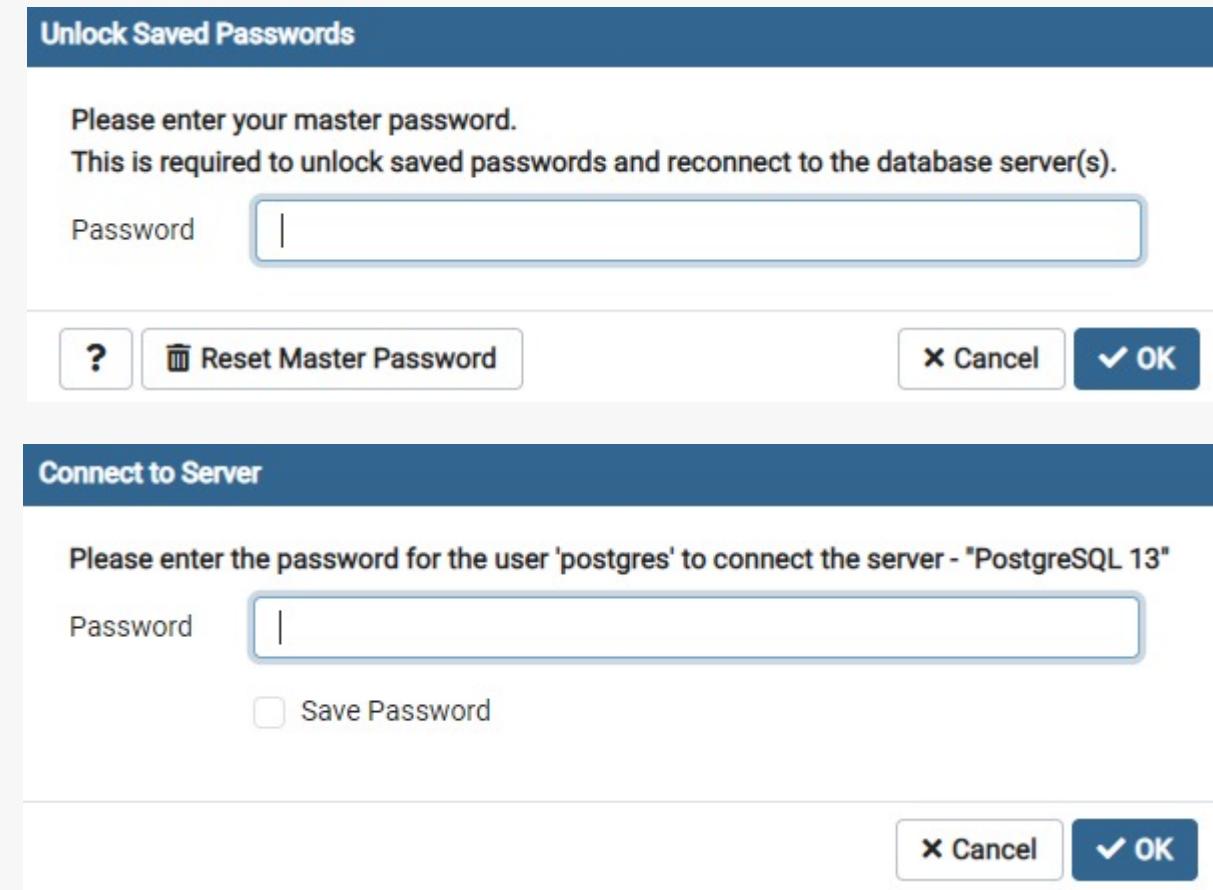
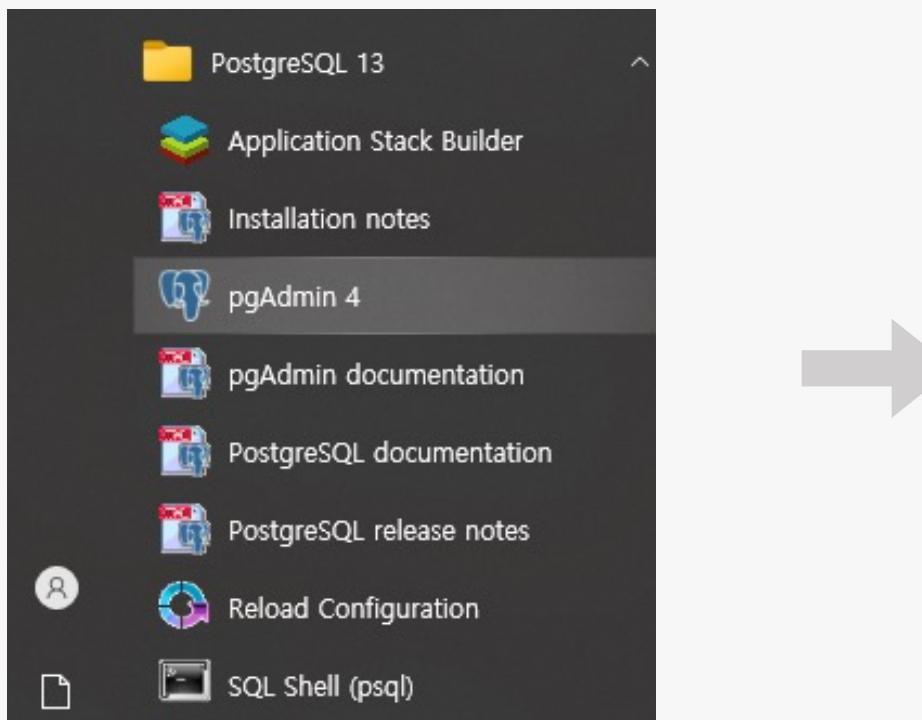
그러던 중 [PostgreSQL Client 비교한 글](#)을 읽어봤는데, 꽤 유용했다!

적당한 툴을 찾지 못해서 PGAdmin, Psequel를 혼용해서 사용하고 있었는데, 이제는 TablePlus으로 정착하려고 한다.

PostgreSQL Client에 정착한 기념으로, 이번 포스팅에서 PostgreSQL Client Tool 을 정리해보고자 한다.

07. pgAdmin4 접속

*PostgreSQL 설치 시 설정했던 암호 입력



07. pgAdmin4 접속

The screenshot shows the pgAdmin4 application window. The top navigation bar includes File, Object, Tools, and Help menus. Below the menu bar is a toolbar with icons for Browser, Dashboard, Properties, SQL, Statistics, Dependencies, and Dependents. The main area is titled "Dependents" and displays the message: "No dependent information is available for the selected object." To the left is a tree view of the database structure under "Servers (1) > PostgreSQL 13 > Databases (1) > postgres". The "Dependents" tab is currently selected. The tree view lists various database objects: Casts, Catalogs, Event Triggers, Extensions, Foreign Data Wrappers, Languages, Publications, Schemas (1), public (Collations, Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Procedures, Sequences, Tables (4)), and Subscriptions. At the bottom of the tree view, there are "Login/Group Roles (9)" entries: pg_execute_server_program, pg_monitor, pg_read_all_settings, pg_read_all_stats, pg_read_server_files, and pg_signal_backend.

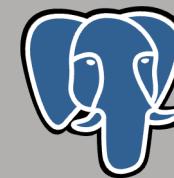
08. pgAdmin4 튜토리얼

공식 튜토리얼: <https://www.pgadmin.org/videos/>

The screenshot shows the pgAdmin website's navigation bar with links for Home, Development, Documentation, Download, and Support. Below the navigation is a sidebar titled 'Quick Links' containing icons for Online Demo, Download, FAQ, Latest Docs, Get Help, and Screenshots. The main content area is titled 'Videos' and features a video thumbnail for '2021-02-23 - ERD Demo (EDB)'. The thumbnail displays a presentation slide for a PostgreSQL tutorial, specifically the pgAdmin Entity-Relationship Diagram (ERD) Tool Demo. The slide has a dark blue background with the text 'PostgreSQL Tutorial' and 'pgAdmin: Entity-Relationship Diagram (ERD) Tool Demo' along with a play button icon. To the right of the video player, there are social sharing icons for YouTube and other platforms.

SQL Shell 명령어

with



PostgreSQL

01. SQL Shell 명령어

명령어	설명	예시 코드
\q	psql을 종료합니다.	-
\l	데이터베이스를 조회합니다.	-
\c	입력한 데이터베이스로 이동	\c DB이름
\e	외부편집기로 SQL쿼리를 입력	-
\dt	현재 데이터베이스에서 테이블을 확인	

명령 프롬프트 - psql -U postgres

```
postgres=# help
PostgreSQL에 대한 명령행 인터페이스인 psql을 사용하고 있습니다.
사용법: \copyright 저작권 정보
        \h SQL 명령 도움말
        \? psql 명령 도움말
        \g 또는 명령 끝에 세미콜론(;) 쿼리 실행
        \q 마침
postgres=#
```

DB 생성 및 삭제하기

with



01. DB 생성 쿼리문

DATABASE 생성 쿼리문

CREATE DATABASE *store*;



데이터베이스 이름

02. DB 접속 SHELL SCRIPT

```
\c store
```



데이터베이스 이름

```
book=# CREATE DATABASE store;  
CREATE DATABASE  
book=# \c store  
접속정보: 데이터베이스="store", 사용자="postgres".  
store=# \l
```

이름	소유주	인코딩	데이터베이스 목록		액세스 권한
			Collate	Ctype	
book	postgres	UTF8	Korean_Korea.949	Korean_Korea.949	
book_store	postgres	UTF8	Korean_Korea.949	Korean_Korea.949	
postgres	postgres	UTF8	Korean_Korea.949	Korean_Korea.949	
store	postgres	UTF8	Korean_Korea.949	Korean_Korea.949	
template0	postgres	UTF8	Korean_Korea.949	Korean_Korea.949	=c/postgres + postgres=CTc/postgres
template1	postgres	UTF8	Korean_Korea.949	Korean_Korea.949	=c/postgres + postgres=CTc/postgres
(6개 행)					

03. DB 삭제 쿼리문

DATABASE 삭제 쿼리문

DROP DATABASE store;

TABLE 생성 및 삭제하기

with
PostgreSQL



01. TABLE 생성 쿼리문

Table 생성 쿼리문

```
CREATE TABLE book_store (
    column_name data_type
    ...
);
```

02. 데이터 타입의 종류

- 숫자형 (Numeric Types)
- 화폐형 (Monetary Types)
- 문자형 (Character Types)
- 날짜 및 시간 (Date & Time)
- 불리언형 (Boolean Types)
- 배열형 (Array Types)
- 제이슨형 (Json Types)

03. 데이터 타입의 종류

데이터 타입	세부 유형	비고
숫자형 (Numeric Types)	INTEGER	SMALLINT 2 bytes & INT 4 bytes
	NUMERIC	소수점 자리 표시 가능 가변적
	FLOAT	부동 소수점 사용 4 bytes, 8 bytes
	SERIAL	INTEGER 기본 값으로 1씩 추가되며 값이 자동 생성

출처: <https://www.postgresql.org/docs/9.5/datatype-numeric.html>

03. 데이터 타입의 종류

데이터 타입	세부 유형	비고
문자형 (Character Types)	VARCHAR(N)	N 길이 내에서의 길이 저장 (Character Varying 약자)
	CHAR(N)	N길이 저장 공백도 처리
	TEXT	길이에 상관없이 모든 문자열을 저장

출처: <https://www.postgresql.org/docs/9.5/datatype-character.html>

03. 데이터 타입의 종류

데이터 타입	세부 유형	비고
날짜 및 시간 (Character Types)	TIMESTAMP	날짜와 시간 모두
	TIMESTAMP	날짜와 시간 모두 시간대 포함
	INTERVAL	시간 간격
	DATE	날짜
	TIME	시간
	TIME	시간, 시간대 첨부

출처: <https://www.postgresql.org/docs/9.5/datatype-datetime.html>

03. 데이터 타입의 종류

데이터 타입	세부 유형	비고
불리언형 (Boolean Types)	TRUE	True, yes, on, 1 모두 참
	FALSE	False, no, off, 0 모두 거짓
	NULL	알 수 없는 정보 또는 일부 불확실

출처: <https://www.postgresql.org/docs/9.5/datatype-boolean.html>

04. TABLE 삭제 쿼리문

Table 생성 쿼리문

```
DROP TABLE book_store;
```

05. 실습

제품정보 테이블 & 주문 테이블을 직접 생성합니다.

제품정보	주문정보
- 제품일련번호(10)	- 주문번호(6)
- 제품명(40)	- 고객아이디(8)
- 생산일자	- 주문상품(40)
- 제품가격	- 수량(1000)
- 공장번호(7)	- 상품가격
	- 주문일시

CAST 연산자

with



01. CAST 연산자 예시

CAST 연산자 예시

- ① **SELECT CAST ('100' AS INTEGER);**
- ② **SELECT '100':: INTEGER;**

01. CAST 연산자 예시

Syntax: CAST (expression AS target_type);

01. CAST 연산자 예시

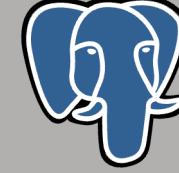
CAST 연산 Sample Code

```
-- phone_book
CREATE TABLE phone_book (
    id INTEGER,
    name VARCHAR,
    phone NUMERIC(11)
);

-- 데이터 insert
-- 원하는 데이터를 추가하도록 합니다.
INSERT INTO phone_book VALUES(1, '정지훈', 01011111234);

-- CAST
SELECT name, CAST (phone AS VARCHAR) FROM phone_book;
```

컬럼 값 제한하기

with 
PostgreSQL

01. 5가지 제약조건

- NOT NULL 제약 조건
- UNIQUE 제약 조건
- PRIMARY KEY
- 외래 키(Foreign KEY)
- CHECK 제약 조건

02. NOT NULL

NOT NULL 제약조건(Constraint)은 빈 값을 허용하지 않는 조건

```
DROP TABLE IF EXISTS contact;
```

```
CREATE TABLE contact(
    id numeric(3) NOT NULL,
    name varchar(15) NOT NULL,
    tel INTEGER[] NOT NULL,
    email VARCHAR
);
```

03. UNIQUE

UNIQUE 제약 조건은 유일한 값을 가져야 하는 것.

```
DROP TABLE IF EXISTS contact;
```

```
CREATE TABLE contact(
    id numeric(3) UNIQUE NOT NULL,
    name varchar(15) NOT NULL,
    tel INTEGER[] NOT NULL,
    email VARCHAR
    UNIQUE
);
```

03. UNIQUE

여러 개가 존재하는 경우 UNIQUE(col_1, col_2, ...)

```
DROP TABLE IF EXISTS contact;
```

```
CREATE TABLE contact(
    id numeric(3) NOT NULL,
    name varchar(15) NOT NULL,
    tel INTEGER[] NOT NULL,
    email VARCHAR,
    UNIQUE (id, tel, email)
);
```

04. PRIMARY KEY

주식별자, 빈값 허용 안됨, PK

```
DROP TABLE IF EXISTS contact;
```

```
CREATE TABLE contact(
    id SERIAL NOT NULL PRIMARY KEY,
    name varchar(15) NOT NULL,
    tel INTEGER[] NOT NULL,
    email VARCHAR
);
```

04. PRIMARY KEY

외래 키 제약조건에 따라, PK가 여러 개일 수 있음

```
CREATE TABLE book(
    book_id SERIAL NOT NULL,
    name varchar(15) NOT NULL,
    admin_no SERIAL NOT NULL REFERENCES contact(id),
    email VARCHAR,
    PRIMARY KEY (book_id, admin_no)
);
```

05. 외래키

부모 테이블 – 자식 테이블 (예)

과목 테이블	
과목_id	NUMERIC(5)
과목_name	VARCHAR(20)

〈부모 테이블〉

교사 정보 테이블	
교사_id	NUMERIC(5)
교사_name	VARCHAR(20)
과목_id	NUMERIC(5)
임용일자	

〈자식 테이블〉

- 3가지 규칙

- ☞ 부모 테이블이 먼저 생성

- ☞ 같은 데이터 타입

- ☞ 참조되는 컬럼은 모두 PK 또는 UNIQUE 제약 조건 형식

05. 외래키

실습 파일 step01_basic/session_02.ipynb 참조

05. 외래키가 포함된 데이터 삭제

```
DELETE FROM subject where sub_id = 0003;
```

--

ERROR: 오류: "subject" 테이블의 자료 갱신, 삭제 작업이
"teacher_sub_code_sub_name_fkey" 참조키(foreign key) 제약 조건 -
"teacher" 테이블 - 을 위반했습니다 DETAIL: (sub_id, sub_name)=(3, IT) 키가
"teacher" 테이블에서 여전히 참조됩니다.

ERROR: update or delete on table "subject" violates foreign key constraint
"teacher_sub_id_fkey" on table "teacher" DETAIL: Key (sub_id)=(2) is still r
eferenced from table "teacher". SQL state: 23503

메시지: 외래 키에 의해 참조된 컬럼은 지울 수 없다는 기본적인 ON DELETE NO ACTION

05. 외래키가 포함된 데이터 삭제

ON DELETE의 다섯 가지 유형

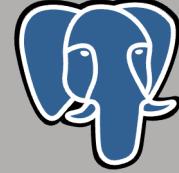
지우면 안되는 경우	ON DELETE NO ACTION
	ON DELETE RESTRICT
지워야 하는 경우	ON DELETE SET NULL
	ON DELETE CASCADE
	ON DELETE SET DEFAULT

06. CHECK

CHECK 뒤에 나오는 식이 BOOLEAN Type의 TRUE 만족

```
ERROR: new row for relation "order_info" violates check constraint "order_info_order_qty_check"
DETAIL: Failing row contains (2, 홍선생, $4,000.00, -2).
SQL state: 23514
```

서브쿼리

with  PostgreSQL

01. 데이터 삽입

step01_basic/countries.sql 파일과 그 외 데이터가 있는지 확인

```
$ psql -U postgres
psql (13.3)
Type "help" for help.
```

```
postgres=# \cd step01_basic
postgres=# \i countries.sql
```

02. 서브쿼리의 정의

쿼리문 안에 쿼리문이 반복되는 구조

02. 서브쿼리의 종류 및 연산자

단일 행
서브쿼리

다중 행
서브쿼리

- 비교연산자(=, <, <=, >=, >>)
- IN, ANY, ALL, EXISTS 사용

더보기: <https://www.postgresql.org/docs/8.1/functions-subquery.html>

Subquery with WHERE and SELECT

사용 데이터: populations, cities, countries

03. Question 1.

조건 1. 2015년 전체 국가의 평균 수명을 계산하십시오.

	avg	double precision
1	71.6763415481105	

03. Question 2.

조건 1. 모든 데이터를 조회한다.

조건 2. 2015년 평균 기대수명의 1.15배보다 높도록 설정한다.

	pop_id [PK] integer	country_code character varying	year integer	fertility_rate real	life_expectancy real	size real
1	21	AUS	2015	1.833	82.45122	2.3789752e+07
2	376	CHE	2015	1.54	83.19756	8.28143e+06
3	356	ESP	2015	1.32	83.380486	4.6443992e+07
4	134	FRA	2015	2.01	82.67073	6.6538392e+07
5	170	HKG	2015	1.195	84.278046	7.3057e+06

03. Question 3.

조건 1. 서브 쿼리의 국가 테이블에 있는 capital 필드를 사용합니다.

조건 2. cities 테이블에서 name, country_code, urbanaarea_pop 필드를 조회한다.

	name [PK] character varying	country_code character varying	urbanarea_pop real
1	Beijing	CHN	2.1516e+07
2	Dhaka	BGD	1.4543124e+07
3	Tokyo	JPN	1.3513734e+07
4	Moscow	RUS	1.2197596e+07
5	Cairo	EGY	1.023035e+07

03. Question 4.

조건 1. 첫번째 주석을 풀고 실행합니다.

조건 2. GROUP BY 코드를 변환하여 SELECT 내부의 하위 쿼리를 사용하도록 한다.
즉, 첫 번째 쿼리에서
GROUP BY 코드를 사용하여 주어진 결과와 일치하는 코드를 작성한다.

조건 3. 다시 city_num 내림차순으로 결과를 정렬한 후, code 오름차순으로 정렬한다.

03. Question 4.

	country character varying	cities_num bigint
1	China	36
2	India	18
3	Japan	11
4	Brazil	10
5	Pakistan	9
6	United States	9
7	Indonesia	7
8	Russian Federation	7
9	South Korea	7

Subquery with FROM

사용 데이터: languages, countries

04. Question 1

- 조건1. 각 국가 코드에 대해 SELECT, FROM 및 GROUP BY를 사용하여 언어 테이블에 나열되는 언어 개수 수를 구한다.
- 조건2. 집계된 필드는 lang_num으로 표시한다. (Alias)

	code character varying	lang_num bigint
1	PRY	2
2	NRU	3
3	MDG	3
4	ASM	5
5	TZA	4

04. Question 2.

- 조건1. 위 쿼리를 FROM 절에 서브쿼리로 사용합니다.
- 조건2. countries 데이터에서 local_name 필드를 조회합니다.
- 조건3. 서브쿼리에서 lang_num을 조건2와 같이 조회합니다.
- 조건4. countries code와 서브쿼리로 사용된 code와 반드시 일치해야 합니다.
- 조건5. lang_num 내림차순으로 정렬합니다.

04. Question 2.

	local_name	lang_num
1	Zambia	19
2	Yeltyop'iya	16
3	Zimbabwe	16
4	Bharat/India	14
5	Nepal	14

04. Question 3.

- 조건1. 다음 쿼리를 실행합니다.

```
SELECT name, continent, inflation_rate  
FROM countries  
    INNER JOIN economies  
        USING (code)  
WHERE year = 2015  
LIMIT 5;
```

	name character varying	continent character varying	inflation_rate real
1	Afghanistan	Asia	-1.549
2	Angola	Africa	10.287
3	Albania	Europe	1.896
4	United Arab Emirates	Asia	4.07
5	Argentina	South America	[null]

04. Question 3.

- 조건2. 해당 쿼리를 FROM 이내의 서브쿼리를 활용한다.
- 조건3. 2015년의 continent 그룹별 inflation_rate의 최댓값 조회한다.

	continent character varying	max_inf real
1	Africa	21.858
2	Asia	39.403
3	South America	121.738
4	North America	7.524
5	Europe	48.684
6	Oceania	9.784

04. Question 3.

- 조건4. 각 대륙별 inflation_rate가 가장 높은 country 조회한다.
- 조건5. 이전 쿼리를 IN 이하절의 서브쿼리로 활용한다.
- 조건6. 주어진 SQL 중, Using(code) 대신 ON 코드로 작성한다.

	name character varying 	continent character varying 	inflation_rate real 
1	Haiti	North America	7.524
2	Malawi	Africa	21.858
3	Nauru	Oceania	9.784
4	Ukraine	Europe	48.684
5	Venezuela	South America	121.738
6	Yemen	Asia	39.403

Window Functions

with
 PostgreSQL

01. 실습 데이터 준비

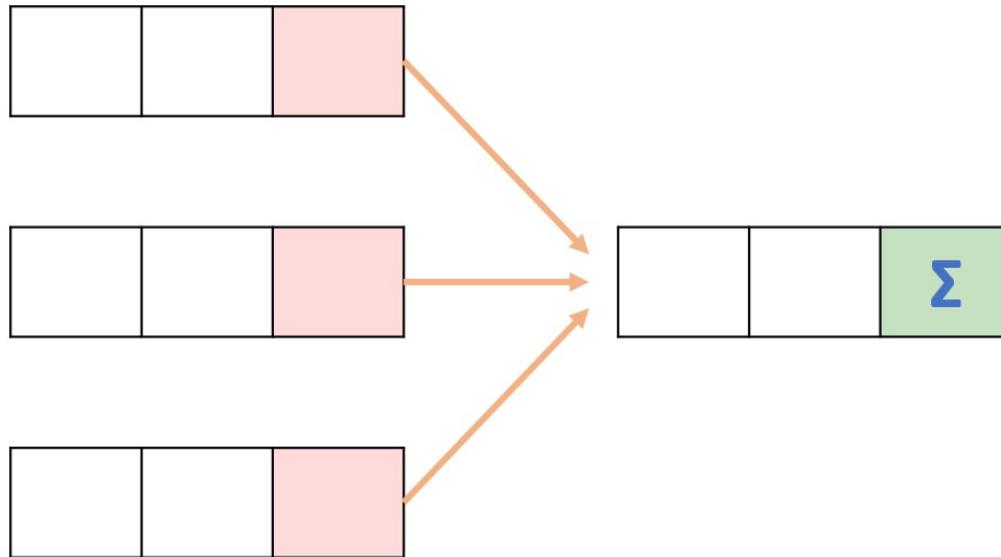
step02_intermediate 내 summer_medals.sql & data 폴더 확인

```
$ psql -U postgres
psql (13.3)
Type "help" for help.
```

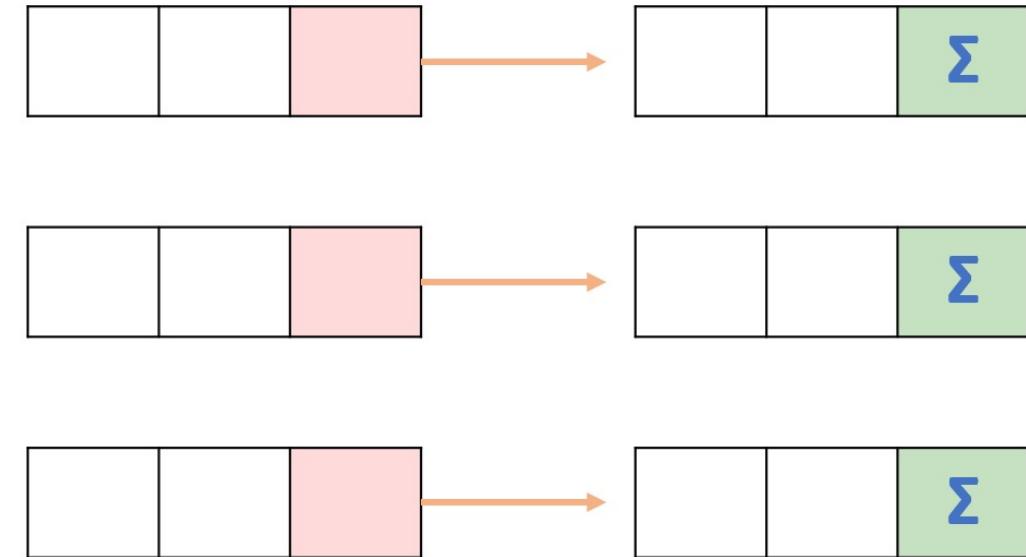
```
postgres=# \i step02_intermediate/summer_medals.sql
CREATE TABLE
ALTER TABLE
COPY 31165
postgres=#
```

02. 기본개념

Aggregation



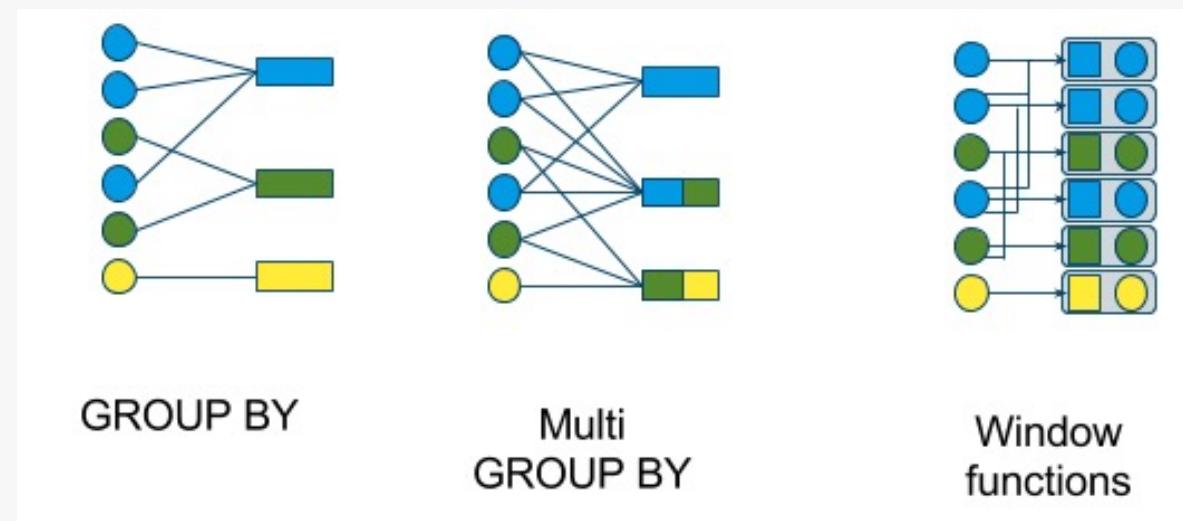
Window Function



출처: <https://towardsdatascience.com/sql-window-function-demonstrated-with-real-interview-questions-from-leetcode-e83e28edaabc>

02. 기본개념

- 현재 행과 관련된 행 집합에서 쿼리 작업 수행
- GROUP BY 집계 함수와 유사하지만, 결괏값에서 모든 행이 존재 할 수
- Ranking, 누적 계산, 이동 평균 계산 진행 가능



출처: <https://sundaskhalid.medium.com/sql-window-function-vs-group-by-b246d14223d2>

03. ROW_NUMBER()

- 각 행에 숫자를 1, 2, 3, ..., N 형태로 추가하도록 합니다.
- Row_N 기준으로 오름차순으로 진행합니다.

```
SELECT
    ROW_NUMBER() OVER() AS Row_N
    *
    ,
FROM Summer_Medals
ORDER BY Row_N ASC;
```

row_n Read-only column	year integer	city character varying (42)	sport character varying (34)	discipline character varying (34)
1	1	1896	Athens	Aquatics
2	2	1896	Athens	Aquatics
3	3	1896	Athens	Aquatics
4	4	1896	Athens	Aquatics
5	5	1896	Athens	Aquatics
6	6	1896	Athens	Aquatics
7	7	1896	Athens	Aquatics
8	8	1896	Athens	Aquatics
9	9	1896	Athens	Aquatics

04. 서브쿼리 with ROW_NUMBER()

- 이번에는 올림픽 연도를 오름차순 순번대로 작성을 하도록 합니다.
- 먼저, 연도에 Distinct를 적용한 후, 오름차순으로 정렬하는 쿼리를 작성합니다.
- 해당 쿼리를 FROM 절에 서브쿼리로 활용합니다.

```
SELECT
    Year
    , ROW_NUMBER() OVER () Row_N
FROM
    (서브쿼리) AS Years
ORDER BY Year ASC;
```

	year integer	row_n bigint
1	1896	1
2	1900	2
3	1904	3
4	1908	4
5	1912	5
6	1920	6
7	1924	7
8	1928	8
9	1932	9
10	1936	10

04. Window Function의 다양한 방법

ORDER BY

사용 데이터: Summer_Medals

04. Example

- Over()의 ORDER BY()는 현재 행과 관련된 행을 정렬합니다.

```
SELECT
    ROW_NUMBER() OVER
    (ORDER BY Year DESC, Event ASC) AS Row_N,
    Year, Event, Country
FROM summer_medals
WHERE
    Medal = 'Gold';
```

	row_n	year	event	country
1	1	2012	+ 100KG	FRA
2	2	2012	+ 67 KG	SRB
3	3	2012	+ 78KG	CUB
4	4	2012	+ 80 KG	ITA
5	5	2012	+ 91KG	GBR
6	6	2012	+105KG	IRI
7	7	2012	+75KG	CHN
8	8	2012	- 48 KG	BRA
9	9	2012	- 49 KG	CHN
10	10	2012	- 58 KG	ESP
11	11	2012	- 60 KG	RUS
12	12	2012	-56KG	PRK

04. Question 1.

- 하계 올림픽이 열린 각 연도에 번호를 할당합니다.
- 가장 최근 연도를 가진 행이 더 낮은 행 수를 갖도록 합니다.

	year	row_n
	integer	bigint
1	1896	27
2	1900	26
3	1904	25
4	1908	24
5	1912	23
6	1920	22
7	1924	21
8	1928	20
9	1932	19
10	1936	18
11	1948	17
12	1952	16

04. Question 2.

- 각 선수에 대해 자신이 획득한 메달 수를 세도록 합니다.

	athlete character varying (94)	medals bigint
1	PHELPS Michael	22
2	LATYNINA Larisa	18
3	ANDRIANOV Nikolay	15
4	MANGIAROTTI Edoardo	13
5	SHAKHLIN Boris	13

04. Question 2.

- 각 선수에 대해 자신이 획득한 메달 수를 세도록 합니다.
- 이전 쿼리에서, 각 선수들의 랭킹을 추가하도록 합니다.

	medals	athlete	row_no
1	22	PHELPS Michael	1
2	18	LATYNINA Larisa	2
3	15	ANDRIANOV Nikolay	3
4	13	MANGIAROTTI Edoardo	4
5	13	ONO Takashi	5
6	13	SHAKHLIN Boris	6
7	12	THOMPSON Jenny	7
8	12	NEMOV Alexei	8
9	12	FISCHER Birgit	9
10	12	KATO Sawao	10
11	12	NURMI Paavo	11
12	12	COUGHLIN Natalie	12

04. Question 3.

- 남자 69KG 역도 경기에서 매년 금메달리스트 조회하도록 합니다.
- 기존 쿼리에서 매년 전년도 챔피언도 같이 조회하도록 합니다.

	year integer	champion character (6)	last_champion character
1	2000	BUL	[null]
2	2004	CHN	BUL
3	2008	CHN	CHN
4	2012	CHN	CHN

04. Window Function의 다양한 방법

PARTITION BY

사용 데이터: Summer_Medals

05. Example

- PARTITION BY는 테이블을 각 칼럼의 값에 따라서 분할을 합니다.
- ROW_NUMBER는 각 파티션에 따라 새롭게 정의를 내립니다.
- LAG는 이전 행이 동일한 파티션에 있는 경우에만 행의 이전 값을 가져옵니다.

05. Example

Query

```
WITH Discus_Gold_Medal AS (
    SELECT
        Year, Event, Country AS Champion
    FROM summer_medals
    WHERE
        Year IN (2004, 2008, 2012)
        AND Gender = 'Men' AND Medal = 'Gold'
        AND Event IN ('Discus Throw', 'Triple Jump')
        AND Gender = 'Men'

    SELECT
        YEAR, Event, Champion,
        LAG(Champion) OVER
            (ORDER BY Event ASC, Year ASC) AS Last_Champion
    FROM Discus_Gold_Medal
    ORDER BY Event ASC, Year ASC;
```

Result

Read-only column	year	event	champion	last_champion
	integer	character varying (98)	character (6)	character
1	2004	Discus Throw	LTU	[null]
2	2008	Discus Throw	EST	LTU
3	2012	Discus Throw	GER	EST
4	2004	Triple Jump	SWE	GER
5	2008	Triple Jump	POR	SWE
6	2012	Triple Jump	USA	POR

05. Example

Query

```
WITH Discus_Gold_Medal AS (
    SELECT
        Year, Event, Country AS Champion
    FROM summer_medals
    WHERE
        Year IN (2004, 2008, 2012)
        AND Gender = 'Men' AND Medal = 'Gold'
        AND Event IN ('Discus Throw', 'Triple Jump')
        AND Gender = 'Men')
```

```
SELECT
    YEAR, Event, Champion,
    LAG(Champion) OVER
    (PARTITION BY Event ORDER BY Event ASC, Year ASC) AS
    Last_Champion
FROM Discus_Gold_Medal
ORDER BY Event ASC, Year ASC;
```

	year integer	event character varying (98)	champion character (6)	last_champion character
1	2004	Discus Throw	LTU	[null]
2	2008	Discus Throw	EST	LTU
3	2012	Discus Throw	GER	EST
4	2004	Triple Jump	SWE	[null]
5	2008	Triple Jump	POR	SWE
6	2012	Triple Jump	USA	POR

05. Question 1.

- 성별에 따라 각 이벤트의 이전 챔피언을 반환합니다.

	gender character varying (10) 	year integer 	champion character (6) 	last_champion character 
1	Men	2000	CZE	[null]
2	Men	2004	NOR	CZE
3	Men	2008	NOR	NOR
4	Men	2012	TTO	NOR
5	Women	2000	NOR	[null]
6	Women	2004	CUB	NOR
7	Women	2008	CZE	CUB
8	Women	2012	CZE	CZE

05. Question 2.

- 성별에 따라 각 이벤트의 이전 챔피언을 반환합니다.
- 성별과 이벤트별로 각 이벤트의 이전 챔피언을 반환합니다.

	gender character varying (10) 	year integer 	event character varying (98) 	champion character (6) 	last_champion character 
1	Men	2000	10000M	ETH	[null]
2	Men	2004	10000M	ETH	ETH
3	Men	2008	10000M	ETH	ETH
4	Men	2012	10000M	GBR	ETH
5	Women	2000	10000M	ETH	[null]
6	Women	2004	10000M	CHN	ETH
7	Women	2008	10000M	ETH	CHN
8	Women	2012	10000M	ETH	ETH
9	Men	2000	100M	USA	[null]
10	Men	2004	100M	USA	USA

06. Window Function의 다양한 방법

Relative vs Absolute

사용 데이터: Summer_Medals

06. 4가지 주요 함수

- Relative
 - ✓ LAG(column, n) LAG()는 현재 행 앞의 행에 있는 열의 값을 반환합니다.
 - ✓ Lead(column, n) 현재 행 뒤의 행에 있는 열의 값을 반환합니다.
- Absolute
 - ✓ FIRST_VALUE(column) 테이블 또는 파티션의 첫번째 값을 반환합니다.
 - ✓ LAST_VALUE(column) 테이블 또는 파티션의 마지막 값을 반환합니다.

06. Question 1.

- 각 연도마다 현재 금메달리스트와 3개 대회 이후의 메달리스트를 같이 조회합니다.

	year integer	athlete character varying (94)	future_champion character varying
1	1992	MARTEN Maritza	SADOVA Natalya
2	1996	WYLUDDA Ilke	BROWN TRAFTON Stephanie
3	2000	ZVEREVA Ellina	PERKOVIC Sandra
4	2004	SADOVA Natalya	[null]
5	2008	BROWN TRAFTON Stephanie	[null]
6	2012	PERKOVIC Sandra	[null]

06. Question 2.

- 모든 선수와 첫 번째 선수를 알파벳 순서대로 조회합니다.

	athlete character varying (94)	first_athlete character varying
1	AABYE Edgar	AABYE Edgar
2	AALTONEN Paavo Johannes	AABYE Edgar
3	AAS Thomas Valentin	AABYE Edgar
4	ABALMASAU Aliaksei	AABYE Edgar
5	ABALO Luc	AABYE Edgar

06. Question 3.

- 각 올림픽 경기가 열렸던 해외 도시를 조회합니다.
- 본 데이터에서 올림픽 경기가 열렸던 마지막 도시를 조회합니다.

	year integer	city character varying (42)	last_city character varying
18	1976	Montreal	London
19	1980	Moscow	London
20	1984	Los Angeles	London
21	1988	Seoul	London
22	1992	Barcelona	London
23	1996	Atlanta	London
24	2000	Sydney	London
25	2004	Athens	London
26	2008	Beijing	London
27	2012	London	London

07. Window Function의 다양한 방법

Ranking

사용 데이터: Summer_Medals

07. Ranking 함수들

- ROW_NUMBER()는 두 행의 값이 동일한 경우에도 항상 고유한 번호를 할당합니다.
- RANK() 동일한 값을 가진 행에 같은 번호를 할당하고 이러한 경우 다음 숫자를 건너뜁니다.
- DENSE_RANK() 값이 동일한 행에 같은 숫자를 할당하지만 다음 숫자를 건너뛰지 않습니다.

```
SELECT
    Country, COUNT(DISTINCT Year) AS Games
FROM summer_medals
Where
    Country IN ('GBR', 'DEN', 'FRA',
    'ITA', 'AUT', 'BEL',
    'NOR', 'JPN', 'KOR')
GROUP BY Country
ORDER BY GAMES DESC;
```

	country character (6)	games bigint
1	GBR	27
2	DEN	26
3	FRA	26
4	ITA	25
5	AUT	24
6	BEL	24
7	NOR	22
8	JPN	20
9	KOR	15

07. Example

```
WITH Country_Games AS (
    SELECT
        Country, COUNT(DISTINCT Year) AS Games
    FROM summer_medals
    WHERE
        Country IN ('GBR', 'DEN', 'FRA',
                    'ITA', 'AUT', 'BEL',
                    'NOR', 'JPN', 'KOR')
    GROUP BY Country
    ORDER BY GAMES DESC)

SELECT
    Country, Games,
    ROW_NUMBER()
        OVER (ORDER BY Games DESC) AS Row_N,
    RANK()
        OVER (ORDER BY Games DESC) AS Rank_N,
    DENSE_RANK()
        OVER (ORDER BY Games DESC) AS Dense_Rank_N
FROM Country_Games
ORDER BY Games DESC, Country ASC;
```

	country character (6)	games bigint	row_n bigint	rank_n bigint	dense_rank_n bigint
1	GBR	27	1	1	1
2	DEN	26	2	2	2
3	FRA	26	3	2	2
4	ITA	25	4	4	3
5	AUT	24	5	5	4
6	BEL	24	6	5	4
7	NOR	22	7	7	5
8	JPN	20	8	8	6
9	KOR	15	9	9	7

07. Question 1.

- 메달 개수에 따라 각 선수의 순위를 매깁니다.
- 숫자가 높을수록, 랭킹이 높을수록, 값이 같을 경우 숫자가 동일하다.

	athlete character varying (94)	medals bigint	rank_n bigint
1	PHELPS Michael	22	1
2	LATYNINA Larisa	18	2
3	ANDRIANOV Nikolay	15	3
4	MANGIAROTTI Edoardo	13	4
5	ONO Takashi	13	4
6	SHAKHLIN Boris	13	4
7	THOMPSON Jenny	12	7
8	NEMOV Alexei	12	7
9	FISCHER Birgit	12	7
10	KATO Sawao	12	7

07. Question 2.

- 일본과 한국 각 나라의 운동선수들을 그들이 획득한 메달의 개수로 순위를 매긴다.

	country character (6) 	athlete character varying (94) 	rank_n bigint 
1	JPN	KITAJIMA Kosuke	1
2	JPN	UCHIMURA Kohei	2
3	JPN	TAKEDA Miho	3
4	JPN	TACHIBANA Miya	3
5	JPN	IRIE Ryosuke	4
6	JPN	TOMITA Hiroyuki	4
7	JPN	SUZUKI Satomi	4
8	JPN	TANI Ryoko	4
9	JPN	KASHIMA Takehiro	4
10	JPN	YOSHIDA Saori	4

44	KOR	JIN Jongoh	1
45	KOR	PARK Taehwan	2
46	KOR	PARK Sung-Hyun	2
47	KOR	LEE Sung Jin	3
48	KOR	YUN Mi-Jin	3
49	KOR	RYU Seungmin	3
50	KOR	HUH Soonyoung	4
51	KOR	HWANG Kyung Sun	4
52	KOR	KIM Dong Moon	4
53	KOR	OH Jin Hyek	4

08. Window Function의 다양한 방법

Paging

사용 데이터: Summer_Medals

08. Paging이란

- 동일한 청크로 데이터 분할
- 대부분의 API가 전송되는 데이터를 줄이기 위해 "페이지"로 데이터 반환
- 사분위수 또는 삼분위수(상위 중간 33%, 하위 3분위)로 데이터를 구분하여 성과 판단
- NTILE()는 데이터를 거의 동일한 페이지로 분할합니다.

08. Example

```
WITH Country_Medals AS (
    SELECT
        Country, COUNT(*) AS Medals
    FROM summer_medals
    GROUP BY Country),

    Thirds AS (
    SELECT
        Country, Medals,
        NTILE(3) OVER (ORDER BY Medals DESC) AS Third
    FROM Country_Medals)

SELECT
    Third,
    ROUND(AVG(Medals), 2) AS Avg_Medals
FROM Thirds
GROUP BY Third
ORDER BY Third ASC;
```

	third integer	avg_medals numeric
1	1	598.74
2	2	22.98
3	3	2.08

08. Question 1.

- 개별 이벤트를 정확히 111개 그룹으로 나누고 이벤트별로 사전순으로 정렬합니다.

	event character varying (98)	page integer
1	- 47.63KG (Flyweight)	1
2	- 47.6KG (Light-Flyweight)	1
3	- 48 KG	1
4	- 48KG	1
5	- 48KG (Extra-Lightweight)	1
6	- 48KG (Light-Flyweight)	1
7	- 49 KG	2
8	- 50.8KG (Flyweight)	2
9	- 51KG (Flyweight)	2

	event character varying (98)	page integer
658	Wg 120 KG	110
659	Wg 55 KG	110
660	Wg 60 KG	110
661	Wg 66 KG	111
662	Wg 74 KG	111
663	Wg 84 KG	111
664	Wg 96 KG	111
665	Yngling - Keelboat	111
666	York Round (100Y - 80Y - 60Y)	111

08. Question 2.

- 메달 개수에 따라 선수를 상위, 중간, 하위 3위로 나눈다.

	athlete character varying (94)	medals bigint	third integer
1	PHELPS Michael	22	1
2	LATYNINA Larisa	18	1
3	ANDRIANOV Nikolay	15	1
4	MANGIAROTTI Edoardo	13	1
5	ONO Takashi	13	1
6	SHAKHLIN Boris	13	1
7	COUGHLIN Natalie	12	1
8	FISCHER Birgit	12	1
9	KATO Sawao	12	1

08. Question 2.

- 메달 개수에 따라 선수를 상위, 중간, 하위 3위로 나눈다.
- 각 그룹의 평균을 반환합니다.

	third integer	avg_medals numeric
1	1	3.79
2	2	2.00
3	3	2.00

Pivot

with
 PostgreSQL

01. 개요

- 상단에 CREATE EXTENSION IF NOT EXISTS tablefunc; 실행합니다.

Function	Returns	Description
crosstab(text sql)	setof record	N개의 값 열에 행 이름을 더한 "피봇 테이블"을 생성합니다. 여기서 N은 호출 쿼리에 지정된 행 유형으로 결정됩니다.
crosstabN(text sql)	setof table_crosstab_N	행 이름과 N개의 값 열이 포함된 "피벗 테이블"을 생성합니다.

출처: <https://www.postgresql.org/docs/9.1/tablefunc.html>

01. Question 1.

- 한국, 중국, 러시아의 2004년, 2008년, 2012년, 조회하도록 합니다.
 - ✓ 조건 1. 금메달만 조회한다.
 - ✓ 조건 2. Sport는 Gymnastics만 조회한다.

	country character (6) 	year integer 	awards bigint 
1	CHN	2004	1
2	CHN	2008	21
3	CHN	2012	9
4	KOR	2012	1
5	RUS	2004	7
6	RUS	2008	7
7	RUS	2012	8

01. Question 2.

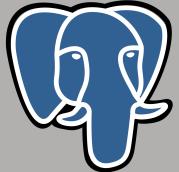
- 한국, 중국, 러시아의 2004년, 2008년, 2012년, 조회하도록 합니다.
 - ✓ 조건 1. 금메달만 조회한다.
 - ✓ 조건 2. Sport는 Gymnastics만 조회한다.
- 이번에는 왼쪽 표를 피벗테이블(오른쪽 그림)로 변환합니다.

	country character (6) 锁定	year integer 锁定	awards bigint 锁定
1	CHN	2004	1
2	CHN	2008	21
3	CHN	2012	9
4	KOR	2012	1
5	RUS	2004	7
6	RUS	2008	7
7	RUS	2012	8



	country character (6) 锁定	2004 integer 锁定	2008 integer 锁定	2012 integer 锁定
1	CHN	2	1	1
2	KOR	3	[null]	[null]
3	RUS	1	2	2

프로시저, 함수, 트리거

with
 PostgreSQL

01. 기본 개요

- 프로시저
 - ✓ 프로시저 언어는 함수와 트리거를 만드는데 사용함.
 - ✓ PostgreSQL의 프로시저 언어로는 PL/pgSQL, PL/TCL, PL/Perl, PL/Python
 - ✓ 프로시저 언어 추가 시, DB에 접속 후 다음과 같이 입력함.

CREATE LANGUAGE 언어이름

출처: <https://www.postgresql.org/docs/9.4/xplang.html>

02. PostgreSQL에서 함수 사용

- 기본 문법

```
CREATE FUNCTION function_name(parameter_1 type, parameter_2 type)
    RETURN return_type AS
$$ BEGIN
    ...
END; $$

LANGUAGE language_name;
```

02. PostgreSQL에서 함수 사용

- 곱셈 함수 만들기

```
CREATE FUNCTION multiple(a INTEGER, b INTEGER)
RETURNS INTEGER AS
$$ BEGIN
    RETURN a * b;
END; $$

LANGUAGE PLpgSQL;
```

- 등록한 함수 확인

```
postgres=# \df
```

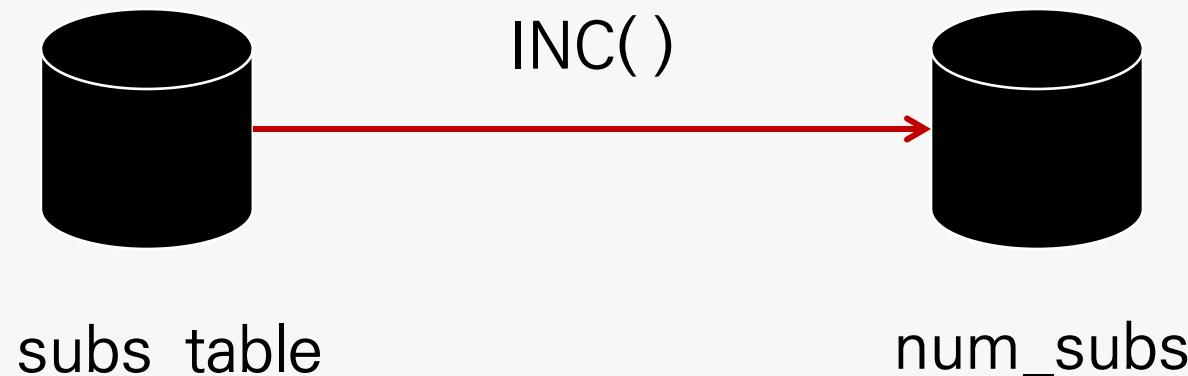
02. PostgreSQL에서 함수 사용

- 실제 적용 하기

```
postgres=# SELECT multiple(3, 4);
multiple
-----
      12
(1 row)
```

03. PostgreSQL에서 트리거 사용

- 유튜브에서 구독자가 구독 버튼 누를 시, 자동으로 구독자수가 갱신되는 시스템
- 구독자가 삽입(Insert) 되면 구독자 수 늘리는 트리거 만든다.



03. PostgreSQL에서 트리거 사용

- 두개의 테이블을 만든다.

```
DROP TABLE subs_table, num_subs;
```

```
CREATE TABLE subs_table(  
    subs_id INTEGER,  
    subs_name VARCHAR(80));
```

```
CREATE TABLE num_subs(  
    subs_num INTEGER);
```

03. PostgreSQL에서 트리거 사용

- 구독자수는 처음에 0이므로 0을 넣어준다.

```
INSERT INTO num_subs VALUES(0);
```

03. PostgreSQL에서 트리거 사용

- 구독자수를 늘려주는 함수 구현

```
CREATE FUNCTION inc()
RETURNS TRIGGER AS
$$ BEGIN
UPDATE num_subs SET subs_num = subs_num + 1;
RETURN NULL;
END; $$

LANGUAGE PLpgSQL;
```

03. PostgreSQL에서 트리거 사용

- 트리거 수행 시, 실행할 함수 INC() 수행하는 코드 입력

```
CREATE TRIGGER subs_insert AFTER INSERT ON subs_table  
FOR EACH ROW EXECUTE PROCEDURE inc();
```

03. PostgreSQL에서 트리거 사용

- subs_table에 임의의 값을 입력하도록 함

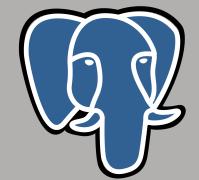
```
INSERT INTO subs_table VALUES  
(1, 'a'),  
(2, 'b'),  
(3, 'c');
```

- num_subs 테이블을 조회 한다.

```
SELECT * FROM num_subs;
```

EXPLAIN 명령어

with
PostgreSQL



01. 기본 개요

- 쿼리문 앞에 설정하여 쿼리문이 어떻게 실행될지 예상한 쿼리 계획(Query Plan) 출력
- Explain 명령어 뒤에 ANALYZE를 붙이면 쿼리 실행 후, 쿼리 계획과 실제 소요 시간 출력

```
EXPLAIN 쿼리문;  
EXPLAIN ANALYZE 쿼리문;
```

02. EXAMPLE

- EXPLAIN

```
EXPLAIN
SELECT * FROM Developers;
QUERY PLAN
-----
Seq Scan on developers (cost=0.00..13.80 rows=380 width=186)
(1 row)
```

- EXPLAIN ANALYZE

```
EXPLAIN ANALYZE SELECT * FROM Developers;
QUERY PLAN
-----
Seq Scan on developers (cost=0.00..13.80 rows=380 width=186)
(actual time=0.025..0.032 rows=9 loops=1)
Planning Time: 0.140 ms
Execution Time: 0.081 ms
(3 rows)
```

02. EXPLAIN과 쿼리 계획

실행계획	설명
Filter	원하는 정보만을 걸러내는 작업
Sequential Scan	순차적으로 정보를 읽어내는 작업
Index Scan	인덱스를 조회하여 원하는 정보를 찾고 테이블에서 해당 정보를 불러오는 작업
Index-Only Scan	인덱스를 조회하여 원하는 정보를 바로 불러오는 작업
Bitmap Heap Scan	Bitmap Index Scan을 통해 얻은 위치 정보로 원하는 정보를 테이블에서 불러오는 작업
Subquery Scan	서브쿼리의 결과로 얻은 테이블을 탐색하는 작업

출처: <https://www.postgresql.org/docs/9.5/using-explain.html>

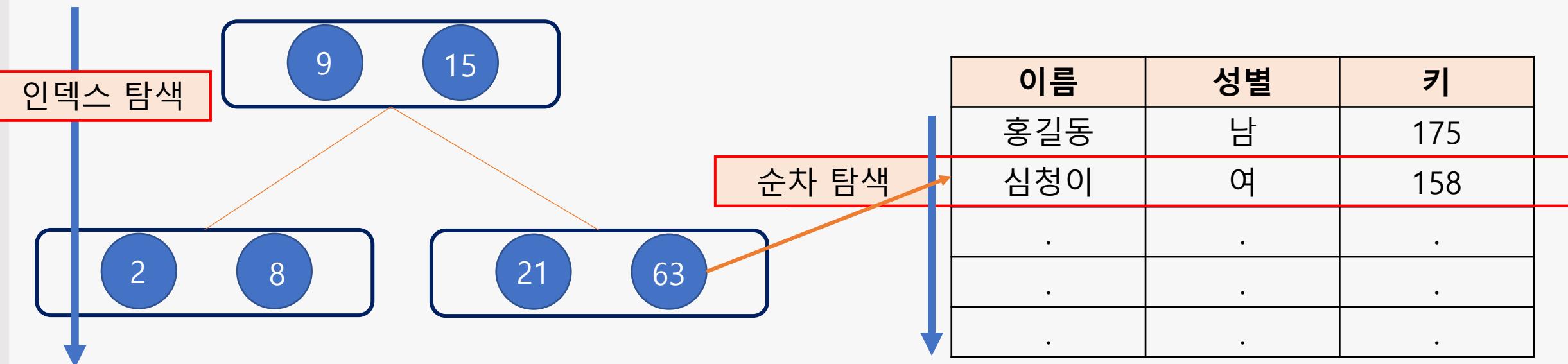
03. 순차 탐색

- 순차 탐색은 말 그대로 순차적으로 테이블 전체 조회
- 가장 직관적이면서 비용이 많이 듈다.
- 인덱스가 없을 경우 또는 불러오려는 정보의 양이 많을 경우 순차 탐색 사용

이름	성별	키
홍길동	남	175
심청이	여	158
.	.	.
.	.	.
.	.	.

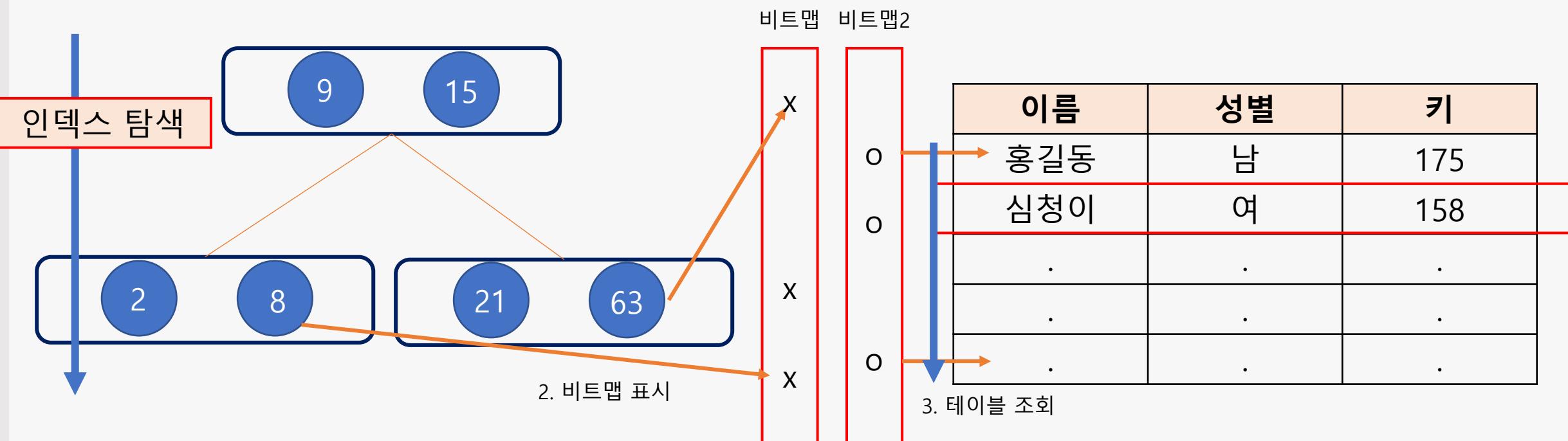
03. 인덱스 탐색

- 인덱스 탐색은 인덱스에서 찾고자 하는 값을 먼저 찾은 후 테이블에 그 값이 들어있는 행을 찾는 작업
- 인덱스 탐색은 보통 일치하는 적절한 개수의로우를 찾는 작업을 할 때 자주 볼 수 있음



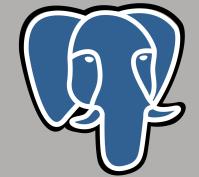
04. 비트맵 탐색

- 비트맵 탐색은 인덱스에서 조회한 내용을 잠시 비트맵에 저장된 내용 기반
- 비트맵 도식은 두 단일 컬럼 인덱스를 조회해서 2개의 비트맵을 만들고 둘 다 가리키는 로우를 테이블에서 찾아내는 과정을 나타내고 있다.



참 고 문 헌

with
PostgreSQL



- 정승호, 박성호, 한도형, 홍수민. (2021). 모두를 위한 PostgreSQL: 누구나 이해할 수 있는 오픈소스 데이터베이스 개발. 서울: 비제이퍼블릭.
- Paul, P. (2019). Beginner's Guide to PostgreSQL. DataCamp. Retrieved From <https://www.datacamp.com/community/tutorials/beginners-introduction-postgresql>
- PostgreSQL 13.3 Documentation, Retrived From <https://www.postgresql.org/docs/13/index.html>
- PostgreSQL Tutorial, <https://www.postgresqltutorial.com/postgresql-subquery/>