# 1 The Everything Store

Columbia Consulting Corporation has been hired by a former hedge fund analyst to create The Everything Store, an online store that will sell everything. The analyst provides Columbia Consulting the following requirements:

- Items have a description and a price.

- Customers have a name and an address.

- When customers create an order, the order must contain multiple items.

From this, they create the following SQL schema:

```
CREATE TABLE Items(
  iid int PRIMARY KEY,
  description text NOT NULL,
  price DECIMAL(8, 2) NOT NULL
);

CREATE TABLE Customers(
  cid int PRIMARY KEY,
  name text NOT NULL,
  address text NOT NULL
);

CREATE TABLE Orders(
  oid int PRIMARY KEY,
  cid INT NOT NULL REFERENCES Customers,
  date date NOT NULL
);

CREATE TABLE OrderLines(
  oid int REFERENCES Orders,
  iid int REFERENCES Items,
  quantity int NOT NULL,
  PRIMARY KEY(oid, iid)
);
```

They created this schema from the following Entity-Relationship Diagram:
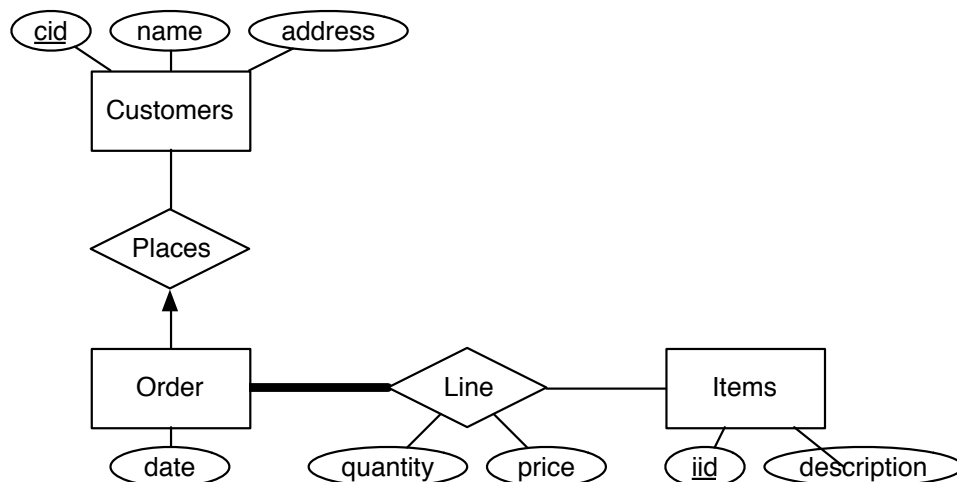
Q: What parts of the ER diagram do not match the SQL schema? For each error, provide an example that is permitted in one model but not in the other. (I found three errors, there may be more!)

Q: List three additional CHECK constraints might make sense for this schema. Describe in one sentence why they might be useful. (I can think of at least one for each table.)

Write SQL queries to answer the following questions. You can go to http://w4111db.eastus.cloudapp.azure.com:8111/orders to run queries on some sample data, or use the psql command with your project account: psql -U (uni) -h w4111db.eastus.cloudapp.azure.com orders. I *strongly* recommend trying to write the queries on paper first, then checking on the computer. This is better practice for exams.

Q: Compute the total number of items and the total dollar value of all items sold.

Q: What are the names of customers who purchased a Nexus 5X?

Q: Compute the products that are purchased together the most often. Rank the items by the number of times the two items appear together in the same order, ignoring the quantity. For example, if an order contains 100 "iPhone 6" and 1000 "MacBook Pro", that counts the same as an order that has 1 iPhone 6 and 1 MacBook Pro.

Q: For each item, compute the number of separate customers who have purchased each one.

Q: For each item, compute the number of orders that include that item, the total quantity purchased, and the total amount charged for them.

Q: Compute the average number of orders per customer.

Q: Which of these questions can be answered in the basic relational algebra model taught in class? Write a relational algebra expression to compute it.

Optional Q: This question is (in my opinion) harder than questions on the midterm. I was not able to write this without multiple attempts. For each customer, display the customer id and name, the total number of items ordered by that customer, and the total cost of those items for each customer.

Hint: What intermediate queries could you write to get you part of the way towards this? (Everyone should be able to write a query to get you part of the way towards this answer.)

2   The Oscars

Columbia Consulting Corporation has created a database to power the official Oscars web site (e.g. http://oscar.go.com/nominees). The want to model the acting awards, which have the following facts:

- Movies have a title. It is possible for multiple movies to have the same title (e.g. they get remade).

- People have names. It is possible for multiple people to have the same name.

- The Oscars has award categories with names like "Actress in a Leading Role" and "Actor in a Supporting Role".

- Each actor in a movie plays a role with a name, such as "Steve Jobs".

- For an award category in a year, there a set of people are nominated. These people are nominated for playing a specific role.

- For each award category in a year, one of the nominees can be marked as a winner.

To model this database, Columbia Consulting created the following database schema:

```
CREATE TABLE Movies(
  mid int PRIMARY KEY,
  title text NOT NULL
);

CREATE TABLE Category(
  cid int PRIMARY KEY,
  name text NOT NULL
);

CREATE TABLE People(
  pid int PRIMARY KEY,
  name text NOT NULL
);

CREATE TABLE Roles(
  pid int REFERENCES People,
  mid int REFERENCES Movies,
  name text NOT NULL,
  PRIMARY KEY(pid, mid)
);

CREATE TABLE Nominees(
  pid int,
  mid int,
  cid int REFERENCES Category,
  year int NOT NULL,
  won bool NOT NULL,
  PRIMARY KEY(pid, mid, cid),
  FOREIGN KEY (pid, mid) REFERENCES Roles
);
```

Q: Draw an entity-relationship diagram for this database.

Q: Describe one real world constraint for the acting awards that is not modeled by this SQL schema. Provide an example of how this is violated. Describe how you could correct the schema to fix this problem. (There are at least two, I'm hoping the class can help me find more!)

Write SQL queries to answer the following questions. You can go to to run queries on some sample data, or use the psql command with your project account: `psql -U (uni) -h w4111db.eastus.cloudapp.azure.com oscars`. I *strongly* recommend trying to write the queries on paper first, then checking on the computer. This is better practice for exams.

Q: Find the names of all movies nominated in 2016 in the database.

Q: Find the people nominated for "Actress in a Leading Role" in 2016 with the movie name and if they won. (Also: try querying for all combinations of Actor/Actress and Leading/Supporting roles)

Q: Find the year, award category, movie, and winning status of all of Jennifer Lawrence's nominations in order of oldest to most recent.

Q: Find the name and number of nominations for each movie in the database.

Q: Find the name and number of wins for each movie in the database.

Q: What is the maximum number of nominations for any movie?

The following questions are (in my opinion) harder than questions on the midterm. I was not able to write these without multiple attempts.

Optional Q: Find the name, number of nominations and number of wins for each movie.

Optional Q: Find the name(s) of the movie(s) with the most nominations in the database.