

# Tech ABC Corp - HR Database

**Project Owner :** Maram Alalyani

**Version :** HRDataBase.V1

**Date :** 24-3-2024



# 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 become 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

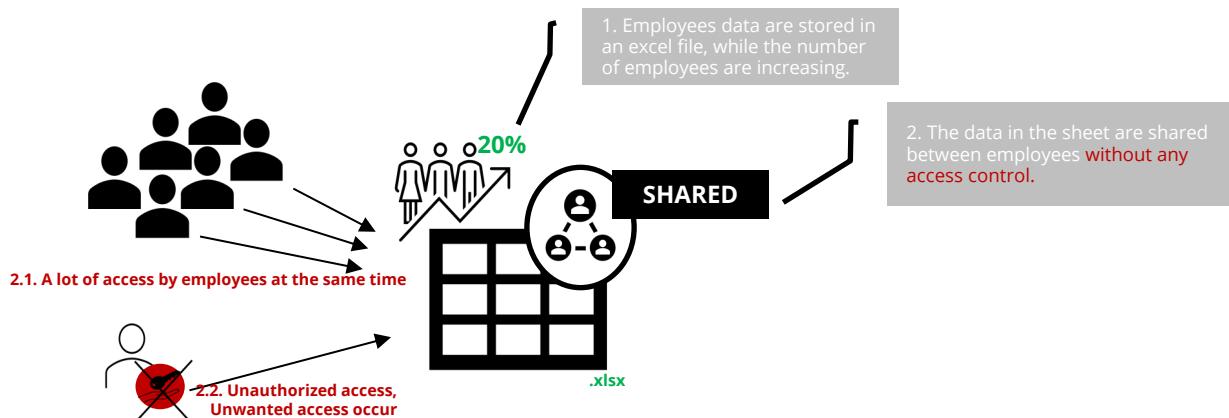
# Data Architect Business Requirement

## Purpose of the new database:

Tech ABC Corp's company projecting a huge growing in the latest 6 months, increasing the number of employees from 10 to 200, published to many locations nationwide. In this kind of situation, only the employees data are stored in an excel sheet file. Which is critical point when we talking about the employees sensitive data, there is no restriction or any kind of privacy since all have access to this sheet, as well as, since the file it's shared between them, the percentage of conflicts occurring. So, there is a need to create a new database to have a single source of truth where all the data stored in, securing the identity and ensuring integrity.

## Describe current data management solution:

### Current Situation:

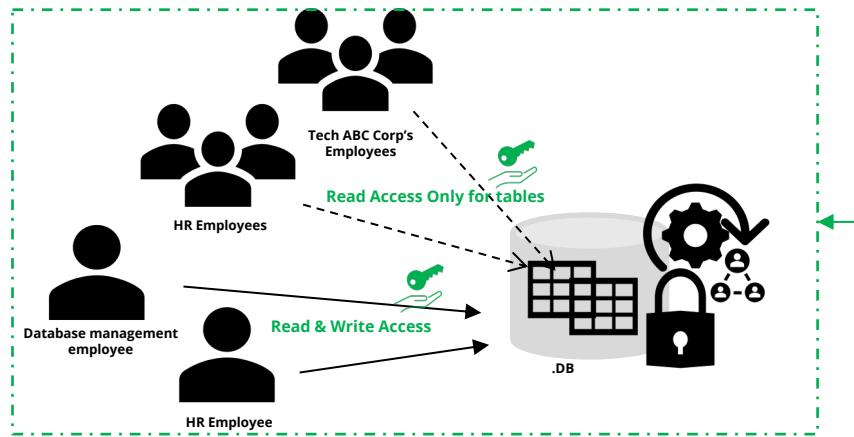


### This situation missing for :

- 1 Data Integrity
- 2 Data Security
- 3 Data scalability & Flexibility
- 4 Data Sharing & Operations
- 5 Single Source Of Truth

These are refer to what Data management Will cover

## Data Management Solution :



# Data Architect Business Requirement

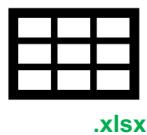
## Current data available:

Employees data, HR related data. Including their names, ID, Emails, hire date , job title , salary, department, manager, start date, end date, location, address, city, state and education level. The current available data as shown below.

EMP_ID	EMP_NM	EMAIL	HIRE_DT	JOB_TITLE	SALARY	DEPARTMENT	MANAGER	START_DT	END_DT	LOCATION	ADDRESS	CITY	STATE	EDUCATION LEVEL

Table1. HR dataset

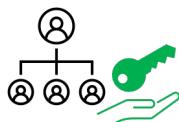
For now, these data enough for us to create the database, we don't need additional data. If any in the future we'll contact you as soon as possible. We'll set some standards and policies for who will own and manage the data, and they will be the Tech ABC Corp's IT department. Giving an access and permissions to the data for change and update, will be with the top of management level access and only one of the HR employee with Read & Write Access



The estimated size of the dataset we have in terms of numbers of rows are 205 of employees data.



Annual growth is projecting a 20% a year for next 5 years.



Employees data are is the most sensitive data, such their Salaries. While This data are to be restricted to a specific employee and Admin setting an access control.

# Data Architect Technical Requirement

The justification behind the creation of new database due to the **Estimated 20% growth with Data integrity and Data Security purposes**

## ● Database objects

The database objects

tables : City(ID, City), State(ID, State,CITY\_ID) , Location(ID, Location, Address, STATE\_ID), Employee(ID,Name,Email,hire\_date,MANAGER\_ID,EDUUCATION\_ID,LOCATION\_ID), Education\_Level(ID, Education\_level), Manager(ID,Manager\_Name), Salary(ID, Salary), Departments(ID,Departments\_name), Job\_Title(ID,Job\_Title) , Employment(Employment\_ID, DEPARTMENT\_ID ,EMP\_ID ,END\_DATE,START\_DATE,SALARY\_ID,JOB\_ID)

## ● Data ingestion

Data ingestion method will be ETL which is The most appropriate approach.

## ● Data governance (Ownership and User access)

**Ownership:** HR

**User Access:**

- **Read & Write Access :** Management level user, One HR employee to the DB.
- **Read Access :** all employees have specific access to specific tables not the sensitive(Salary).

# Data Architect Technical Requirement

- **Scalability**

Replication is the most suitable scalability approach for this kind of situation.

- **Flexibility**

Describe measures taken to ensure future data integration if needed

- **Storage & retention**

**Storage (disk or in-memory):** disk

**Retention:** the data have to be kept for 7 years long.

- **Backup**

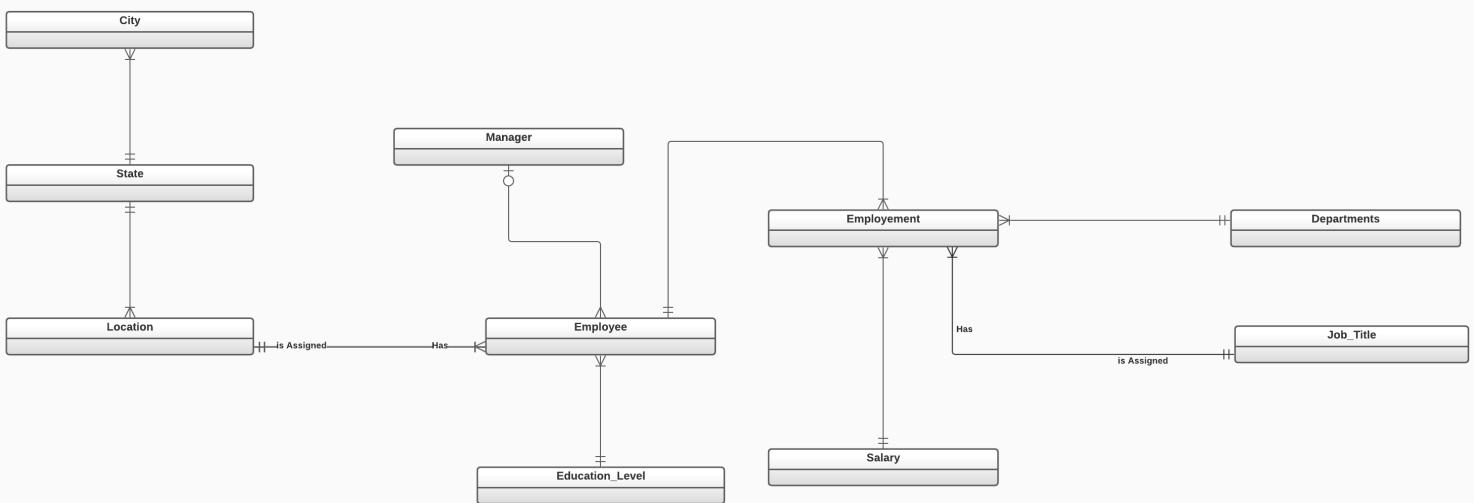
Backup schedule is full backup 1x per week, incremental backup daily.

---

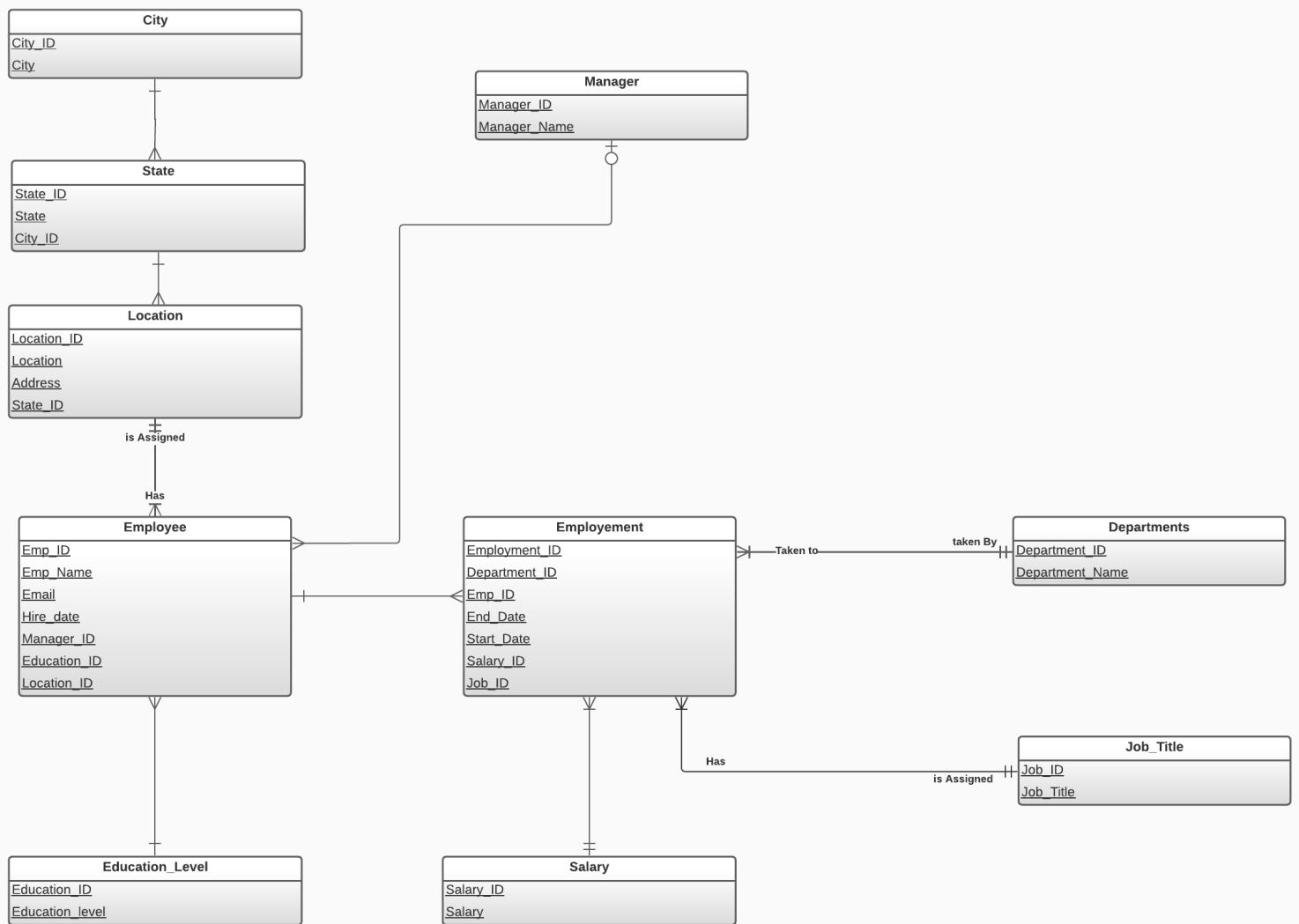
# **Step 2**

## Relational Database Design

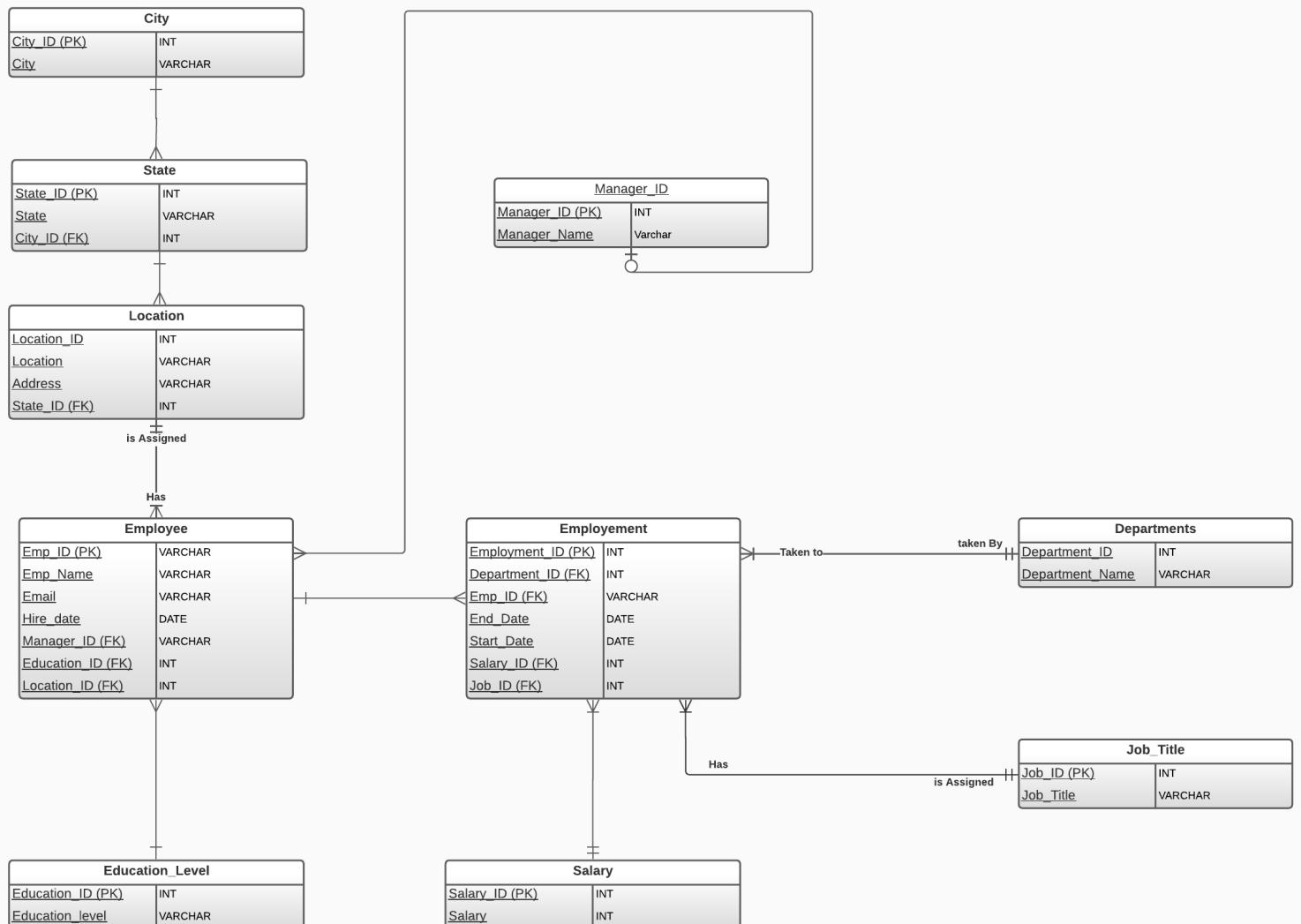
# ERD Conceptual Model



# ERD Logical Model



# ERD Physical Model





## Step 3

# Create A Physical Database

In this step, Start turning HR database model into a physical database.

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

# DDL

DDL SQL script capable of building the HR Database that designed in Step 2

```
Query  Query History

1  create table proj_stg(Emp_ID varchar(8),
2  Emp_NM varchar(50),
3  Email  varchar(100),
4  hire_dt date,
5  job_title varchar(100),
6  salary int,
7  department_nm varchar(50),
8  manager varchar(50),
9  start_dt date,
10 end_dt date,
11 location varchar(50),
12 address varchar(100),
13 city varchar(50),
14 state varchar(2),
15 education_lvl varchar(50)
16 );
17
18
19 insert into proj_stg (Emp_ID, Emp_NM, email, hire_dt, job_title, salary, department_nm, manager, start_
values ('E17469', 'Haifa Hajiri', 'Haifa.Hajiri@TechCorp.com', ' 2003-12-17 ','Administrative Assis
20 insert into proj_stg (Emp_ID, Emp_NM, email, hire_dt, job_title, salary, department_nm, manager, start_
21 values('E27621', 'Wendell Mobley', 'Wendell.Mobley@TechCorp.com', ' 2013-11-27 ','Administrative A
22 insert into proj_stg (Emp_ID, Emp_NM, email, hire_dt, job_title, salary, department_nm, manager, start_
23 values('E27909', 'Michael Sperduti', 'Michael.Sperduti@TechCorp.com', ' 2014-06-20 ','Administrati
24 insert into proj_stg (Emp_ID, Emp_NM, email, hire_dt, job_title, salary, department_nm, manager, start_
25 values('E35053', 'Ashley Bergman', 'Ashley.Bergman@TechCorp.com', ' 2009-03-01 ','Administrative A
26 insert into proj_stg (Emp_ID, Emp_NM, email, hire_dt, job_title, salary, department_nm, manager, start_
27 values('E51723', 'Carlos Lopez', 'Carlos.Lopez@TechCorp.com', ' 2014-05-23 ','Administrative Assis
28 insert into proj_stg (Emp_ID, Emp_NM, email, hire_dt, job_title, salary, department_nm, manager, start_
29 values('E59688','Jason Wingard', 'Jason.Wingard@TechCorp.com', ' 2006-01-11 ','Administrative Assis
30
31
32 insert into proj_stg (Emp_ID, Emp_NM, email, hire_dt, job_title, salary, department_nm, manager, start_
33 values('E186971', 'Anita Deluise', 'Anita.Deluise@TechCorp.com', ' 1995-06-01 ','Administrative
```

The “Proj Stg” table created based on the table from provided excel sheet, inserting the value

# DDL

DDL SQL script capable of building the HR Database that designed in Step 2

Query    Query History

```
335 -- Insert City , ETL --
336 create table City(
337     City_ID SERIAL PRIMARY KEY,
338     City varchar(50));
339
340 INSERT INTO City(City)
341 SELECT DISTINCT(city) FROM proj_stg;
342
343
344 select * from City;
345
346 -- Insert City , ETL --
347 create table State(
348     State_ID SERIAL PRIMARY KEY,
349     State varchar(2),
350     City_ID int references City(City_ID));
351
352 INSERT INTO State(State, City_ID)
353 SELECT DISTINCT ps.state, c.City_ID
354 FROM proj_stg ps
355 JOIN City c
356 ON ps.city = c.City
357
358 select * from State;
```

City, State Tables creation & Ingested data use of ETL method.

# DDL

DDL SQL script capable of building the HR Database that designed in Step 2

```
Query Query History
360 -- Insert Location , ETL --
361 create table Location(
362     Location_ID SERIAL PRIMARY KEY,
363     location varchar(50),
364     Address varchar(100),
365     State_ID int references State(State_ID));
366
367 INSERT INTO Location(location , Address , State_ID)
368 SELECT DISTINCT
369     ps.location,
370     ps.address,
371     s.State_ID
372 FROM proj_stg ps
373 JOIN City c
374 ON ps.city = c.City
375 JOIN State s
376 ON c.City_ID = s.City_ID;
377
378 select * from Location;
379
380
381 -- Insert Job , ETL --
382 create table Job(
383     job_ID SERIAL PRIMARY KEY,
384     job_title varchar(100));
385
386 INSERT INTO Job(job_title)
387 SELECT DISTINCT(job_title)
388 FROM proj_stg;
389
390 select * from Job;
391
```

```
Query Query History
392 -- Insert Departments , ETL --
393 create table Departments(
394     Department_ID SERIAL PRIMARY KEY,
395     Department_nm varchar(50));
396
397 INSERT INTO Departments(Department_nm)
398 SELECT DISTINCT(department_nm)
399 FROM proj_stg;
400
401 select * from Departments;
402
403 -- Insert Salary , ETL --
404 create table Salary(
405     Salary_ID SERIAL PRIMARY KEY,
406     Salary int);
407
408 INSERT INTO Salary(Salary)
409 SELECT DISTINCT(salary)
410 FROM proj_stg;
411
412 select * from Salary;
413
414 -- Insert Education_Level , ETL --
415 create table Education_level(
416     Education_ID SERIAL PRIMARY KEY,
417     Education_lvl varchar(50));
418
419 INSERT INTO Education_level(Education_lvl)
420 SELECT DISTINCT(education_lvl)
421 FROM proj_stg;
422
423 select * from Education_level;
```

# DDL

DDL SQL script capable of building the HR Database that designed in Step 2

Query    Query History

```
425 -- Insert Manager , ETL --
426 CREATE TABLE Manager (
427     Manager_ID SERIAL PRIMARY KEY,
428     Manager_Name varchar(50));
429
430 INSERT INTO Manager (Manager_Name)
431 SELECT DISTINCT manager
432 FROM proj_stg
433 WHERE manager IS NOT NULL AND TRIM(manager) <> '';
434
435 select * from Manager;
436
437 -- Insert Employee , ETL --
438 create table Employee(
439     EMP_ID varchar(8) PRIMARY KEY,
440     Emp_Name varchar(50),
441     Email varchar(50),
442     hire_dt date,
443     Manager_ID int references Manager(Manager_ID),
444     Education_ID int references Education_level(Education_ID),
445     Location_ID int references Location(Location_ID));
446
447 INSERT INTO Employee(EMP_ID, Emp_Name, Email, hire_dt, Manager_ID, Education_ID, Location_ID)
448 SELECT
449     ps.Emp_ID,
450     ps.Emp_NM,
451     ps.Email,
452     ps.hire_dt,
453     m.Manager_ID,
454     el.Education_ID,
455     l.Location_ID
456 FROM
```

# DDL

DDL SQL script capable of building the HR Database that designed in Step 2

```
Query Query History
459 LEFT JOIN Education_level el ON el.Education_lvl = ps.education_lvl
460 LEFT JOIN Location l ON l.location = ps.location AND l.Address = ps.address
461 ON CONFLICT (EMP_ID) DO NOTHING;
462
463 select * from Employee;
464
465 -- Insert Employment , ETL --
466 create table Employment(
467     Employment_ID SERIAL PRIMARY KEY,
468     Department_ID int references Departments(Department_ID),
469     Emp_ID varchar(8) references Employee(Emp_ID),
470     start_dt date,
471     end_dt date,
472     Salary_ID int references Salary(Salary_ID),
473     job_ID int references Job(job_ID));
474
475 INSERT INTO Employment(Department_ID, Emp_ID, start_dt, end_dt, Salary_ID, job_ID)
476 SELECT
477     d.Department_ID,
478     e.EMP_ID,
479     ps.start_dt,
480     ps.end_dt,
481     s.Salary_ID,
482     j.job_ID
483 FROM
484     proj_stg ps
485 JOIN Departments d ON d.Department_nm = ps.department_nm
486 JOIN Employee e ON e.EMP_ID = ps.Emp_ID
487 JOIN Salary s ON s.Salary = ps.salary
488 JOIN Job j ON j.job_title = ps.job_title;
489
490 select * from Employment;
```

```
Query Query History
452     ps.hire_dt,
453     m.Manager_ID,
454     el.Education_ID,
455     l.Location_ID
456 FROM
457     proj_stg ps
458 LEFT JOIN Manager m ON m.Manager_Name = ps.manager
459 LEFT JOIN Education_level el ON el.Education_lvl = ps.education_lvl
460 LEFT JOIN Location l ON l.location = ps.location AND l.Address = ps.address
461 ON CONFLICT (EMP_ID) DO NOTHING;
462
463 select * from Employee;
464
465 -- Insert Employment , ETL --
466 create table Employment(
467     Employment_ID SERIAL PRIMARY KEY,
468     Department_ID int references Departments(Department_ID),
469     Emp_ID varchar(8) references Employee(Emp_ID),
470     start_dt date,
471     end_dt date,
472     Salary_ID int references Salary(Salary_ID),
473     job_ID int references Job(job_ID));
474
475 INSERT INTO Employment(Department_ID, Emp_ID, start_dt, end_dt, Salary_ID, job_ID)
476 SELECT
477     d.Department_ID,
478     e.EMP_ID,
479     ps.start_dt,
480     ps.end_dt,
481     s.Salary_ID,
482     j.job_ID
483 FROM
```

# CRUD

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

```
490 select * from Employment;
491
492 /* Question 1: Return a list of employees with Job Titles and Department Names */
493 SELECT e.Emp_Name, j.job_title, d.Department_nm
494 FROM Employee e
495 JOIN Employment em ON e.EMP_ID = em.Emp_ID
496 JOIN Job j ON em.job_ID = j.job_ID
497 JOIN Departments d ON em.Department_ID = d.Department_ID;
```

Data Output Messages Notifications

	emp_name character varying (50) 	job_title character varying (100) 	department_nm character varying (50) 
1	Haifa Hajiri	Administrative Assistant	Distribution
2	Wendell Mobley	Administrative Assistant	Distribution
3	Michael Sperduti	Administrative Assistant	Distribution
4	Ashley Bergman	Administrative Assistant	Distribution
5	Carlos Lopez	Administrative Assistant	Distribution
6	Jason Wingard	Administrative Assistant	Distribution
7	Anita Delruise	Administrative Assistant	HQ
8	Joseph Donohue	Administrative Assistant	HQ
9	John Certa	Administrative Assistant	HQ
10	Raven Landis	Administrative Assistant	HQ
11	Randy Myers	Administrative Assistant	HQ
12	Danny Godiksen	Administrative Assistant	HQ
13	Raj Prudvi	Administrative Assistant	IT
14	Alexis Fitzpatrick	Administrative Assistant	IT

Total rows: 205 of 205    Query complete 00:00:00.163

# CRUD

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

```
499 /* Question 2: Insert Web Programmer as a new job title */
500 INSERT INTO Job(job_title)
501 VALUES('Web Programmer');
502 select * from Job;
503
```

Data Output    Messages    Notifications

	job_id	job_title
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	11	Web Programmer

Total rows: 11 of 11    Query complete 00:00:00.271

# CRUD

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

```
503
504 /* Question 3: Correct the job title from web programmer to web developer */
505 UPDATE Job SET job_title = 'Web Developer'
506 where job_title = 'Web Programmer';
507 select * from Job;
```

Data Output    Messages    Notifications

	job_id [PK] integer	job_title 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	11	Web Developer

Total rows: 11 of 11    Query complete 00:00:00.266

# CRUD

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

```
508
509 /* Question 4: Delete the job title Web Developer from the database */
510 DELETE FROM Job WHERE job_title = 'Web Developer';
511 select * from Job;
512
```

Data Output **Messages** Notifications

DELETE 1

Query returned successfully in 156 msec.

Total rows: 11 of 11

Query complete 00:00:00.156

# CRUD

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

```
513 /* Question 5: How many employees are in each department? */
514 SELECT d.Department_nm, count(em.Emp_ID) as Employee_Count
515 FROM Employment em
516 JOIN Departments d on em.Department_ID = d.Department_ID
517 GROUP BY d.Department_nm;
518
```

Data Output    Messages    Notifications

	department_nm character varying (50)	employee_count bigint
1	Product Development	70
2	HQ	13
3	Distribution	27
4	Sales	41
5	IT	54

Total rows: 5 of 5 | Query complete 00:00:00.176

# 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.**

```
519 /* Question 6: Write a query that returns current and past jobs
520 (include employee name, job title, department, manager name, start and end date for position)
521 for employee Toni Lembeck */
522
523 SELECT
524     e.Emp_Name,
525     j.job_title,
526     d.Department_nm,
527     m.Manager_Name,
528     em.start_dt,
529     em.end_dt
530 FROM Employment em
531 JOIN Employee e ON em.Emp_ID = e.EMP_ID
532 JOIN Job j ON em.job_ID = j.job_ID
533 JOIN Departments d ON em.Department_ID = d.Department_ID
534 LEFT JOIN Manager m ON e.Manager_ID = m.Manager_ID |
535 WHERE e.Emp_Name = 'Toni Lembeck';
536
537
```

Data Output    Messages    Notifications



	emp_name character varying (50)	job_title character varying (100)	department_nm character varying (50)	manager_name character varying (50)	start_dt date	end_dt date
1	Toni Lembeck	Database Administrator	IT	Jacob Lauber	2001-07-18	2100-02-02
2	Toni Lembeck	Network Engineer	IT	Jacob Lauber	1995-03-12	2001-07-18

Total rows: 2 of 2    Query complete 00:00:00.162

# CRUD

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

Based on the IT Depts. Best Practices Guides of the Tech ABC Corp, The security of the database is based at the user level. Which is rather than granting the user access to the database, grant user access to all tables in the database, then revoke access to any tables holding restricted data.



## Step 4

Above and Beyond  
(optional)

---

# Appendix

# Standout Suggestion 1

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

Query    Query History

```
541 CREATE VIEW Employee_Details AS
542 SELECT
543     e.Emp_ID,
544     e.Emp_Name,
545     e.Email,
546     e.hire_dt,
547     j.job_title,
548     d.Department_nm,
549     s.Salary,
550     el.Education_lvl,
551     l.location,
552     l.Address,
553     c.City,
554     st.State
555 FROM Employee e
556 JOIN Employment em ON e.Emp_ID = em.Emp_ID
557 JOIN Job j ON em.job_ID = j.job_ID
558 JOIN Departments d ON em.Department_ID = d.Department_ID
559 JOIN Salary s ON em.Salary_ID = s.Salary_ID
560 JOIN Education_Level el ON e.Education_ID = el.Education_ID
561 JOIN Location l ON e.Location_ID = l.Location_ID
562 JOIN State st ON l.State_ID = st.State_ID
563 JOIN City c ON st.City_ID = c.City_ID;
564
```

Data Output    Messages    Notifications

CREATE VIEW

Query returned successfully in 198 msec.

Query    Query History    Scratch Pad X

```
556     st.State
557 FROM Employee e
558 JOIN Employment em ON e.Emp_ID = em.Emp_ID
559 JOIN Job j ON em.job_ID = j.job_ID
560 JOIN Departments d ON em.Department_ID = d.Department_ID
561 JOIN Salary s ON em.Salary_ID = s.Salary_ID
562 JOIN Education_Level el ON e.Education_ID = el.Education_ID
563 JOIN Location l ON e.Location_ID = l.Location_ID
564 JOIN State st ON l.State_ID = st.State_ID
565 JOIN City c ON st.City_ID = c.City_ID;
566
567 select * from Employee_Details;
568 /* */
569 /* */
570
```

Data Output    Messages    Notifications

emp\_id character varying (8) | emp\_name character varying (50) | email character varying (50) | hire\_dt date | job\_title character varying (100) | department\_nm character varying (50) | salary integer | education\_lv character varying (50)

emp_id	emp_name	email	hire_dt	job_title	department_nm	salary	education_lv
1 E17469	Haifa Hajiri	Haifa.Hajiri@TechCorp.com	2003-12-17	Administrative Assistant	Distribution	47418	No College
2 E27621	Wendell Mobley	Wendell.Mobley@TechCorp.com	2013-11-27	Administrative Assistant	Distribution	28969	Some College
3 E27909	Michael Sperduti	Michael.Sperduti@TechCorp.com	2014-06-20	Administrative Assistant	Distribution	43778	Associates Degree
4 E35053	Ashley Bergman	Ashley.Bergman@TechCorp.com	2009-03-01	Administrative Assistant	Distribution	41090	No College
5 E51723	Carlos Lopez	Carlos.Lopez@TechCorp.com	2014-05-23	Administrative Assistant	Distribution	35825	No College
6 E59688	Jason Wingard	Jason.Wingard@TechCorp.com	2006-01-11	Administrative Assistant	Distribution	34809	Some College
7 E18697	Anita Deluse	Anita.Deluse@TechCorp.com	1995-06-01	Administrative Assistant	HQ	45325	Associates Degree
8 E25640	Joseph Donohue	Joseph.Donohue@TechCorp.com	2007-12-27	Administrative Assistant	HQ	40647	No College
9 E35075	John Certa	John.Certa@TechCorp.com	2007-08-29	Administrative Assistant	HQ	51633	Some College

Total rows: 205 of 205    Query complete 00:00:00.155    Ln 568, Col 1

# Standout Suggestion 2

## Implement user security on the restricted salary attribute.

```
592 /* Implement user security on the restricted salary attribute.  
593 Create a non-management user named NoMgr */  
595  
596 create user NoMgr with password 'password';  
597  
598 GRANT USAGE  
599 ON SCHEMA public TO NoMgr;  
600 /* Grant access to all tables excluding the Salary */  
601 GRANT SELECT ON ALL TABLES IN SCHEMA public TO NoMgr;  
602  
603 CREATE VIEW employee_view AS  
604 SELECT emp_id, emp_name, email, hire_dt, manager_id, location_id  
605 FROM Employee;  
606  
607 GRANT SELECT ON employee_view TO NoMgr;  
608  
609
```

Data Output Messages Notifications

GRANT

Query returned successfully in 128 msec.

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

```
592 /* Implement user security on the restricted salary attribute.  
593 Create a non-management user named NoMgr */  
595  
596 create user NoMgr with password 'password';  
597  
598 GRANT USAGE  
599 ON SCHEMA public TO NoMgr;  
600 /* Grant access to all tables excluding the Salary */  
601 GRANT SELECT ON ALL TABLES IN SCHEMA public TO NoMgr;  
602  
603 CREATE VIEW employee_view AS  
604 SELECT emp_id, emp_name, email, hire_dt, manager_id, location_id  
605 FROM Employee;  
606  
607 grant select on employee_view to NoMgr;  
608  
609 revoke all privileges on Salary from NoMgr;
```

Data Output Messages Notifications

REVOKE

Query returned successfully in 312 msec.