

▶ Chapter 03: PostgreSQL 기본 용어와 DB 관리

SQL 활용 프로그래밍

SQL Application Programming



- CHAPTER 03 PostgreSQL 기본 용어와 DB 관리
 - SECTION 01 요구사항 분석과 시스템 설계 그리고 모델링
 - 1.1 정보시스템 구축 절차 요약
 - 1.2 데이터베이스 모델링과 필수 용어
 - SECTION 02 PostgreSQL을 이용한 데이터베이스 구축
 - 2.1 데이터베이스 구축/관리 및 활용의 전반적인 절차
 - 2.2 샘플 데이터베이스 설치
 - 2.3 데이터 활용 - 출력, 입력, 삭제
 - 2.4 백업과 복원 - 환경변수 설정, cmd, pgAdmin4



CHAPTER 03 PostgreSQL 기본 용어와 DB 관리

실무에서 발생하는 상황과 비슷한 설정을 하여 응용프로그램과 연동해본다.

SECTION 01 요구사항 분석과 시스템 설계 그리고 모델링

1-1. 정보시스템 구축 절차 요약

- 분석, 설계, 구현, 시험, 유지보수의 5가지 단계
- 분석
 - 구현하고자 하는 프로젝트의 가장 첫 번째 단계
 - 시스템 분석 또는 요구사항 분석이라고 불림
 - 요구사항 분석은 현재 우리가 '무엇을(What)' 할 것인지 결정
 - 사용자의 인터뷰와 업무 조사 등을 수행
 - 프로젝트의 첫 단추를 끼우는 중요한 단계
- 설계
 - 시스템 설계 또는 프로그램 설계라고 불림
 - 구축하고자 하는 시스템을 '어떻게(How)' 할 것인지 결정
 - 대부분의 프로젝트에서 분석과 설계의 과정이 전체 공정의 50% 이상 차지

SECTION 01 요구사항 분석과 시스템 설계 그리고 모델링

1-1. 정보시스템 구축 절차 요약

- 구현

- 설계 단계의 결과물들을 기반으로 명령문을 실행하며 데이터베이스를 실제로 생성
- 데이터베이스 스키마를 만들고 데이터 입력

- 시험

- 데이터베이스 구현이 요구사항에 따라 잘 개발되었는지, 요구사항대로 잘 작동하는지 확인

- 유지보수

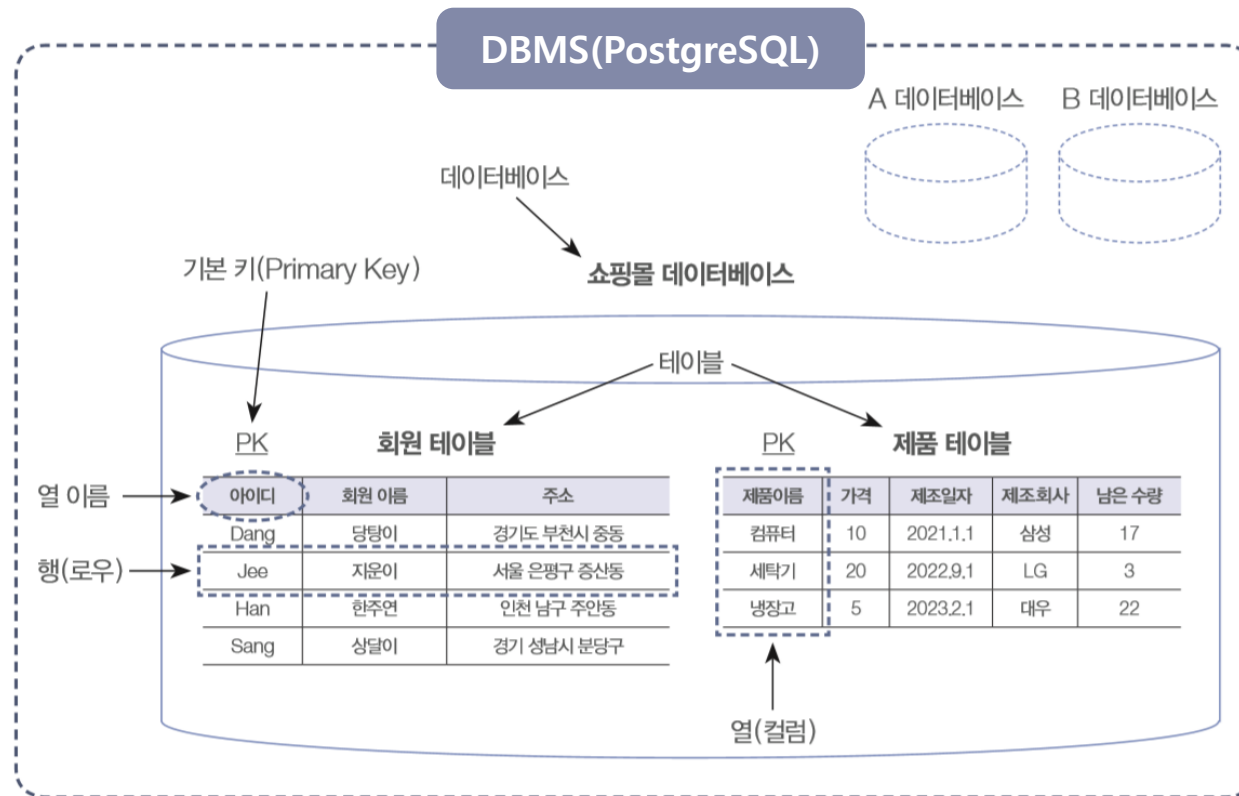
- 품질 관리 및 모니터링
- 보안 정책 수립

SECTION 01 요구사항 분석과 시스템 설계 그리고 모델링

1-2. 데이터베이스 모델링과 필수 용어

◦ 데이터베이스 모델링

- 현실세계에서 사용되는 데이터를 PostgreSQL에 어떻게 옮겨 놓을 것인지를 결정하는 과정
- 저장할 정보는 **테이블(Table)**이라는 형식에 맞춰 저장
- Ex) 쇼핑몰 데이터 베이스



[그림 3-1] 테이블의 구조와 관련 용어

SECTION 01 요구사항 분석과 시스템 설계 그리고 모델링

1-2. 데이터베이스 모델링과 필수 용어

- 데이터
 - 정보는 있으나 아직 체계화 되지 못한 상태
 - Ex) 회원 테이블에서 당당이, 컴퓨터, 2023.2.1과 같이 하나하나의 단편적인 정보
- 테이블
 - 회원이나 제품의 데이터를 입력하기 위해 표 형태로 표현한 것
 - Ex) 회원 정보 테이블, 제품 정보 테이블
- 데이터베이스(DB)
 - 테이블이 저장되는 저장소
- DBMS (DataBase Management System)
 - 데이터베이스를 관리하는 시스템 또는 소프트웨어
 - 앞서 설치한 PostgreSQL이 DBMS임

SECTION 01 요구사항 분석과 시스템 설계 그리고 모델링

1-2. 데이터베이스 모델링과 필수 용어

- 열(column=field)

- 각 테이블은 열로 구성
- 회원 테이블의 경우에는 아이디, 회원 이름, 주소 3개의 열로 구성

열(column)

| 아이디 | 회원 이름 | 주소 |
|---------|-------|-----------|
| Seonwoo | 이선우 | 서울특별시 동작구 |

행(row)

- 열 이름

- 각 열을 구분하기 위한 이름
- 각 테이블 내에서는 중복되지 않고 고유해야 함

```
CREATE TABLE 테이블 이름 (컬럼명 데이터타입 조건, ... , 컬럼명 데이터타입 조건);  
Ex)
```

```
CREATE TABLE Student (  
ID VARCHAR(50) NOT NULL UNIQUE,  
Name VARCHAR(30) NOTNULL,  
... );
```

- 데이터 형식

- 회원 테이블에서 회원 이름 열의 데이터는 숫자가 아닌 문자 형식이어야 함
- 제품 테이블에서 가격 열은 숫자 형식이어야 함. 가격에 "비쌘"과 같은 문자가 들어가면 안됨
- 테이블을 생성할 때 열 이름과 함께 지정

- 행(row=record)

- 실질적인 데이터이며 'Seonwoo/이선우/서울특별시 동작구' 가 하나의 행으로 행 데이터라고도 부름
- 회원 테이블에서 4개의 행 데이터가 있을 때, 4명의 회원이 존재함을 의미

SECTION 01 요구사항 분석과 시스템 설계 그리고 모델링

1-2. 데이터베이스 모델링과 필수 용어

- 기본 키 (Primary Key) 열
 - 기본 키(또는 주키) 열은 각 행을 구분하는 유일한 열
 - 중복되어서는 안되며, 비어 있어서도 안 됨
 - 회원 테이블에서 아이디는 기본 키가 될 수 있는 반면, 회원 이름은 동명이인이 있을 수 있으므로 기본키가 될 수 없음
- 외래 키(Foreign Key) 필드
 - 다른 테이블의 기본키를 참조하는 키
- SQL (Structured Query Language)
 - 구조화된 질의 언어
 - 사람과 DBMS가 소통하기 위한 언어

SECTION 01 요구사항 분석과 시스템 설계 그리고 모델링

1-2. 데이터베이스 모델링과 필수 용어

◦ 슈퍼 키 (Super Key)

- 각 행을 유일하기 식별할 수 있는 속성들의 집합
- ex) [학번] (O) : 학번만 가지고 학생들을 구분할 수 있으므로 슈퍼키임
[주민등록번호] (O) : 주민등록번호만 가지고도 학생들을 구분할 수 있으므로 슈퍼키임
[이름] (X) : 이름은 서로 같을 수 있기 때문에 구분하지 못하므로 슈퍼키가 아님
[학번, 이름] (O) : 학번과 이름의 조합으로 학생들을 구분할 수 있으므로 슈퍼키임

◦ 후보 키(Candidate Key)

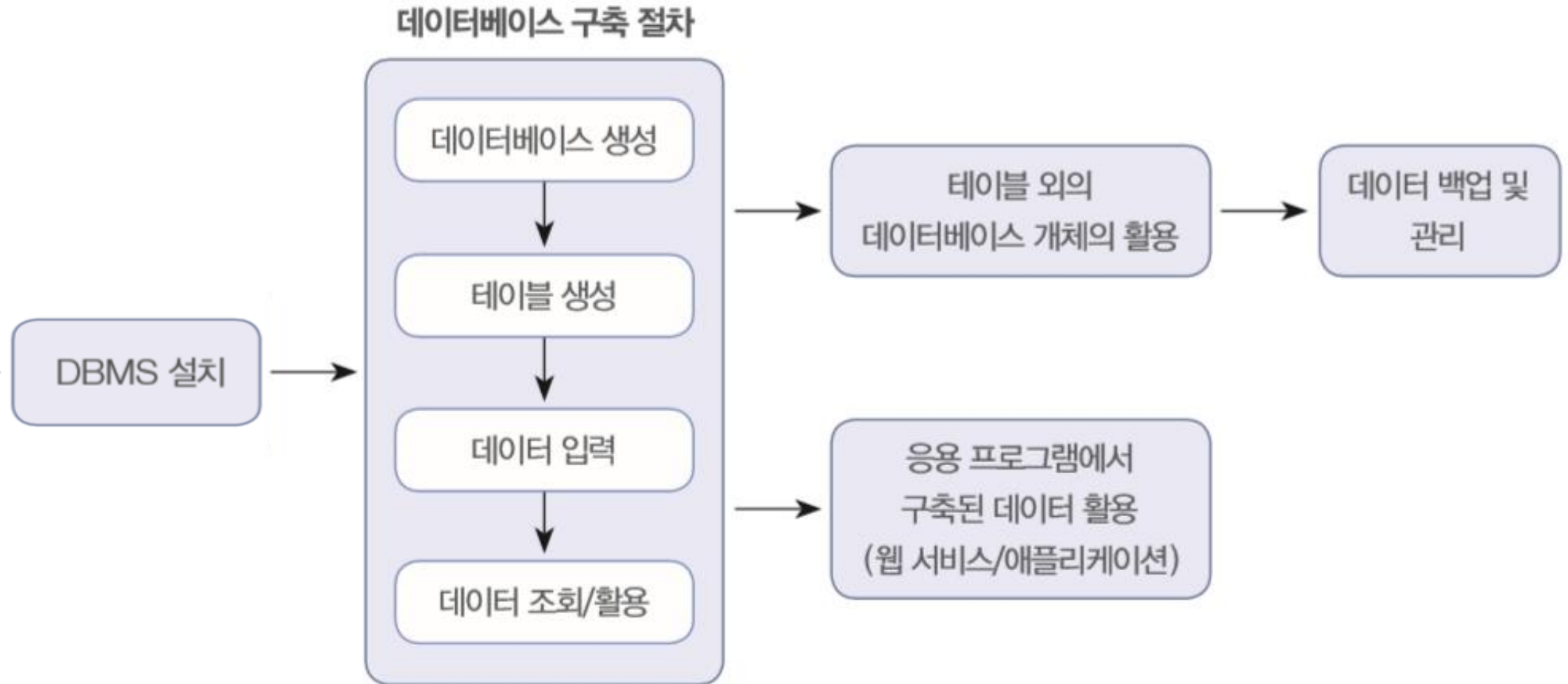
- 각 행을 유일하게 식별할 수 있는 '최소한의' 속성들의 집합
- 서로 구분할 수 있으면서 불필요한 속성들이 없어야 함
- ex) [학번] (O) : 학생들을 구분할 수 있고 속성이 하나 뿐이므로 후보키임
[주민등록번호] (O) : 학생들을 구분할 수 있고 속성이 하나 뿐이므로 후보키임
[이름] (X) : 이름은 서로 같을 수 있기 때문에 후보키가 아님
[학번, 이름] (X) : 학생들을 구분할 수는 있으나 [학번] 하나 또는 [이름] 하나만 가지고도 구분할 수 있으므로 후보키가 아님

학생 테이블

| 학번 | 이름 | 주민등록번호 | 주소 |
|----------|-----|--------------|----|
| 12201818 | 원터 | 000902-***** | 인천 |
| 12152025 | 카리나 | 950608-***** | 인천 |
| 12091834 | 닝닝 | 891101-***** | 부산 |
| 12220852 | 카리나 | 021228-***** | 서울 |

SECTION 02 PostgreSQL을 이용한 데이터베이스 구축 절차

2-1. 데이터베이스 구축/관리 및 활용의 전반적인 절차

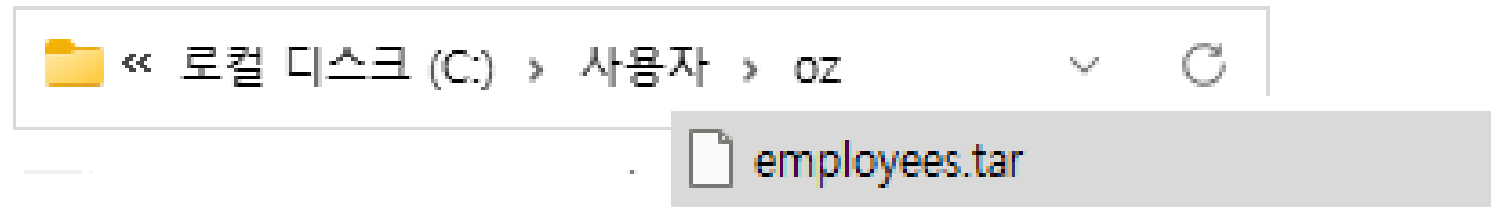


[그림 3-2] 데이터베이스 구축/관리 및 활용의 전반적인 절차

SECTION 02 PostgreSQL을 이용한 데이터베이스 구축 절차

2-2. 샘플 데이터베이스 설치

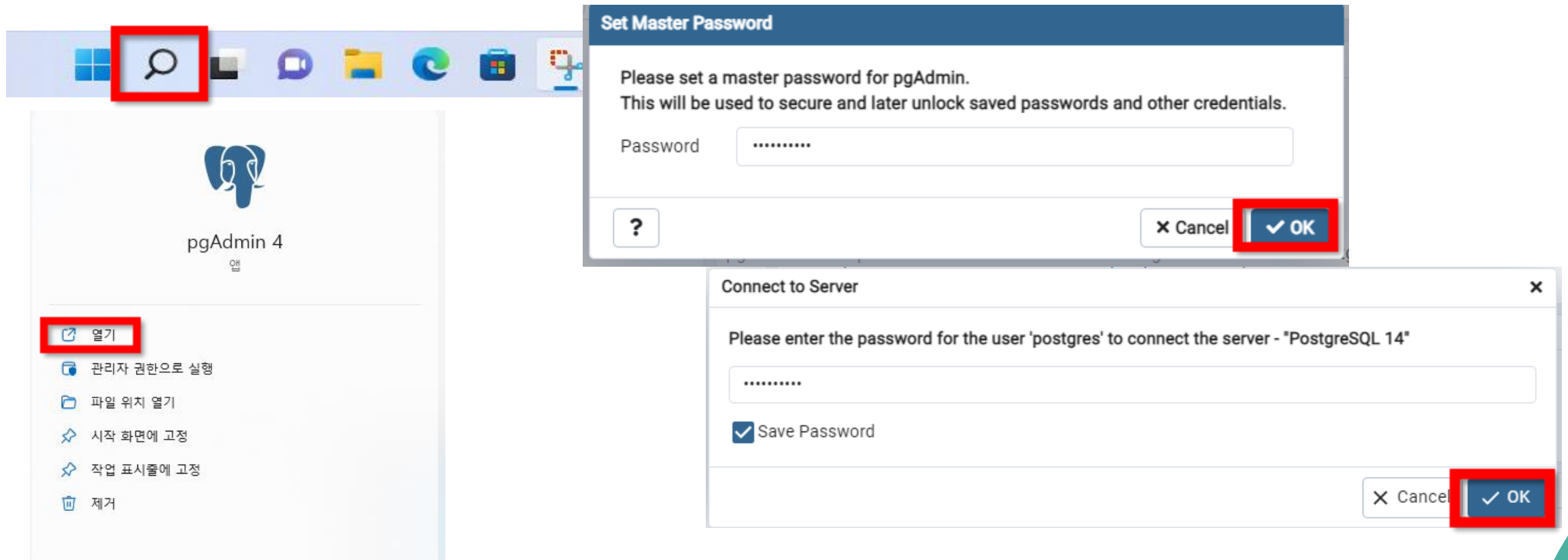
- employees 샘플데이터
 - 실습 시 대량의 데이터가 필요
 - employees.tar 파일을 다운로드하여 사용



SECTION 02 PostgreSQL을 이용한 데이터베이스 구축 절차

2-2. 샘플 데이터베이스 설치

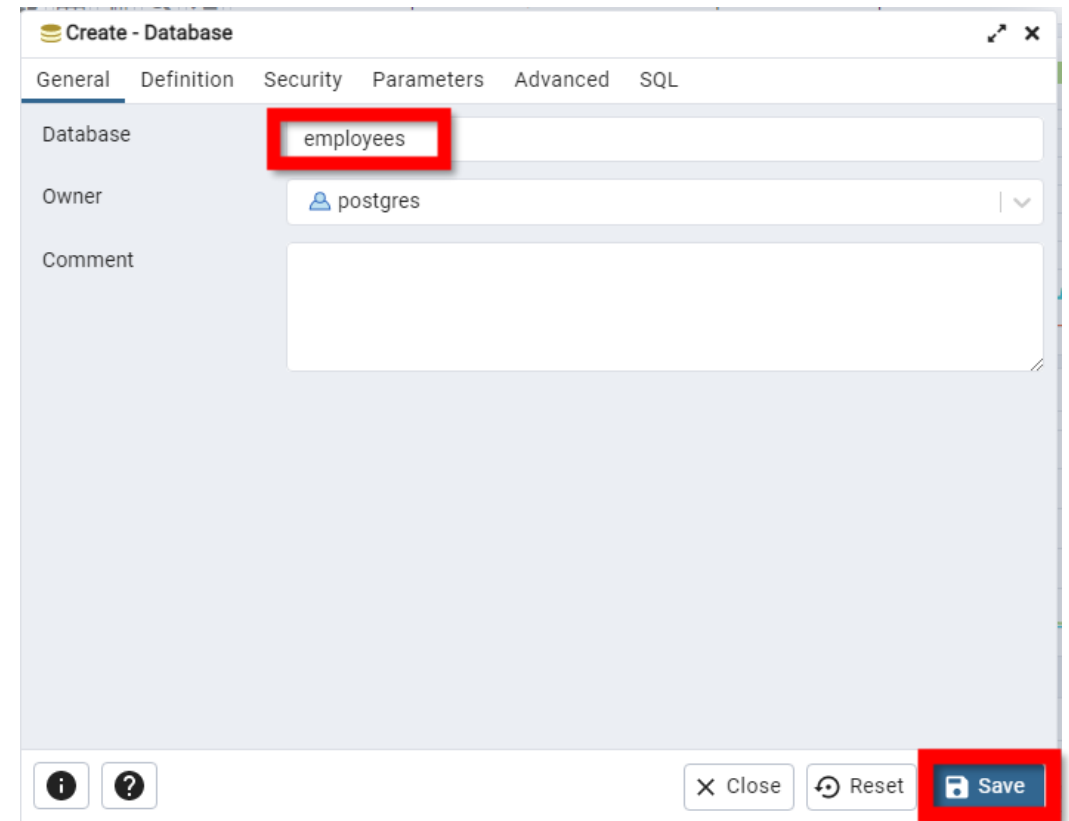
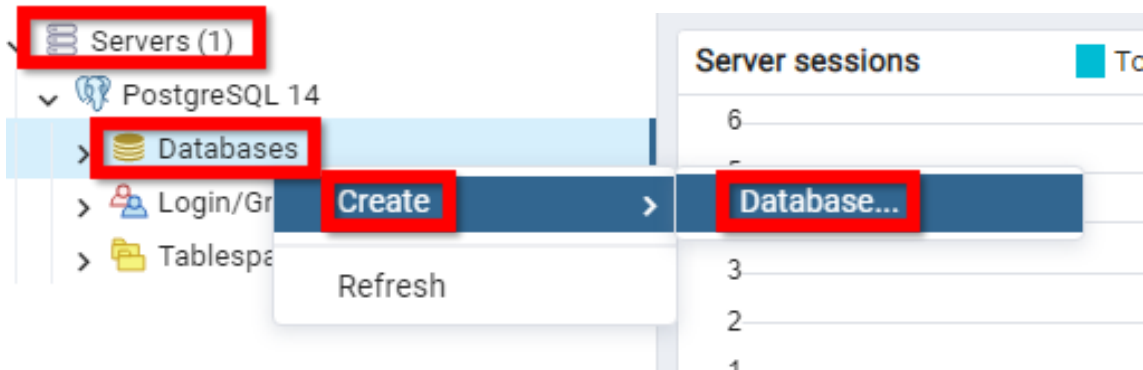
- 작업표시줄의 검색기능에 'pgAdmin' 검색 후 실행 >> 비밀번호 설정
>> PostgreSQL 설치 시 설정했던 비밀번호 입력



SECTION 02 PostgreSQL을 이용한 데이터베이스 구축 절차

2-2. 샘플 데이터베이스 설치

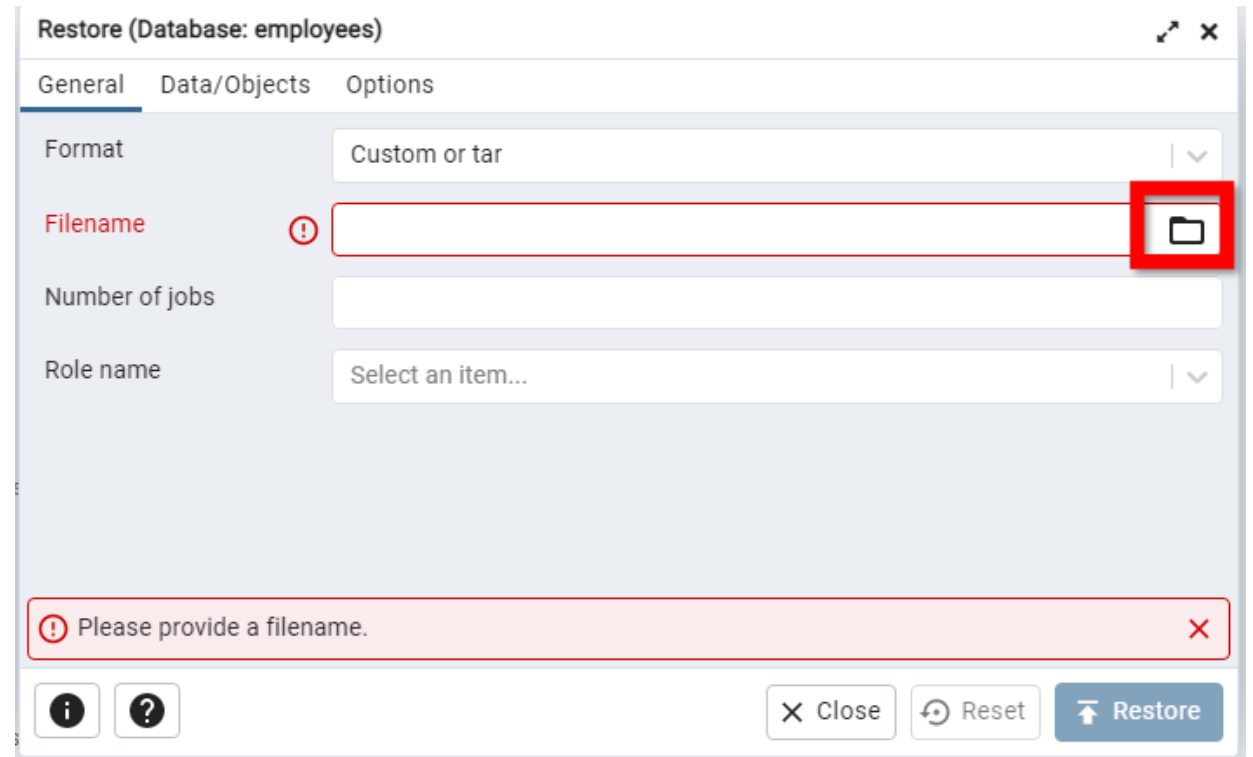
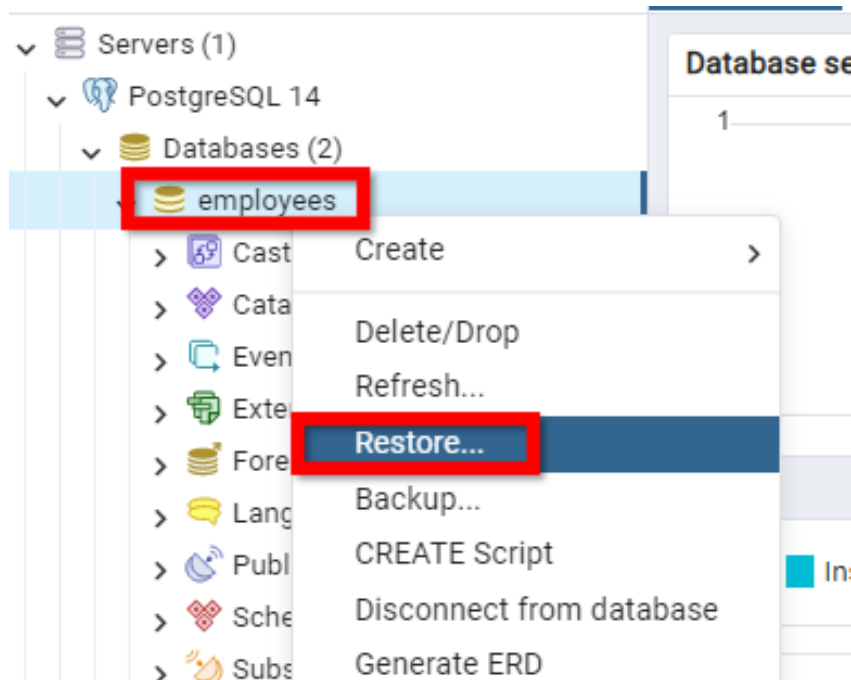
- 좌측의 내비게이터에서 Servers >> DataBases 우클릭 >> Create >> Database >> Database 이름 입력 >> Save



SECTION 02 PostgreSQL을 이용한 데이터베이스 구축 절차

2-2. 샘플 데이터베이스 설치

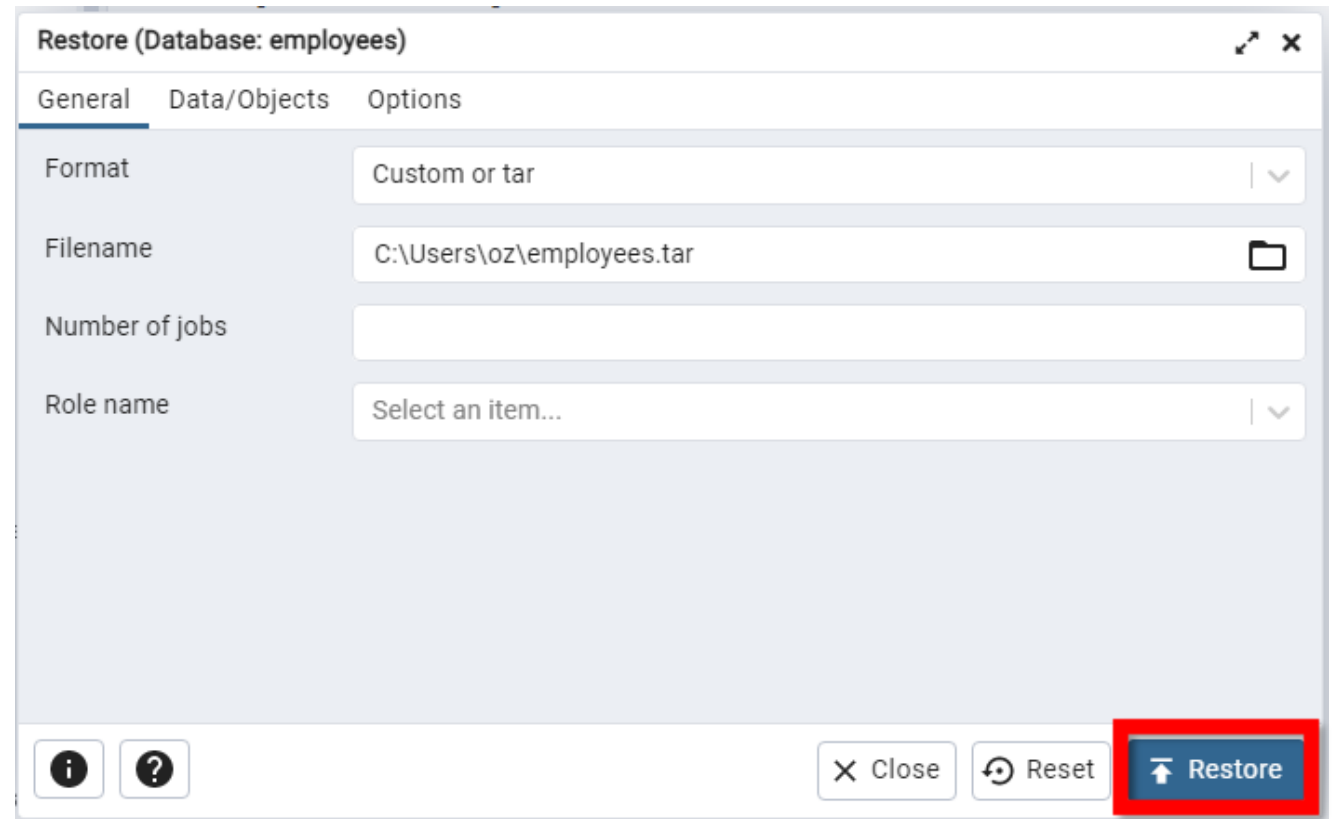
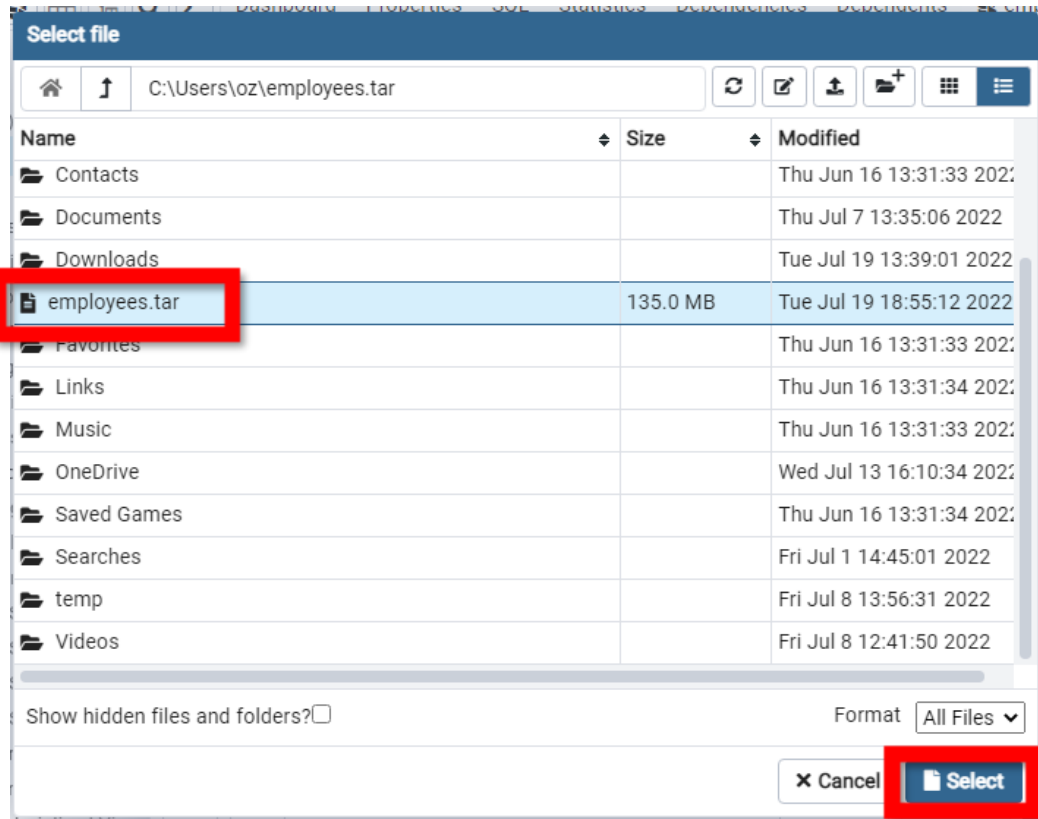
- 생성된 데이터베이스 우클릭 >> Restore >>  버튼



SECTION 02 PostgreSQL을 이용한 데이터베이스 구축 절차

2-2. 샘플 데이터베이스 설치

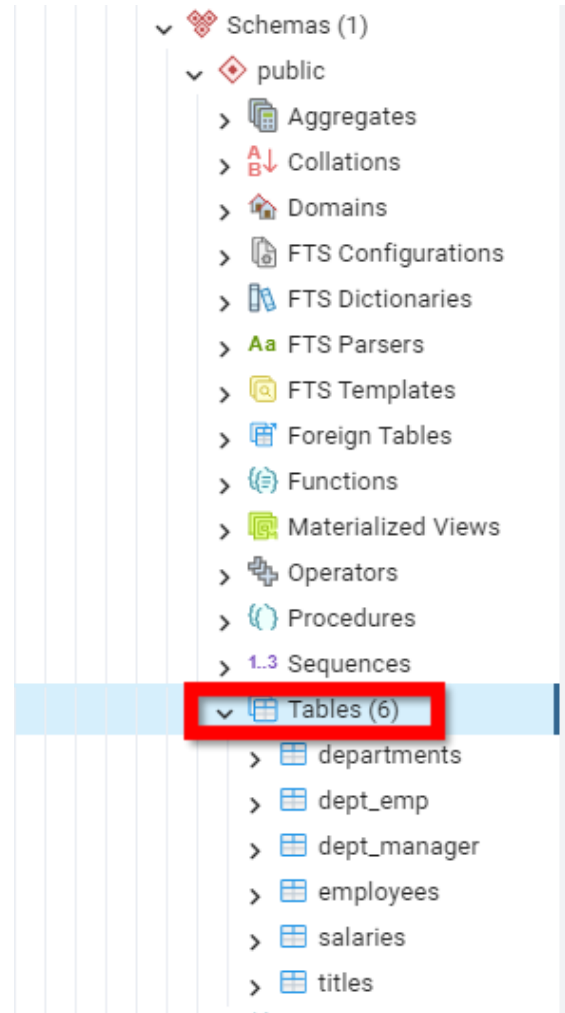
- Employees.tar파일 선택 >> Select >> Restore



SECTION 02 PostgreSQL을 이용한 데이터베이스 구축 절차

2-2. 샘플 데이터베이스 설치

- Tables를 열어보면 각 테이블들이 나열되어 보여집니다.

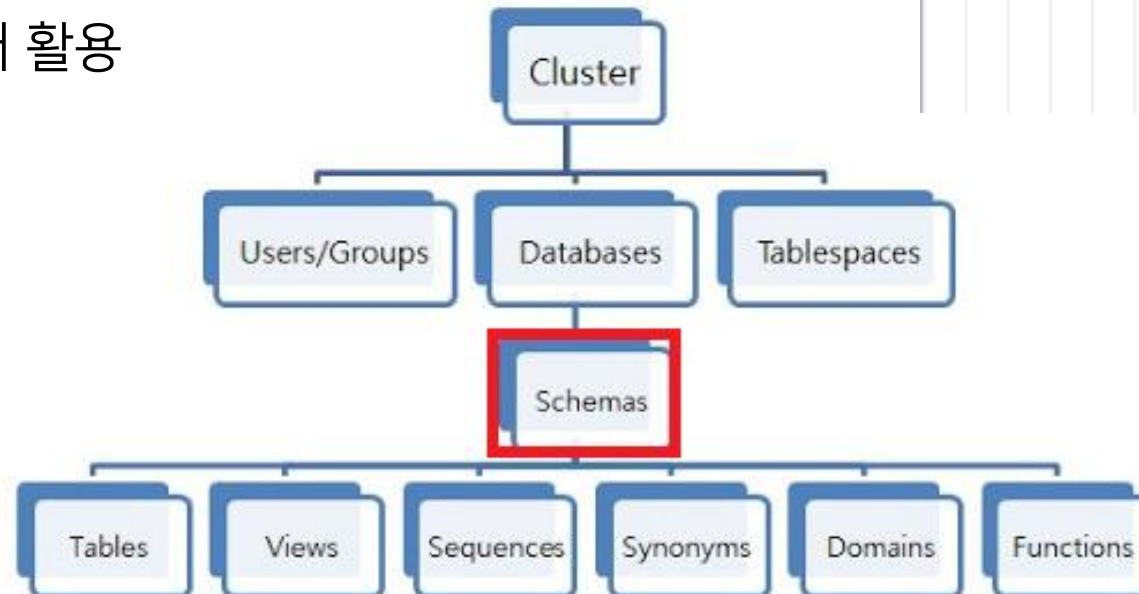
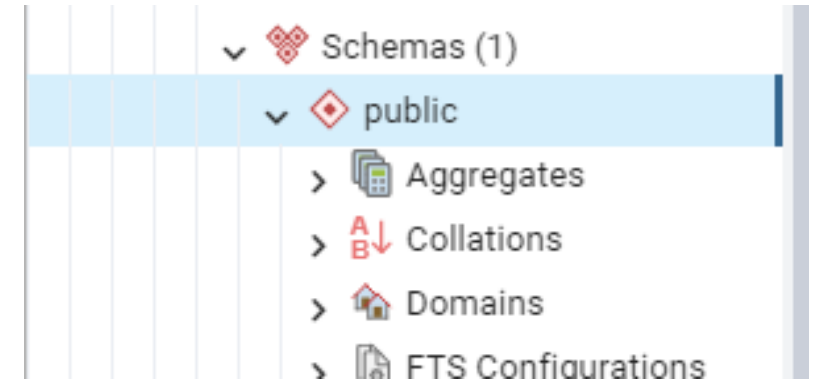


SECTION 02 PostgreSQL을 이용한 데이터베이스 구축 절차

2-3. 데이터 활용

```
nested exception is org.postgresql.util.PSQLException: ERROR: relation "TABLE_NAME" does not exist
```

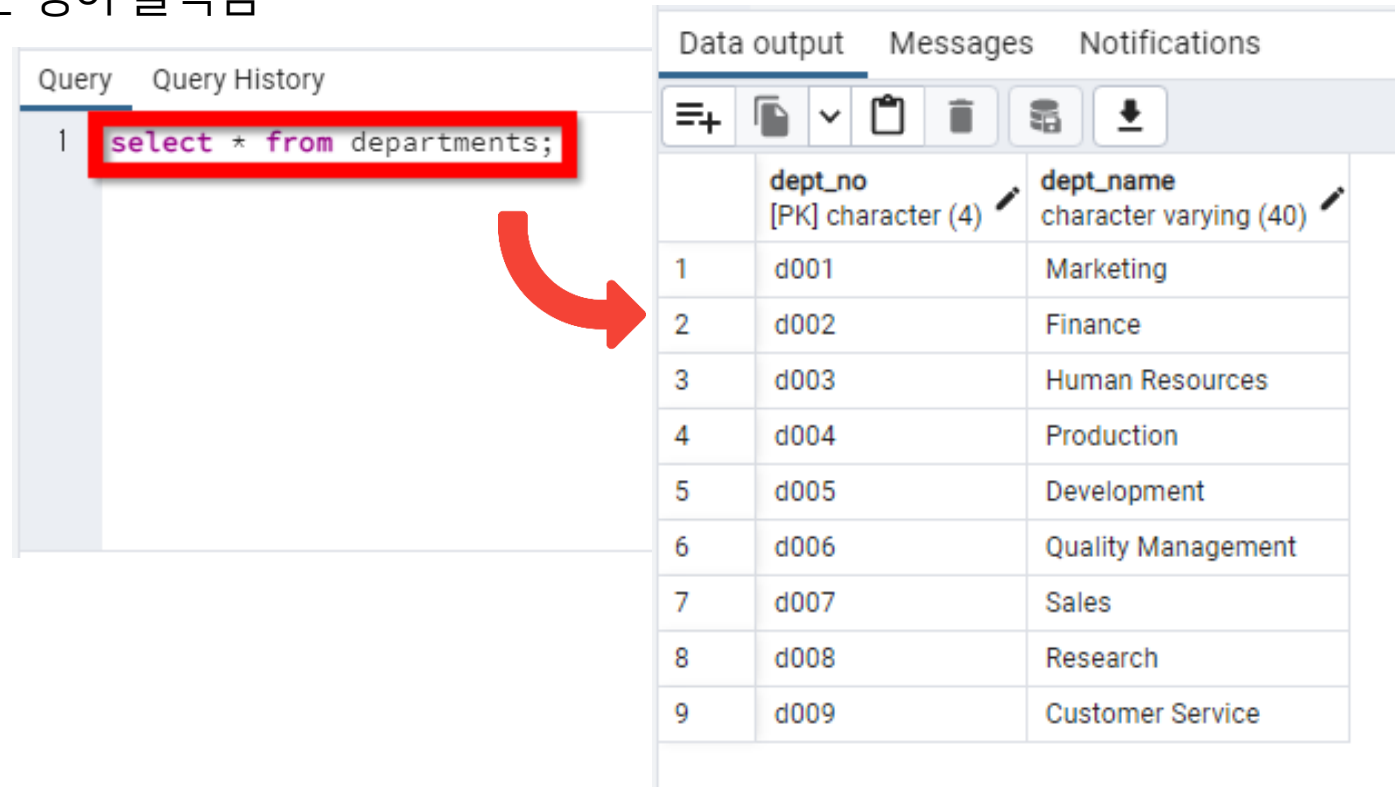
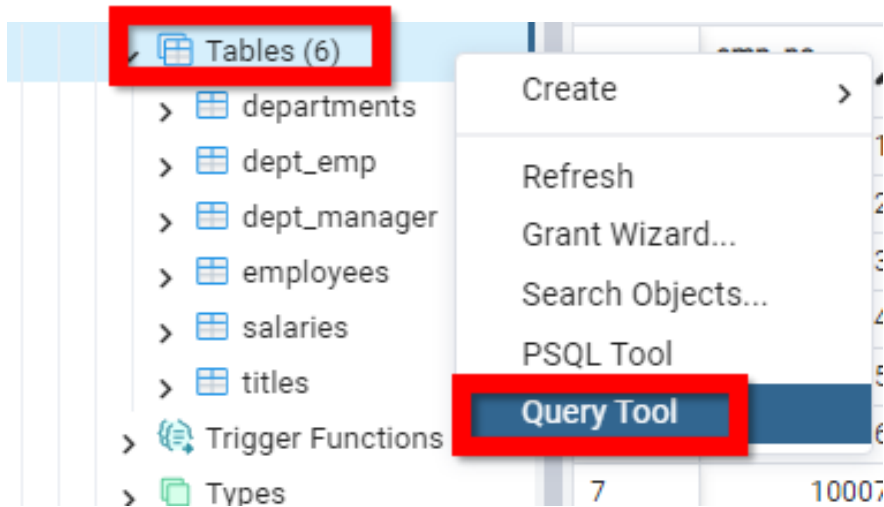
- PostgreSQL에서는 대소문자 구분을 위한 큰따옴표를 사용하지 않을 경우 오류가 발생하므로 테이블과 컬럼명은 가급적 소문자로 선언하는 것이 좋음
- 명령어는 대문자 소문자 상관없이 사용 가능
- Schema
 - 기능: PostgreSQL 내부에서 Database를 논리적으로 구분하여 관리
- 주로 select 문 사용해 데이터 활용



SECTION 02 PostgreSQL을 이용한 데이터베이스 구축 절차

2-3. 데이터 활용 - 출력


- departments 테이블의 데이터 출력
 - Tables 우클릭 >> Query Tool >> 쿼리 창에 `select * from departments;` 입력 >> F5 버튼
 - 결과 창에 departments 테이블의 모든 행이 출력됨

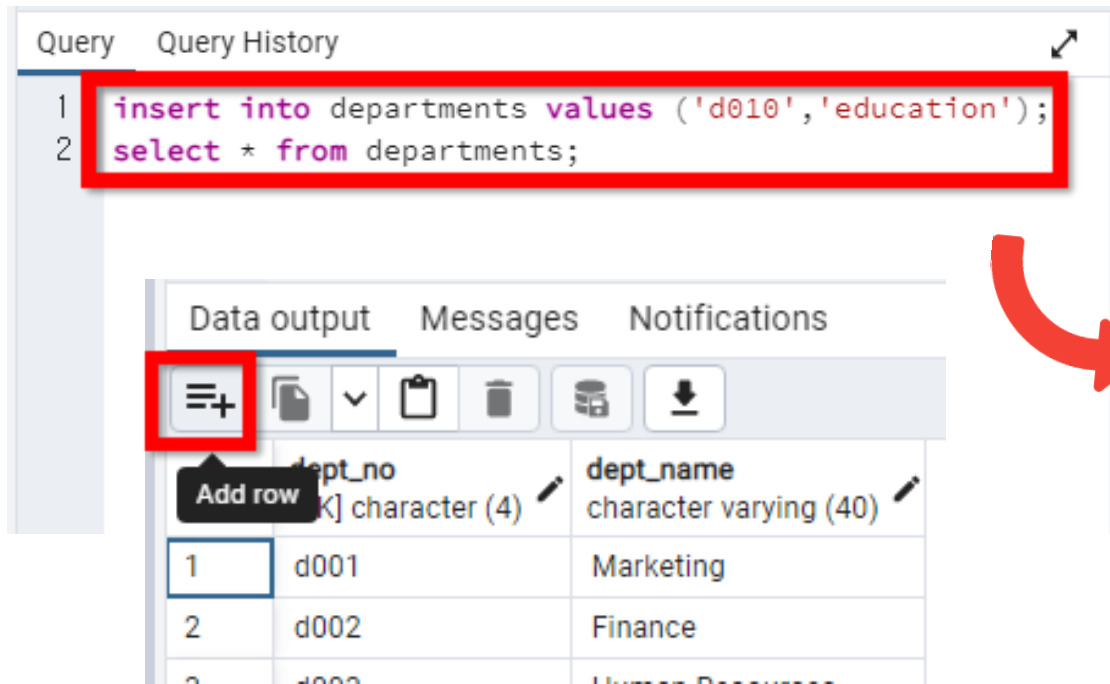


SECTION 02 PostgreSQL을 이용한 데이터베이스 구축 절차

2-3. 데이터 활용 - 입력

- departments 테이블의 데이터 입력







- 쿼리 창에 `insert into departments values ('d010','education');` 입력 또는  버튼
>> `select * from departments;` >> F5 버튼을 통해 행이 추가된 것을 확인



Query Query History

```
1 insert into departments values ('d010','education');
2 select * from departments;
```

Data output Messages Notifications

Add row


| | dept_no [PK] character (4) | dept_name character varying (40) |
|---|-------------------------------|-------------------------------------|
| 1 | d001 | Marketing |
| 2 | d002 | Finance |
| 3 | d003 | Human Resources |

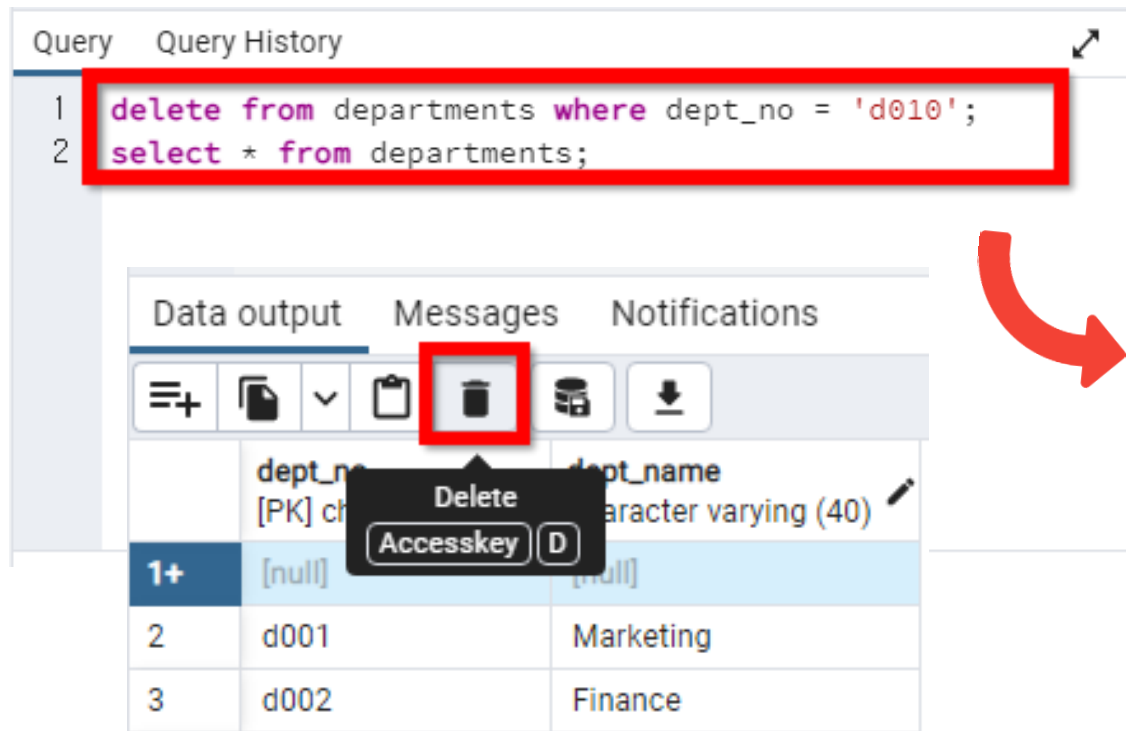
| | dept_no [PK] character (4) | dept_name character varying (40) |
|----|-------------------------------|-------------------------------------|
| 1 | d001 | Marketing |
| 2 | d002 | Finance |
| 3 | d003 | Human Resources |
| 4 | d004 | Production |
| 5 | d005 | Development |
| 6 | d006 | Quality Management |
| 7 | d007 | Sales |
| 8 | d008 | Research |
| 9 | d009 | Customer Service |
| 10 | d010 | education |

SECTION 02 PostgreSQL을 이용한 데이터베이스 구축 절차

2-3. 데이터 활용 - 삭제

- departments 테이블의 데이터 삭제

- 쿼리 창에 `delete from departments where dept_no = 'd010';` 입력 또는  버튼
>> `select * from departments;` >> F5 버튼을 통해 dept_no가 'd010' 인 행이 삭제된 것을 확인

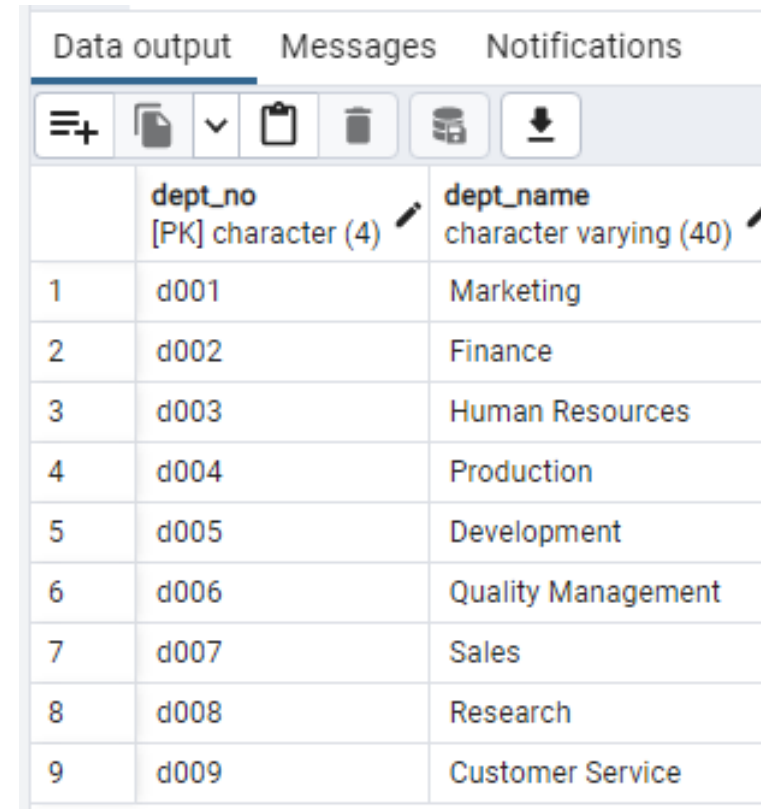


The screenshot shows the PostgreSQL query editor with two queries entered:

```
1 delete from departments where dept_no = 'd010';
2 select * from departments;
```

The queries are highlighted with a red box. Below the queries, the 'Data output' tab is active, showing a table with columns 'dept_no' and 'dept_name'. The first row is highlighted in blue, and a tooltip indicates that the 'Delete' button (trash icon) can be used to delete the selected row. A red arrow points from the 'Delete' button in the query editor to the 'Delete' button in the data output table.

| | dept_no [PK] character (4) | dept_name character varying (40) |
|----|-------------------------------|-------------------------------------|
| 1+ | [null] | [null] |
| 2 | d001 | Marketing |
| 3 | d002 | Finance |



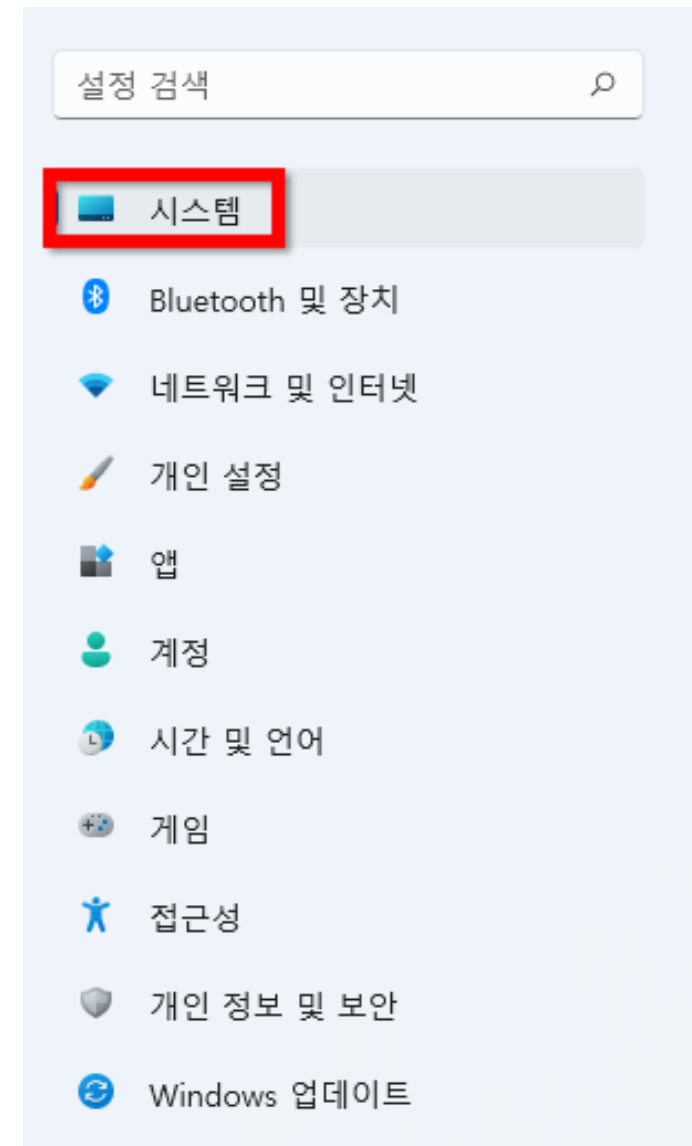
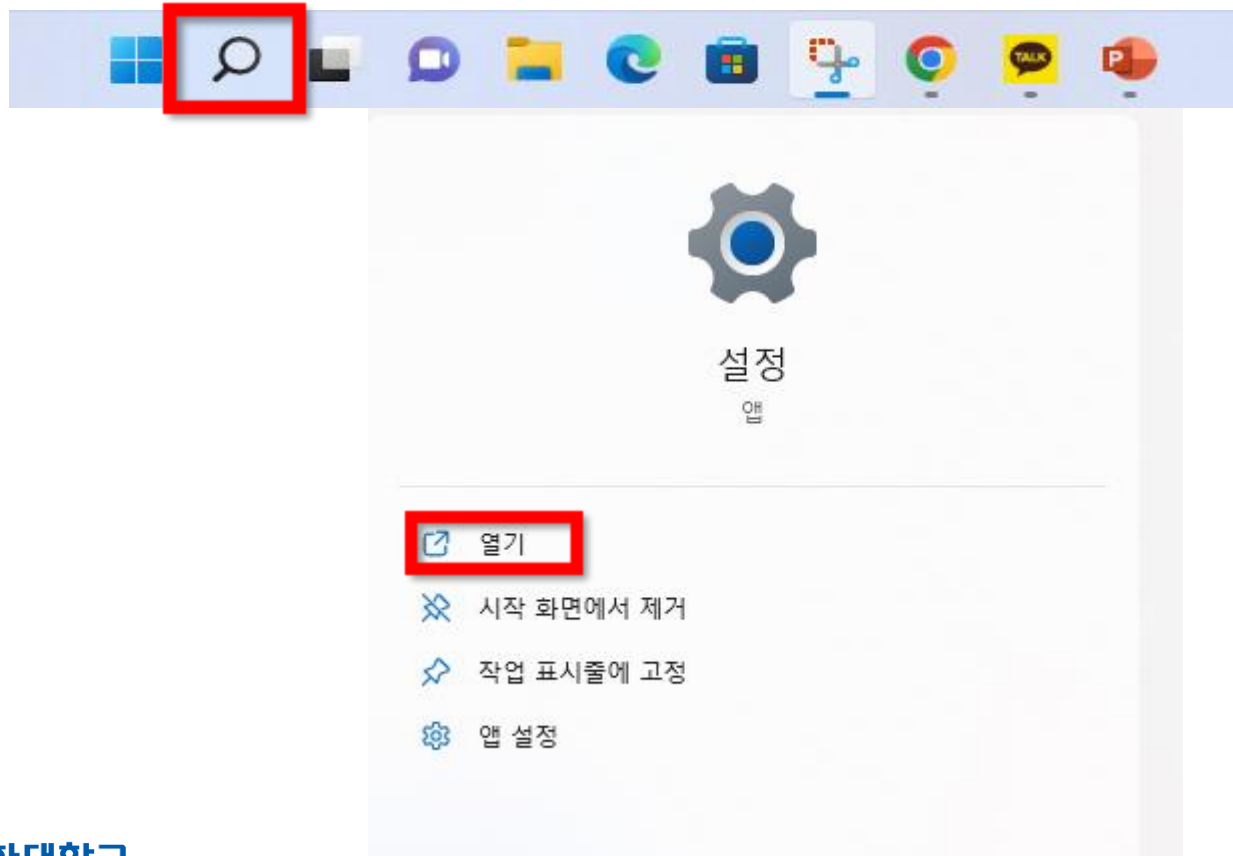
The screenshot shows the 'Data output' tab with a table containing 9 rows of department data. The columns are 'dept_no' and 'dept_name'.

| | dept_no [PK] character (4) | dept_name character varying (40) |
|---|-------------------------------|-------------------------------------|
| 1 | d001 | Marketing |
| 2 | d002 | Finance |
| 3 | d003 | Human Resources |
| 4 | d004 | Production |
| 5 | d005 | Development |
| 6 | d006 | Quality Management |
| 7 | d007 | Sales |
| 8 | d008 | Research |
| 9 | d009 | Customer Service |

SECTION 02 데이터베이스 백업 및 관리

2-4. 백업과 복원 - 환경변수 설정

- 백업과 복원을 위해서는 환경변수 설정이 필요
- 작업표시줄의 검색기능에 '설정' 검색 후 실행 >> 시스템



SECTION 02 데이터베이스 백업 및 관리

2-4. 백업과 복원 - 환경변수 설정

- 정보 >> 고급 시스템 설정

실시간 및 복사 기록, 동기화, 시뮬레이션

정보 >

장치 사양, PC 이름 바꾸기, Windows 사양

장치 사양

복사

| | | |
|---------|--|----------|
| 장치 이름 | DESKTOP-5R0HGKU | |
| 프로세서 | AMD Ryzen 5 5600G with Radeon Graphics | 3.90 GHz |
| 설치된 RAM | 16.0GB(15.9GB 사용 가능) | |
| 장치 ID | 7AA130F1-EA24-4A65-A2FA-EAAB28C6A0B0 | |
| 제품 ID | 00331-20020-00000-AA416 | |
| 시스템 종류 | 64비트 운영 체제, x64 기반 프로세서 | |
| 펜 및 터치 | 이 디스플레이에 사용할 수 있는 펜 또는 터치식 입력이 없습니다. | |

관련 링크 도메인 또는 작업 영역 시스템 보호 **고급 시스템 설정**

Windows 사양

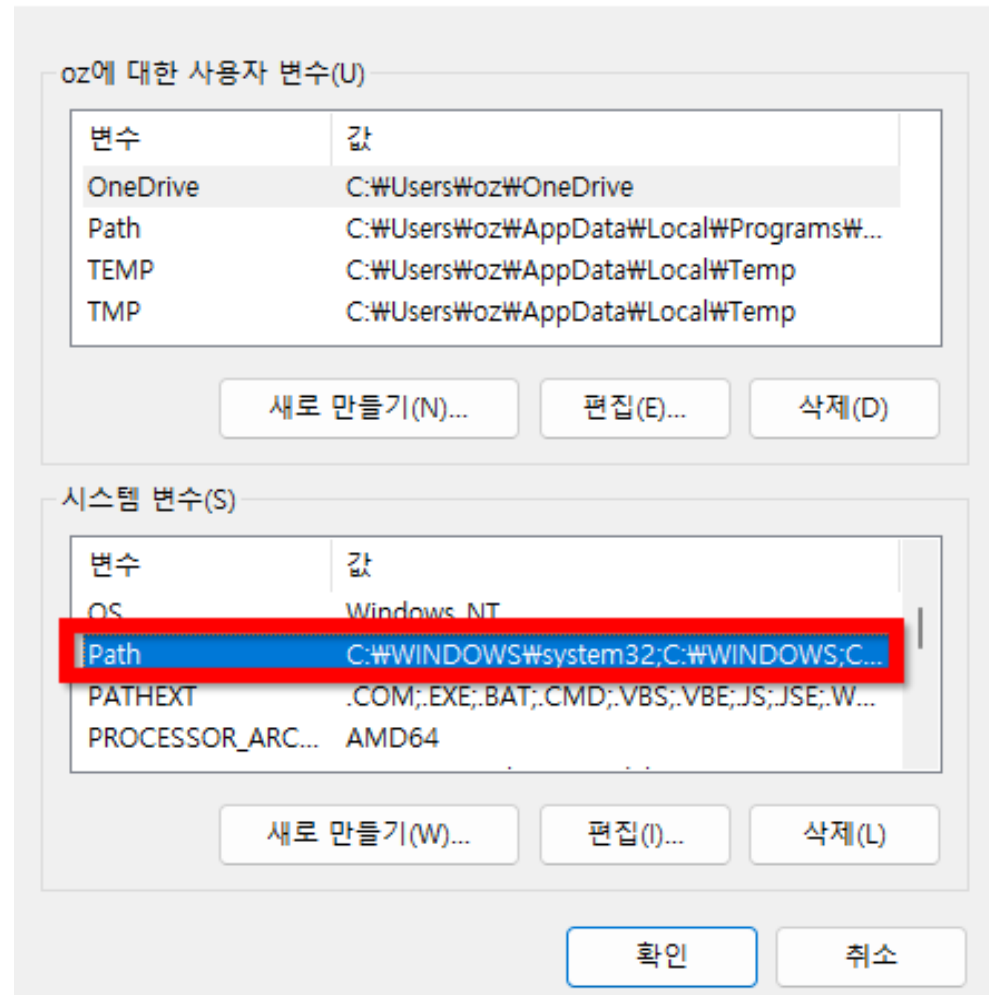
복사

SECTION 02 데이터베이스 백업 및 관리

2-4. 백업과 복원 - 환경변수 설정

- 환경 변수 >> 시스템 변수에서 변수 'Path'를 찾아 더블 클릭

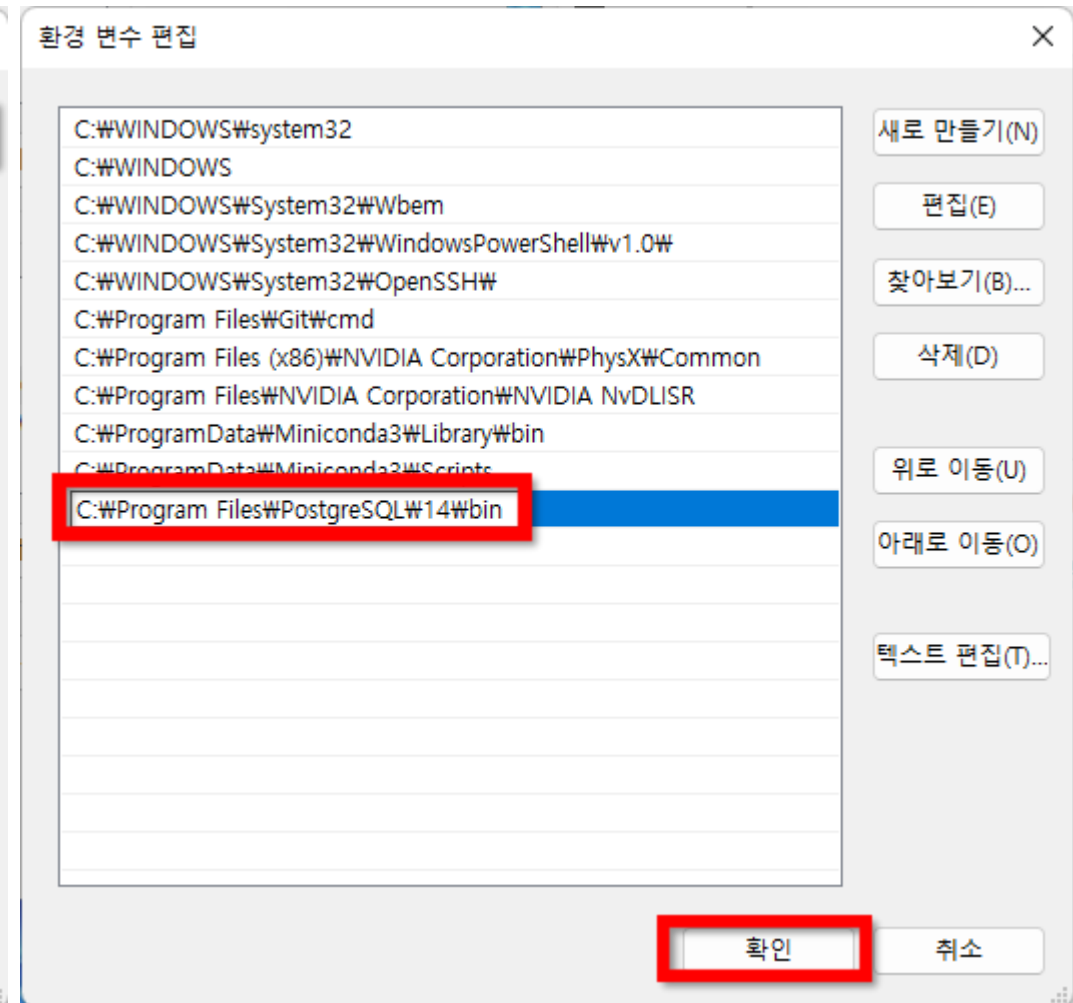
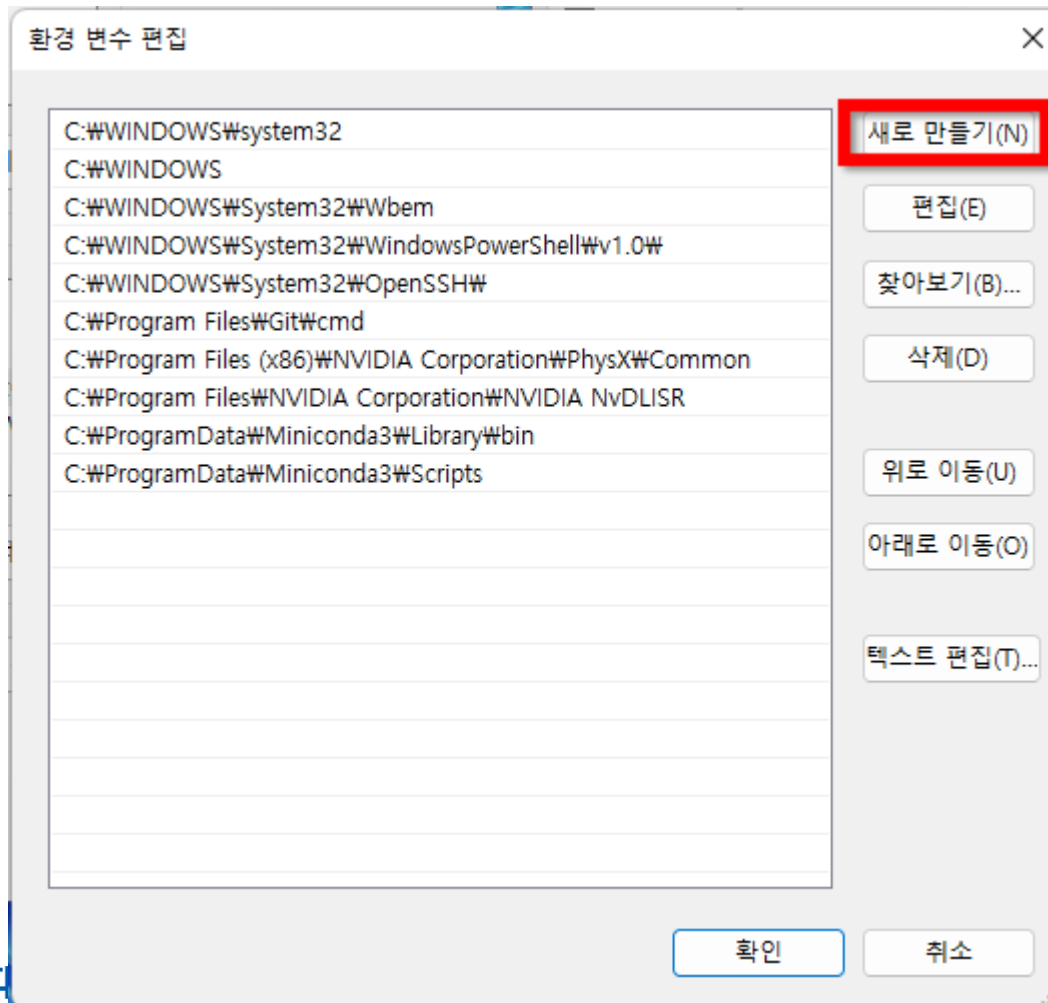
환경 변수



SECTION 02 데이터베이스 백업 및 관리

2-4. 백업과 복원 - 환경변수 설정

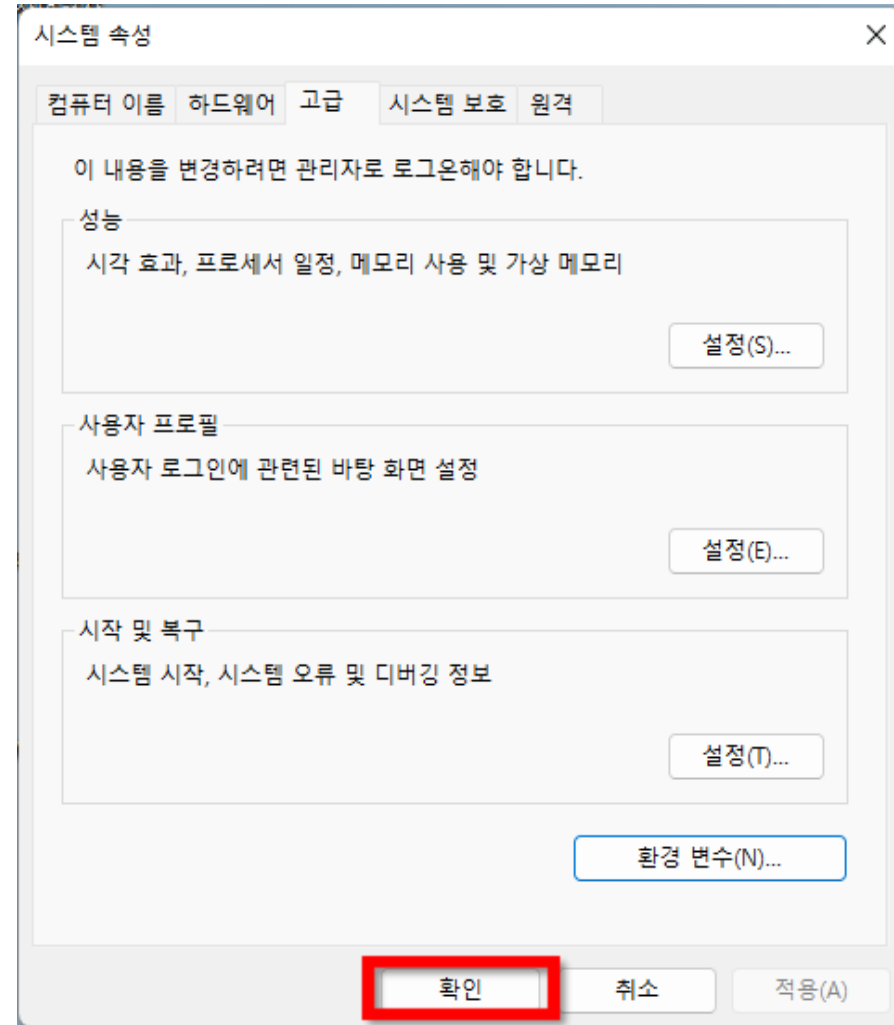
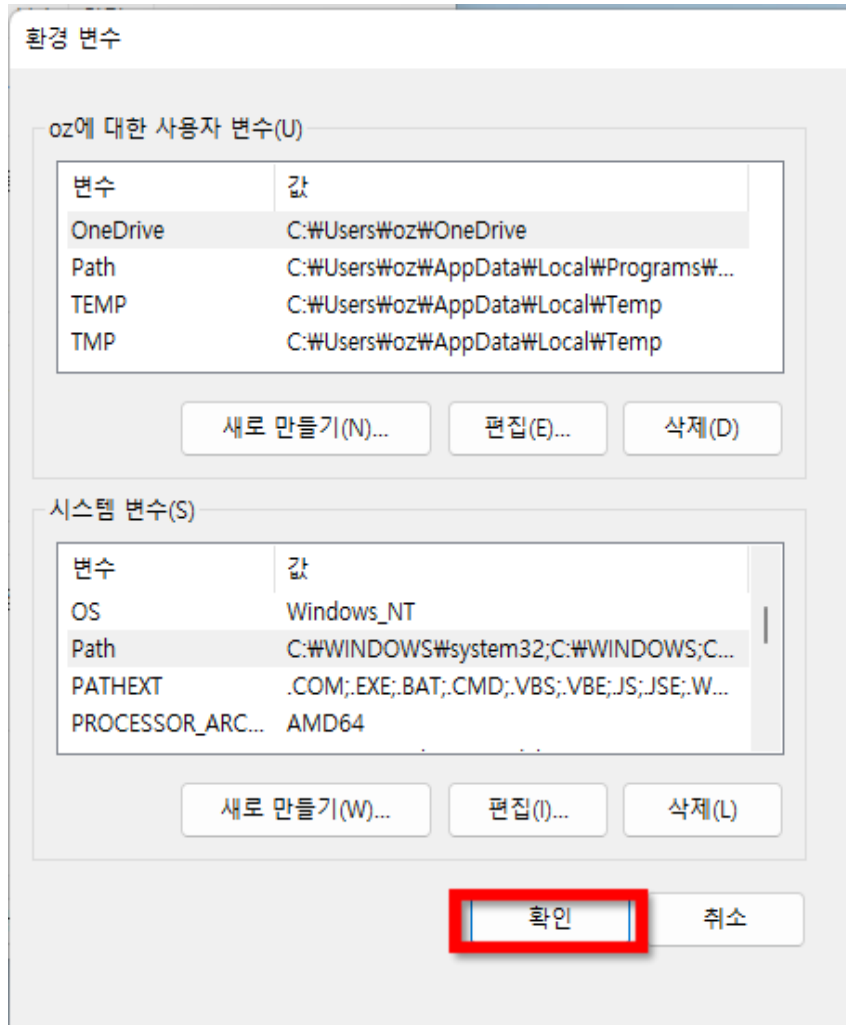
- 새로 만들기 >> 'C:\Program Files\PostgreSQL\14\bin' 입력 >> 확인



SECTION 02 데이터베이스 백업 및 관리

2-4. 백업과 복원 - 환경변수 설정

- 확인 >> 확인



SECTION 02 데이터베이스 백업 및 관리

2-4. 백업과 복원

- 백업과 복원은 관리자가 해야 할 가장 중요한 일
 - 중요한 데이터를 모두 잃어버리는 불의의 사고에 대비해야 함
- 백업
 - 현재의 데이터베이스를 다른 곳에 보관하는 작업
 - `pg_dump -d DB명 -U 사용자명 -F 파일형식 -v > 파일경로/백업파일명`: 한 개의 데이터베이스 백업
 - `pg_dumpall -U 사용자명 -F 파일형식 -v > 파일경로/백업파일명`: 모든 데이터베이스 백업

< 옵션 >

- (1) d : 백업할 DB 이름
- (2) U : DB에 접속하는 사용자 이름
- (3) F : 백업 파일의 형식
- (4) v : 진행 상황에 대한 자세한 정보를 보여줌

< 파일형식 옵션 >

- (1) p : plain text 형식이며 확장자 .sql 권장
- (2) c : custom 형식이며 확장자 .dump 권장
- (3) t : tar 형식이며 확장자 .tar 권장
- (4) d : directory 형식

SECTION 02 데이터베이스 백업 및 관리

2-4. 백업과 복원

◦ 복원

- 데이터베이스에 문제 발생 시 다른 곳에 백업된 데이터를 이용해 원상태로 돌려놓는 작업
- 데이터베이스 복원을 위해서는 먼저 빈 데이터베이스가 생성되어야 함
- `pg_restore -C -d DB명 -U 사용자명 -F 파일형식 -v < 백업파일명` : 비 텍스트 파일(c,t,d)과
pg_dump로 백업된 파일 복원
- `psql -d DB명 -U 사용자명 -F 파일형식 -v < 백업파일명` : 텍스트 파일(p)과
pg_dump, pg_dumpall로 백업된 파일 복원

< 옵션 >

- (1) d : 복원할 DB 이름
- (2) U : DB에 접속하는 사용자 이름
- (3) F : 복원 파일의 형식
- (4) v : 진행 상황에 대한 자세한 정보를 보여줌

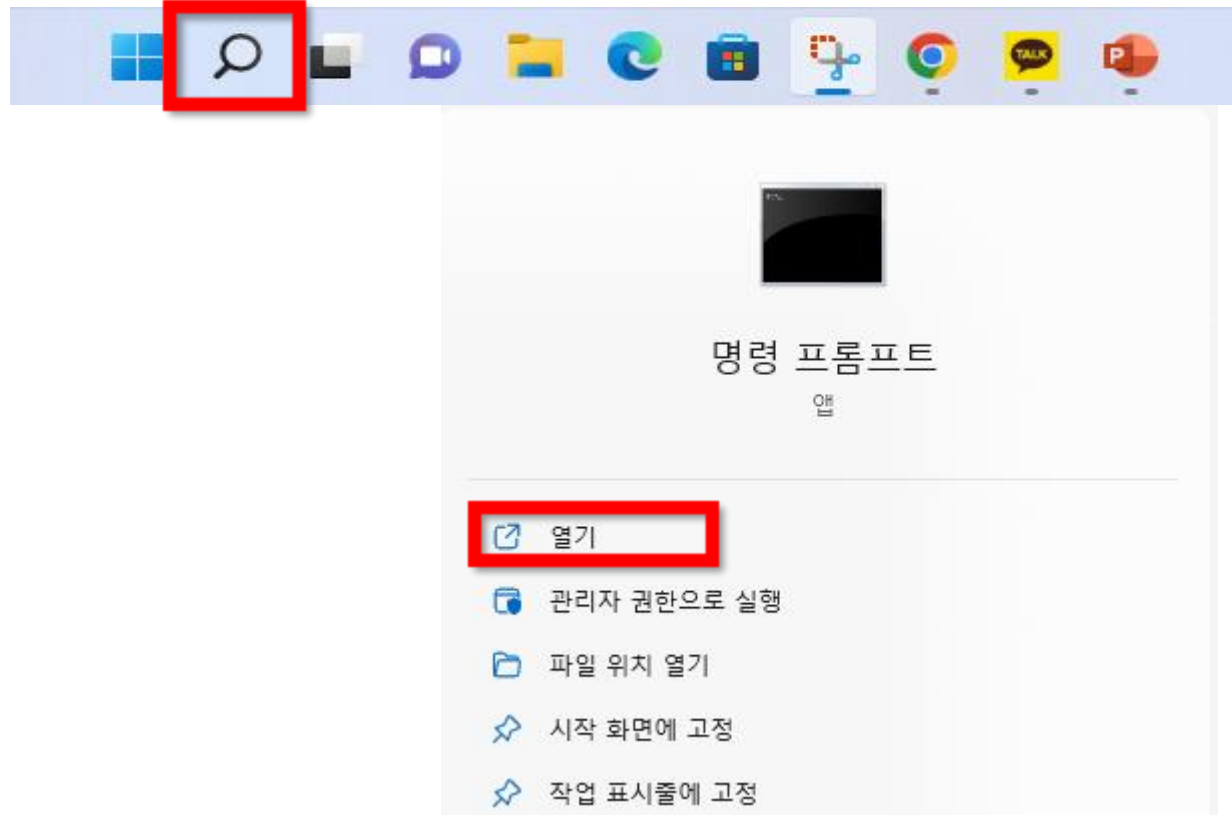
< pg_restore의 C옵션 >

- (1)기능: 사용자가 직접 빈 DB를 생성하지 않아도
자동으로 복원을 위한 데이터베이스를 만듦
- (2)조건: DB명 자리에 postgres, 사용자명 자리에
postgres가 들어가야 함

SECTION 02 데이터베이스 백업 및 관리

2-4. 백업과 복원 - cmd에서의 백업

- tar형식으로 백업과 복원해보기
- 작업표시줄의 검색기능에 'cmd' 검색 후 실행



SECTION 02 데이터베이스 백업 및 관리

2-4. 백업과 복원 - cmd에서의 백업

- 'pg_dump -d employees -U postgres -F t -v > backup_employees.tar' >> 엔터 키
 - >> 암호 입력 >> 엔터 키

```
C:\Users\zzz>pg_dump -d employees -U postgres -F t -v > backup_employees.tar
암호: 
pg_dump: 마지막 내장 OID는 16383
pg_dump: 확장 기능은
pg_dump: 확장 멤버를 스캔
pg_dump: 스키마들을 읽기
pg_dump: 사용자 정의 테이블 함수
pg_dump: 사용자 정의 함수
pg_dump: 사용자 정의 자료형
pg_dump: 프로시저 언어를 읽기
pg_dump: 사용자 정의 집계 함수
pg_dump: 사용자 정의 전역 변수
pg_dump: 사용자 정의 상수
pg_dump: 사용자 정의 시스템 객체
pg_dump: 사용자 정의 테이블
```

SECTION 02 데이터베이스 백업 및 관리

2-4. 백업과 복원 - cmd에서의 복원

- 기존 DB를 삭제하고 백업해둔 DB 복원해보기
- 'psql -U postgres' >> 엔터 키 >> 암호 입력 >> 엔터 키
>> 'DROP DATABASE employees;' >> 엔터 키

```
C:\Users\moz> psql -U postgres
postgres 사용자의 암호: 
psql (14.4)
도움말을 보려면 "help"를 입력하십시오.

postgres=# DROP DATABASE employees;
DROP DATABASE
```

SECTION 02 데이터베이스 백업 및 관리

2-4. 백업과 복원 - cmd에서의 복원

- 'W|' 명령어를 통해 DB가 삭제된 것을 확인
- 'W|' >> 엔터키 >> 'Wq' >> 엔터 키

소문자 l

psql을 빠져나가기 위한 명령어

```
postgres=# W|

```

| 이름 | 소유주 | 인코딩 | 데이터베이스 목록 Collate | Ctype | 액세스 권한 |
|-----------|----------|------|----------------------|------------------|---------------|
| postgres | postgres | UTF8 | Korean_Korea.949 | Korean_Korea.949 | |
| template0 | postgres | UTF8 | Korean_Korea.949 | Korean_Korea.949 | =c/postgres + |
| template1 | postgres | UTF8 | Korean_Korea.949 | Korean_Korea.949 | =c/postgres + |

(3개 행)

```
postgres=# Wq
C:\Users\moz>
```


SECTION 02 데이터베이스 백업 및 관리

2-4. 백업과 복원 - cmd에서의 복원

- 'pg_restore -C -d postgres -U postgres -v < backup_employees.tar' >> 엔터 키
>> 암호 입력 >> 엔터 키

```
C:\Users\Woz>pg_restore -C -d postgres -U postgres -v < backup_employees.tar
pg_restore: 복원 작업을 위해 데이터베이스에 접속 중
암호: 
pg_restore: DATABASE "employees" 만드는 중
pg_restore: "employees" 새 데이터베이스에 접속중
pg_restore: TYPE "public.gen" 만드는 중
pg_restore: TABLE "public.departments" 만드는 중
pg_restore: TABLE "public.dept_emp" 만드는 중
pg_restore: TABLE "public.dept_manager" 만드는 중
pg_restore: TABLE "public.employees" 만드는 중
pg_restore: TABLE "public.salaries" 만드는 중
pg_restore: TABLE "public.titles" 만드는 중
pg_restore: "public.departments" 테이블의 자료를 처리 중
pg_restore: "public.dept_emp" 테이블의 자료를 처리 중
pg_restore: "public.dept_manager" 테이블의 자료를 처리 중
pg_restore: "public.employees" 테이블의 자료를 처리 중
pg_restore: "public.salaries" 테이블의 자료를 처리 중
pg_restore: "public.titles" 테이블의 자료를 처리 중
```

SECTION 02 데이터베이스 백업 및 관리

2-4. 백업과 복원 - cmd에서의 복원

- 'W |' 과 'Wdt' 명령어를 통해 DB가 복원된 것을 확인
- 'psql -U postgres' >> 엔터 키 >> 암호 입력 >> 엔터 키

```
C:\Users\moz>psql -U postgres
postgres 사용자의 암호: 
psql (14.4)
도움말을 보려면 "help"를 입력하십시오.
```

SECTION 02 데이터베이스 백업 및 관리

2-4. 백업과 복원 - cmd에서의 복원

- 'ㄱ' >> 엔터 키 >> 'dt' >> 엔터 키

```
postgres=# ㄱ
```

| 이름 | 소유주 | 인코딩 | 데이터베이스 Collate | 목록 Ctype | 액세스 권한 |
|-----------|----------|------|------------------|------------------|-------------------------|
| employees | postgres | UTF8 | Korean_Korea.949 | Korean_Korea.949 | |
| postgres | postgres | UTF8 | Korean_Korea.949 | Korean_Korea.949 | |
| template0 | postgres | UTF8 | Korean_Korea.949 | Korean_Korea.949 | =c/postgres + |
| template1 | postgres | UTF8 | Korean_Korea.949 | Korean_Korea.949 | postgres=CTc/postgres + |

(4개 행)

```
postgres=# ㄱdt
```

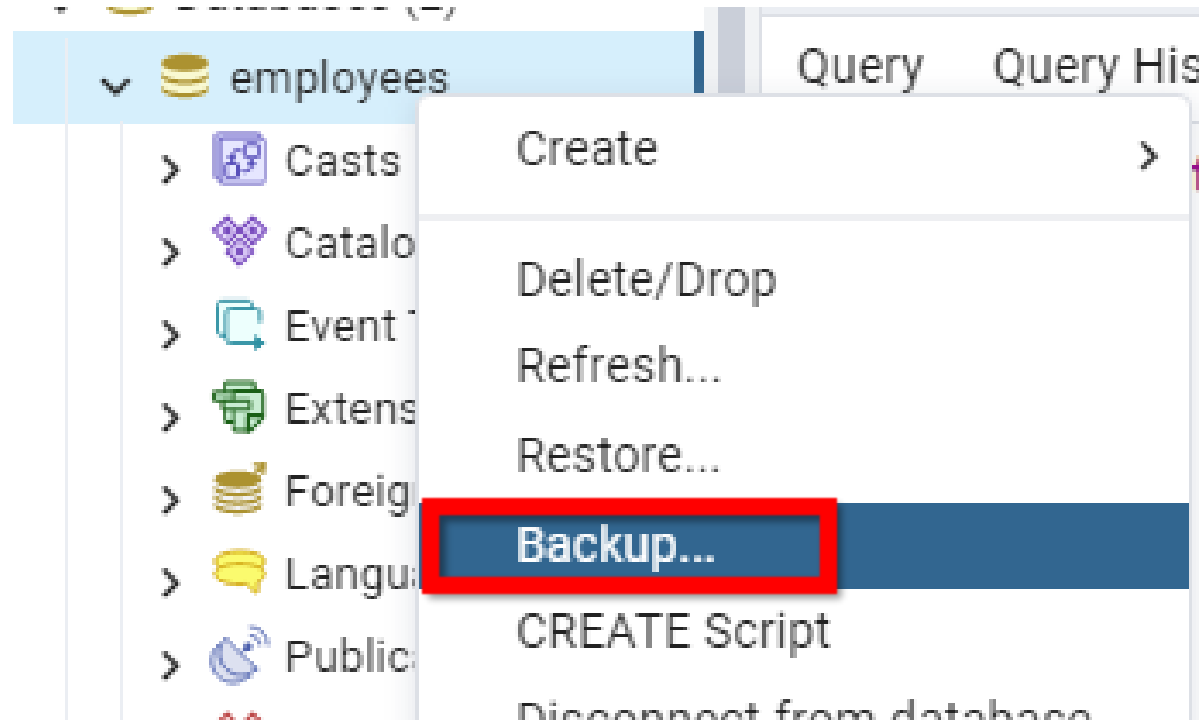
| 스키마 | 데이터베이스 레이션(relation) 이름 | 종류 | 소유주 |
|--------|-------------------------|-----|----------|
| public | departments | 테이블 | postgres |
| public | dept_emp | 테이블 | postgres |
| public | dept_manager | 테이블 | postgres |
| public | employees | 테이블 | postgres |
| public | salaries | 테이블 | postgres |
| public | titles | 테이블 | postgres |

(6개 행)

SECTION 02 데이터베이스 백업 및 관리


2-4. 백업과 복원 - pgAdmin4에서의 백업

- 백업할 데이터베이스 우클릭 >> Backup





SECTION 02 데이터베이스 백업 및 관리

2-4. 백업과 복원 - pgAdmin4에서의 백업

-  버튼 >> 백업할 파일 이름 입력 >> Create

Backup (Database: employees)








General Data/Objects Options





Filename  

Format Custom

Compression ratio

Encoding Select an item...

  C:\Users\oz backup_employees     

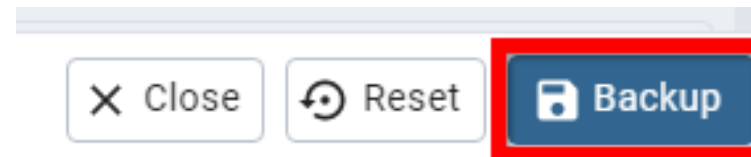
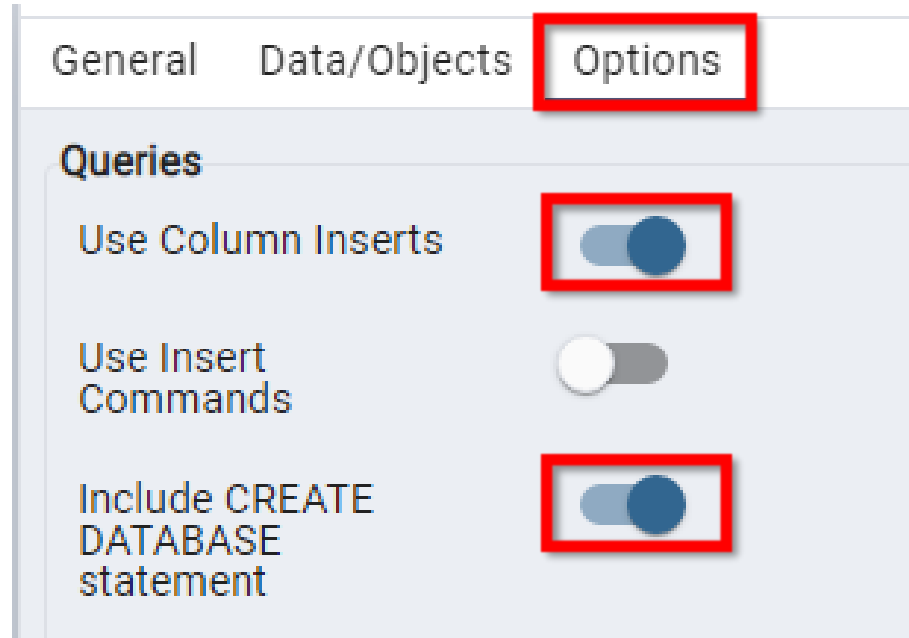
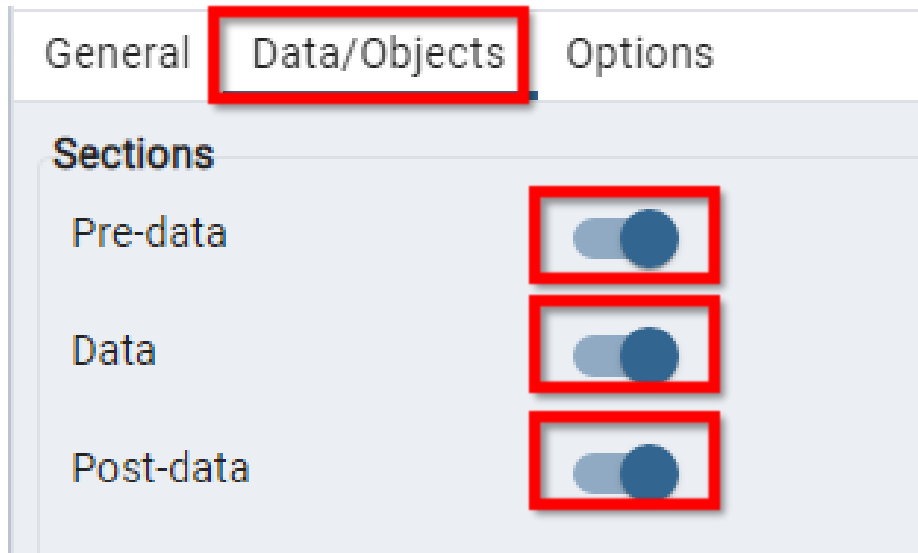
| Name | Size | Modified |
|---|---------|--------------------------|
|  .bash_history | 96.0 B | Fri Jul 8 13:46:33 2022 |
|  .gitconfig | 205.0 B | Fri Jul 8 13:03:59 2022 |
|  .lessht | 20.0 B | Fri Jul 8 13:58:44 2022 |
|  anseI | | Tue Jul 12 14:29:27 2022 |

 Cancel  Create

SECTION 02 데이터베이스 백업 및 관리

2-4. 백업과 복원 - pgAdmin4에서의 백업

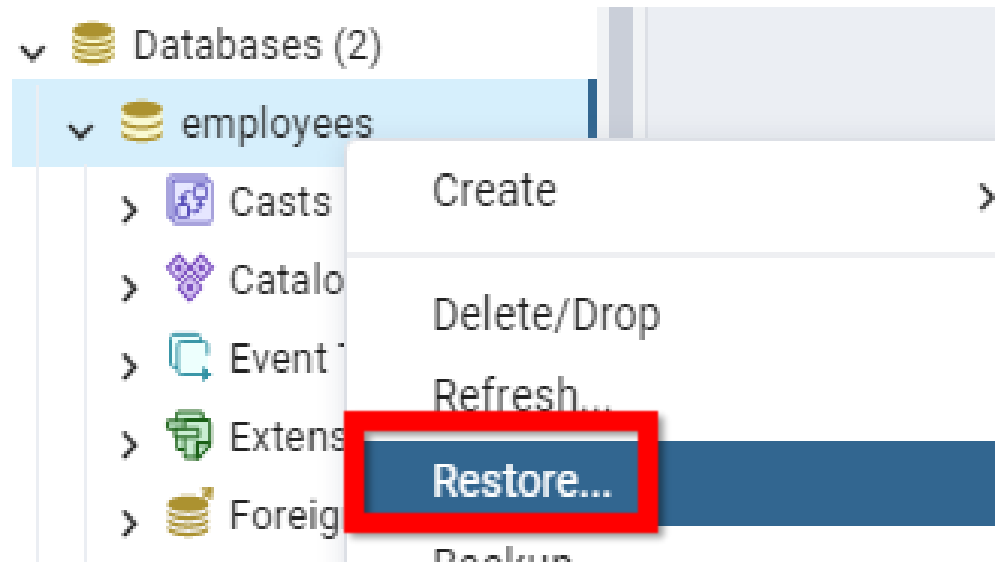
- 상단의 Data/Objects >> 옵션 체크 >> Options >> 옵션 체크 >> Backup



SECTION 02 데이터베이스 백업 및 관리

2-4. 백업과 복원 - pgAdmin4에서의 복원

- employees DB 우클릭 >> Restore



SECTION 02 데이터베이스 백업 및 관리



2-4. 백업과 복원 - pgAdmin4에서의 복원

-  버튼 >> 백업된 파일 선택 >> Select

Restore (Database: employees)







General Data/Objects Options



Format Custom or tar

Filename  

Number of jobs

Role name Select an item...

| | | |
|--|---------|--------------------------|
|  .bash_history | 96.0 B | Fri Jul 8 13:46:33 2022 |
|  .gitconfig | 205.0 B | Fri Jul 8 13:03:59 2022 |
|  .lessht | 20.0 B | Fri Jul 8 13:58:44 2022 |
|  ansel | | Tue Jul 12 14:29:27 2022 |
|  backup_employees | 38.1 MB | Fri Jul 22 19:07:30 2022 |
|  Contacts | | Thu Jun 16 13:31:33 2022 |

 Cancel  Select

SECTION 02 데이터베이스 백업 및 관리

2-4. 백업과 복원 - pgAdmin4에서의 복원

- Restore

