1. How many records (rows of raw data) does the above table store, and how many fields (columns or attributes) are in each record? (2 Pts)

7 rows

5 column attributes

2. What problem would you encounter if you wanted to list the records in order of the manager’s last name, or if you sometimes wanted to omit the first name or middle name in a display or printout? This design fault is referred to as a **composite attribute**. Show the table structure of an altered table that will correct this problem? Show all columns and rows of raw data in this revised table. (2 Pts)

The first, middle and last name are all in a single field making them all part of a composite attribute for name.

Therefore before sorting by last name we would probably want to recreate/redesign the table columns as follows:

Column Fields – PROJECT\_CODE

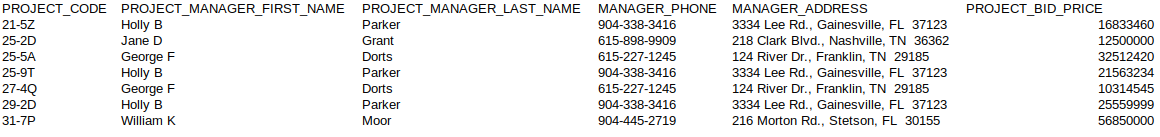
PROJECT\_MANAGER\_LAST\_NAME

PROJECT\_MANAGER\_FIRST\_NAME

MANAGER\_PHONE

MANAGER\_ADDRESS

PROJECT\_BID\_PRICE

The newly revised table with data should look something like this:

3. What problem would you encounter if you wanted to list the records in order of the street address, city, state, or zip, or area code? Building upon the improvements that you’ve already made, show the table structure of an altered table that also corrects this problem? Show all columns and rows in this revised table, including the new ones from Step 2.

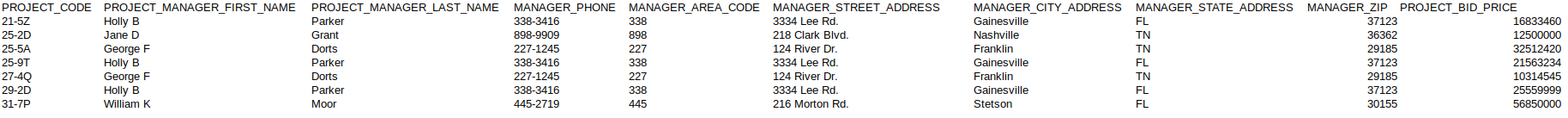
(3 Pts)

Similarly a redesign would improve the situation for these also composite attributes (address, phone)

The new column fields should contain at the very least the following:

MANAGER\_STREET\_ADDRESS, MANAGER\_ADDRESS\_CITY, MANAGER\_ADDRESS\_STATE, MANAGER\_ADDRESS\_ZIP, MANAGER\_AREA\_CODE.

E.g. Revised Table:



4. What data redundancies do you detect; i.e., what unnecessary repetitions are occurring? How could these redundancies lead to update anomalies, delete anomalies, or insert anomalies? (2 Pts)

The data redundancies in this table fall under the three categories described (update/delete/insert)

Update Anomaly exists in the table when data is entered multiple times for the same manager. This redundancy may create a need to go back and change the data in multiple places for for the same person.

Deletion anomaly is present (not really associated with the redundancy though) in which manager data could be removed from the database when removing a project from the table.

Insert anomaly is present as well in the sense that creating a place for a new manager here would require creating a project for that manager which may or may not actually be the case in reality, but would possibly be required. (Question - could this depend on whether or not null is allowed for Project Code?)

5. Using two relational tables, PROJECT and MANAGER, eliminate the redundancies you identified in Problem 4. Create a ManagerID column in both tables so you can link the two tables with the ManagerID being the primary key in MANAGER and a foreign key in PROJECT. Identify the primary key in each table. With words, show how the two tables join together by a foreign key that references a primary key. A format that would be useful is

Tablename.Columnname references Tablename.Columnname.

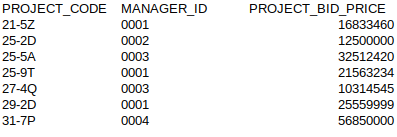
*Foreign Key Primary Key*

In this problem, show the column names across the top of each table and the rows of raw data below the column names. The columns must correct all faults (composite attributes and redundancies) that you saw above. (5 Pts)

PROJECT.MANAGER\_ID references MANAGER.MANAGER\_ID

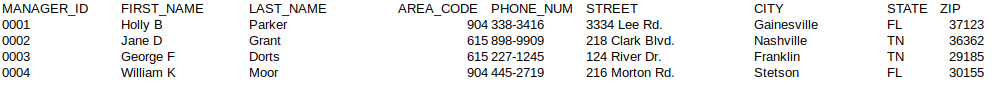
*Foreign Key* *Primary Key*

Project Table: PK FK



Managers Table:

PK



In the 2 tables redundancy has been reduced and all relevant information remains with accuracy here.