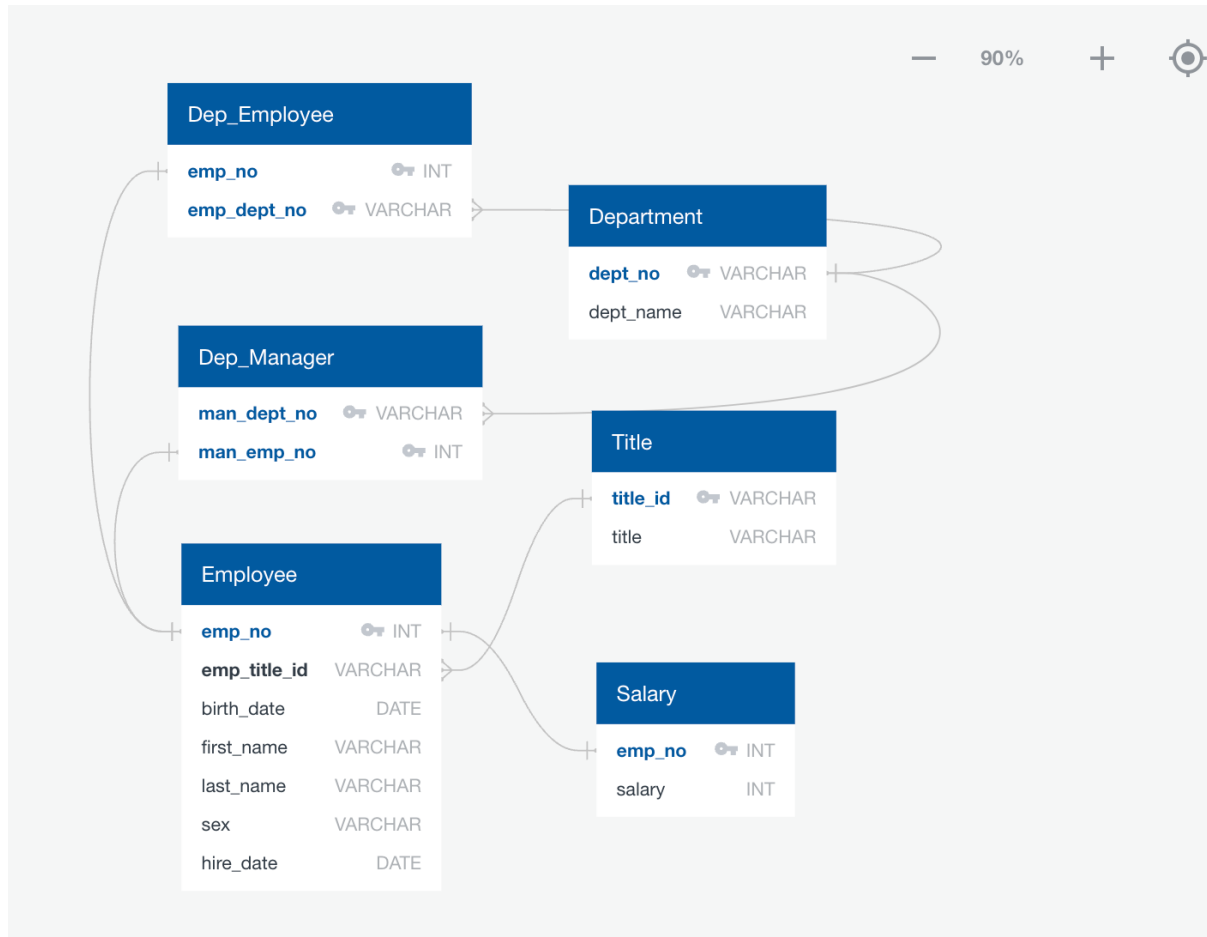


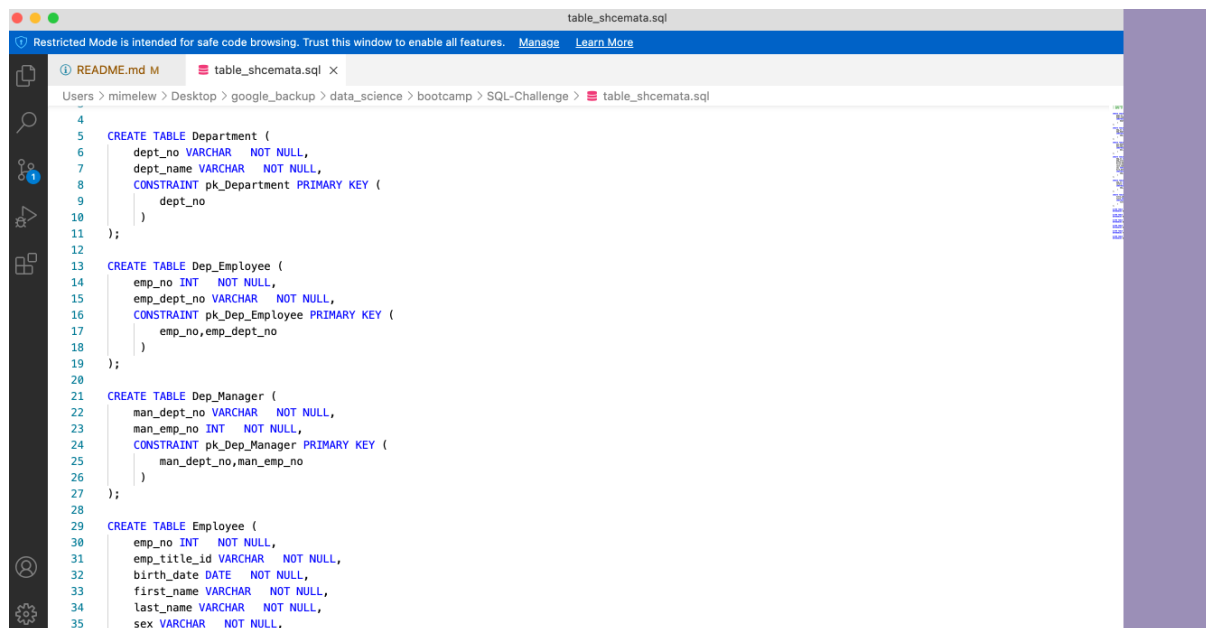
## Project Summary

In this project, I conducted the full process of data engineering/modelling.

First of all, with all the separated tables with employer information, I created a DBD chart to inform the realtions between all the seperated tables that hosted different type of employee data.

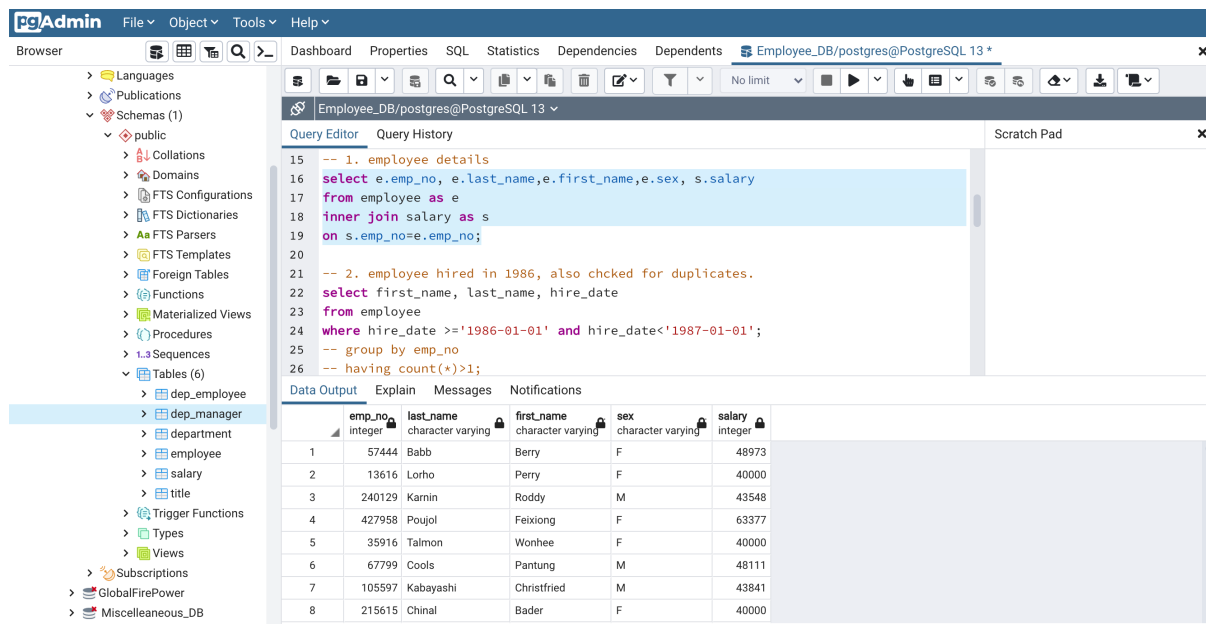


Second, I wrote schemas to create a database that could house all the tables, named employee\_DB.



```
4
5 CREATE TABLE Department (
6   dept_no VARCHAR NOT NULL,
7   dept_name VARCHAR NOT NULL,
8   CONSTRAINT pk_Department PRIMARY KEY (
9     dept_no
10  )
11 );
12
13 CREATE TABLE Dep_Employee (
14   emp_no INT NOT NULL,
15   emp_dept_no VARCHAR NOT NULL,
16   CONSTRAINT pk_Dep_Employee PRIMARY KEY (
17     emp_no, emp_dept_no
18   )
19 );
20
21 CREATE TABLE Dep_Manager (
22   man_dept_no VARCHAR NOT NULL,
23   man_emp_no INT NOT NULL,
24   CONSTRAINT pk_Dep_Manager PRIMARY KEY (
25     man_dept_no, man_emp_no
26   )
27 );
28
29 CREATE TABLE Employee (
30   emp_no INT NOT NULL,
31   emp_title_id VARCHAR NOT NULL,
32   birth_date DATE NOT NULL,
33   first_name VARCHAR NOT NULL,
34   last_name VARCHAR NOT NULL,
35   sex VARCHAR NOT NULL,
```

Third, with the DND, I was able to make queries to different tables in the database through manipulation, such as joining, filtering etc, to retrieve data I need. I can also create a new table to house these filtered data if required in the future.



The screenshot shows the pgAdmin interface with the following components:

- Left Panel (Browser):** Shows the database structure for 'Employee\_DB/postgres@PostgreSQL 13'. The 'Tables (6)' folder is expanded, showing 'dep\_employee', 'dep\_manager', 'department', 'employee', 'salary', and 'title'.
- Query Editor:** Contains two SQL queries:

```
-- 1. employee details
select e.emp_no, e.last_name, e.first_name, e.sex, s.salary
from employee as e
inner join salary as s
on s.emp_no=e.emp_no;

-- 2. employee hired in 1986, also checked for duplicates.
select first_name, last_name, hire_date
from employee
where hire_date >='1986-01-01' and hire_date <'1987-01-01';
-- group by emp_no
-- having count(*)>1;
```
- Data Output:** Displays the results of the first query in a table with 5 columns: emp\_no, last\_name, first\_name, sex, and salary. The results are as follows:

emp_no	last_name	first_name	sex	salary	
1	57444	Babb	Berry	F	48973
2	13616	Lorho	Perry	F	40000
3	240129	Karmin	Roddy	M	43548
4	427958	Poujol	Feixiong	F	63377
5	35916	Talmon	Wonhee	F	40000
6	67799	Cools	Pantung	M	48111
7	105597	Kabayashi	Christfried	M	43841
8	215615	Chinal	Bader	F	40000

Admin

FileObjectToolsHelp

DashboardPropertiesSQLStatisticsDependenciesDependentsEmployee\_DB/postgres@PostgreSQL 13\*

Browser

Query EditorQuery HistoryScratch Pad

19on s.emp\_no=e.emp\_no;

20

21-- 2. employee hired in 1986, also chcked for duplicates.

22select first\_name, last\_name, hire\_date

23from employee

24where hire\_date >='1986-01-01' and hire\_date<'1987-01-01';

25-- group by emp\_no

26-- having count(\*)>1;

27

28-- 3. manager info

29select dm.man\_dept\_no, d.dept\_name, dm.man\_emp\_no, e.last\_name,e.first\_name

30from dep\_manager as dm

Data OutputExplainMessagesNotifications

	first_name	last_name	hire_date	
	character varying	character varying	date	
1	Eran	Cusworth	1986-11-14	
2	Bojan	Zalocco	1986-10-14	
3	Nevio	Demizu	1986-05-18	
4	Ziva	Vecchi	1986-07-03	
5	Mohit	Speak	1986-01-14	
6	Qunsheng	Speer	1986-02-13	
7	Dines	Encarnacion	1986-08-02	
8	Hamgdar	Swick	1986-05-28	

Successfully run. Total query runtime: 131 msec. 36150 rows affected.