

Homework #3

Complete By: Monday, March 11th @ 5:00pm

Submission: submitted via digitally via codio
You may submit a zip of files through Blackboard

Reading Tables

In a file named buildIndustry.sql, put the following commands.

Create a database named Industry if it does not exist. For each of the tables on the following page, drop them if they already exist, then create tables by those names with appropriate data types and cardinality restrictions. In particular, this means salary should be a number with two decimal places.

In a file named populateIndustry.sql, put the SQL necessary to populate the tables created in buildIndustry.sql with the data from the following page. This data has been included in codio in the file Industry Data.csv for your convenience, so that you can copy and paste instead of retyping the data. Instructions on how to load sql files into mysql have been included in Populate Database.txt, with the data from the University database used by the textbook given as an example. Note that the largeRelationsInsertFile.sql file takes a long time to load and should be preloaded in the database University, you may test the commands with the smallRelationsInsertFile.sql file.

Employee Table

PERSON_NAME	STREET	CITY
Jacob	West Taylor	Miami
Alexander	Loomis	Chicago
Jessica	Morgan	New York
Andrew	Polk	Chicago
Evan	Michigan	Austin
Nicholas	Roosevelt	Los Angeles
Amy	Roosevelt	Springfield
Cassy	Clinton	Nashville
William	Polk	Lincoln

Employment Table

PERSON_NAME	COMPANY_NAME	SALARY
Jacob	Amazon	15,000
Alexander	American Airline	9,500
Jessica	Amazon	12,000
Andrew	Starbucks	9,000
Evan	Facebook	18,000
Nicholas	First Bank	20,000
Amy	American Airline	9,000
Cassy	Central Bank	14,000
William	Nestlé	9,800

Company Table

COMPANY_NAME	CITY
First Bank	Miami
Amazon	Chicago
Apple	Los Angeles
Central Bank	Nashville
Nestlé	Lincoln
Facebook	Austin
Starbucks	Chicago
American Airline	San Francisco

For each of the remaining questions, put the answer in a file named Q#.sql, where # is the number of the question. If order is not specified, order results in alphabetical order (A-Z) for the first string attribute. Give the SQL queries that are equivalent to the following relational algebra expressions.

1. $\Pi_{PERSON_NAME}(Employee)$
2. $\Pi_{COMPANY_NAME}(\sigma_{SALARY > 10000}(Employment))$
3. $\Pi_{PERSON_NAME, CITY}(Employment \bowtie Company)$
4. $\Pi_{PERSON_NAME, COMPANY_NAME}(\sigma_{CITY = "Chicago"}(Employment \bowtie Employee))$
5. $\Pi_{PERSON_NAME, COMPANY_NAME}(\sigma_{CITY = "Chicago"}(Employment \bowtie Company))$

6. Write a SQL query which returns the employee name and company name for every currently employed person in the database.
7. Write a SQL query which returns the name and salary of each person recorded in the database, but only for salaries less than 10,000, ordered from highest to lowest salary.
8. Write a SQL query which returns the names of both the employees and the companies they work at, but only for employees who work at in the same city they live in. Employees live in the city indicated in the Employee table. Employees work at the city indicated in the Company table.
9. Write a SQL query which returns the names of both the employees and the companies they work at, but only for employees who work in a different city from the one they live in.
10. Write a SQL query which returns a table listing all known cities, both locations where people live as well as the cities where companies are headquartered.
11. Write a SQL query which lists the names of all companies along with the total salary that company pays. If a company has no known employees, the result should be NULL. Title the column containing the total salary as TOTAL_SALARY in the table your query generates.