

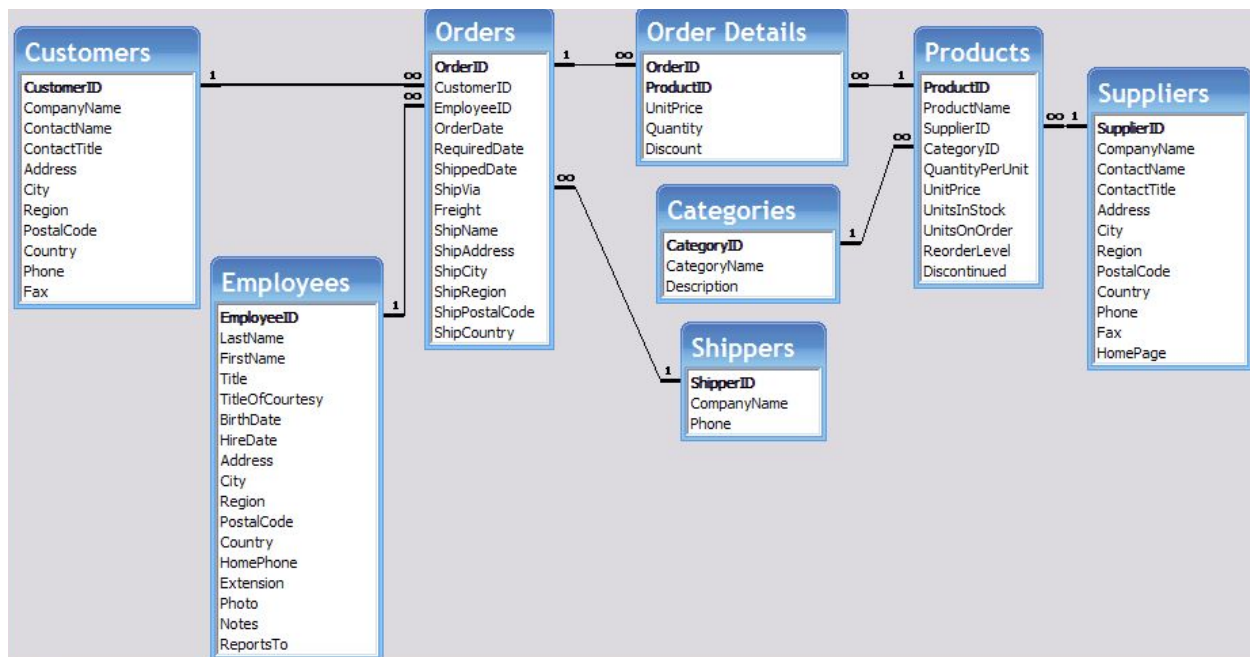
ORIE 3120: Industrial Data and Systems Analysis
Spring 2019
Homework #2

Due Date: 2/12/20 (Wednesday) 2:30pm

To submit, upload **1 file** to Gradescope containing the queries run (in a format that can be copied and pasted) and screenshots of the resulting tables.

1. Modifying a Relational Database Scheme

Recall this database schema from class, which could be used to keep track of orders from customers. Recall that boldface indicates a primary key. Also recall that a line marked by infinity indicates a foreign key whose corresponding column is indicated by the field linked by that line.



Describe how you would change this schema to keep track of multiple addresses for each customer. Your solution should allow a customer to have any number of addresses. Draw the portion of the schema that you are changing. Underline the primary key in any table you change, and clearly indicate any foreign keys with a line and the correct “1” and “infinity” labels.

2. Creating a Relational Database Schema

Draw a relational database schema to track the organizational structure, finances, and events for a student club. For example, you might consider tracking current and past members of the club, along with information about that member, when they joined, and whether they are currently active; official positions within the club and who has held those positions and when; events organized by the club, along with who organized the event, who attended the event, and

what items were purchased as part of that event; elections for officers of the club; and information about prospective members including a member contact person.

The schema must consist of at least four tables with a total of at least 20 fields (eg. five fields each), and there must be at least three one-to-many relationships illustrated. Label the "1" and the "many" ends of each relationship. The "1" typically belongs to the end that is a key for the corresponding table. Also, underline the primary key of each table.

Then, create tables in SQLite Studio that correspond to your schema, making sure to create foreign keys and primary keys that correspond to your schema. Populate each table with at least 4 rows.

Include:

- Your schema (you can draw this on paper and take a photo)
- A screenshot of the "Structure" page from SQLite Studio for each of your tables
- A screenshot of the "Data" page for each of your tables
- A screenshot of the "Constraints" page for each of your tables

To help the TAs understand the foreign key relationships in your schema, please include a 1-line description of each foreign key relationship. This is optional, but will help the TAs understand whether your foreign key relationships make sense.

3. Finding Valid Foreign Keys

We have two tables, Dept and Empl, with the following fields and records.

Dept

DeptID	Name	Manager
1	Finance	2
2	Ops	4
3	IT	1

Empl

EmplID	Name	Dept
1	Hiroki	3
2	Qing	1
3	JR	2
4	Jon	2

List the foreign keys that may be created in this database. For each, state the foreign key's table and field, and the referenced table and field. Do not include foreign key relationships from a field to itself.

4. Aggregation

Using the Orders table in the [Northwind database](#), linked in Resources on the class website, write a query that (1) calculates the number of orders handled by each employee, (2) the date of the earliest order, and (3) the date of the last order. Your results should be sorted in descending order of the number of orders handled, and employees should be listed by

Employeeid. The first few rows of your query result should look like the following, and the names of the fields should also match:

Employeeid	NumOrders	FirstOrderDate	LastOrderDate
4	156	2012-07-08	2014-05-06
3	127	2012-07-08	2014-04-30
1	123	2012-07-17	2014-05-06

5. Creating Histograms

We now show you some SQL queries that are very useful for creating histograms.

a. Write a single SQL query that counts up, for each whole dollar amount, the number of products whose unit price rounds to that amount. Sort your query result in ascending order of the rounded unit price. Use the Products table in the [Northwind database](#). The first few entries will look like this:

RoundedUnitPrice	N
3	1
5	1
6	1
7	2
8	1
9	2
10	6
12	1

b. Modify your query so that we count the number of products whose unit price falls into each of the following bins: [-5, 5), [5, 15), [15, 25), Output the center of each bin and the count. HINT: it may be useful to use “cast” to ensure that unitprice is treated as a floating point number: cast(unitprice as double). It may also be useful to use the round() function. Your result will look like this:

BinCenter	N
0	2
10	22
20	24
30	12
40	7
50	4
60	2
80	1
100	1
120	1
260	1

c. Using excel or a google sheet create a bar chart showing a histogram based on the results from part (b).