Tech ABC Corp - HR Database

Michał Jańczyk - 20.02.2021

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:**

  The recipient of our product will be our internal HR department. The aim will be to make the work of managing the staff of our company more effective. The database has to meet various requirements in order to be useful.

- **Describe current data management solution:**

  Currently, our HR department uses an extensive Excel spreadsheet to create and manage information on more than 200 employees in our company.

- **Describe current data available:**

  The sheet contains basic information about the company's employees along with the salaries they receive.

- **Additional data requests:**

  Only one administrator will have permission to edit data in the database. The rest of the users will have read level rights. One of the things to consider at this stage of the database design - will be to connect the database to the payroll department's system.

- **Who will own/manage data**

  Our main user with the most rights (permissions to edit the database) will be Mrs Sarah Collins - Head of HR

- **Who will have access to database**

  Request for database will be granted only HR and Management Level Employees will own and manage (have access to the data and have access to write permission) data in the database. Other employees who do not belong in the department mentioned above will have read access but which is restricted to only reading the Salary Table.

Data Architect Business Requirement

- **Estimated size of database**

  The current number of employees in the company is 200. We will definitely devote one row per employee in our main table. Next, we will need to consider all relationships to intermediate tables. In general, it will be a small database that will not require any special extension in terms of memory.

- **Estimated annual growth**

  The company has a very successful business so it will have no problem attracting new people. Expected growth is around 20% over the next five years.

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

  The current solution is not secure enough. It is exposed to data leakage. However, the data that is being processed is sensitive data that can be used to identify individuals in the event of loss. Personal data such image-related information (i.e. photos), phone numbers, metadata profiling internal company positions and user preferences, and location data - to determine a person's whereabouts.

- **Data retention and backup requirements**

  For the sensitive data of which our database is composed, the retention time for this information is 7 years, required by federal regulations to maintain this period of time. As for the cyclical backup of the database, this will take place weekly.

Data Architect Technical Requirement

- **Justification for the new database**

  One of the main reasons is security. A shared sheet with a list of employees can easily fall into the wrong hands. The second issue is the not too convenient way to share information, in the case of editing data. It is necessary to send out a new version of the sheet to all stakeholders.

- **Database objects**

  The database will consist of four main tables: Employee, Manager, Department, Location. Additionally, the database will contain a view that returns all employee attributes. It will also facilitate, stored procedures make the database appear to work effortlessly. Instead of creating a complicated query, they just need to provide a name to a stored procedure to get the results they want. In accordance with company policy, we will consider creating a fifth table containing the employee's salary data. Separate table for salary makes applying security rules for that information possible.

- **Data ingestion**

  Data will be entered into the database on the basis of the ETL process. Our IT department also recommend this option. ETL is the current best practice for working with flat files. If the flat file will be regularly updated, an automated ETL process can be set up. In addition, there will be only one person authorized to perform this type of action (editing). In the future, we are considering an API connection with the finance department in order to manage employee salaries more easily.

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

  Ownership: The data will be owned and maintained by our internal HR department.
  User Access: All employees logged into our domain will have access to the database - basic access. Senior managers will additionally have access to the table containing employees' salaries - extended access.

Data Architect Technical Requirement

- **Scalability**

  Even if a system is working reliably today, that doesn't mean it will necessarily work reliably in the future. One common reason for degradation is increased load: perhaps the system has grown from 1,000 concurrent users to 10,000 concurrent users. The data which we are keeping does not change very often, then replication is easier, we just need to copy the data to every node once.

- **Flexibility**

  More Flexibility Now = More Flexibility Later. The best Data Management systems keep the user in mind. In the case of corporate data, the user is, of course, the consumer. Our user will be made available with the payroll department's system in the future.

- **Storage & retention**

  Storage (disk or in-memory): This data is a very important and reliable source of information in our form. However, they do not require fast data conversion operations so it is a good idea to recommend our IT department and choose - Standard disc partition.

  Retention: Required by federal regulations to maintain this data for at least 7 years.

- **Backup**

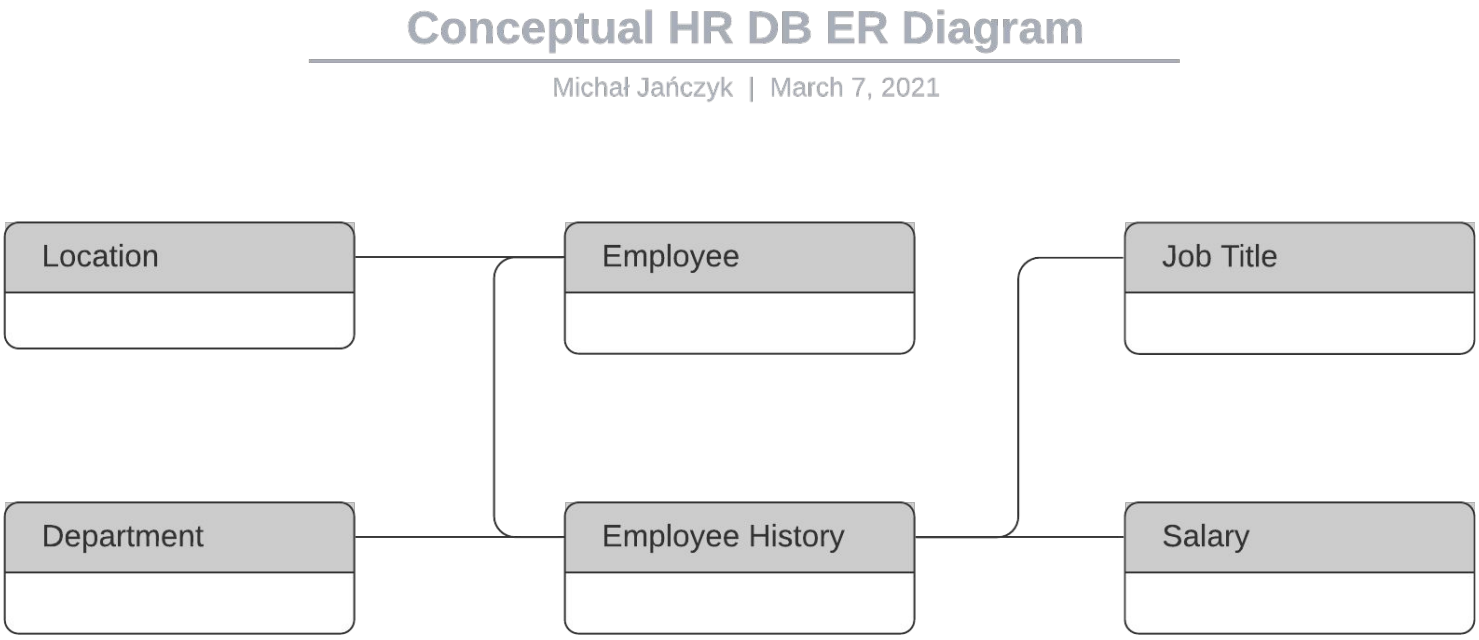
  Critical. Due to the importance of the data, it is recommended that the entire database is backed up once a week. However, incremental backup should be done daily.

Step 2

Relational Database Design

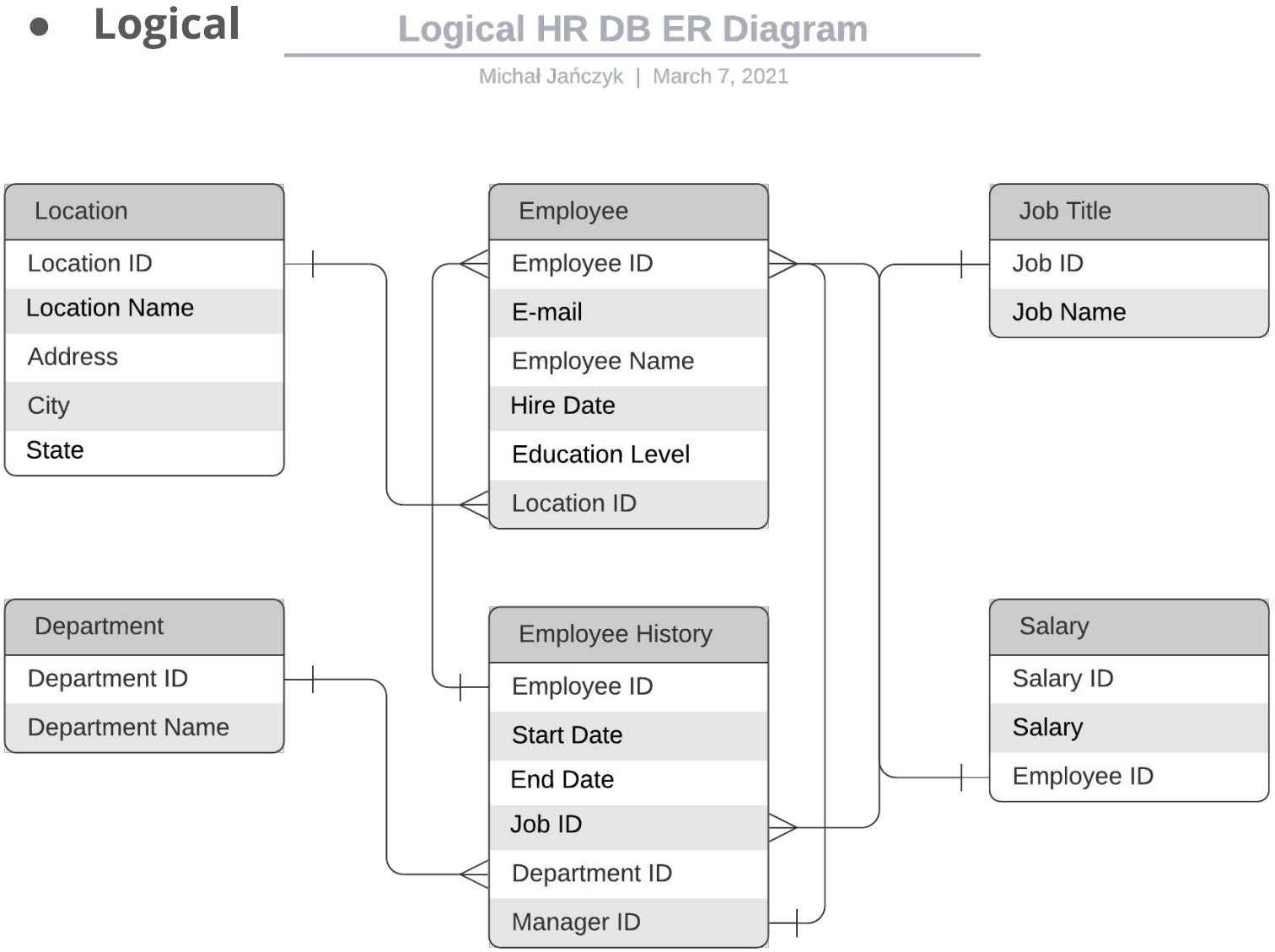
ERD

- **Conceptual**



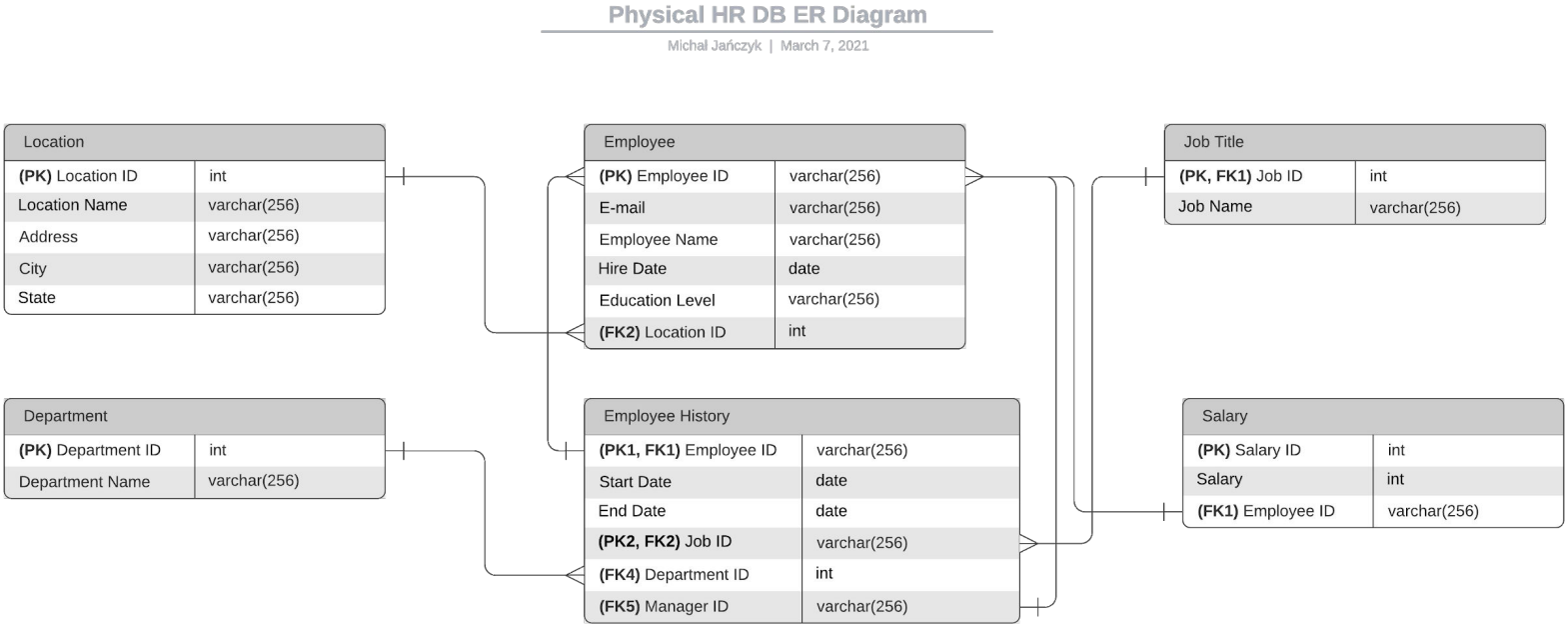
ERD

- **Logical**


Logical HR DB ER Diagram
Michał Jańczyk | March 7, 2021
Location
Location ID
Location Name
Address
City
State
Employee
Employee ID
E-mail
Employee Name
Hire Date
Education Level
Location ID
Job Title
Job ID
Job Name
Department
Department ID
Department Name
Employee History
Employee ID
Start Date
End Date
Job ID
Department ID
Manager ID
Salary
Salary ID
Salary
Employee ID

ERD

- **Physical**



Step 3

Create A Physical Database

DDL

```
11 -- create statement for employee table
12 create table employee
13 (
14     employee_id     varchar(256),
15     email           varchar(256),
16     employee_name   varchar(256),
17     hire_date       date,
18     education_level varchar(256),
19     location_id     serial
20 );
21
22 -- create statement for employee history table
23 create table employee_history
24 (
25     employee_id   varchar(256),
26     start_date    date,
27     end_date      date,
28     job_id        serial,
29     department_id serial,
30     manager_id    serial
31 );
32
```

CRUD

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

```
207 select emp.employee_name, emp_job.job_name, dep.department_name
208 from employee as emp
209 left join employee_history as emp_hist on emp.employee_id = emp_hist.employee_id
210 left join job as emp_job on emp_hist.job_id = emp_job.job_id
211 left join department as dep on emp_hist.department_id = dep.department_id
```

root@e604cf943752: /home/w

```
     employee_name      |         job_name          |   department_name
------------------------+---------------------------+---------------------
 Krishna Burli          | Administrative Assistant | Distribution
 Jacob Lauber           | Administrative Assistant | Distribution
 Dakeem Coleman         | Administrative Assistant | Distribution
 Keith Ingram           | Administrative Assistant | Distribution
 Marie Dumadara         | Administrative Assistant | Distribution
 Nick Gowen             | Administrative Assistant | Distribution
 Erica Davis            | Administrative Assistant | HQ
 Patricia DuBois        | Administrative Assistant | HQ
 Dennis Fredrich        | Administrative Assistant | HQ
```

CRUD

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

```
214
215 -- Question 2: Insert Web Programmer as a new job title
216 insert into job (job_name) values ('Web Programmer');
217
218 select job_name
219 from job
220 where job_name = 'Web Programmer';
```

root@e604cf943752: /home/w

```
INSERT 0 205
INSERT 0 199
INSERT 0 205
INSERT 0 205
INSERT 0 1
    job_name
----------------
 Web Programmer
(1 row)

postgres@e604cf943752:~$
```

CRUD

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

```
216 -- Question 2: Insert Web Programmer as a new job title
217 insert into job (job_name) values ('Web Programmer');
218 select job_name
219 from job
220 where job_name = 'Web Programmer';
221
222 -- Question 3: Correct the job title from web programmer to web developer
223 update job set job_name = 'Web Developer'
224 where job_name = 'Web Programmer';
225 select job_name
226 from job
227 where job_name = 'Web Developer';
```

root@aa0c2a604600: /home/v

```
INSERT 0 1
    job_name
----------------
 Web Programmer
(1 row)

UPDATE 1
   job_name
---------------
 Web Developer
(1 row)
```

CRUD

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

```
update job set job_name = 'Web Developer'
where job_name = 'Web Programmer';
select job_name
from job
where job_name = 'Web Developer';

-- Question 4: Delete the job title Web Developer from the database
delete from job where job_name = 'Web Developer';
select job_name
from job
where job_name = 'Web Developer';
```

```
root@aa0c2a604600: /home/v

   job_name
---------------
 Web Developer
(1 row)

DELETE 1
 job_name
----------
(0 rows)

postgres@aa0c2a604600:~$
```

CRUD

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

```sql
-- Question 5: How many employees are in each department?
select dep.department_name as "'Department Name'"
    , count(distinct emp_hist.employee_id) as "'Number of employees'"
from employee_history as emp_hist
left join department as dep on emp_hist.department_id = dep.department_id
group by dep.department_name

```

root@aa0c2a604600: /home/v

```
  'Department Name'   | 'Number of employees'
----------------------+-----------------------
 Distribution         |                    27
 HQ                   |                    13
 IT                   |                    53
 Product Development  |                    70
 Sales                |                    41
(5 rows)

postgres@aa0c2a604600:~$
```

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.**

```
226 -- 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.
227 select emp.employee_name as "Employee Name"
228     , emp_job.job_name as "Job Title"
229     , dep.department_name as "Department Name"
230     , emp_man.employee_name as "Manager Name"
231     , emp_hist.start_date as "Start Date"
232     , emp_hist.end_date as "End Date"
233 from employee as emp
234 left join employee_history as emp_hist on emp.employee_id = emp_hist.employee_id
235 left join job as emp_job on emp_hist.job_id = emp_job.job_id
236 left join department as dep on emp_hist.department_id = dep.department_id
237 left join employee as emp_man on emp_hist.manager_id = emp_man.employee_id
238 where emp.employee_name = 'Toni Lembeck';
```

root@7771bffc631e: /home/w

```
INSERT 0 199
INSERT 0 205
INSERT 0 205
 Employee Name |       Job Title        | Department Name | Manager Name | Start Date |  End Date
---------------+------------------------+-----------------+--------------+------------+------------
 Toni Lembeck  | Database Administrator | IT              | Jacob Lauber | 2001-07-18 | 2100-02-02
 Toni Lembeck  | Network Engineer       | IT              | Jacob Lauber | 1995-03-12 | 2001-07-18
(2 rows)

postgres@7771bffc631e:~$
```

CRUD

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

To prevent a leak of information form our database that involved data which caused a conflict within our company, even if it was not personally identifiable information. We need to organize data access for our teams and review who has access to what information. We are going to use service accounts in this database for executive and other employees. We will create two service account users – one for the management group and the other for the employee group. The administrators in this case will have windows authentication access to salary table. These service accounts will also have a role each – other for employees and management for management staff.

```
USE master
GO
CREATE LOGIN [exampleManagement] WITH PASSWORD = 'passwordisexample'
CREATE USER [exampleManagementUser] FROM LOGIN [exampleManagement]
CREATE LOGIN [exampleEmployee] WITH PASSWORD = 'passwordexample'
CREATE USER [exampleEmployeeUser] FROM LOGIN [exampleEmployee]

USE HR_DB
GO
CREATE ROLE [management]
CREATE ROLE [employee]
ALTER ROLE [management] ADD MEMBER [exampleManagementUser]
ALTER ROLE [employee] ADD MEMBER [exampleEmployeeUser]
```

Step 4

Above and Beyond

(optional)