Tech ABC Corp - HR Database

[SRIJANA THAPA & 21-01-2025]

Business Scenario

Business requirement

Tech ABC Corp saw explosive growth with a sudden appearance onto the gaming scene with their new AI-powered video game console. As a result, they have gone from a small 10 person operation to 200 employees and 5 locations in under a year. HR is having trouble keeping up with the growth, since they are still maintaining employee information in a spreadsheet. While that worked for ten employees, it has becoming increasingly cumbersome to manage as the company expands.

As such, the HR department has tasked you, as the new data architect, to design and build a database capable of managing their employee information.

Dataset

The HR dataset you will be working with is an Excel workbook which consists of 206 records, with eleven columns. The data is in human readable format, and has not been normalized at all. The data lists the names of employees at Tech ABC Corp as well as information such as job title, department, manager's name, hire date, start date, end date, work location, and salary.

IT Department Best Practices

The IT Department has certain Best Practices policies for databases you should follow, as detailed in the Best Practices document.

Step 1

Data Architecture Foundations

Step 1: Data Architecture Foundations

Hi,

Welcome to Tech ABC Corp. We are excited to have some new talent onboard. As you may already know, Tech ABC Corp has recently experienced a lot of growth. Our AI powered video game console WOPR has been hugely successful and as a result, our company has grown from 10 employees to 200 in only 6 months (and we are projecting a 20% growth a year for the next 5 years). We have also grown from our Dallas, Texas office, to 4 other locations nationwide: New York City, NY, San Francisco, CA, Minneapolis, MN, and Nashville, TN.

While this growth is great, it is really starting to put a strain on our record keeping in HR. We currently maintain all employee information on a shared spreadsheet. When HR consisted of only myself, managing everyone on an Excel spreadsheet was simple, but now that it is a shared document I am having serious reservations about data integrity and data security. If the wrong person got their hands on the HR file, they would see the salaries of every employee in the company, all the way up to the president.

After speaking with Jacob Lauber, the manager of IT, he suggested I put in a request to have my HR Excel file converted into a database. He suggested I reach out to you as I am told you have experience in designing and building databases. When you are building this, please keep in mind that I want any employee with a domain login to be have read only access the database. I just don't want them having access to salary information. That needs to be restricted to HR and management level employees only. Management and HR employees should also be the only ones with write access. By our current estimates, 90% of users will be read only.

I also want to make sure you know that am looking to turn my spreadsheet into a live database, one I can input and edit information into. I am not really concerned with reporting capabilities at the moment. Since we are working with employee data we are required by federal regulations to maintain this data for at least 7 years; additionally, since this is considered business critical data, we need to make sure it gets backed up properly.

As a final consideration. We would like to be able to connect with the payroll department's system in the future. They maintain employee attendance and paid time off information. It would be nice if the two systems could interface in the future

I am looking forward to working with you and seeing what kind of database you design for us.

Thanks,
Sarah Collins
Head of HR

Data Architect Business Requirement

- **Purpose of the new database:**

  What is the business partner requesting

- **Describe current data management solution:**

  What is the current method data storage/management

- **Describe current data available:**

  What data does the business currently have available

- **Additional data requests:**

  Does the user have future data requests

- **Who will own/manage data**

  What department will own / manage the data in the database

- **Who will have access to database**

  List user types that will have access; also list any restrictions to access.

Data Architect Business Requirement

- **Estimated size of database**

  List the size of the database in terms of numbers of rows. Business users often understand row or column size instead of GBs or MBs

- **Estimated annual growth**

  List any expected growth to the data

- **Is any of the data sensitive/restricted**

  List any data that may be sensitive or restricted from particular users

Data Architect Technical Requirement

- **Justification for the new database**

  Provide at least two justifications for building a database

- **Database objects**

  List the database objects (tables, views, special procedures) that will be created for the database.

  Hint - you may want to circle back to this answer after completing the logical ERD in step 2.

- **Data ingestion**

  Select a data ingestion method (ETL, Direct feed, API) based on the information provided.

Data Architect Technical Requirement

- **Data governance (Ownership and User access)**

  Ownership: who will own and maintain the data

  User Access: who will and will not have access to the data

- **Scalability**

  Should replication or sharding be used to ensure scalability based on user needs

- **Flexibility**

  Describe measures taken to ensure future data integration if needed

- **Storage & retention**

  Storage (disk or in-memory): check IT best practices document

  Retention: how long does the data have to be kept for?

- **Backup**

  IT Best Practices document lists Backup schedule requirements

Step 2

Relational Database Design

Step 2: Relational Database Design

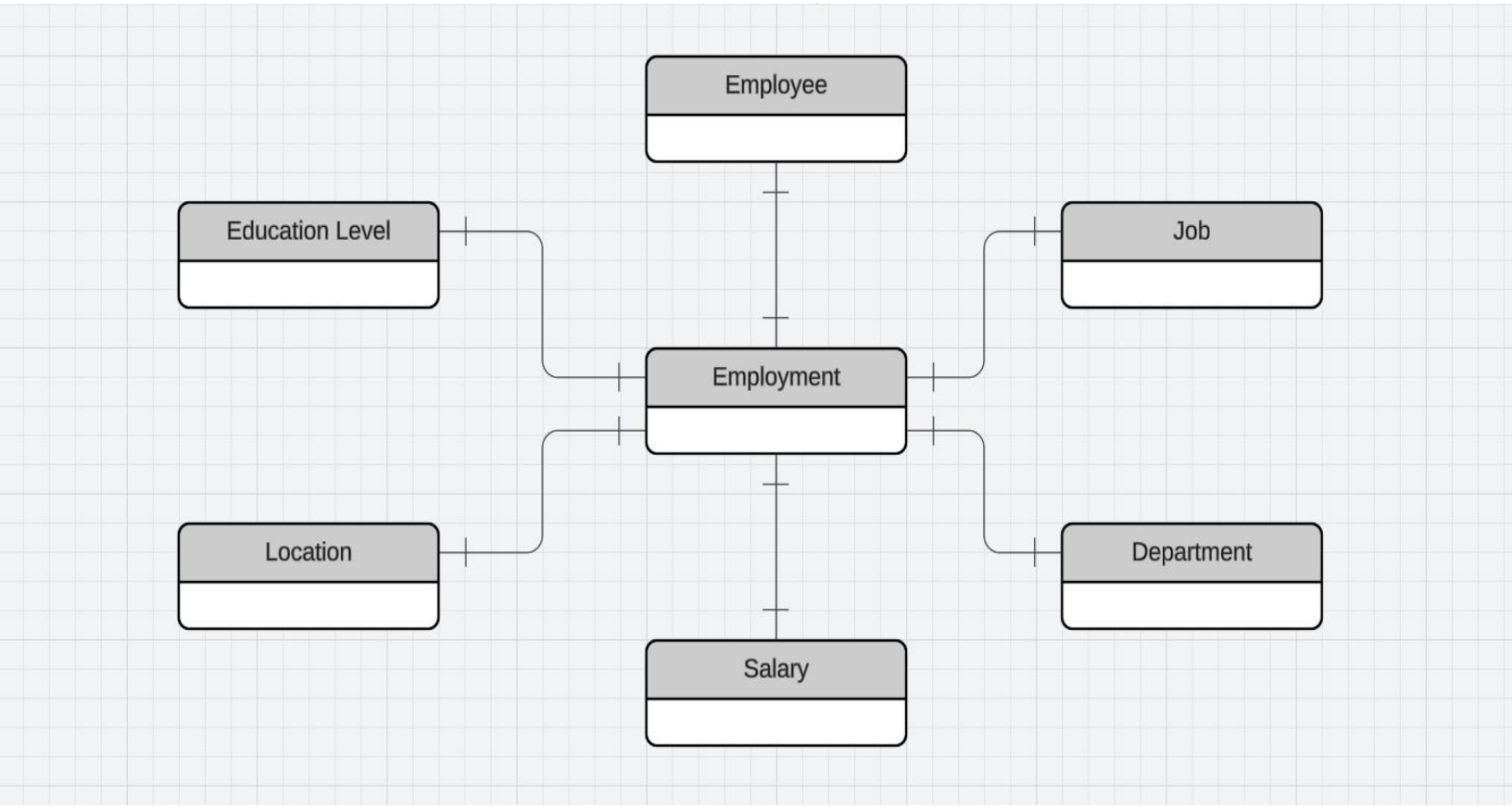
This step is where you will go through the process of designing a new database for Tech ABC Corp's HR department. Using the dataset provided, along with the requirements gathered in step one, you are going to develop a relational database set to the 3NF.

Using Lucidchart, you will create 3 entity relationship diagrams (ERDs) to show how you developed the final design for your data.

You will submit a screenshot for each of the 3 ERDs you create. You will find detailed instructions for developing each of the ERDs over the next several pages.

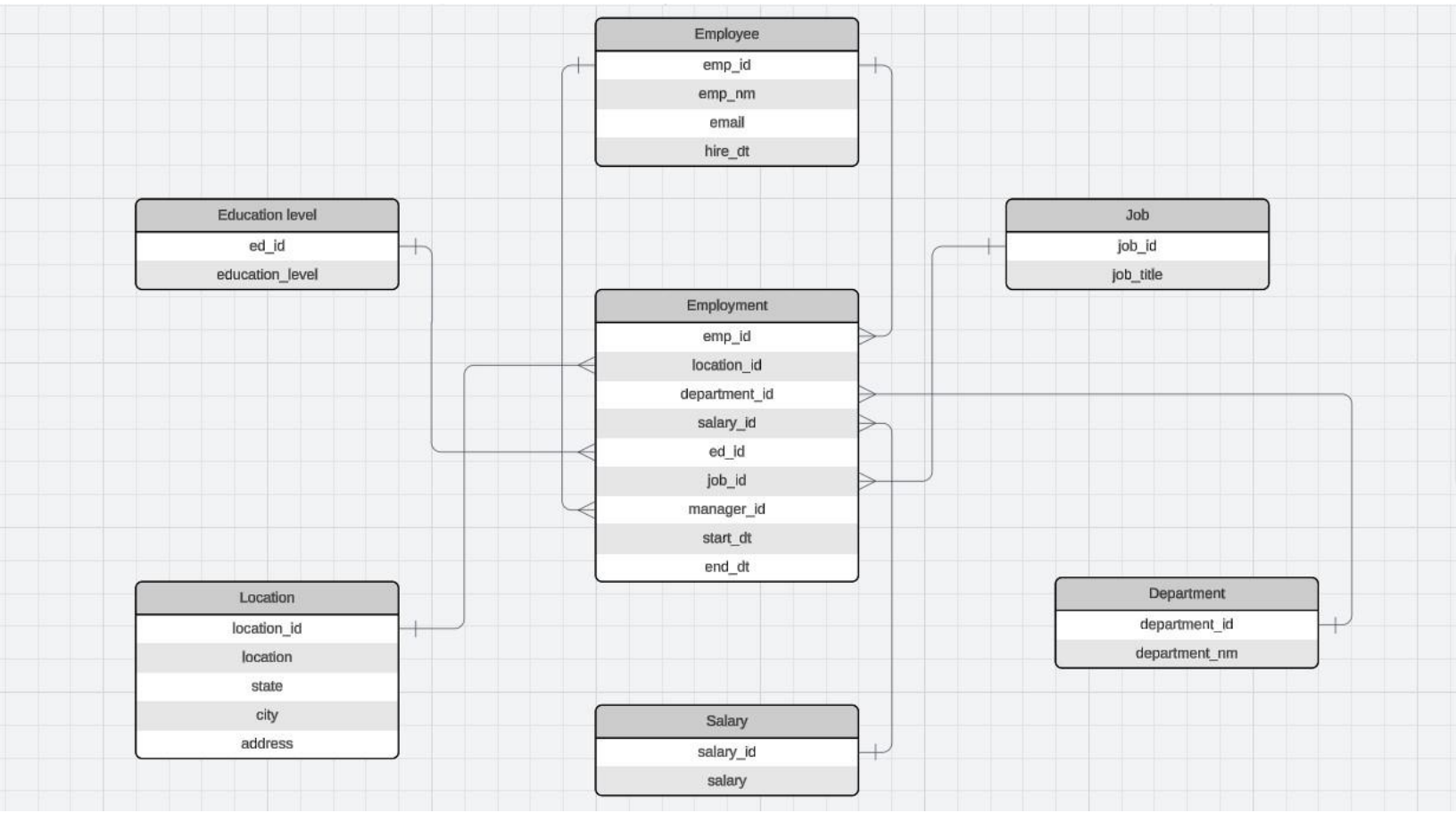
ERD

- **Conceptual**



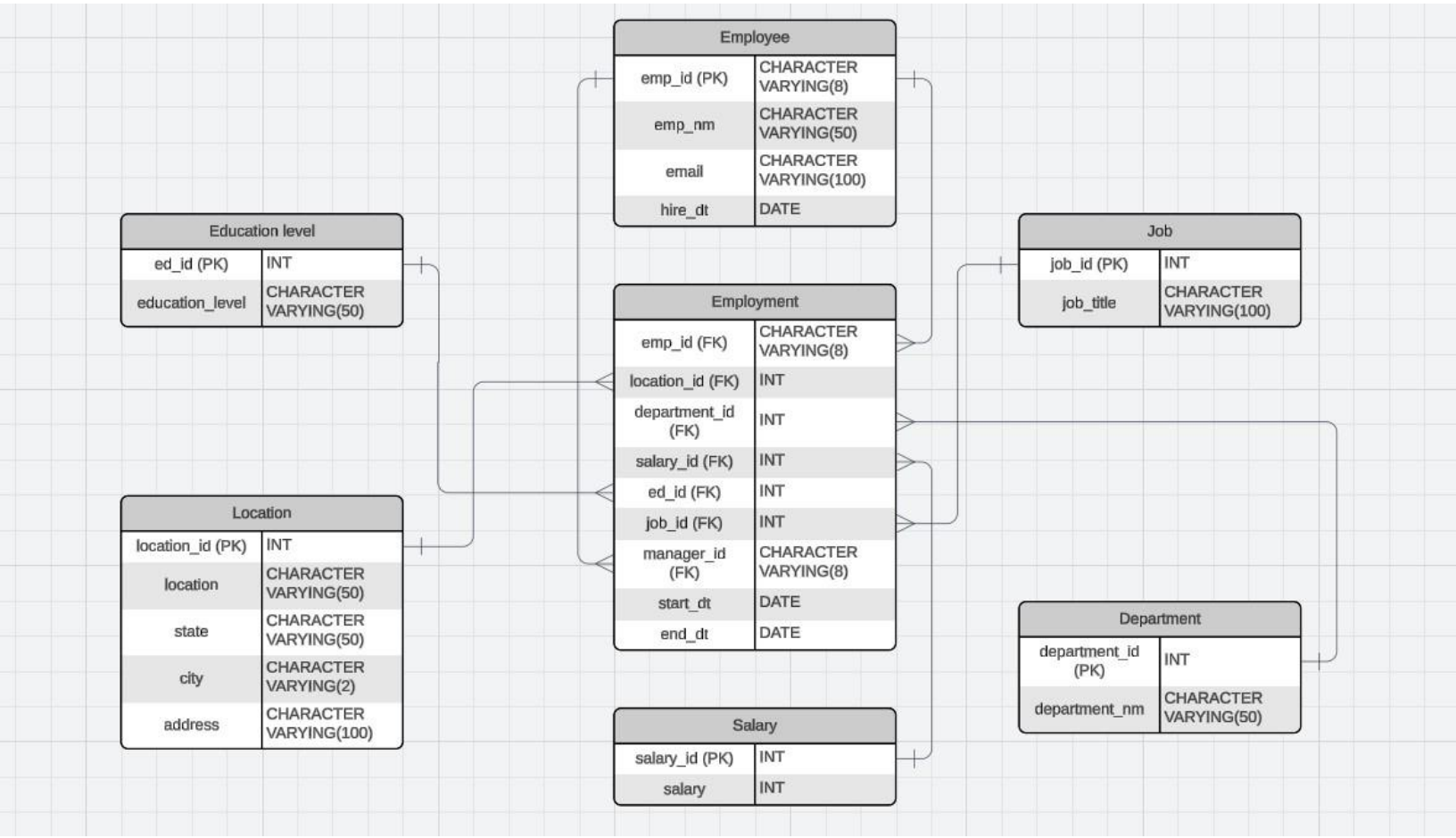
ERD

- **Logical**


Employee
emp_id
emp_nm
email
hire_dt
Education level
ed_id
education_level
Job
job_id
job_title
Employment
emp_id
location_id
department_id
salary_id
ed_id
job_id
manager_id
start_dt
end_dt
Location
location_id
location
state
city
address
Department
department_id
department_nm
Salary
salary_id
salary

ERD

- **Physical**



Step 3

Create A Physical Database

Step 3: Create A Physical Database

In this step, you will be turning your database model into a physical database.

You will:

- Create the database using SQL DDL commands
- Load the data into your database, utilizing flat file ETL
- Answer a series of questions using CRUD SQL commands to demonstrate your database was created and populated correctly

Submission
For this step, you will need to submit SQL files containing all DDL SQL scripts used to create the database.

You will also have to submit screenshots showing CRUD commands, along with results for each of the questions found in the starter template.

Hints
Your DDL script will be graded by running the code you submit. Please ensure your SQL code runs properly!

Foreign keys cannot be created on tables that do not exist yet, so it may be easier to create all tables in the database, then to go back and run modify statements on the tables to create foreign key constraints.

After running CRUD commands like update, insert, or delete, run a `SELECT*` command on the affected table, so the reviewer can see the results of the command.

DDL

Create a DDL SQL script capable of building the database you designed in Step 2

scratchpad.sql | StageTableLoad.sql

```sql
/*Create a DDL and DML SQL script capable of building the database I designed in the ERD*/

CREATE TABLE Employee (
    emp_id CHARACTER VARYING(8) PRIMARY KEY,
    emp_nm CHARACTER VARYING(50),
    email CHARACTER VARYING(100),
    hire_dt DATE);


INSERT INTO Employee(emp_id, emp_nm, email, hire_dt)
SELECT DISTINCT emp_id, emp_nm, email, hire_dt FROM proj_stg;


CREATE TABLE Job (
    job_id SERIAL PRIMARY KEY,
    job_title CHARACTER VARYING(100));

INSERT INTO Job(job_title)
SELECT DISTINCT job_title FROM proj_stg;


CREATE TABLE Department (
    department_id SERIAL PRIMARY KEY,
    department_nm CHARACTER VARYING(50));

INSERT INTO Department(department_nm)
SELECT DISTINCT department_nm FROM proj_stg;

CREATE TABLE Salary (
    salary_id SERIAL PRIMARY KEY,
    salary INTEGER);

INSERT INTO Salary(salary)
SELECT salary FROM proj_stg;

CREATE TABLE Location (
    location_id SERIAL PRIMARY KEY,
    location CHARACTER VARYING(50),
    state CHARACTER VARYING(2),
    city CHARACTER VARYING(50),
    address CHARACTER VARYING(100));

INSERT INTO Location(location, state, city, address)
SELECT DISTINCT location, state, city, address FROM proj_stg;

CREATE TABLE education_level (
    ed_id SERIAL PRIMARY KEY,
    education_level CHARACTER VARYING(50));

INSERT INTO education_level(education_level)
SELECT DISTINCT education_lvl FROM proj_stg;

```

DDL

```
53 CREATE TABLE Employment (
54     emp_id CHARACTER VARYING(8),
55     location_id INTEGER,
56     department_id INTEGER,
57     salary_id INTEGER,
58     ed_id INTEGER,
59     job_id INTEGER,
60     manager_id CHARACTER VARYING(8),
61     start_dt DATE,
62     end_dt DATE);
63
64
65
66 CREATE VIEW manager
67 AS SELECT a.emp_id AS manager_id,
68 ps.manager AS manager_name
69 FROM proj_stg AS ps
70 FULL JOIN (SELECT DISTINCT emp_id, emp_nm FROM proj_stg
71 WHERE emp_nm IN (SELECT DISTINCT manager FROM proj_stg)) AS a
72 ON ps.manager=a.emp_nm;
73
```

CRUD

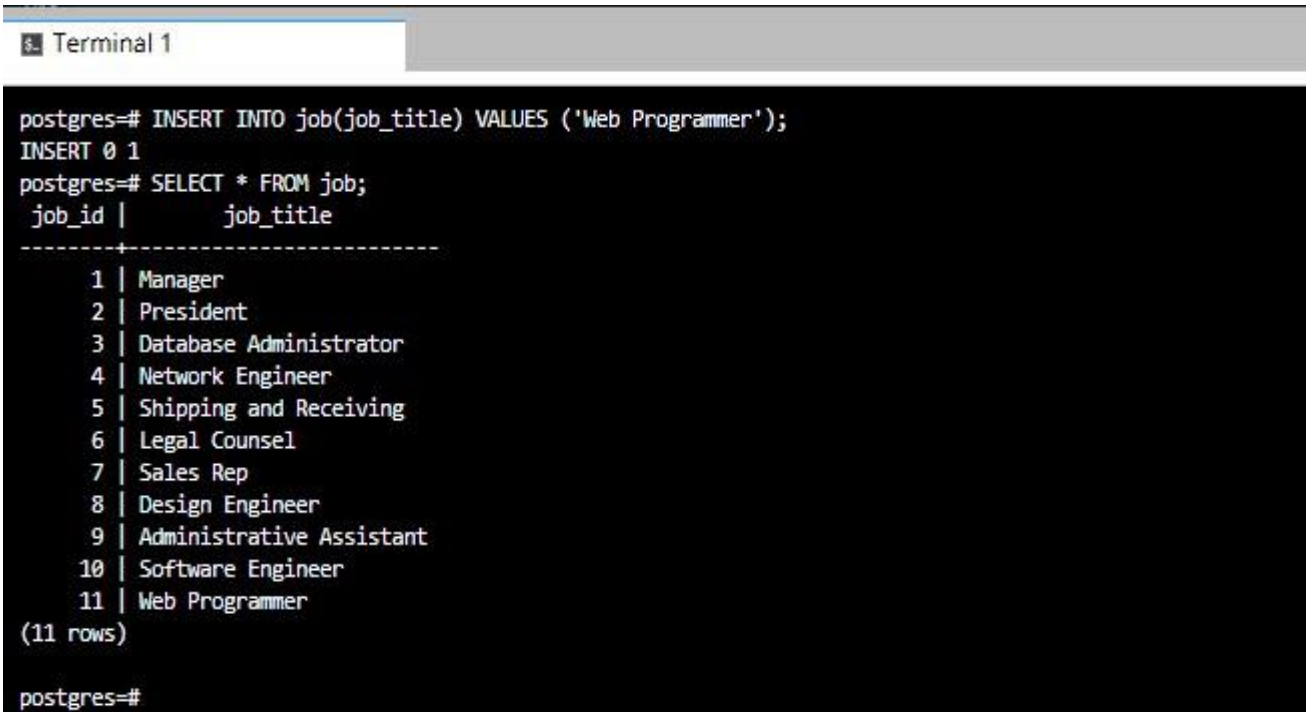
- **Question 1: Return a list of employees with Job Titles and Department Names**

Terminal 1

```
postgres=# SELECT e.emp_id, j.job_title, d.department_nm
postgres-# FROM employee AS e
postgres-# JOIN employment AS em
postgres-# ON e.emp_id = em.emp_id
postgres-# JOIN job AS j
postgres-# ON j.job_id = em.job_id
postgres-# JOIN department AS d
postgres-# ON d.department_id = em.department_id;
 emp_id |        job_title         |    department_nm
--------+--------------------------+--------------------
 E10033 | Software Engineer        | Product Development
 E10407 | Sales Rep                | Product Development
 E11678 | Network Engineer         | Product Development
 E11920 | Sales Rep                | Sales
 E12397 | Network Engineer         | Product Development
 E12562 | Administrative Assistant | Product Development
 E12890 | Software Engineer        | Product Development
 E13085 | Shipping and Receiving   | Distribution
 E13160 | Database Administrator   | IT
 E13160 | Network Engineer         | Product Development
 E13596 | Sales Rep                | Product Development
 E14737 | Shipping and Receiving   | Distribution
 E14913 | Network Engineer         | Product Development
 E15267 | Shipping and Receiving   | Distribution
 E15292 | Software Engineer        | IT
 E15292 | Shipping and Receiving   | Distribution
 E16276 | Sales Rep                | Sales
 E16346 | Administrative Assistant | Product Development
 E16678 | Database Administrator   | IT
 E16678 | Network Engineer         | IT
 E16995 | Sales Rep                | Product Development
 E17054 | President                | HQ
 E17372 | Sales Rep                | Sales
 E17469 | Administrative Assistant | Distribution
 E18659 | Software Engineer        | Product Development
 E18697 | Administrative Assistant | HQ
 E20101 | Sales Rep                | Sales
```

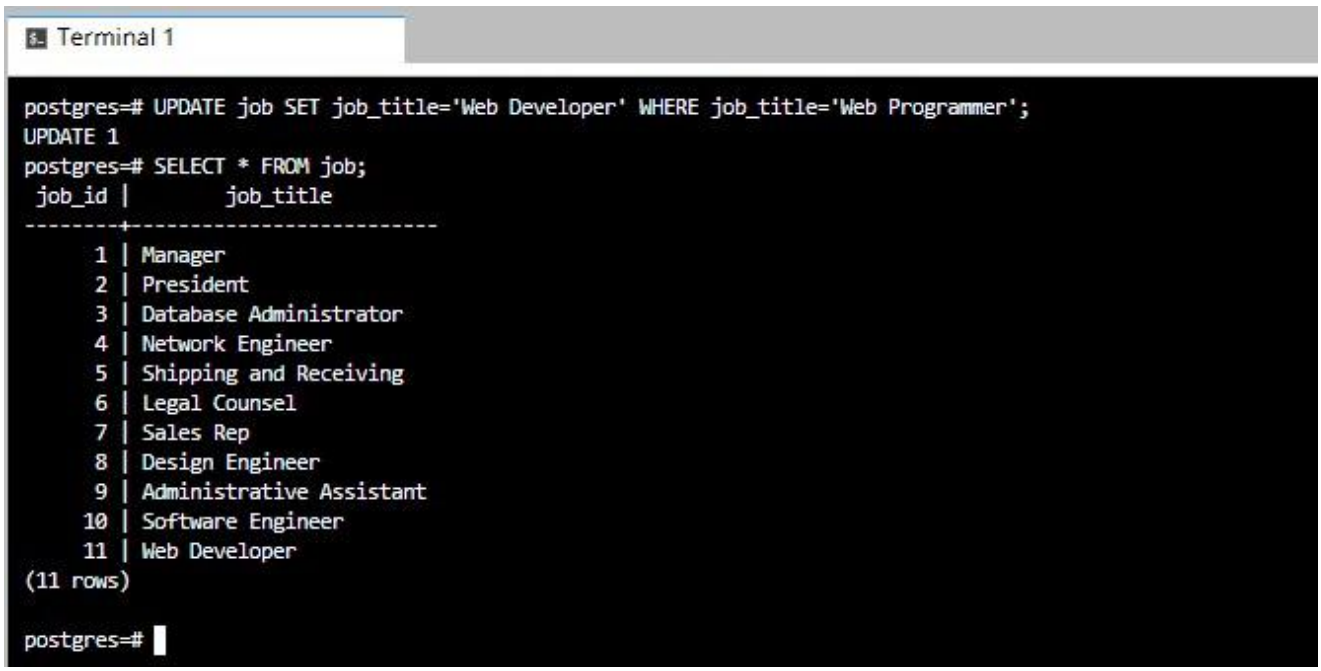
CRUD

- **Question 2: Insert Web Programmer as a new job title**



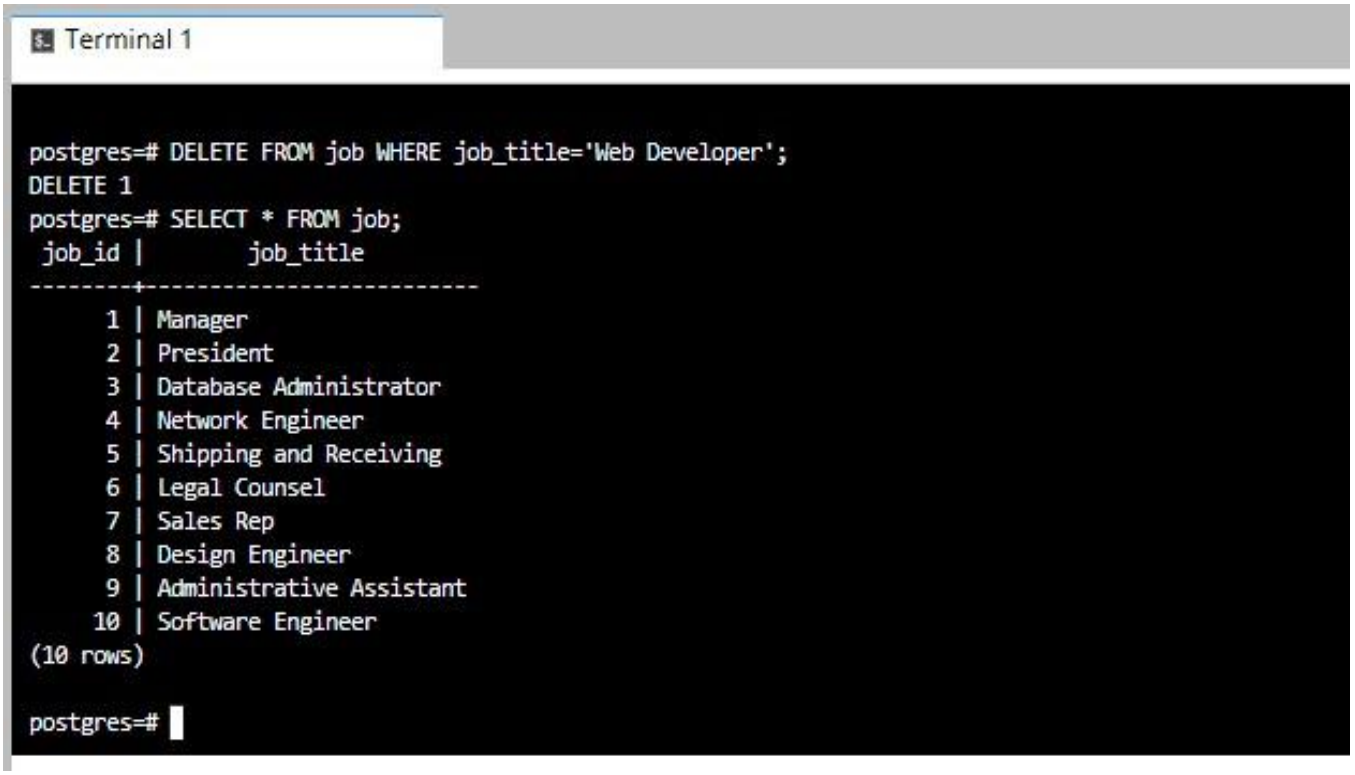
CRUD

- **Question 3: Correct the job title from web programmer to web developer**



CRUD

- **Question 4: Delete the job title Web Developer from the database**



CRUD

- **Question 5: How many employees are in each department?**

Terminal 1

```
postgres=# SELECT d.department_nm, COUNT(e.emp_id)
postgres-# FROM department AS d
postgres-# JOIN employment AS em
postgres-# ON d.department_id = em.department_id
postgres-# JOIN employee AS e
postgres-# ON e.emp_id = em.emp_id
postgres-# GROUP BY d.department_nm;
    department_nm    | count
---------------------+-------
 IT                  |    54
 Product Development |    70
 HQ                  |    13
 Distribution        |    27
 Sales               |    41
(5 rows)
```

CRUD

- **Question 6: Write a query that returns current and past jobs (include employee name, job title, department, manager name, start and end date for position) for employee Toni Lembeck.**

Terminal 1

```
postgres-# SELECT DISTINCT e.emp_nm, j.job_title, d.department_nm, s.manager, em.start_dt, em.end_dt
postgres-# FROM employee AS e
postgres-# JOIN employment AS em
postgres-# ON e.emp_id = em.emp_id
postgres-# JOIN department AS d
postgres-# ON d.department_id = em.department_id
postgres-# JOIN sub AS s
postgres-# ON s.manager_id = em.manager_id
postgres-# JOIN job AS j
postgres-# ON j.job_id = em.job_id
postgres-# WHERE e.emp_nm = 'Toni Lembeck';
    emp_nm    |       job_title        | department_nm |   manager    | start_dt  |   end_dt
--------------+------------------------+---------------+--------------+-----------+------------
 Toni Lembeck | Network Engineer       | IT            | Jacob Lauber | 1995-03-12 | 2001-07-18
 Toni Lembeck | Database Administrator | IT            | Jacob Lauber | 2001-07-18 | 2100-02-02
(2 rows)
```

CRUD

- **Question 7: Describe how you would apply table security to restrict access to employee salaries using an SQL server.**

To restrict access to employee salaries in SQL Server, we create a **NonManagement** role for users who should not access salary data. Next, we create a user **NoMgr** and assign them to this role. We grant **SELECT** permissions to necessary tables like **Employee**, **Department**, and **Location**, and explicitly deny **SELECT** access to the **Salary** table. This ensures **NoMgr** can view essential employee information but cannot access sensitive salary data.

Step 4

Above and Beyond

(optional)

Step 4: Above and Beyond

This last step is called Above and Beyond. In this step, I have proposed 3 challenges for you to complete, which are above and beyond the scope of the project. This is a chance to flex your coding muscles and show everyone how good you really are.

These challenge steps will bring your project even more in line with a real-world project, as these are the kind of “finishing touches” that will make your database more usable. Imagine building a car without air conditioning or turn signals. Sure, it will work, but who would want to drive it.

I encourage you to take on these challenges in this course and any future courses you take. I designed these challenges to be a challenge to your current abilities, but I ensured they are not an unattainable challenge. Remember, these challenges are completely optional - you can pass the project by doing none of them, or just some of them, but I encourage you to at least attempt them!

Standout Suggestion 1

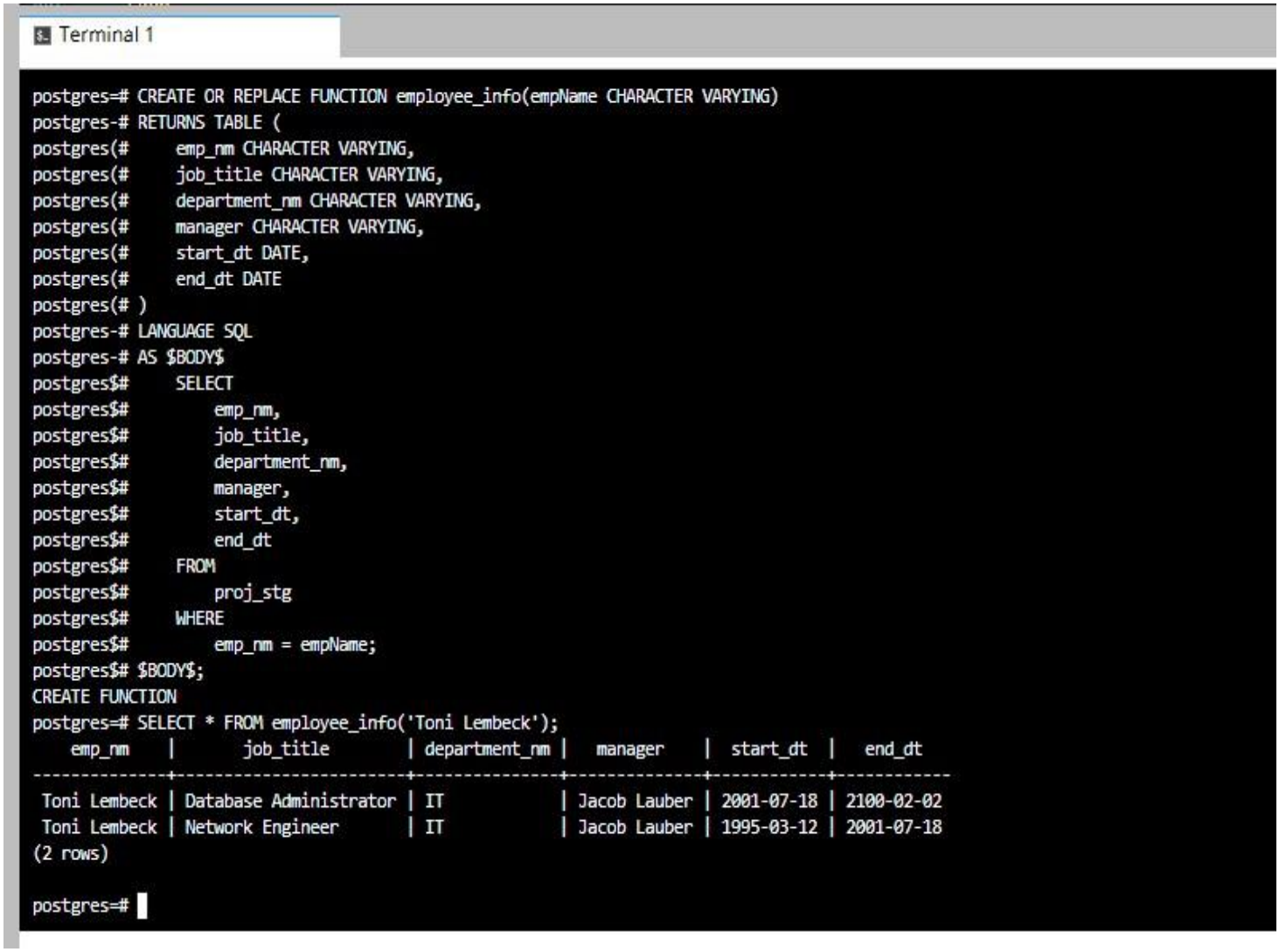
Create a view that returns all employee attributes; results should resemble initial Excel file

Terminal 1

```
postgres=# CREATE VIEW Employeefullinfoview AS SELECT e.emp_id,
postgres-# e.emp_nm,
postgres-# e.email,
postgres-# e.hire_dt,
postgres-# j.job_title,
postgres-# s.salary,
postgres-# d.department_nm,
postgres-# sub.manager,
postgres-# em.start_dt,
postgres-# em.end_dt,
postgres-# l.location,
postgres-# l.address,
postgres-# l.city,
postgres-# l.state,
postgres-# edu.education_level
postgres-# FROM employee AS e
postgres-# JOIN employment AS em
postgres-# ON e.emp_id = em.emp_id
postgres-# JOIN salary AS s
postgres-# ON s.salary_id = em.salary_id
postgres-# JOIN location AS l
postgres-# ON l.location_id = em.location_id
postgres-# JOIN (SELECT DISTINCT emp.emp_id AS manager_id, emp.emp_nm AS manager
postgres(# FROM employee AS emp
postgres(# JOIN employment AS empl
postgres(# ON emp.emp_id = empl.manager_id) AS sub
postgres-# ON sub.manager_id = em.manager_id
postgres-# JOIN job AS j
postgres-# ON j.job_id= j.job_id
postgres-# JOIN department AS d
postgres-# ON d.department_id=em.department_id
postgres-# JOIN education_level AS edu
postgres-# ON edu.ed_id = em.ed_id;
CREATE VIEW
postgres=# SELECT * FROM  Employeefullinfoview;
 emp_id |       emp_nm        |              email              | hire_dt    |       job_title        | salary |    department_nm    |      manager       | start_dt   |  end_dt    | location |      address       |
   city      | state |         education_level
--------+---------------------+---------------------------------+------------+------------------------+--------+---------------------+--------------------+------------+------------+----------+--------------------+--
------------+-------+-----------------------------------
 E10033 | Jermaine Massey     | Jermaine.Massey@TechCorp.com    | 2016-03-07 | Legal Counsel          | 111681 | Product Development | Conner Kinch       | 2016-03-07 | 2100-07-08 | HQ       | 1 Tech ABC Corp Way | D
allas       | TX    | Bachelors Degree
 E10033 | Jermaine Massey     | Jermaine.Massey@TechCorp.com    | 2016-03-07 | Sales Rep              | 111681 | Product Development | Conner Kinch       | 2016-03-07 | 2100-07-08 | HQ       | 1 Tech ABC Corp Way | D
allas       | TX    | Bachelors Degree
 E10033 | Jermaine Massey     | Jermaine.Massey@TechCorp.com    | 2016-03-07 | Design Engineer        | 111681 | Product Development | Conner Kinch       | 2016-03-07 | 2100-07-08 | HQ       | 1 Tech ABC Corp Way | D
allas       | TX    | Bachelors Degree
 E10033 | Jermaine Massey     | Jermaine.Massey@TechCorp.com    | 2016-03-07 | Manager                | 111681 | Product Development | Conner Kinch       | 2016-03-07 | 2100-07-08 | HQ       | 1 Tech ABC Corp Way | D
allas       | TX    | Bachelors Degree
 E10033 | Jermaine Massey     | Jermaine.Massey@TechCorp.com    | 2016-03-07 | Database Administrator | 111681 | Product Development | Conner Kinch       | 2016-03-07 | 2100-07-08 | HQ       | 1 Tech ABC Corp Way | D
allas       | TX    | Bachelors Degree
 E10033 | Jermaine Massey     | Jermaine.Massey@TechCorp.com    | 2016-03-07 | Network Engineer       | 111681 | Product Development | Conner Kinch       | 2016-03-07 | 2100-07-08 | HQ       | 1 Tech ABC Corp Way | D
allas       | TX    | Bachelors Degree
 E10033 | Jermaine Massey     | Jermaine.Massey@TechCorp.com    | 2016-03-07 | Software Engineer      | 111681 | Product Development | Conner Kinch       | 2016-03-07 | 2100-07-08 | HQ       | 1 Tech ABC Corp Way | D
allas       | TX    | Bachelors Degree
 E10033 | Jermaine Massey     | Jermaine.Massey@TechCorp.com    | 2016-03-07 | President              | 111681 | Product Development | Conner Kinch       | 2016-03-07 | 2100-07-08 | HQ       | 1 Tech ABC Corp Way | D
allas       | TX    | Bachelors Degree
 E10033 | Jermaine Massey     | Jermaine.Massey@TechCorp.com    | 2016-03-07 | Shipping and Receiving | 111681 | Product Development | Conner Kinch       | 2016-03-07 | 2100-07-08 | HQ       | 1 Tech ABC Corp Way | D
allas       | TX    | Bachelors Degree
 E10033 | Jermaine Massey     | Jermaine.Massey@TechCorp.com    | 2016-03-07 | Administrative Assistant | 111681 | Product Development | Conner Kinch       | 2016-03-07 | 2100-07-08 | HQ       | 1 Tech ABC Corp Way | D
allas       | TX    | Bachelors Degree
 E10033 | Jermaine Massey     | Jermaine.Massey@TechCorp.com    | 2016-03-07 | Web Programmer         | 111681 | Product Development | Conner Kinch       | 2016-03-07 | 2100-07-08 | HQ       | 1 Tech ABC Corp Way | D
allas       | TX    | Bachelors Degree
```

Standout Suggestion 2

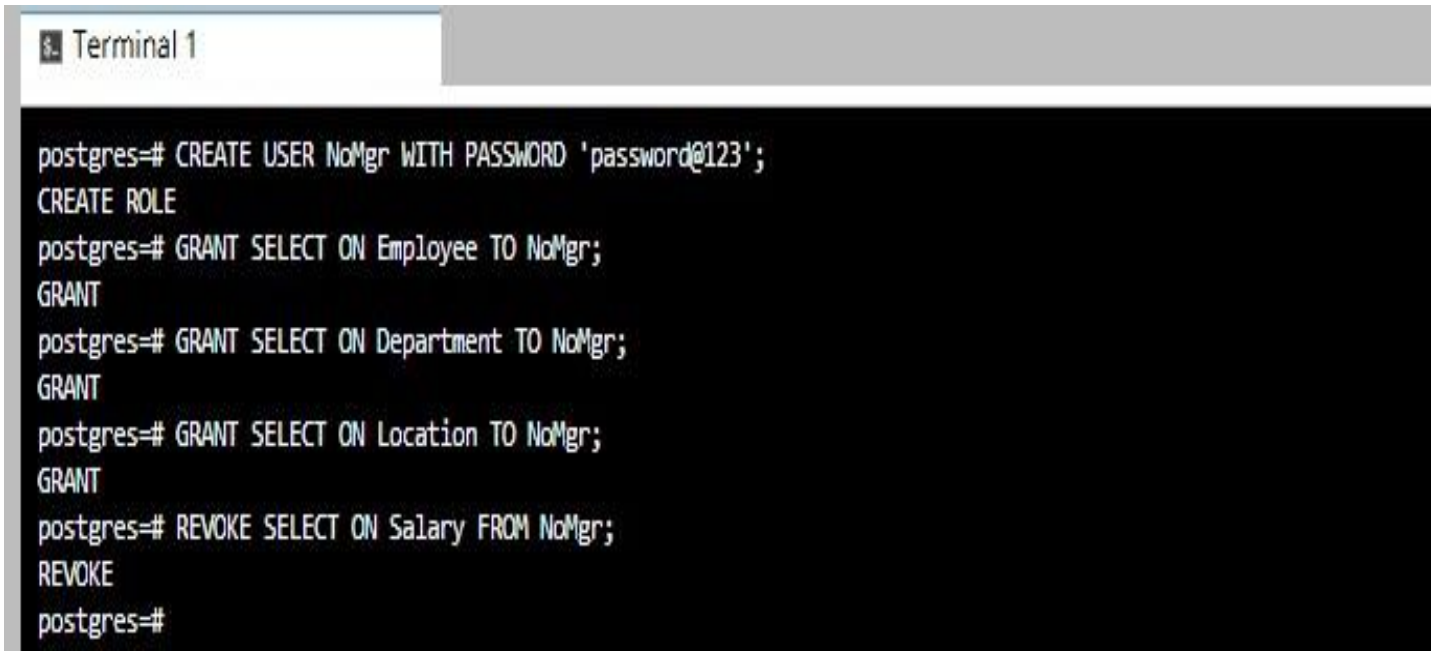
Create a stored procedure with parameters that returns current and past jobs (include employee name, job title, department, manager name, start and end date for position) when given an employee name.



Standout Suggestion 3

Implement user security on the restricted salary attribute.

I try to grant the privilege only for those table that don’t contain salary amount.



Appendix