**Business Data Management**

**Assignment I**

**Due Date: Oct 6, 2021**

**FIGURE P1.1 The File Structure for Problems 1-3**



**1. (5 pts) How many records does the file contain? How many fields are there per record?**

**2. (10 pts) What problem would you encounter if you wanted to produce a listing by city? How would you solve this problem by altering the file structure?**

**3. (10 pts) What data redundancies do you detect? How could those redundancies lead to**

**anomalies?**

**4. (10 pts) Identify and discuss the serious data redundancy problems exhibited by the file structure shown in Figure P1.5.**

**FIGURE P1.5 The File Structure for Problems 4-5**



**5. (10 pts) How would you reorganize the data – i.e., what new files should you create to help eliminate the data redundancies found in the file shown in Figure P1.5?**

**6. (15 pts) Give an example of each of the three types of relationships (1:1, 1:N, M:N).**

**Using Figure P2.4 as your guide, work Problems 7–8. The DealCo relational diagram shows the initial entities and attributes for the DealCo stores, located in two regions of the country.**



**Figure P2.4 The DealCo relational diagram**

**7. (10 pts) Identify each relationship type and write all of the business rules.**

**8. (10 pts) Create the basic Crow’s Foot ERD for DealCo. (You can hand in a hand-drawn**

**diagram for this, but please make sure you have your name noted on the page, if you do so.)**

**9. (20 pts) Write the business rules that are reflected in the ERD shown in Figure P2.15.**

**(Remember that the ERD is always read from the “1” to the “M” side, regardless of the**

**orientation of the ERD components.) Will there be any problems in representing real-world data through this model (i.e., are any of the assumptions problematic in the real-world)?**

**FIGURE P2.15 The Crow’s Foot ERD for Problem 9**



1. There are 7 records and 5 fields per record.

2. (1)The detail of the MANAGER\_ADDRESS are concluded, and the value of city was in it. It is not convinent to get the value since the length of the stree and the number of strees varies from city to city. Secondly, we could use string match the first "," and the second "," , and get the value between the two comma. It is feasible in this file since there are few records but would costs unnecessary computer time.

(2)We can split the MANAGER\_ADDRESS to sevreal columns -- ST\_NUM, ST, CITY, STATE, ZIP.

3.(1)PROJECT\_MANAGER, MANAGER\_PHONE, MANAGER\_ADDRESS.

(2)If the manager change phone number or appartment and forget to report or the update of the list's all the record, could end up database anomalies. Same for other circumstances, such as the manager resigned or the project was assigned a new manager.

4. PROJ\_CODE, PROJ\_NAME, EMP\_NUM, EMP\_NAME, JOB\_CODE, JOB\_CHG\_HOUR, EMP\_PHONE. There various data redundancy could lead to the burden of the database capasity. And the relationship between employees and project is M:N, which is not acceptable by mordan relationship database.

5. We should create another table in between to help transfer the M:N relationship to 1:M:1

6. 1:1 person:SSN, 1:N employer:employee, M:N studet:class

7. REGION STORE EMPLOYEE JOB

REGION-STORE - 1:M, STORE-EMPLOYEE - 1:M, EMPLOYEE-JOB - 1:M

8.

[1] A region contains at least one or more stores.

Region must have a "REGION\_CODE" attribute.

Region should contain a descript.

REGION\_CODE is a foreign key.

[2] A store must have a "STORE\_CODE" attribute.

A store must belong to one and only one region.

A store contains at least one or more employees.

STORE\_CODE is a foreign key.

[3] Employee must have a "EMP\_CODE" attribute.

"Title" indicates the employee's position in the company.

A employee must belong to one and only one store which presented by the store code.

Employee has a job code to show his job type in the store.

Employee can only has one job type.

job code is a foreign key.

[4]job must have a "JOB\_CODE" attribute.

job must hava a base payment.

9.(1) All the relationship between entities are weak relationships.

[1] A publisher contain a attribute "p\_id" as a primary key.

A publisher has other attributes such as "p\_address", "p\_phone", "p\_fax", "p\_name", etc.

[2] A book contain a attribute "ISBN" as a primary key.

A book has other attributes such as "b\_name", "b\_category", "a\_id", "p\_id",etc.

"a\_id", "p\_id" are foreign keys.

The value of "a\_id", "p\_id" must no be empty.

A book has one and only one author and publisher.

[3] A author contain a attribute "a\_id" as a primary key.

A autho has other attributes such as "a\_lname", "a\_fname"

[4] A contract contain a attribute "c\_id" as a primary key.

A contract has other attributes such as "a\_id", "p\_id", "c\_price", "c\_s\_date",etc.

"a\_id", "p\_id" are foreign keys.

The value of "a\_id", "p\_id" must no be empty.

(2)

[1] A book can be published by sevreal publisheres, and written by several authors.

[2] Since the reason given by [1], the contracts could have more than one publisher and it is a M:N relationship. It can not be handled by the modern relationship database,in order to do so, wo shuold invole another table called submission to covert the M:N to 2 1:M relationship. Same as the tables of contract and author.