Washington State University

School of Electrical Engineering and Computer Science

CptS 451 – Introduction to Database Systems

Online

Dr. Sakire Arslan Ay

Homework-5

Design Theory and Normal Forms

Name: ______________________________

Student Number:______________________

Question:	Max points:	Score:
1	45	
2	55	
Total	100	

(45pts) Question1: Identifying Functional Dependencies

Your friend designed a simple database to keep track of the expected monthly sales in his small store. He created the following database:

```
MySales (pname, discount, month, price)
```

He inserted the projected data into the database, then he realized that there is something wrong with his table: it is difficult to update. He asked your help to fix his data management problems. He gave you the file `mySalesData.csv` and asked you to revise the table schema. You agreed to help and realized that you need to normalize his database. Unfortunately you cannot sit down and talk to him to find out what functional dependencies make sense in his business. Instead, you will reverse engineer the functional dependencies from his data instance. You should do the following steps:

a) Create a table “MySales “ in the database and load the data from the provided file into that table (use PostgreSQL). Please refer to the Milestone-1 description for instructions on how to import a .csv file in PostgreSQL.

b) Find all functional dependencies in the relation. This is a reverse engineering task, so expect to proceed in a trial and error fashion. First search for the simple dependencies, (e.g., one attribute determines other attribute(s)) then try the more complex ones, (e.g. two or more attributes determine some other attribute(s)), as needed. To check each functional dependency, you have to write a SQL query. Your challenge is to write this SQL query for every candidate functional dependency that you check, such that you can determine whether the FD holds or not by looking at the query's answer.

Try to be clever in order not to check too many dependencies, but don't miss potential relevant dependencies.

You should turn in all functional dependencies that you found, and for each of them the SQL query that discovered it.

c) Decompose the table to BCNF relations and create SQL tables for the decomposed schema. Create keys and foreign keys where appropriate. **Turn in the SQL commands for creating the tables.**

d) Populate your BCNF tables from the original data. For this you need to write SQL queries that insert the data into the tables you created at part (c) from the table you created at part (a). **Here, turn in the SQL queries that load the tables, and the number of tuples in each table after loading them** (obtained by running SELECT * FROM Table).

(55pts) Question2: BCNF Decomposition

Consider the following relations and functional dependencies. For each relation do the following:

a) Identify the minimal key(s) for the relation

b) Identify whether the relation is in BCNF, if not in BCNF decompose it into a set of BCNF relations.

c) If you decomposed the relation in part b, argue whether the composition preserved functional dependencies.

1. (30pts) **R(A,B,C,D,E,F)** and FD's **A** → **BC**, **D** → **AF**
2. (25pts) **S(A,B,C,D)** and FD's **ABC** → **D**, **D** → **A**

Submission Instructions:

Homework 5 will be submitted electronically on Blackboard to HW5-DROPBOX. You may either type/draw your HW in an editor or handwrite it and then scan it. Do not take pictures of your handwritten pages. Name your file CptS451_HW5_<yourname>.pdf Please submit only PDF files.