Tech ABC Corp - HR Database

[Tran Thi Anh Hong & March 30, 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:**
  Tech ABC Corp is experiencing significant growth with an increase in headcount from 10 to 200 in only 6 months. It's posing challenges in managing employee information while the current method would not help the organization overcome and satisfy new expectations. Thereby, the new database design or a DBMS (Database Management System) solution would be beneficial to the company in achieving data integrity and data security.

- **Describe current data management solution:**

  The company is in fact using a shared spreadsheet to maintain all employee information.

- **Describe current data available:**

  An HR file containing employee data in a human-readable format 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.

Data Architect Business Requirement

- Additional data requests:
  - Live database
  - Maintain data for at least 7 years
  - Backed up properly
  - Connect with the payroll department's system in the future
- Who will own/manage data

  HR department
- Who will have access to database
  - Employees with domain login: read only, no permission to salary information
  - Management and HR employees: write access, salary information permitted

Data Architect Business Requirement

- **Estimated size of database**

  Depending on the memo from Sarah Collins and HR data in the excel file, estimated row of some tables below:

  - Employee: 200 rows
  - Location: 6 rows

  So database size is estimated up from around 206 rows

- **Estimated annual growth**

  Size of database is expected to grow 20% a year for the next 5 years in corresponding to the expected growth in employee size

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

  Salary information is restricted to HR and management employees only

Data Architect Technical Requirement

- **Justification for the new database**
  - Ensure data integrity and data consistency as there are many more employees contribute to HR data now and it would cause many chances in duplicated information, different information for one employee,...
  - Ensure data security that prevent unauthorized person from accessing the data and make unexpected changes.
  - Allow to connect with other system from other department in the future.
  - Save time to get information needed as information are now stored in a structured database.

Data Architect Technical Requirement

- **Database objects**

  Entities:

  - Employee: basic information of employee such as name, email, hired date and education level
  - Department: department information
  - Location: regions where all offices located
  - City: cities where all offices located
  - State: states where all offices located
  - Position: position history of all employees as one employee can be able to move from a job title to another one in the company
  - Salary: Salary amount

  View:

  - Full information of employees in human-readable format

  Function/Store procedure:

  - Get full information of employee by giving function/stored procedure employee name

- **Data ingestion**

  ETL is best practice as of now to ingest HR data to database from excel file.

  Direct feed may be applied in the future as *it allows the two systems (HR and Payroll departments) being able to interface directly.*

Data Architect Technical Requirement

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

  Ownership: Sarah Collins (Head of HR) will own and maintain the data

  User Access:
  - Employee (with domain login): read access
  - Management and HR employees: read and write access
  - Other employees without domain login: no access

- **Scalability**

  Because 90% of users have read access as estimation, so replicated database is best practice to speed up reading capability for that user amount.

- **Flexibility**

  To ensure future data integration if needed, we can consider these measurements:

  - Type of users with different needs, for example:
    - Employees only needs to view their personal information
    - HR manager needs aggregate data to make report
    - HR employees needs full HR data even from other department to update employee info

Data Architect Technical Requirement

- **Storage & retention**

  Storage: store database in **spinning disk** because there's no advanced computation required

  Retention: data needs to be kept for at least 7 years

- **Backup**

  Since HR data is considered business critical data, so critical backup plan is required as follow:

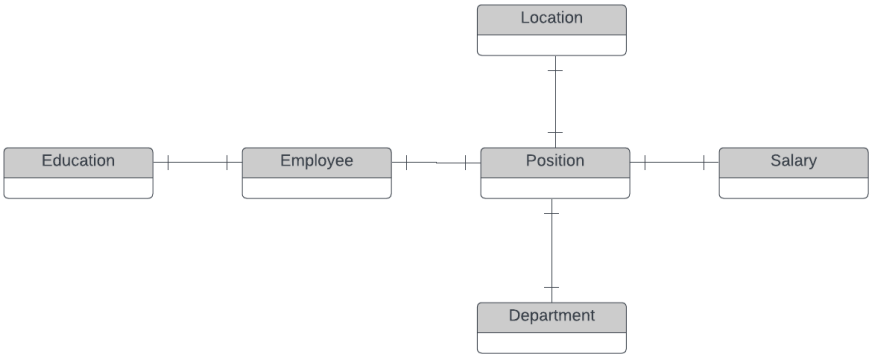
  - Back up schedule is full backup 1x per week, incremental backup daily

Step 2

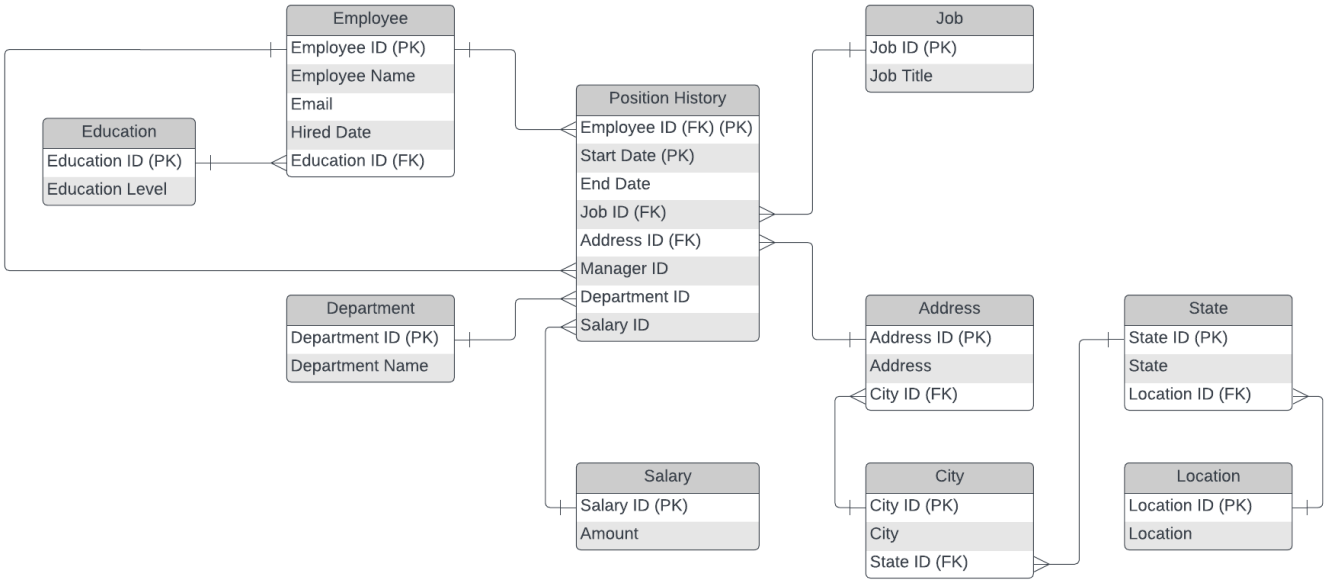
Relational Database Design

ERD

- **Conceptual**

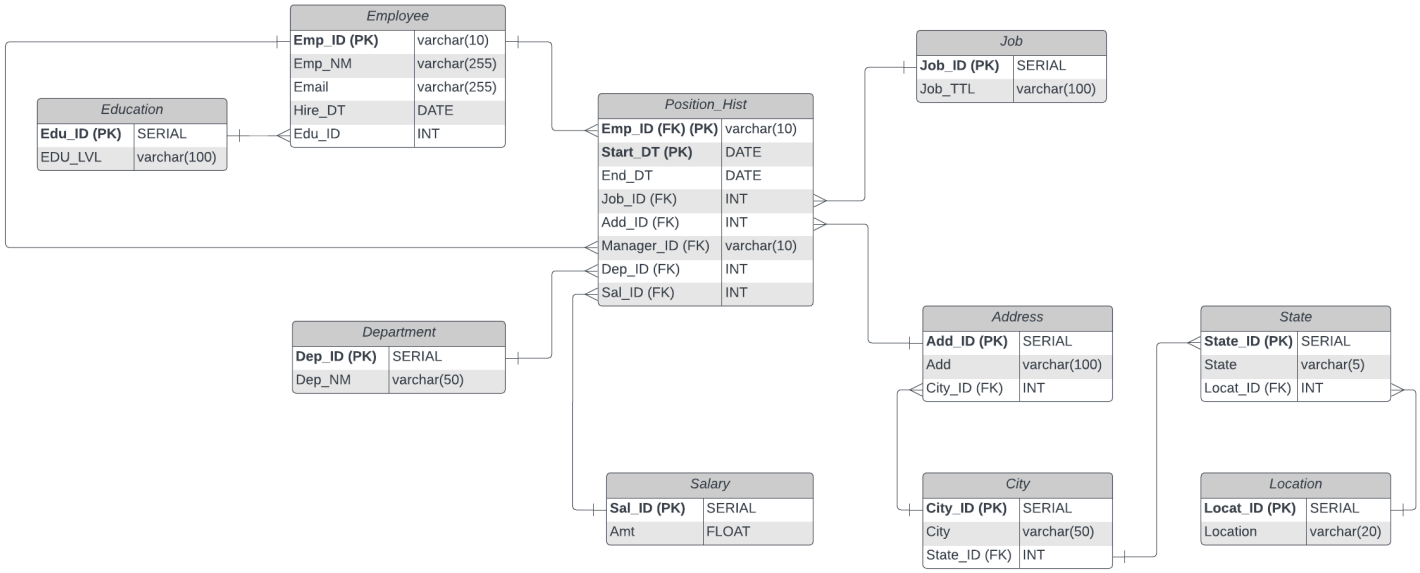


- **Logical**



ERD

- **Physical**



Step 3

Create A Physical Database

DDL

Create a database named whatever in pgAdmin4 (PostgreSQL), connect to that database and run DDL below to build the database designed in Step 2

project1/postgres@PostgreSQL 15

No limit

Query | Query History

```sql
--create table
create table Job(Job_ID serial primary key,
                 Job_TTL varchar(100));
create table Department(Dep_ID serial primary key,
                        Dep_NM varchar(50));
create table Location(Locat_ID serial primary key,
                      Location varchar(20));
create table State(State_ID serial primary key,
                   State varchar(5),
                   Locat_ID int references Location(Locat_ID));
create table City(City_ID serial primary key,
                  City varchar(50),
                  State_ID int references State(State_ID));
create table Address(Add_ID serial primary key,
                     Add varchar(100),
                     City_ID int references City(City_ID));
create table Education(Edu_ID serial primary key,
                       Edu_LVL varchar(100));
create table Employee(Emp_ID varchar(10) primary key,
                      Emp_NM varchar(255),
                      Email varchar(255),
                      Hire_DT date,
                      Edu_ID int references Education(Edu_ID));
create table Salary(Sal_ID serial primary key,
                    Amt float);
create table Position_Hist(Emp_ID varchar(10) references Employee(Emp_ID),
                           Start_DT date,
                           primary key(Emp_ID,Start_DT),
                           End_DT date,
                           Job_ID int references Job(Job_ID),
                           Add_ID int references Address(Add_ID),
                           Manager_ID varchar(10) references Employee(Emp_ID),
                           Dep_ID int references Department(Dep_ID),
                           Sal_ID int references Salary(Sal_ID));
```

Data Output | Messages | Notifications

```
CREATE TABLE

Query returned successfully in 103 msec.
```

Total rows: 0 of 0 | Query complete 00:00:00.103

DDL

Create a stage table named "staging_tbl"

```sql
--create a stage table for falt file ETL
create table staging_tbl(emp_id varchar(10),
                         emp_nm varchar(255),
                         email varchar(255),
                         hire_dt date,
                         job_title varchar(100),
                         salary float,
                         department varchar(50),
                         manager varchar(255),
                         start_dt date,
                         end_dt date,
                         location varchar(20),
                         address varchar(100),
                         city varchar(50),
                         state varchar(5),
                         education_level varchar(100));
```

Data Output | Messages | Notifications

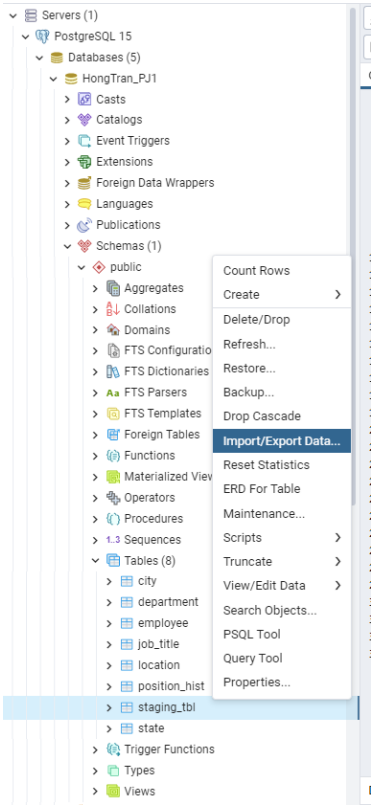
```
CREATE TABLE

Query returned successfully in 75 msec.
```

Total rows: 0 of 0 | Query complete 00:00:00.075

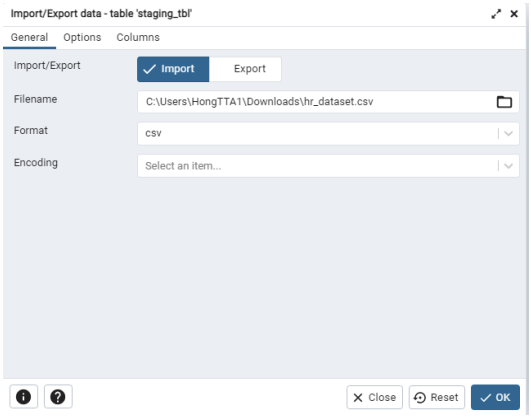
ETL

Import HR data from excel into the stage table, following these steps:

- Right click on **staging_tbl** table in pgAgmin4 (PostgreSQL)
- Select Import/Export Data...



- Load local file in your machine and Click OK



ETL

Load data from the stage table into the database

project1/postgres@PostgreSQL 15

No limit

Query | Query History

```sql
insert into location(location)
select distinct location from staging_tbl;

insert into state(state,locat_id)
select distinct s.state,l.locat_id
from staging_tbl s join location l on s.location=l.location;

insert into city(city,state_id)
select distinct s.city,st.state_id
from staging_tbl s join state st on s.state=st.state;

insert into address(add,city_id)
select distinct s.address,c.city_id
from staging_tbl s join city c on s.city=c.city;

insert into job(job_ttl)
select distinct job_title from staging_tbl;

insert into department(dep_nm)
select distinct department from staging_tbl;

insert into salary(amt)
select distinct salary from staging_tbl;

insert into education(edu_lvl)
select distinct education_level from staging_tbl;

insert into employee(emp_id,emp_nm,email,hire_dt,edu_id)
select distinct s.emp_id,s.emp_nm,s.email,s.hire_dt,e.edu_id
from staging_tbl s join education e on s.education_level=e.edu_lvl;

insert into position_hist(emp_id,start_dt,end_dt,job_id,add_id,manager_id,dep_id,sal_id)
select distinct e.emp_id,s.start_dt,s.end_dt,j.job_id,a.add_id,m.manager_id,d.dep_id,sa.sal_id
from staging_tbl s
    join job j on s.job_title=j.job_ttl
```

Data Output | Messages | Notifications

```
INSERT 0 205

Query returned successfully in 89 msec.
```

Total rows: 0 of 0 | Query complete 00:00:00.089

CRUD

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

Properties SQL Statistics Dependencies Dependents Processes CRUD.sql

project1/postgres@PostgreSQL 15

No limit

Query Query History

```sql
--Question 1: Return a list of employees with Job Titles and Department Names
select p.emp_id,e.emp_nm,j.job_ttl,d.dep_nm
from position_hist p
     join department d on p.dep_id=d.dep_id
     join job j on p.job_id=j.job_id
     join employee e on p.emp_id = e.emp_id
where p.end_dt is null;
```

Data Output Messages Notifications

	emp_id character varying (10)	emp_nm character varying (255)	job_ttl character varying (100)	dep_nm character varying (50)
1	E60752	Michael Tholstrup	Design Engineer	Product Development
2	E27909	Michael Sperduti	Administrative Assistant	Distribution
3	E93871	Travis Black	Sales Rep	Product Development
4	E22680	Cassidy Bancroft	Sales Rep	Product Development

CRUD

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

Properties SQL Statistics Dependencies Dependents Processes CRUD.sql

project1/postgres@PostgreSQL 15

No limit

Query Query History

```sql
--Question 2: Insert Web Programmer as a new job title
insert into job(job_ttl) values ('Web Programmer');
select*from job;
```

Data Output Messages Notifications

	job_id [PK] integer	job_ttl character varying (100)
1	1	Shipping and Receiving
2	2	Sales Rep
3	3	Administrative Assistant
4	4	Design Engineer
5	5	Database Administrator
6	6	Software Engineer
7	7	Manager
8	8	Legal Counsel
9	9	President
10	10	Network Engineer
11	12	Web Programmer

CRUD

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

Properties SQL Statistics Dependencies Dependents Processes CRUD.sql*

project1/postgres@PostgreSQL 15

No limit

Query Query History

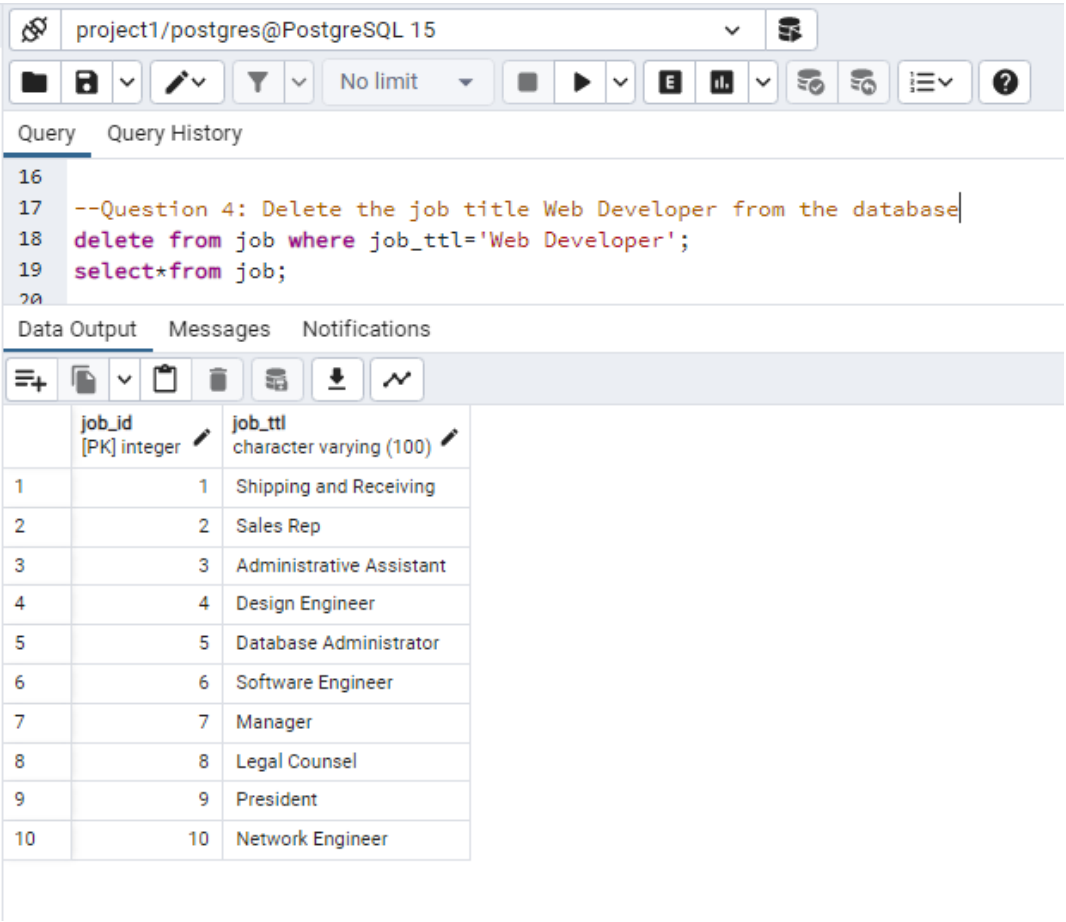
```sql
--Question 3: Correct the job title from web programmer to web developer
update job set job_ttl='Web Developer' where job_ttl='Web Programmer';
select*from job;
```

Data Output Messages Notifications

	job_id [PK] integer	job_ttl character varying (100)
1	1	Shipping and Receiving
2	2	Sales Rep
3	3	Administrative Assistant
4	4	Design Engineer
5	5	Database Administrator
6	6	Software Engineer
7	7	Manager
8	8	Legal Counsel
9	9	President
10	10	Network Engineer
11	12	Web Developer

CRUD

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



	job_id [PK] integer	job_ttl character varying (100)
1	1	Shipping and Receiving
2	2	Sales Rep
3	3	Administrative Assistant
4	4	Design Engineer
5	5	Database Administrator
6	6	Software Engineer
7	7	Manager
8	8	Legal Counsel
9	9	President
10	10	Network Engineer

CRUD

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

project1/postgres@PostgreSQL 15

No limit

Query | Query History

```
--Question 5: How many employees are in each department?
select d.dep_nm,count(distinct emp_id) as num_of_emp
from position_hist p join department d on p.dep_id=d.dep_id
where p.end_dt is null
group by d.dep_nm;
```

Data Output | Messages | Notifications

	dep_nm character varying (50)	num_of_emp bigint
1	Distribution	25
2	HQ	13
3	IT	52
4	Product Development	69
5	Sales	40

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.**

```
select e.emp_nm,j.job_ttl,d.dep_nm as department, em.emp_nm as manager,p.start_dt,p.end_dt
from position_hist p
        join employee e on p.emp_id=e.emp_id
        left join employee em on p.manager_id=em.emp_id
        join job j on p.job_id=j.job_id
        join department d on p.dep_id=d.dep_id
where e.emp_nm='Toni Lembeck';
```

Data Output | Messages | Notifications

	emp_nm character varying (255)	job_ttl character varying (100)	department character varying (50)	manager character varying (255)	start_dt date	end_dt 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.**

  Salary information is stored in the table Salary. To prevent specific roles (users) from accessing this sensitive information, I did the following:

  - Create a role with domain login
  - Grant select priviledge to all tables in the database for the role
  - Revoke select on the table Salary so the role does not have permission to read this table

Step 4

Above and Beyond

(optional)

Standout Suggestion 1

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

project1/postgres@PostgreSQL 15

No limit

Query | Query History | Scratch Pad

```
--View: return all employee attributes; result should resemble initial Excel file
create or replace view emp_info as
    select p.emp_id,e.emp_nm,e.email,e.hire_dt,j.job_ttl,sa.amt as salary,
           d.dep_nm as department, em.emp_nm as manager,
           p.start_dt,p.end_dt,l.location,a.add as address,c.city,s.state,
           edu.edu_lvl
    from position_hist p
         join employee e on p.emp_id=e.emp_id
         left join employee em on p.manager_id=em.emp_id
         join job j on p.job_id=j.job_id
         join department d on p.dep_id=d.dep_id
         join education edu on e.edu_id=edu.edu_id
         join salary sa on p.sal_id=sa.sal_id
         join address a on p.add_id=a.add_id
         join city c on a.city_id=c.city_id
         join state s on c.state_id=s.state_id
         join location l on s.locat_id=l.locat_id;

--query view
select*from emp_info;
```

Data Output | Messages | Notifications

	emp_id character varying (10)	emp_nm character varying (255)	email character varying (255)	hire_dt date	job_ttl character varying (100)	salary double precision	department character varying (50)	manager character varying (255)	start_dt date	end_dt date
1	E60752	Michael Tholstrup	Michael.Tholstrup@TechCorp.com	2010-02-27	Design Engineer	138695	Product Development	Conner Kinch	2010-02-27	[null]
2	E27909	Michael Sperduti	Michael.Sperduti@TechCorp.com	2014-06-20	Administrative Assistant	43778	Distribution	Allison Gentle	2014-06-20	[null]
3	E93871	Travis Black	Travis.Black@TechCorp.com	2002-04-30	Sales Rep	88910	Product Development	Conner Kinch	2002-04-30	[null]
4	E22680	Cassidy Bancroft	Cassidy.Bancroft@TechCorp.com	2013-10-26	Sales Rep	196637	Product Development	Conner Kinch	2013-10-26	[null]
5	E13596	Kenneth Dewitt	Kenneth.Dewitt@TechCorp.com	2012-04-09	Sales Rep	180913	Product Development	Conner Kinch	2012-04-09	[null]
6	E24100	Zondra Peck	Zondra.Peck@TechCorp.com	1997-03-06	Network Engineer	98994	IT	Jacob Lauber	1997-03-06	[null]
7	E71128	Stan Frank	Stan.Frank@TechCorp.com	1998-01-28	Shipping and Receiving	48749	Distribution	Allison Gentle	1998-01-28	[null]
8	E70374	Norman Guerrero	Norman.Guerrero@TechCorp.com	2007-08-19	Sales Rep	123946	Sales	Jennifer De La Garza	2007-08-19	[null]
9	E12562	Keith Ingram	Keith.Ingram@TechCorp.com	1996-04-14	Administrative Assistant	48910	Product Development	Conner Kinch	1996-04-14	[null]
10	E21696	Mallory Russo	Mallory.Russo@TechCorp.com	1997-12-06	Legal Counsel	167887	Product Development	Conner Kinch	1997-12-06	[null]
11	E61947	Charles Wirry	Charles.Wirry@TechCorp.com	2010-03-04	Sales Rep	212353	Sales	Jennifer De La Garza	2010-03-04	[null]
12	E35075	John Certa	John.Certa@TechCorp.com	2007-08-29	Administrative Assistant	51633	HQ	Tyrone Hutchison	2007-08-29	[null]

Total rows: 205 of 205 | Query complete 00:00:00.063

Standout Suggestion 2

Create a function 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.

project1/postgres@PostgreSQL 15

No limit

Query | Query History

```
--Function: return current and past jobs when given an employee name
create or replace function getJobHist(empName varchar(255))
returns table(emp_nm varchar(255),
              job_ttl varchar(100),
              department varchar(50),
              manager varchar(255),
              start_dt date,
              end_dt date)
language 'plpgsql'
as $$
    begin
        return query
        select e.emp_nm,e.job_ttl,e.department,e.manager,e.start_dt,e.end_dt
        from emp_info e
        where e.emp_nm=empName;
    end
$$;

--Call function
select*from getJobHist('Edward Eslser');
```

Data Output | Messages | Notifications

	emp_nm character varying	job_ttl character varying	department character varying	manager character varying	start_dt date	end_dt date
1	Edward Eslser	Software Engineer	IT	Jacob Lauber	2006-07-26	[null]
2	Edward Eslser	Shipping and Receiving	Distribution	Allison Gentle	2002-04-16	2006-07-25

Standout Suggestion 3

Implement user security on the restricted salary attribute

project1/postgres@PostgreSQL 15

No limit

Query | Query History

```
--Create a role that can login to the database
create role NoMgr
login password '12345';
--Grant select privilege to all tables in the database
grant select
on all tables in schema "public"
to NoMgr;
--Revoke select privilege from the table Salary (sensitive information)
revoke select
on Salary
from NoMgr;
```

Data Output | Messages | Notifications

```
REVOKE

Query returned successfully in 107 msec.
```

SQL Shell (psql)

```
Server [localhost]:
Database [postgres]: project1
Port [5432]:
Username [postgres]: nomgr
Password for user nomgr:
psql (15.2)
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

project1=> select*from salary;
ERROR:  permission denied for table salary
project1=>
```

Appendix

Additional Info

Slide 29: I created a function instead of stored procedure. This is because stored procedure in PostgreSQL does not return a result set from select statement as I intended. Using function is much more flexible and support that task.