Each question is worth 30 points. You may work with one partner and sign both of your names to your paper. Homeworks should be legible.

You are asked to design the database for a shipping application of a manufacturer. An order consists of a set of line items all directed to a particular customer address. Each line item consists of a particular item (e.g. a red polo shirt size large) and a quantity. An item will appear only once in the order. An instance of an item is at one location (a factory, warehouse, or customer site). However, different instance of an item could be in several locations. For example, there could be several warehouses with red polo shirt size large. Each shipment takes a set of item instances from one location to another. A shipment pertains to a particular order.

1. Design the tables for this application. Describe the functional dependencies you assume. Ensure the tables are in third normal form. You should not need to do a formal decomposition for this purpose. That comes later in the homework.

2. Write a query (or sequence of queries) in SQL to discover where every item instance (e.g. a particular red polo shirt) in an order is at a particular time. An item instance may be in a location or in a shipment. Note that this may entail your adding fields to the tables. For example, you have to distinguish between items (large red polo shirts) and instances of those items (large red polo shirt number 27). That is, a particular item instance identifies exactly one physical shirt, so if there are 2000 large red polo shirts, then each one will have a different instance id.

3. Assume the existence of a table distance(loc1, loc2, dist) which tells the minimum distance between every possible loc1 and loc2. (Perhaps you get this from a mapping program.) Given a ship-to location x and an order o that has been translated down to item instances, write a query that finds the item instance(s) in the order (i.e. the instances are not yet in any shipment) whose distance to x is greatest.

4. Suppose we are given five attributes: A B C D E and the following functional dependencies.
   \[ A \rightarrow C, \quad B \rightarrow C, \quad A \rightarrow E, \quad CE \rightarrow D, \quad DE \rightarrow C, \quad CE \rightarrow A. \]
   - Find a minimal equivalent set of functional dependencies. (You need not use any special algorithm, though you should be able to follow along based on the notes.)
   - Find a key of A B C D E given these functional dependencies.
   - Find a lossless, dependency-preserving third normal form decomposition of these attributes.
   - Answer part a again if we add the dependency \( C \rightarrow E \).

5. In our quest for canonical covers, we proceeded as follows (in my preferred variant of the algorithm): Given a set of FD’s with only one attribute on the right hand side of any FD, repeat the following two steps until there is no change:
eliminate all redundant FD’s,
remove all extraneous left hand side attributes.

Give an example showing that the second step can introduce new redundant FDs (and is therefore useful, because those discovered FDs can then be removed).

6. You are given the task of designing a relational database for hotel reservations. Each hotel has several room types (e.g. singles, doubles, doubles with views). For that hotel and room type, there are a certain number of rooms available and the room type has a description for that hotel. (Thus the same room type may have a different description for different hotels.) Hotel reservations reserve a certain number of rooms of each type for certain dates. Numberavailable is the number of rooms of that type in the hotel and is constant over time. Numbertaken is the number taken of that type for the hotel for a particular date. (So Numbertaken is always less than or equal to Numberavailable.)

Here is the initial schema:

hotelroomres( hotelid, roomtypeid, description, numberavailable, date, numbertaken)

Find the functional dependencies and decompose this table if necessary into a third normal form design. Identify the keys. Explain the advantages of your decomposition.

7. Now, use XQuery and FLWOR expressions to formulate the following query from hispo.xml:
Compute the total sales per partNum. (10 points extra credit if you run it and show a printout of the run).