



ITESM & Trilogy

Alumno:

Ing. Juan Ramón Félix Álvarez Tostado

Profesores:

Mr. Alejandro Estrella Gabilondo

Mr. Jorge Luis Ramos Zavaleta

SQL Home work 7

Data Analytics Bootcamp

27 de Junio de 2020

---For a better representation and understanding of the Keys and junctions I have use the following command in quickdatabasediagrams ---

#ERD

Titles

-

emp\_title\_id INT PK FK - Employees.emp\_title\_id

title VARCHAR

Salaries

-

emp\_id\_s INTEGER FK - Employees.emp\_no

salary INTEGER

Manager

-

dept\_no\_m INTEGER FK - Departments.dept\_no

emp\_id\_m INTEGER FK - Employees.emp\_no

Employees

-

emp\_no INTEGER PK

emp\_title\_id INTEGER

bith\_date DATE

first\_name VARCHAR

last\_name VARCHAR

sex VARCHAR

hire\_date DATE

Departments

-

dept\_no INTEGER PK

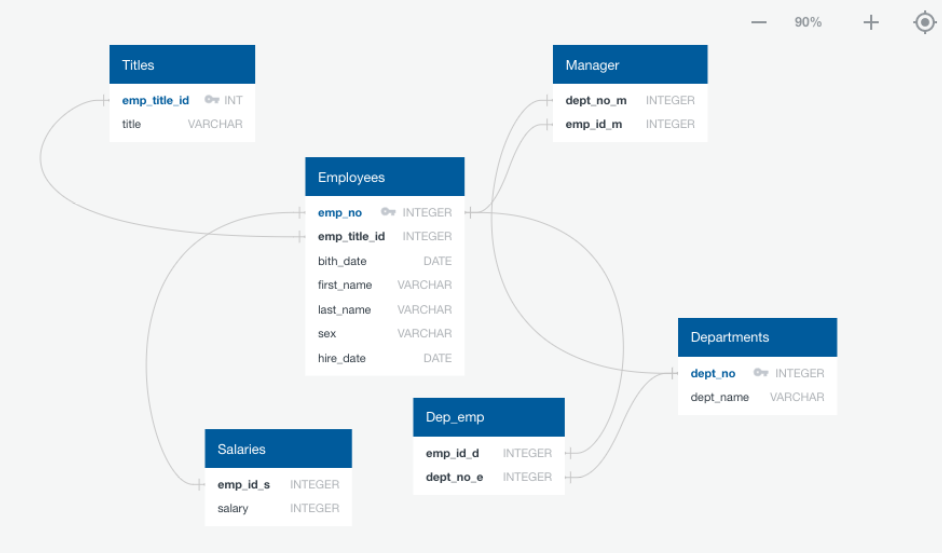
dept\_name VARCHAR

Dep\_emp

-

emp\_id\_d INTEGER FK - Employees.emp\_no

dept\_no\_e INTEGER FK - Departments.dept\_no



Once I have knowledge of how the junctions and keys need to be made I have created the tables.

--The first table was the Titles as this one has a primary key that is related to employee—

--This one is because this id is a unique information form the data and that will be used for other tables there for I need it to be stable as unique and primary key—

CREATE TABLE Titles (

emp\_title\_id VARCHAR unique,

title VARCHAR,

PRIMARY KEY (emp\_title\_id)

);

--The second Table was employee as this one is the one with more information and with more junctions with the other tables-- --This one is because this id is a unique information form the data and that will be use for other tables there for I need it to be stable as unique and primary key—

--This one is the foreign key of the table title and is reference to that table --

CREATE TABLE Employees (

emp\_no Int unique,

emp\_title\_id VARCHAR REFERENCES Titles(emp\_title\_id),

birth\_date VARCHAR,

first\_name VARCHAR,

last\_name VARCHAR,

sex VARCHAR,

hire\_date VARCHAR,

PRIMARY KEY(emp\_no)

);

--The third table is the Departments table this one is the last one to have a primary key—

--This one is because this id is a unique information form the data and that will be use for other tables there for I need it to be stable as unique and primary key—

CREATE TABLE Deparments (

dept\_no VARCHAR unique,

dept\_name VARCHAR,

PRIMARY KEY (dept\_no)

);

--The fourth table is the Department and Employee definition this one has two relations with foreign keys first the employee id and secondly the department id--

CREATE TABLE Dep\_emp (

emp\_id\_d INT REFERENCES employees(emp\_no) ,

dept\_no\_e VARCHAR REFERENCES Deparments(dept\_no)

);

--The fifth table is the Salaries this one has one relation with a foreign key the employee id--

CREATE TABLE Salaries (

emp\_id\_s INT REFERENCES Employees(emp\_no),

salary INT

);

--The six table is the one of the Salaries this one has a relation with the foreign key the employee id--

CREATE TABLE Salaries (

emp\_id\_s INT REFERENCES Employees(emp\_no),

salary INT

);

--The last table is the one of the manager in each division or department this one has two relations with foreign keys the employee id and the department id--

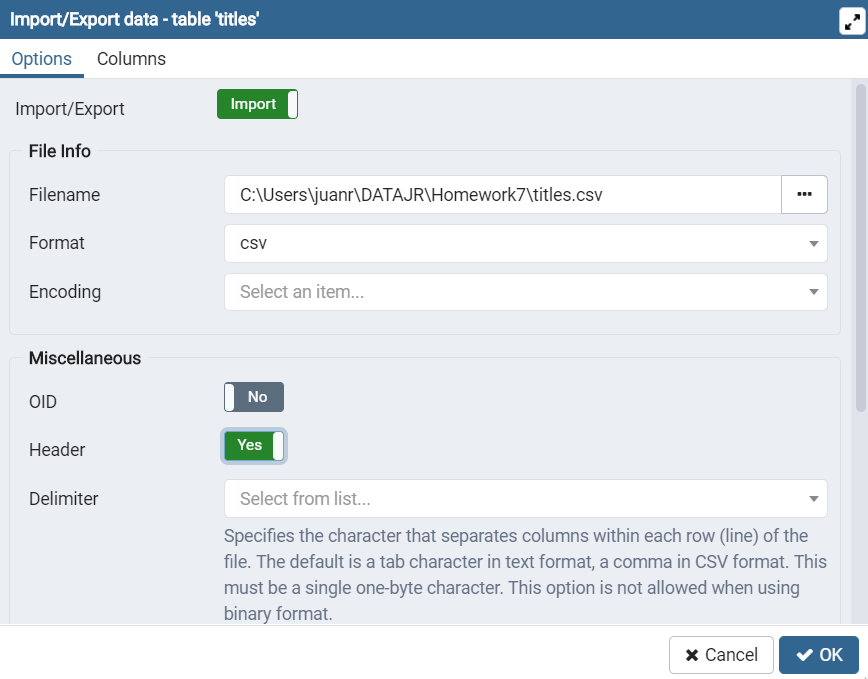
CREATE TABLE Manager (

dept\_no\_m VARCHAR REFERENCES Deparments(dept\_no),

emp\_id\_m INT REFERENCES Employees(emp\_no)

);

Once I have crated the tables I have upload the information of its corresponding CVS.



--After I have uploaded the information I notice that my dates information in the CVS where not legible and in order to be able to use them I have altered the Table of employees so the Dates are in their corresponding data type--

ALTER TABLE employees

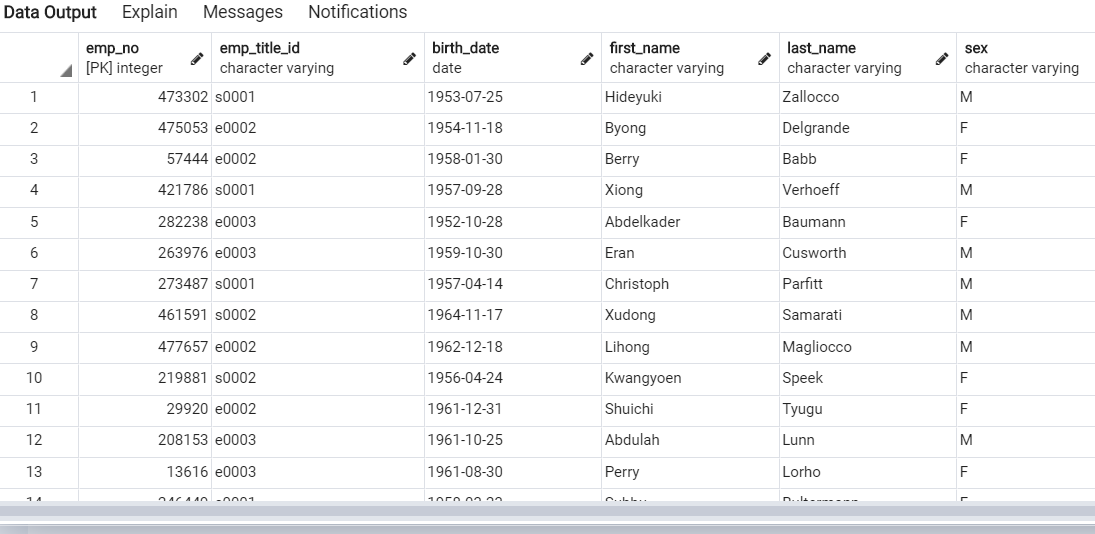
ALTER COLUMN birth\_date TYPE DATE USING(to\_date(employees.birth\_date,'mm/dd/yyyy')); salary INT

ALTER TABLE employees

ALTER COLUMN hire\_date TYPE DATE USING(to\_date(employees.hire\_date,'mm/dd/yyyy'));

SELECT \* FROM employees;

--Here you can see the dates in their corresponding order ‘YYYY-MM-DD’ to difference of the CVS that was ‘MM/DD/YYYY’--



--For each of the list I have created a view in order to have track of the task that has been done and to have the image of those nee tables—

--The first list was employee number, last name, first name, sex, and salary. I use the tables employees and salaries and join them together extracting only the required columns--

CREATE VIEW List\_1 AS

SELECT employees.emp\_no,

employees.first\_name,

employees.last\_name,

employees.sex,

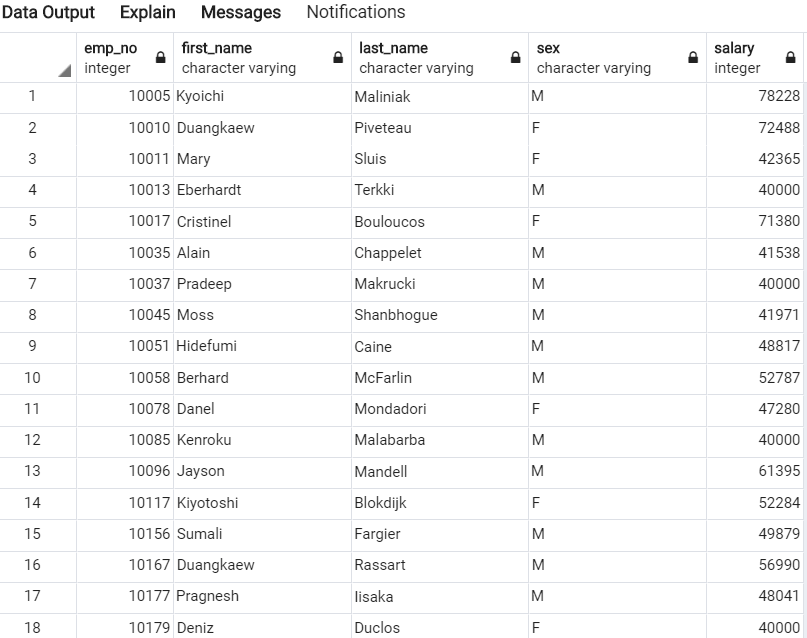
salaries.salary

FROM employees

INNER JOIN salaries ON

employees.emp\_no = salaries.emp\_id\_s;

SELECT \* FROM List\_1;



--The second list was employees who were hired in 1986. I use the table employees and use the condition of hire\_ date Between '1986-01-01' and '1986-12-31' in other to get employees hire in that year--

CREATE VIEW List\_2 AS

SELECT employees.first\_name,

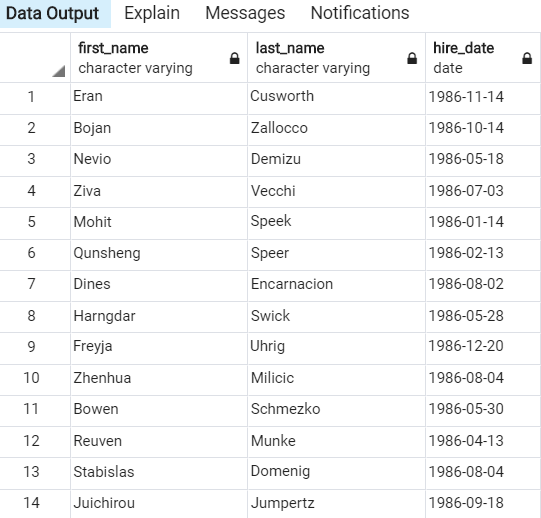
employees.last\_name,

employees.hire\_date

FROM employees

WHERE hire\_date BETWEEN '1986-01-01' and '1986-12-31';

SELECT \* FROM List\_2;



--The third list was department number, department name, the manager’s employee number, last name, first name. I Have use two views for this one first I have make a join between manger table and department and then I have used that view to join later with employees to get the names--

CREATE VIEW List\_3 AS;

SELECT manager.dept\_no\_m,

manager.emp\_id\_m,

deparments.dept\_name

FROM manager

INNER JOIN deparments ON manager.dept\_no\_m = deparments.dept\_no;

CREATE VIEW List\_4 AS

SELECT List\_3.dept\_no\_m,

List\_3.emp\_id\_m,

List\_3.dept\_name,

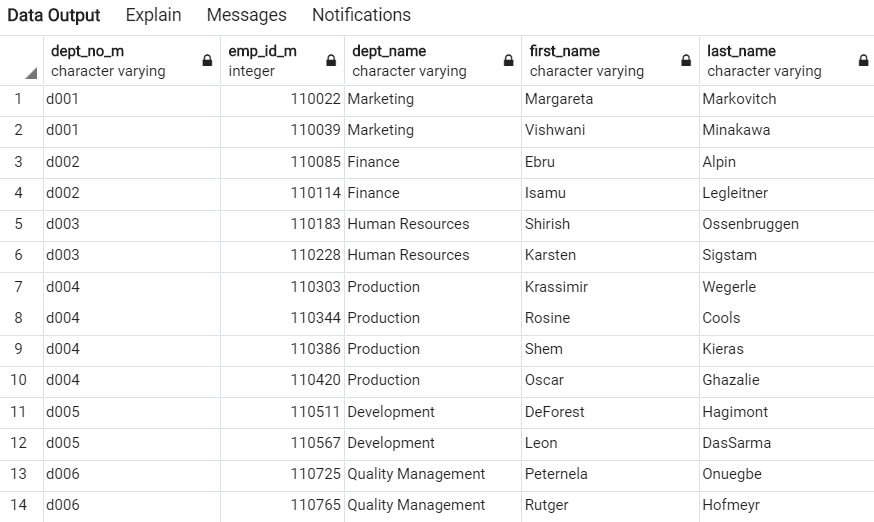
employees.first\_name,

employees.last\_name

FROM List\_3

INNER JOIN employees ON List\_3.emp\_id\_m = employees.emp\_no;

SELECT \* FROM List\_4;



--The fourth list was employee number, last name, first name, and department name. I Have use two views for this one too first I have made a join between table dep\_emp and department and then I have used that view to join later with employees to get the names--

CREATE VIEW List\_5 AS

SELECT dep\_emp.emp\_id\_d,

dep\_emp.dept\_no\_e,

deparments.dept\_name

FROM dep\_emp

INNER JOIN deparments ON dep\_emp.dept\_no\_e = deparments.dept\_no;

CREATE VIEW List\_6 AS

SELECT List\_5.emp\_id\_d,

List\_5.dept\_no\_e,

List\_5.dept\_name,

employees.first\_name,

employees.last\_name

FROM List\_5

INNER JOIN employees ON List\_5.emp\_id\_d = employees.emp\_no;

SELECT \* FROM List\_6;



--The fifth list was first name is “Hercules” and last names begin with “B.” for this one I used only the employees table and I have use a condition that first gives me all the Hercules and then I used AND to add the second condition that their last name stated with B using the %B code--

CREATE VIEW List\_7 AS

SELECT employees.first\_name,

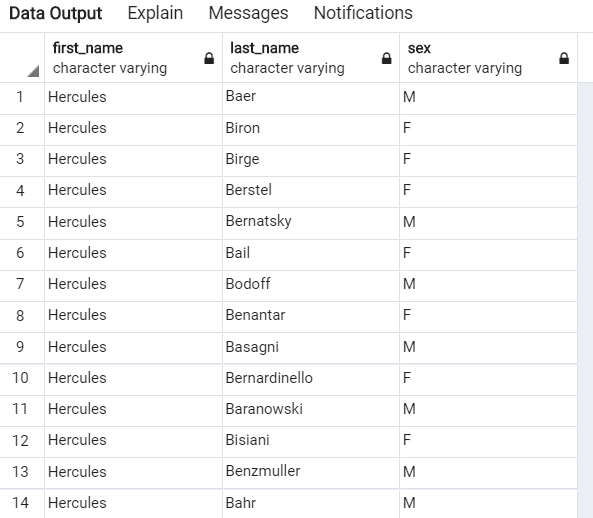
employees.last\_name,

employees.sex

FROM employees

WHERE first\_name = 'Hercules' and last\_name LIKE 'B%';

SELECT \* FROM List\_7;



--For list 6 I used the Create view of the list 5 as this one convenable has all the information that we required there for was more essay to make the extraction of the names and departments after that I have ose the condition that they are from the SALES department--

CREATE VIEW List\_8 AS

SELECT List\_6.emp\_id\_d,

List\_6.last\_name,

List\_6.first\_name,

List\_6.dept\_name

FROM List\_6

WHERE dept\_name = 'Sales';

SELECT \* FROM List\_8;



--For list 7 I have used the same code form above but I have added the OR in other to have the condition that the employees must be from either SALES or DEVELOPMENT--

CREATE VIEW List\_9 AS

SELECT List\_6.emp\_id\_d,

List\_6.last\_name,

List\_6.first\_name,

List\_6.dept\_name

FROM List\_6

WHERE dept\_name = 'Sales' or dept\_name ='Development';

SELECT \* FROM List\_9;



--For the Last list I have crated a count of the last names then group it by the last names and then order it by the count in descending order--

CREATE VIEW List\_10 AS

SELECT

last\_name,

COUNT (last\_name)

FROM

employees

GROUP BY

last\_name

ORDER BY count DESC;

SELECT \* FROM List\_10;

