Tech ABC Corp - HR Database

[Do Thi Kim Uyen & 12-Apr-2023]

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

  Answer: Tech ABC Corps has dramatically expanded their resources with its new AI powered video game in just within 6 months. Also because of the plan to open office in 4 new locations in a year, the HR team is in trouble managing the employment data. Therefore, the Tech ABC Corp HR department request an HR database design to in capable of keeping up with the growth.

- **Describe current data management solution:** What is the current method data storage/management

  Answer: All the employee information is stored in a shared spreadsheet.

- **Describe current data available:** What data does the business currently have available

Answer: The shared excel workbook has 206 records, witth 11 columns. The dataset is in denorrmalized form. The data contains information such as employee name, email, job title, department, manager's name, hire date, start date, end date, work location and salary.

Data Architect Business Requirement

- **Additional data requests:** Does the user have future data requests

  Answer: Future request is to integrate with payroll system. So the amployee attendance and paid time off information can be accessed from HR department in the future.

- **Who will own/manage data:** What department will own / manage the data in the database

  Answer: Some managers in HR department

- **Who will have access to database:** List user types that will have access; also list any restrictions to access.

  Answer: -Employees, may have read accessed to database, but must not access to the salary information.

  - HR, may no restriction with some managers, and maybe some restriction for lower level in HR department.

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.

  Answer: Currently the data just had 205 records, and related to 15 fields of information.

- **Estimated annual growth:** List any expected growth to the data

  Answer: Expected grouth every year is 20% for the next 5 years

- **Is any of the data sensitive/restricted:** List any data that may be sensitive or restricted from particular users

  Answer: Salary data is definitely is the most sensitive data, needed to be in good secure.

Data Architect Technical Requirement

- **Justification for the new database:** Provide at least two justifications for building a database
  Answer:

  - 20% growth expected a year in human resources for the next 5 years.
  - Data security and data integrity can be resolved.

- **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.

  Answer: Tables in the database and its columns
  - Employment (employee_id, location_id, job_id, department_id, salary_id, education_level_id, start_date, end_date)
  - Employee (employee_id, employee_name, email, hire_date, manager_id, manager_name)
  - Job (job_id, job_title)
  - Department (department_id, department)
  - Location (location_id, location, address, city, state)
  - Salary (salary_id, salary)
  - Education Level (education_level_id, education_level)

Data Architect Technical Requirement

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

  Answer: ETL is the appropriate ingestion technique to transform the flat files. We can create an automated ETL process to ingest data into the database from excel spreadsheet.

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

  Ownership: who will own and maintain the data

  → Answer: HR

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

  → Answer: -Employees, may have read accessed to database, but must not access to the salary information.

  - HR, has the write access.

Data Architect Technical Requirement

- **Scalability:** Should replication or sharding be used to ensure scalability based on user needs

  Answer: Replication should be used in this case. As the growth of the company , the number of employees will be expanded, and the high amount of employees need access to database. The high access to the database will slow down the database query performance. So need to replica the database to distribute load the database and improve the query perfomance.

- **Flexibility:** Describe measures taken to ensure future data integration if needed

  Answer: ETL could be suitable in the future to integrate with the database from payroll system.

Data Architect Technical Requirement

- **Storage & retention**

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

  → Answer: Disk based storage should be used to decrease the cost. And also, this database is just to store the data for futher management, not require for any analytical purposes, so we don't need an on-premise database for easily scale up and down for good query performance.

  Retention: how long does the data have to be kept for?
  → Answer: As per federal regulations, the employee information need to be kept for 7 years.

- **Backup:** IT Best Practices document lists Backup schedule requirements.

  Answer: As the head of HR department needs the live database, that she can often input and edit information, so we need to incremental backup daily, and a full backup 1X per week. And also, need to backup the changes as often as possible.

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.

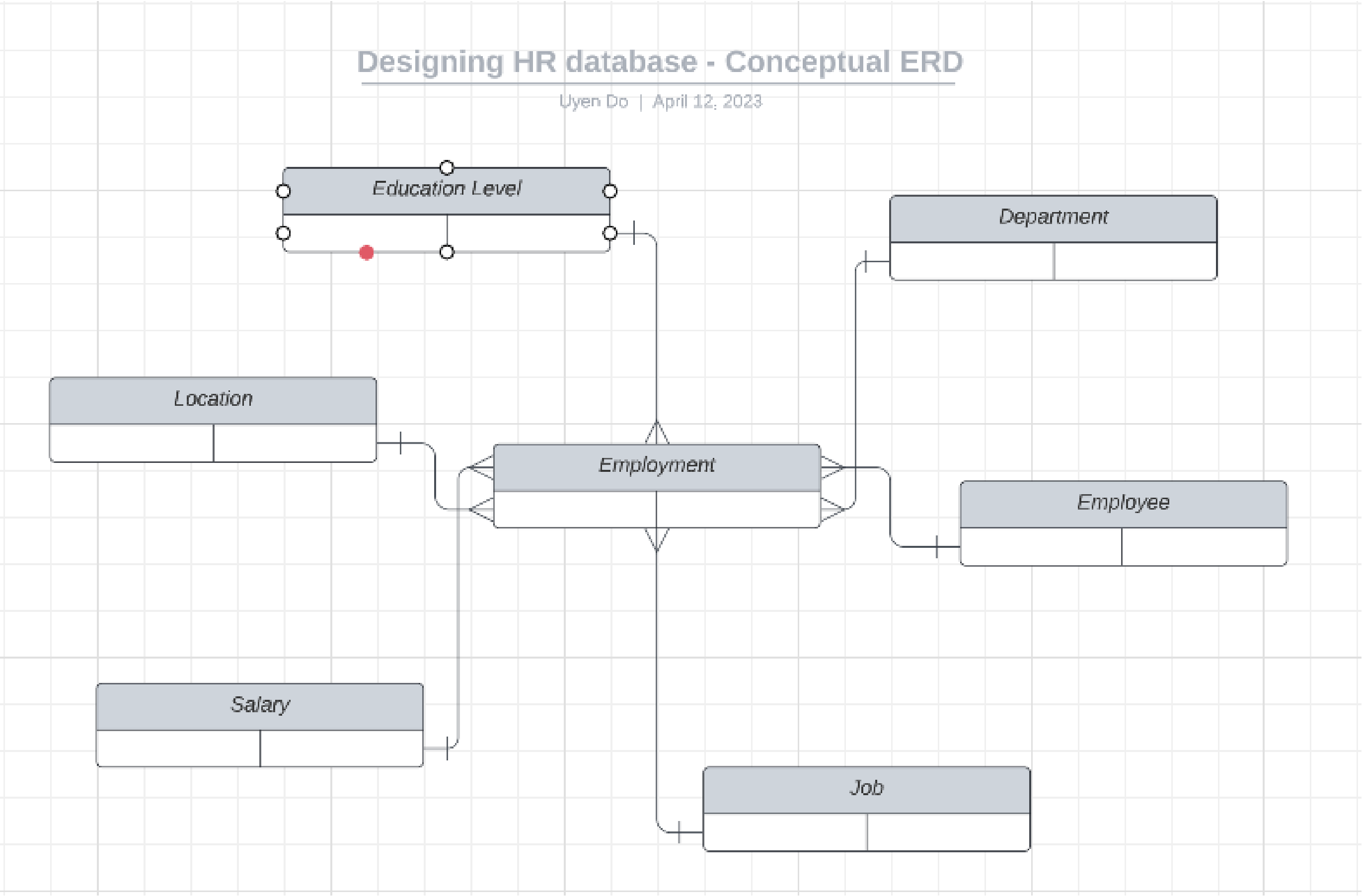
You will submit a screenshot for each of the 3 ERDs you create. You will find detailed instructions for developing each of the ERDs over the next several pages.

ERD

- ## Conceptual

This is the most general level of data modeling. At the conceptual level, you should be thinking about creating entities that represent business objects for the database. Think broadly here. Attributes (or column names) are not required at this point, but relationship lines are required (although Crow's foot notation is not needed at this level). Create at least three entities for this model; thinking about the 3NF will aid you in deciding the type of entities to create.

Use Lucidchart’s built-in template for DBMS ER Diagram UML.

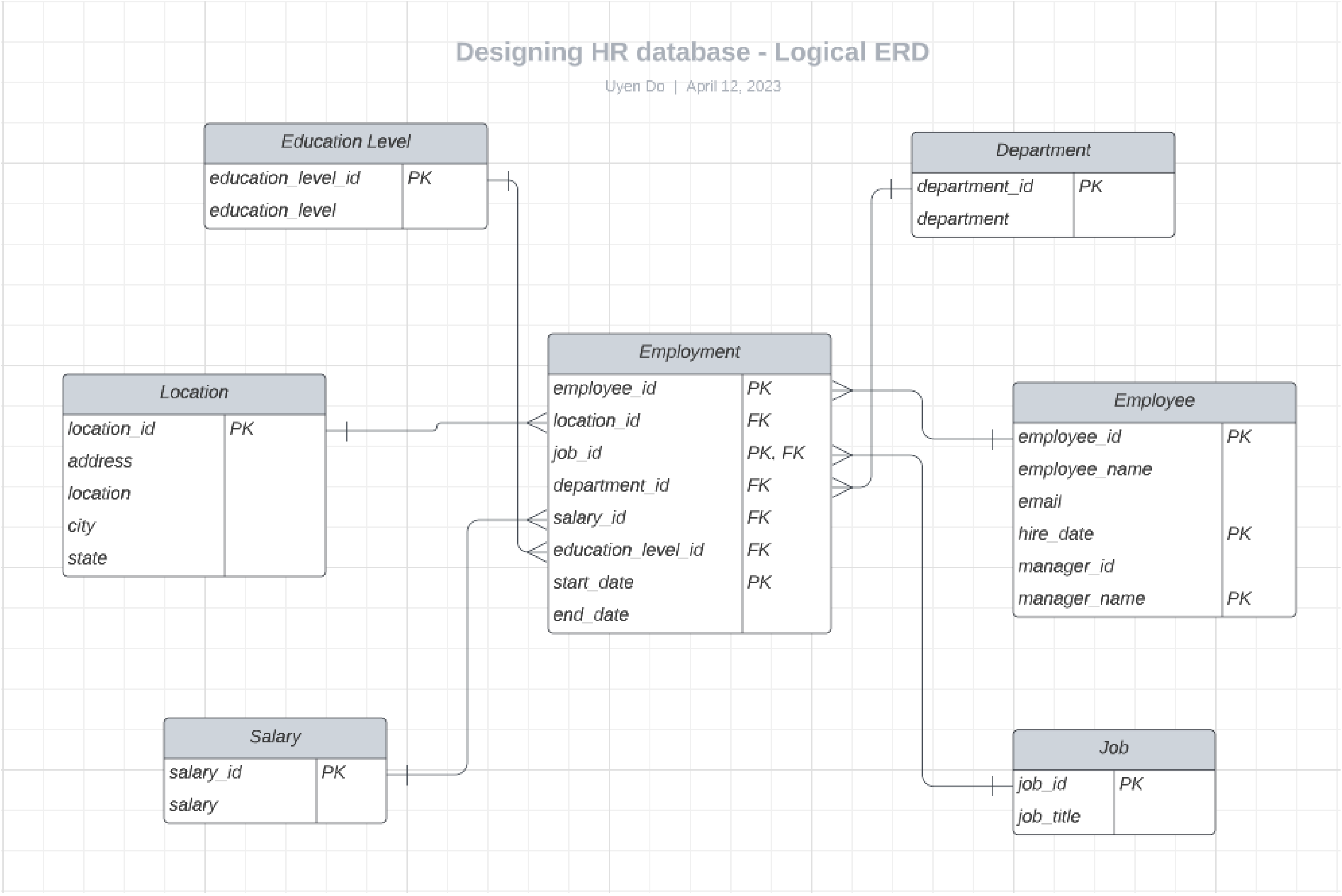


ERD

- **Logical**

The logical model is the next level of refinement from the conceptual ERD. At this point, you should have normalized the data to the 3NF. Attributes should also be listed now in the ERD. You can still use human-friendly entity and attribute names in the logical model, and while relationship lines are required, Crow's foot notation is still not needed at this point.

Use Lucidchart’s built-in template for DBMS ER Diagram UML.



Designing HR database - Logical ERD

Uyen Do | April 12, 2023

Education Level	
education_level_id	PK
education_level	

Department	
department_id	PK
department	

Location	
location_id	PK
address	
location	
city	
state	

Employment	
employee_id	PK
location_id	FK
job_id	PK, FK
department_id	FK
salary_id	FK
education_level_id	FK
start_date	PK
end_date	

Employee	
employee_id	PK
employee_name	
email	
hire_date	PK
manager_id	
manager_name	PK

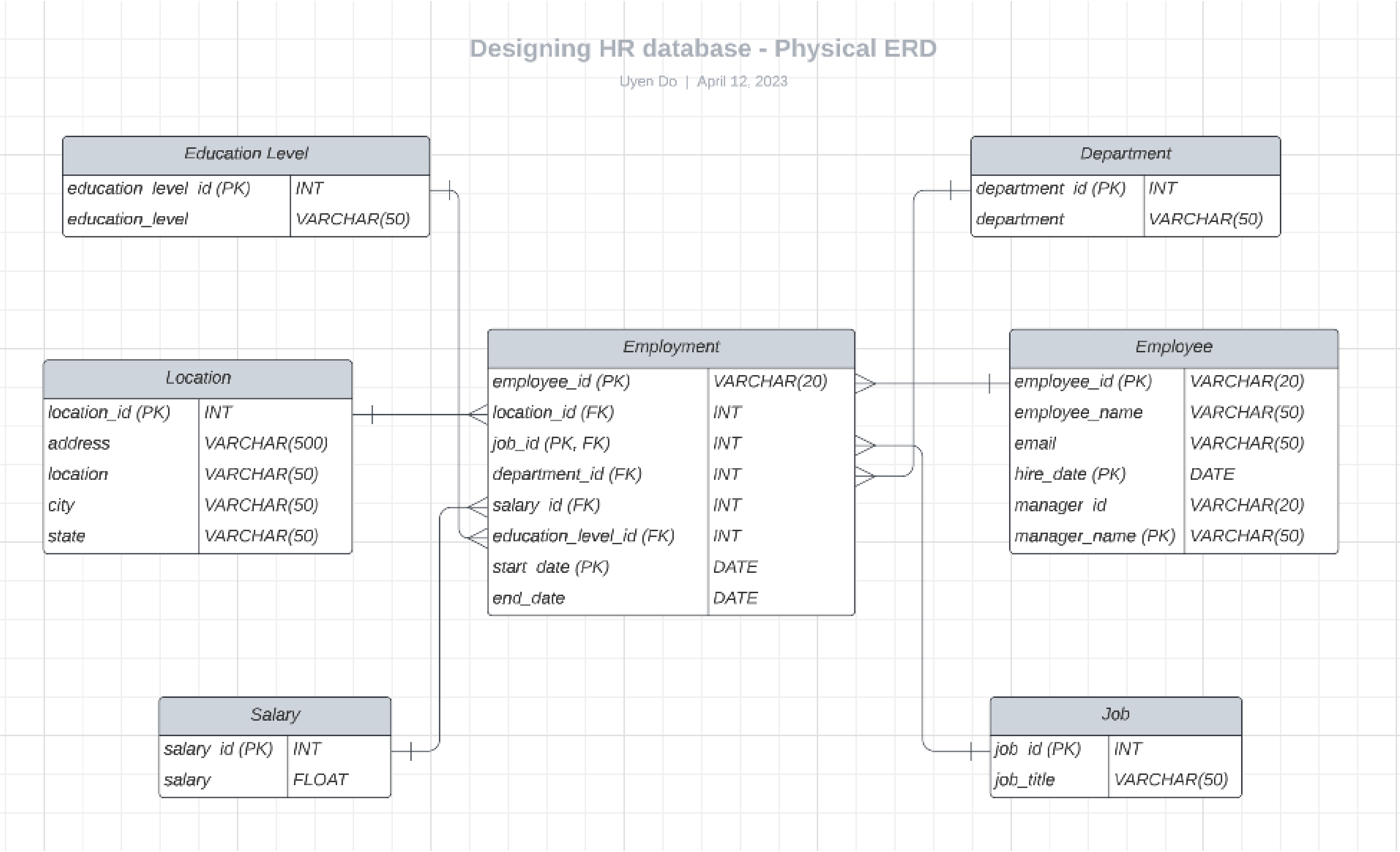
Salary	
salary_id	PK
salary	

Job	
job_id	PK
job_title	

ERD

- ## Physical

The physical model is what will be built in the database. Each entity should represent a database table, complete with column names and data types. Primary keys and foreign keys should also be represented here. Primary keys should be in bold type with the (PK) designation following the field name. Foreign keys should be in normal type face, but have the designation (FK) after the column name. Finally, in the physical model, Crow's foot notation is important.



Designing HR database - Physical ERD

Uyen Do | April 12, 2023

Education Level	
education_level_id (PK)	INT
education_level	VARCHAR(50)

Department	
department_id (PK)	INT
department	VARCHAR(50)

Location	
location_id (PK)	INT
address	VARCHAR(500)
location	VARCHAR(50)
city	VARCHAR(50)
state	VARCHAR(50)

Employment	
employee_id (PK)	VARCHAR(20)
location_id (FK)	INT
job_id (PK, FK)	INT
department_id (FK)	INT
salary_id (FK)	INT
education_level_id (FK)	INT
start_date (PK)	DATE
end_date	DATE

Employee	
employee_id (PK)	VARCHAR(20)
employee_name	VARCHAR(50)
email	VARCHAR(50)
hire_date (PK)	DATE
manager_id	VARCHAR(20)
manager_name (PK)	VARCHAR(50)

Salary	
salary_id (PK)	INT
salary	FLOAT

Job	
job_id (PK)	INT
job_title	VARCHAR(50)

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.

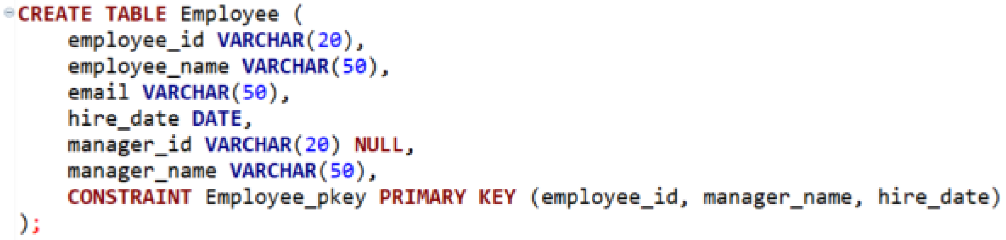
DDL

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

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

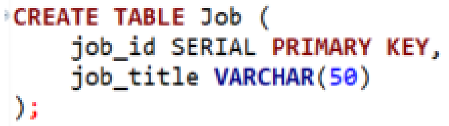
Foreign keys cannot be created on tables that do not exist yet, so it may be easier to create all tables in the database, then to go back and run modify statements on the tables to create foreign key constraints.

- **Employee Table:**



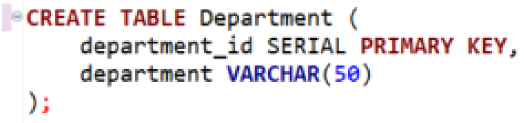
```
CREATE TABLE Employee (
    employee_id VARCHAR(20),
    employee_name VARCHAR(50),
    email VARCHAR(50),
    hire_date DATE,
    manager_id VARCHAR(20) NULL,
    manager_name VARCHAR(50),
    CONSTRAINT Employee_pkey PRIMARY KEY (employee_id, manager_name, hire_date)
);
```

- **Job Table:**



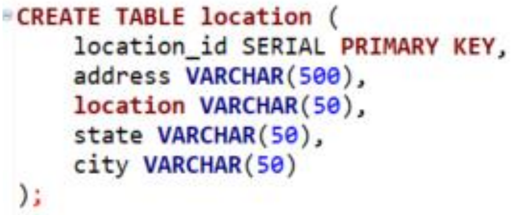
```
CREATE TABLE Job (
    job_id SERIAL PRIMARY KEY,
    job_title VARCHAR(50)
);
```

- **Department Table:**



```
CREATE TABLE Department (
    department_id SERIAL PRIMARY KEY,
    department VARCHAR(50)
);
```

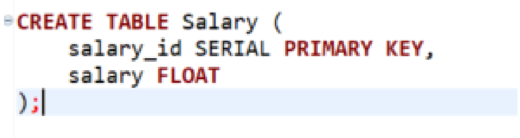
- **Location Table:**



```
CREATE TABLE location (
    location_id SERIAL PRIMARY KEY,
    address VARCHAR(500),
    location VARCHAR(50),
    state VARCHAR(50),
    city VARCHAR(50)
);
```

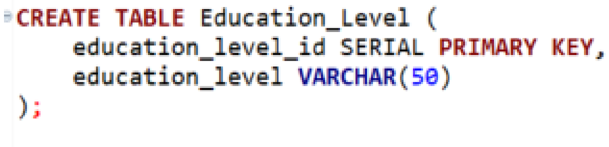
DDL

- **Salary Table:**



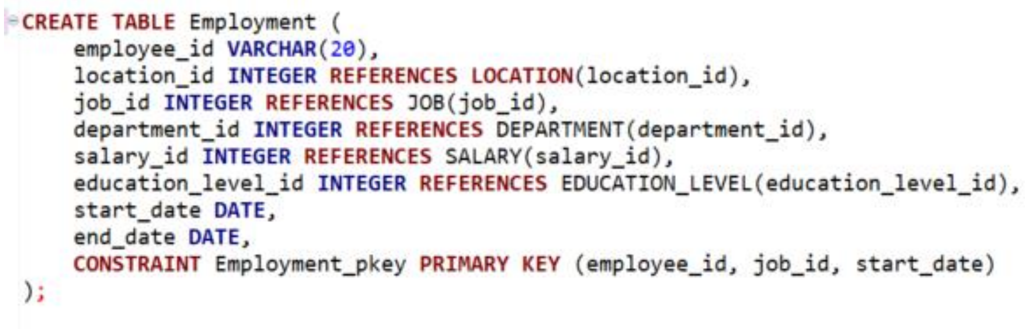
```
CREATE TABLE Salary (
    salary_id SERIAL PRIMARY KEY,
    salary FLOAT
);
```

- **Educational Level Table:**



```
CREATE TABLE Education_Level (
    education_level_id SERIAL PRIMARY KEY,
    education_level VARCHAR(50)
);
```

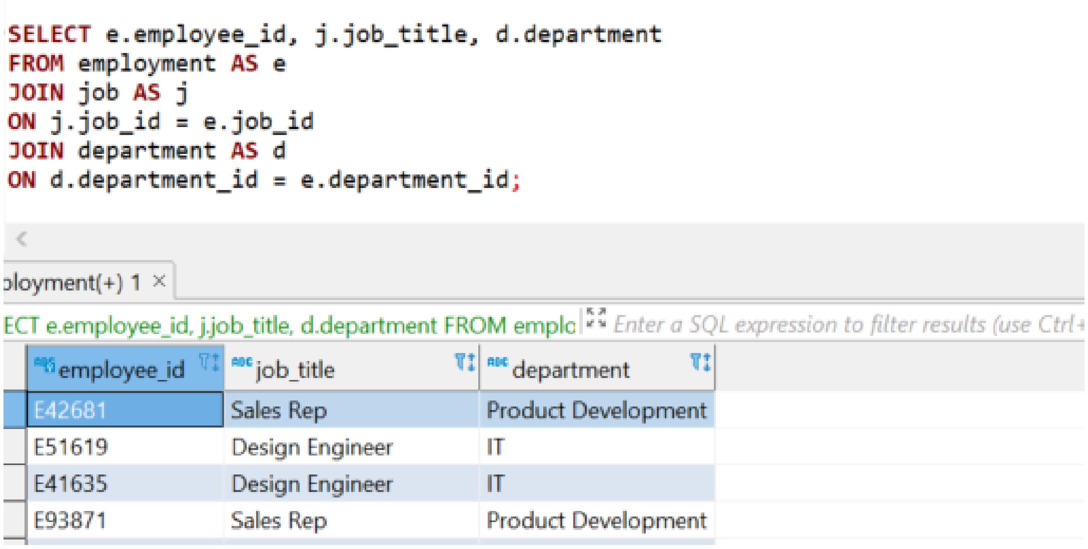
- **Employment Table:**



```
CREATE TABLE Employment (
    employee_id VARCHAR(20),
    location_id INTEGER REFERENCES LOCATION(location_id),
    job_id INTEGER REFERENCES JOB(job_id),
    department_id INTEGER REFERENCES DEPARTMENT(department_id),
    salary_id INTEGER REFERENCES SALARY(salary_id),
    education_level_id INTEGER REFERENCES EDUCATION_LEVEL(education_level_id),
    start_date DATE,
    end_date DATE,
    CONSTRAINT Employment_pkey PRIMARY KEY (employee_id, job_id, start_date)
);
```

CRUD

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



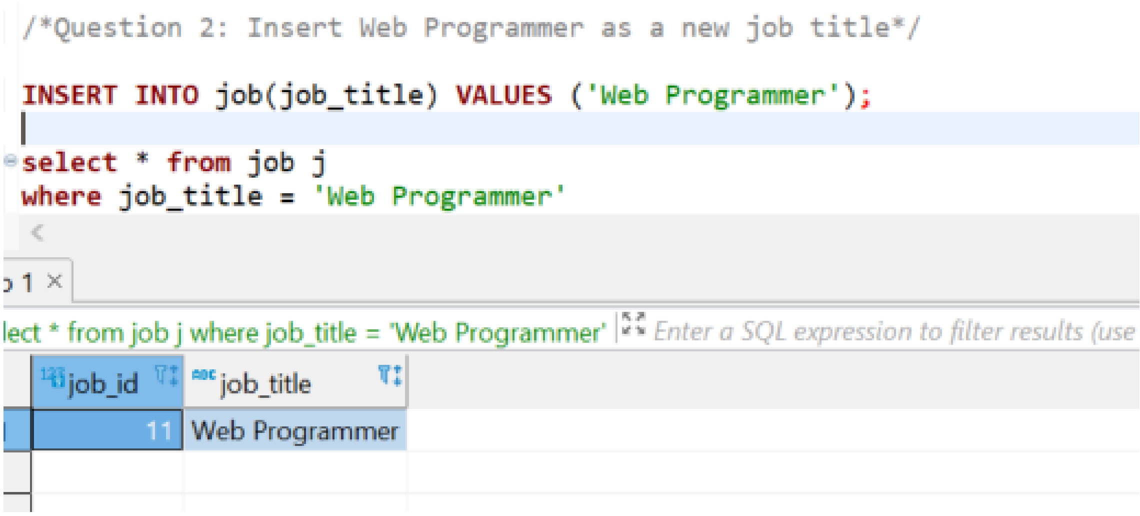
```
SELECT e.employee_id, j.job_title, d.department
FROM employment AS e
JOIN job AS j
ON j.job_id = e.job_id
JOIN department AS d
ON d.department_id = e.department_id;
```

ployment(+) 1 ×
ECT e.employee_id, j.job_title, d.department FROM emplo
Enter a SQL expression to filter results (use Ctrl+

employee_id	job_title	department
E42681	Sales Rep	Product Development
E51619	Design Engineer	IT
E41635	Design Engineer	IT
E93871	Sales Rep	Product Development

CRUD

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



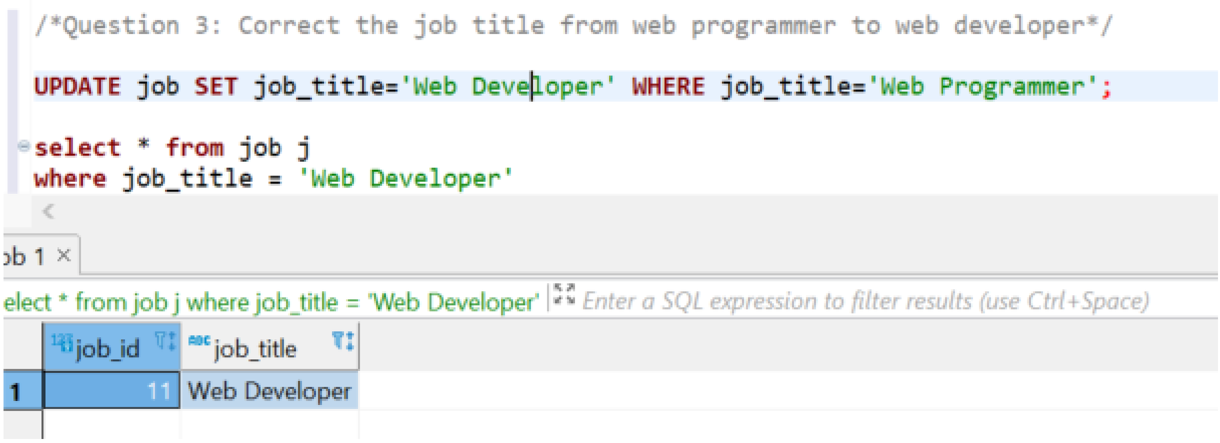
```
/*Question 2: Insert Web Programmer as a new job title*/

INSERT INTO job(job_title) VALUES ('Web Programmer');

select * from job j
where job_title = 'Web Programmer'
```

```
lect * from job j where job_title = 'Web Programmer'
```

job_id	job_title
11	Web Programmer

CRUD

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



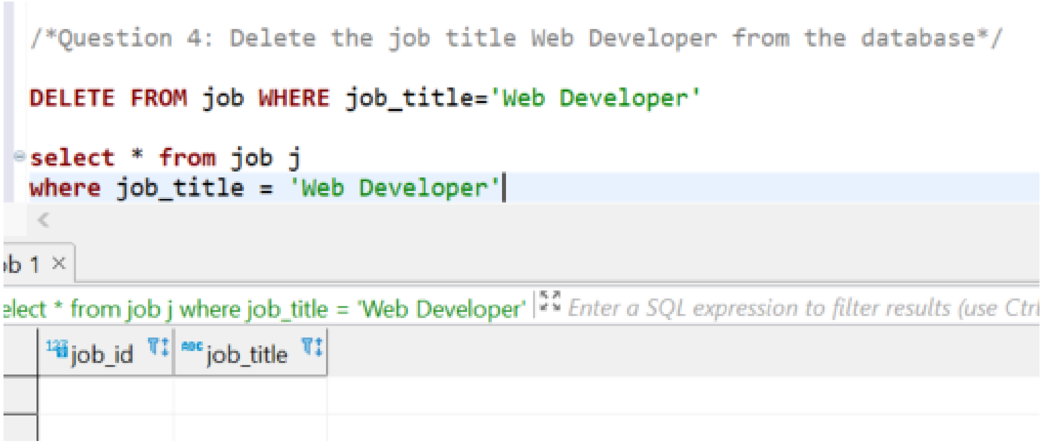
```
/*Question 3: Correct the job title from web programmer to web developer*/

UPDATE job SET job_title='Web Developer' WHERE job_title='Web Programmer';

select * from job j
where job_title = 'Web Developer'
```

elect * from job j where job_title = 'Web Developer'

	job_id	job_title
1	11	Web Developer

CRUD

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



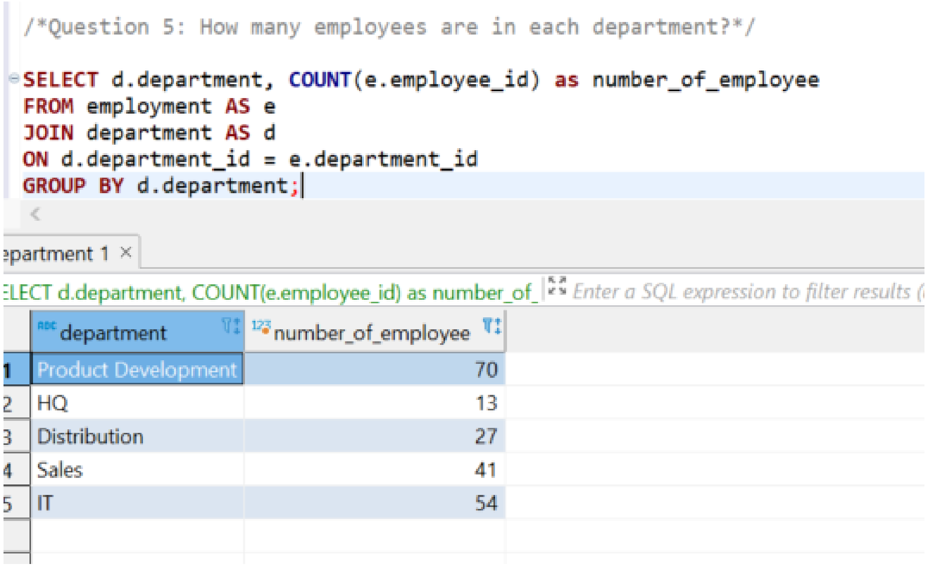
```
/*Question 4: Delete the job title Web Developer from the database*/

DELETE FROM job WHERE job_title='Web Developer'

select * from job j
where job_title = 'Web Developer'
```

```
elect * from job j where job_title = 'Web Developer'
```

job_id	job_title

CRUD

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



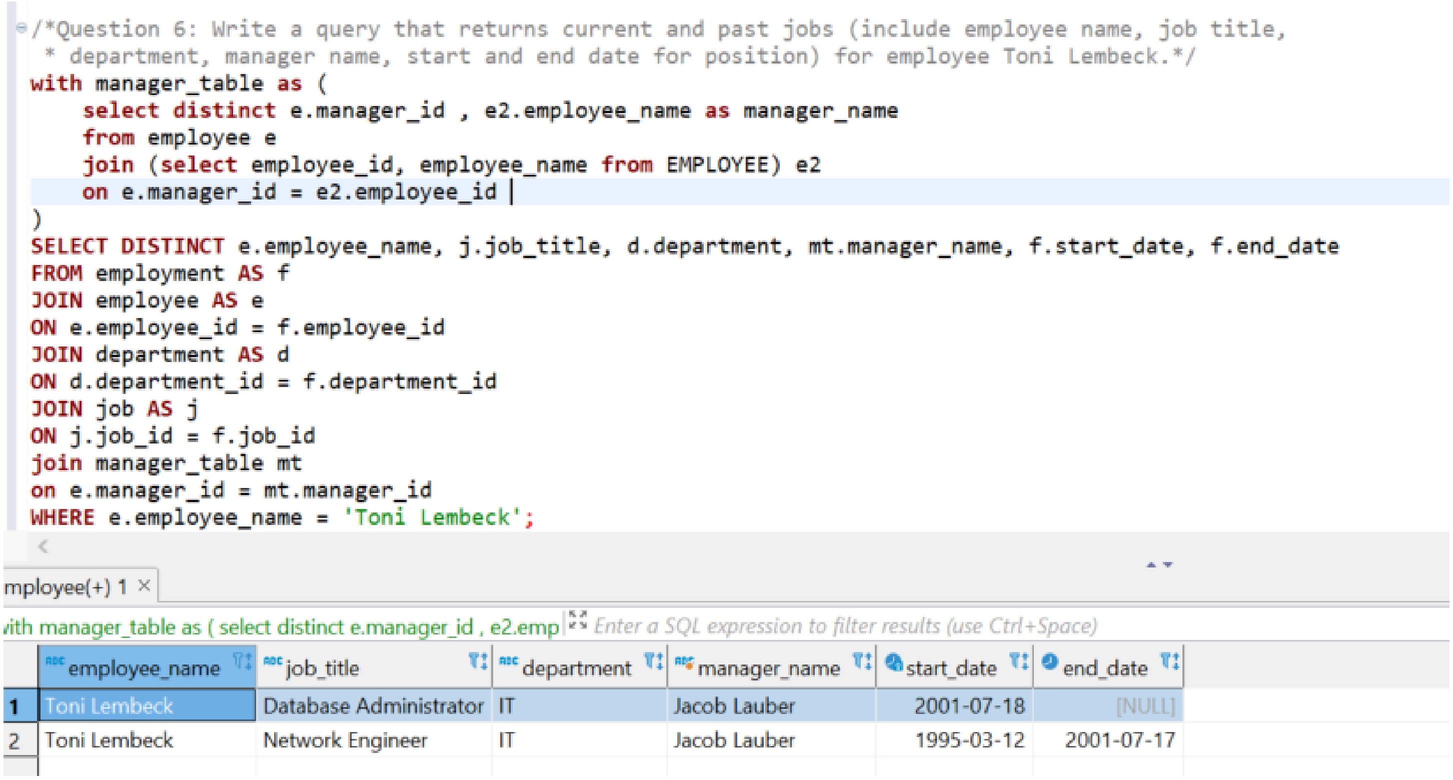
```
/*Question 5: How many employees are in each department?*/

SELECT d.department, COUNT(e.employee_id) as number_of_employee
FROM employment AS e
JOIN department AS d
ON d.department_id = e.department_id
GROUP BY d.department;
```

```
ELECT d.department, COUNT(e.employee_id) as number_of_
```

	department	number_of_employee
1	Product Development	70
2	HQ	13
3	Distribution	27
4	Sales	41
5	IT	54

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.**



```
/*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.*/
with manager_table as (
    select distinct e.manager_id , e2.employee_name as manager_name
    from employee e
    join (select employee_id, employee_name from EMPLOYEE) e2
    on e.manager_id = e2.employee_id
)
SELECT DISTINCT e.employee_name, j.job_title, d.department, mt.manager_name, f.start_date, f.end_date
FROM employment AS f
JOIN employee AS e
ON e.employee_id = f.employee_id
JOIN department AS d
ON d.department_id = f.department_id
JOIN job AS j
ON j.job_id = f.job_id
join manager_table mt
on e.manager_id = mt.manager_id
WHERE e.employee_name = 'Toni Lembeck';
```

mployee(+) 1 ×
ith manager_table as (select distinct e.manager_id , e2.emp
Enter a SQL expression to filter results (use Ctrl+Space)

	employee_name	job_title	department	manager_name	start_date	end_date
1	Toni Lembeck	Database Administrator	IT	Jacob Lauber	2001-07-18	[NULL]
2	Toni Lembeck	Network Engineer	IT	Jacob Lauber	1995-03-12	2001-07-17

CRUD

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

  Answer: The company has an domain for employee to log in. This domain will authenticate the employee, and limit their access to any database or tables. In this case, we can revoke user access of the employees from viewing the Salary table.

Appendix

Additional Info