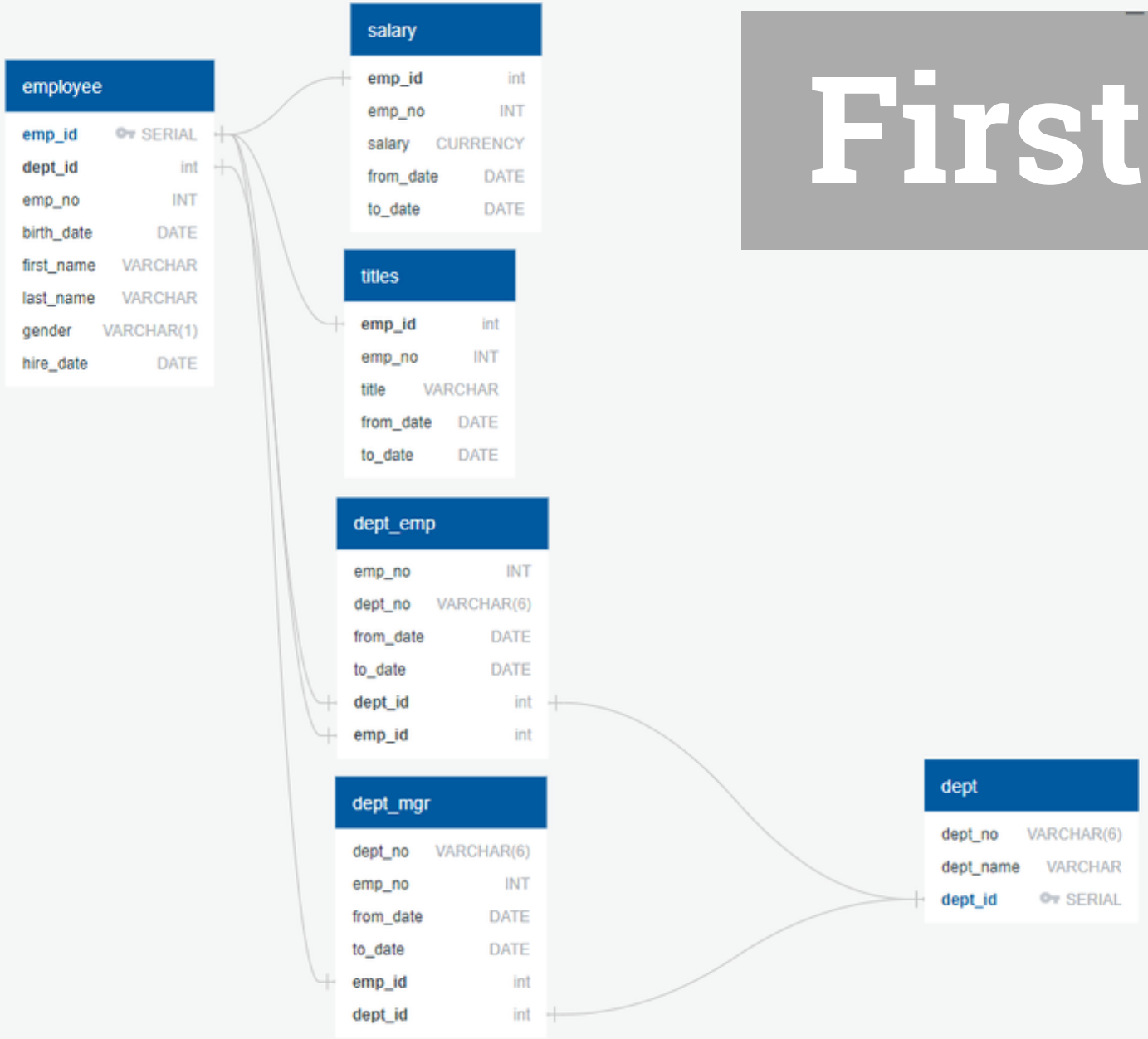
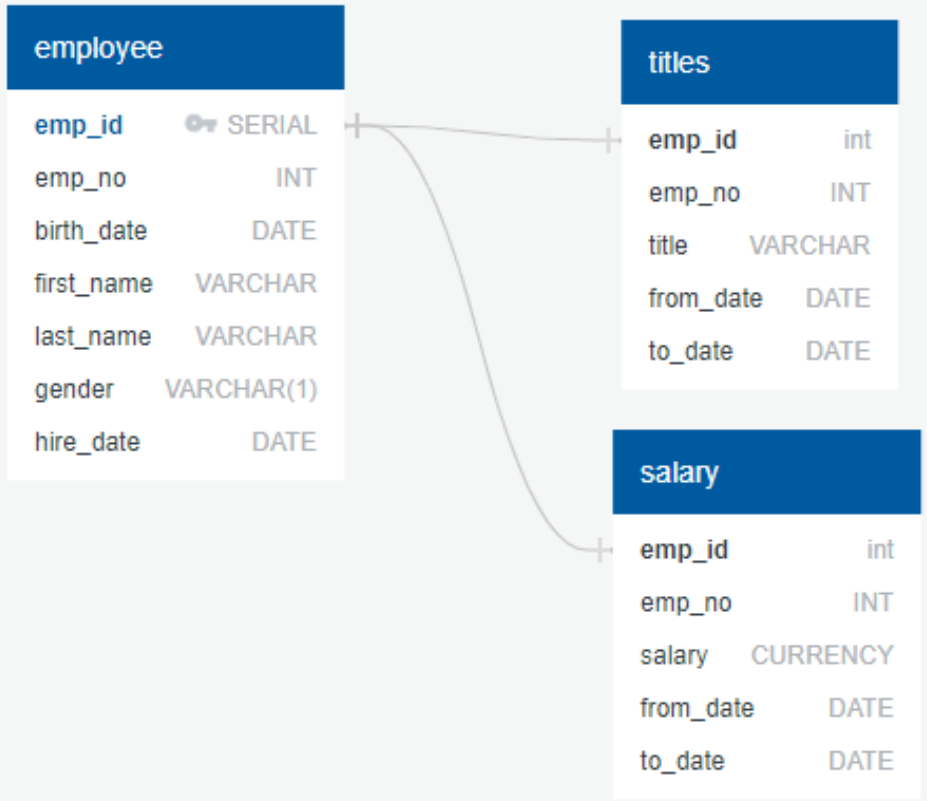
First Draft



Thought I had to repeat ID throughout the tables in order to link them for a query.

But, I noticed I was wrong and could use the ID #'s once if it was mapped effective.

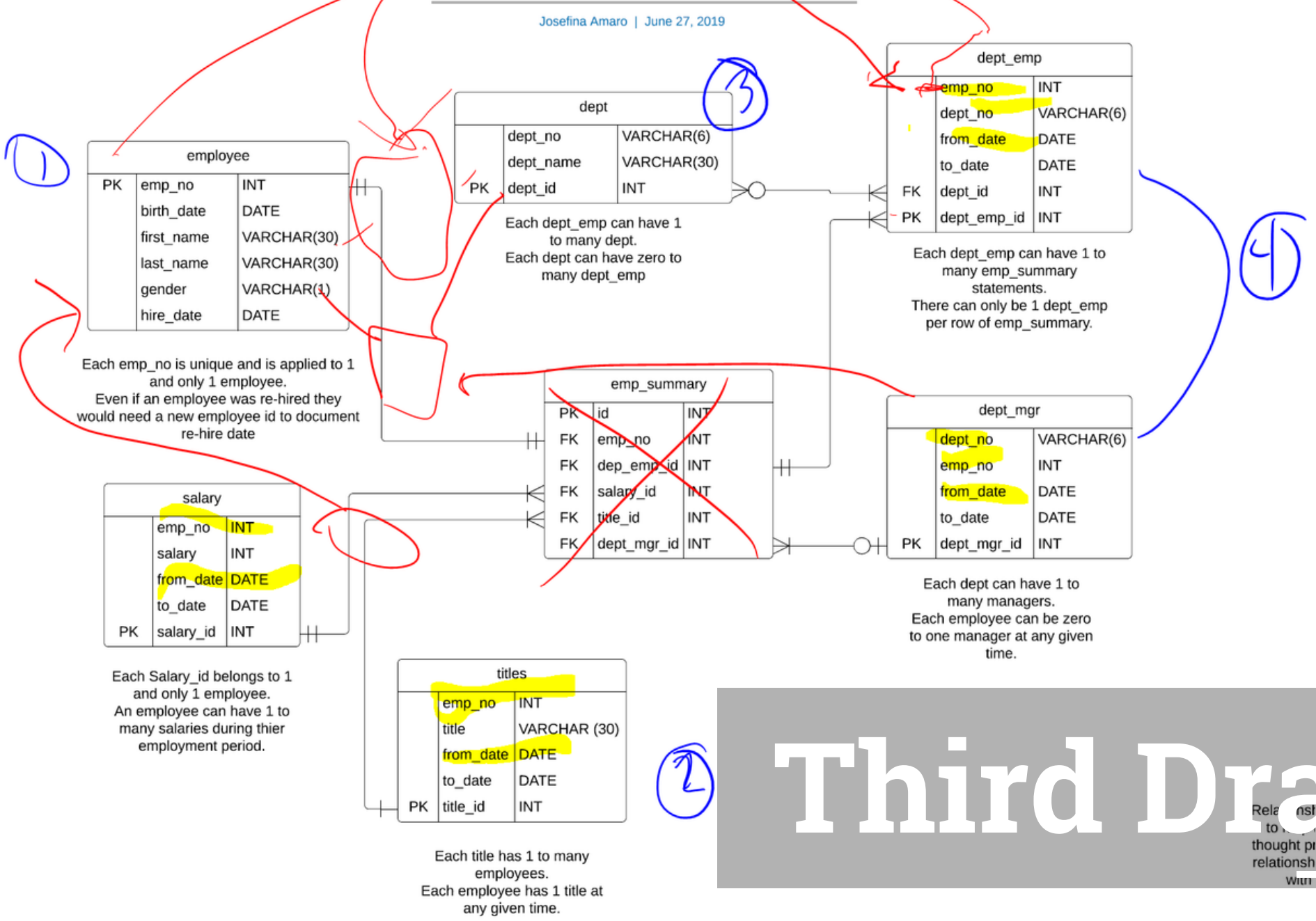


Second Draft

Then I thought I had to make an ID for the emp_no. After research, I figured I could use the emp_no as a primary key.

I also was looking at examples that made me think that the ID or "keys" referred to values in the columns not rows.

Josefina Amaro | June 27, 2019



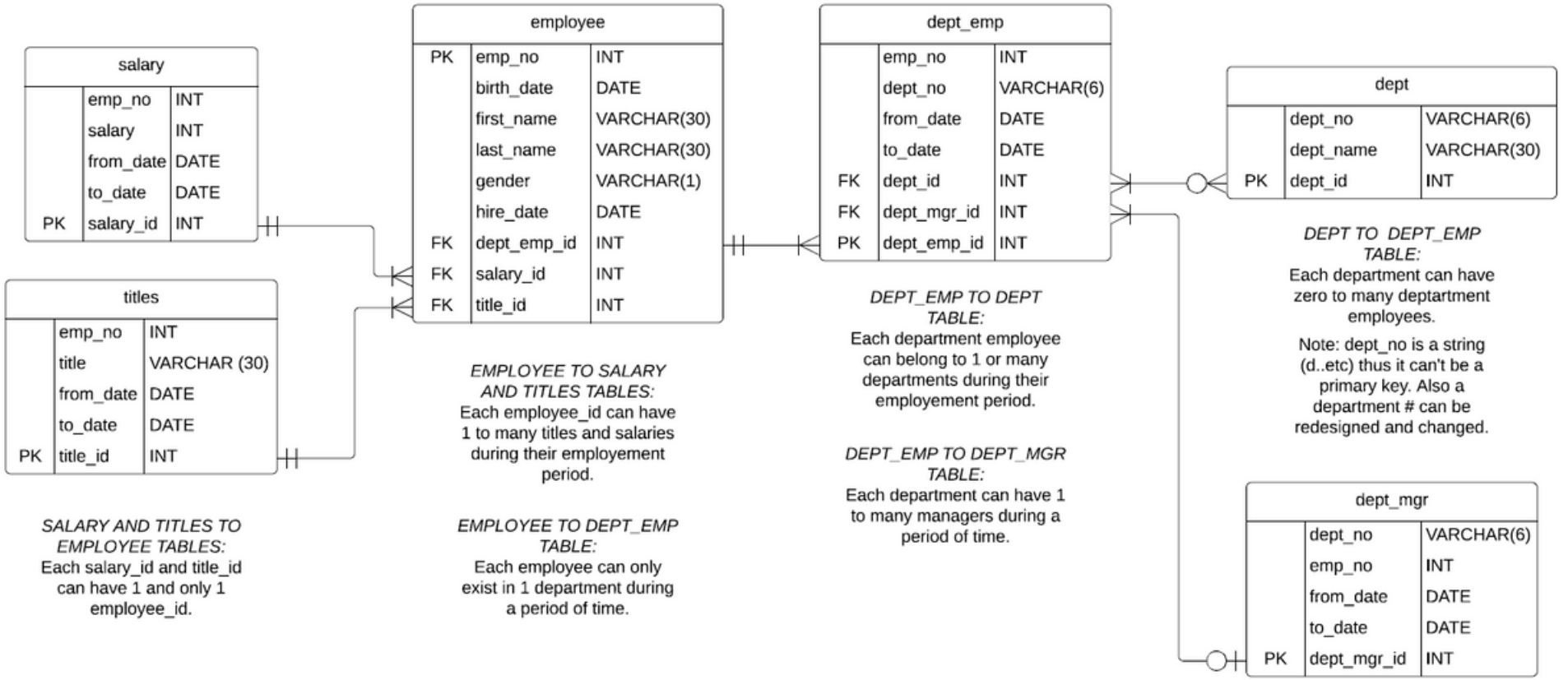
Third Draft

Spoke with the TA about my ERD on Saturday and my note was to simplify my tables. I could accomplish more with less and that the junction table wasn't necessary.

I wasn't sure how the queries would be run and what was the objective. I though each new query would be used to make a new table or CSV. Overall, still wasn't clear on the follow through.

SQL Homework

Josefina Amaro | July 1, 2019



SALARY AND TITLES TO EMPLOYEE TABLES:
Each salary_id and title_id can have 1 and only 1 employee_id.

EMPLOYEE TO SALARY AND TITLES TABLES:
Each employee_id can have 1 to many titles and salaries during their employement period.

EMPLOYEE TO DEPT_EMP TABLE:
Each employee can only exist in 1 department during a period of time.

DEPT_EMP TO DEPT TABLE:
Each department employee can belong to 1 or many departments during their employement period.

DEPT_EMP TO DEPT_MGR TABLE:
Each department can have 1 to many managers during a period of time.

DEPT TO DEPT_EMP TABLE:
Each department can have zero to many deptartment employees.

Note: dept_no is a string (d..etc) thus it can't be a primary key. Also a department # can be redesigned and changed.

DEPT_MGR TO DEPT_EMP TABLE:
Each employee can be zero t[illegible]ne manager a[illegible] time.

NOTES
Composite Primary Keys:
Since the from date could be changed due to an adjustment, I decided to use a serial id instead.

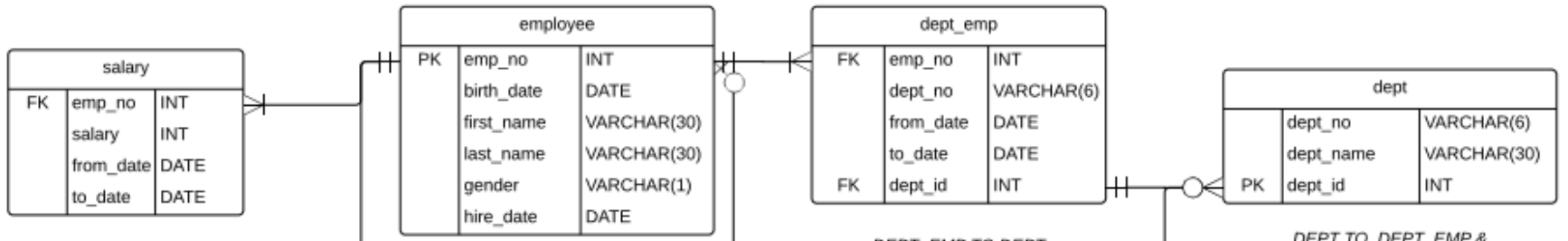
Fourth Draft

Once I started to write my queries I realized my PKs were not nessecary due to the relationships between emp_no.


300
YEARS
LATER

SQL Homework

Josefina Amaro | July 3, 2019



SALARY TO EMPLOYEE TABLE:

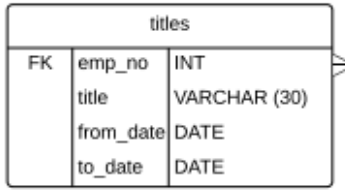
Each salary can have 1 to many emp_no instances during their time of employement.

EMPLOYEE TO SALARY AND TITLES TABLES:
Each emp_no can only have 1 salary & title during a period of time.

DEPT_EMP TO DEPT TABLE:
Each department employee can belong to 1 or many departments during their employement period.

DEPT TO DEPT_EMP & DEPT_MGR TABLE:
Each department can have zero to many deptartment employees and managers.

Note: dept_no is a string (d..etc) thus it can't be a primary key. Also a department # can be redesigned and changed.

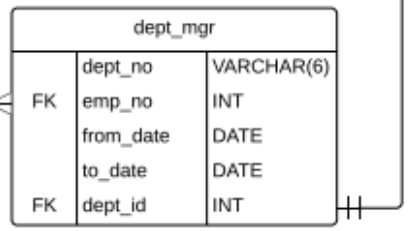


TITLE TO EMPLOYEE TABLE:
Each emp_no can have 1 to many titles during thier time of employement.

EMPLOYEE TO DEPT_EMP TABLE:
Each employee can only exist in 1 department during a period of time.

EMPLOYEE TO DEPT_MGR TABLE:
Each employee can be zero to one manager at any given time.

DEPT_EMP to DEPT TABLE:
Each dept_emp can only be in 1 department during a period of time.



DEPT_MGR TO EMPLOYEE TABLE:
dept and can 1 to many mangers during a period of time.

DEPT_MGR TO DEPT TABLE:
Each dept_mgr can only have 1 dept_id during a period of time.

NOTES

Composite Primary Keys:
The suggestion arose to make the from date and the emp_no a composite key.

However,
1. I didn't see the functional aspect of doing that for this example, since there is emp_no and dept_no in every table. Queries just have to written to their respective tables with the perspective of the desired result.

2. I made serial_id for each table to establish a primary key for each table. But, again it didn't really serve a functional purpose since we did have emp_no and dept_no that were unique.
Also, to summarize the relationships a junction table appeared to be a solution, but it just added complexity without functionality.

Thus I kept these tables simple.

Finale