Warmup #1  (Textbook Problem 7.14)
Explain the distinctions among the terms primary key, candidate key, and superkey.

Warmup #2  (Textbook Problem 7.29)
Explain the distinction between total and partial constraints.

Warmup #3  (Textbook Problem 8.9)
Given the database schema $R(a, b, c)$, and a relation $r$ on the schema $R$, write a SQL query to test whether the functional dependency $b \rightarrow c$ holds on relation $r$. Also write an SQL assertion that enforces the functional dependency; assume that no null values are present.

Warmup #4  (Textbook Problem 7.20)
Consider the E-R diagram in Figure 7.29, which models an online bookstore.

a List the entity sets and their primary keys.

b Suppose the bookstore adds Blu-ray discs and downloadable video to its collection. The same item may be present in one or both formats, with differing prices. Extend the E-R diagram to model this addition, ignoring the effect on shopping baskets.

c Now extend the E-R diagram, using generalization, to model the case where a shopping basket may contain any combination of books, Blu-ray discs, or downloadable video.
Problem 5 (To Be Graded)

In this problem you are going to design a relational database for a retailer. The retailer has the following requirements:

- The retailer has a number of regions, which represent different distribution points. Every region has a unique id, name and address (containing street, city, state and zip code).

- Every region serves a number of customers. Every customer has a unique id, first and last name, address (street, city, state and zip code) and account balance. Every customer belongs to a single region.
• Customers can make multiple orders. Every order has a unique id, date, status ("Processing", "Paid", "Delivered", etc.).

• Every order contains one or more items, where each item represents something that every region sells. Each item has a unique id, name and price. An order can contain more than one of the same item. At this point you can assume that items are the same for every region.

1. Assuming the requirements above construct an E-R diagram and a relational schema for the database.

The retailer decided to change requirements as follows:

• Every region can have different items for sale now. The basic properties of the items remain the same (unique id, name, price), but now every region has its own listing of items and different regions might have different quantities for the same item in stock. Different items from the same order might be fulfilled by different regions.

• For analytical purposes the retailer decides to keep some historical information about every order. Every historical record contains information about the order, the customer, the total amount and the date of the order.

2. Describe the changes you have to make to the E-R diagram and the schema to accommodate the new requirements.

Note: As always, multiple designs are possible, but you should try to come up with a “better” one in a general sense. For instance, it is a better practice to avoid information duplication in a relation. Thus, putting all information about an order, including all its items, in a single relation is a bad idea.

Problem 6 (To Be Graded)

Following is the schema for the database of music festivals:

• musician(mid, name, address)
• musician_in_band(mid, bid, instrument)
• band(bid, bandname, home_state, home_zip)
• shows(sid, bid, ticket_price, qty_sold, total_profit)

Where:

• total_profit is the total amount of money that the show generated
• qty_sold refers to the number of tickets sold at the given ticket_price (each show only has a single ticket_price initially).
• Every show contains exactly one band, but bands may play in multiple shows.

1. Assume the following tuples in musician_in_band:

<table>
<thead>
<tr>
<th>mid</th>
<th>bid</th>
<th>instrument</th>
</tr>
</thead>
<tbody>
<tr>
<td>m1</td>
<td>b2</td>
<td>&quot;Fiddle&quot;</td>
</tr>
<tr>
<td>m1</td>
<td>b3</td>
<td>&quot;Guitar&quot;</td>
</tr>
<tr>
<td>m2</td>
<td>b3</td>
<td>&quot;Guitar&quot;</td>
</tr>
</tbody>
</table>

What are the primary keys, candidate keys and superkeys in musician_in_band? Please list all possibilities.

2. Assume the following requirement for band: a state will contain multiple zip codes, but a zip code will always be in the same state. Different bands can have the same name, the same states, or the same zip code. What additional functional dependencies does this new requirement bring?