**Assignment3** **– Relational Databases**

Name: FUYIJUN

Student ID: 200552722

Group No.9

**Part 1:**

Step 1 – You have been provided a SQL script containing data for a table called my\_contacts. Import this data into a new database that you create for the purpose of this assignment – name the database appropriately. Please include the importation process in your GroupNumber\_Assignment3.sql file

Create a database -> use the database -> create the table 'my\_contacts' -> insert the data in it

To improve the efficiency of operation, I have imposed appropriate varchar character restrictions on some commonly used string data types (name, email, friends etc.). In addition, for the birthday column, I used date as the data type.

For more details information, please check my SQL. Document

A screenshot of a computer

Description automatically generated

Step 2 – Using one insert statement, add 10 more users to this table. Please devise fictitious contact information for this purpose please. Decide whether your import statement requires field mapping. If it does, include it, and if it does not, exclude it.

I used one insert statement to add 10 more users, since the values match the corresponding columns in right orders. There is no need for field mapping.

For more details, please check my SQL. Document

A screenshot of a computer

Description automatically generated

Step 3 – In the *my\_contacts* table you will find two columns that are violating First Normal Form by storing multiple values within each column.

Use your knowledge of working with string values, multi-table operations and joins to extract the data from *my\_contacts* and store the data according to this schema; there is an ERD provided below to assist you.

Please establish appropriate composite primary IDs for the junction tables, and otherwise use the ERD should explain the primary and foreign keys that need to be established.

Please remove any empty/unwanted columns after you have completed your work

**Execution: 18 marks Accuracy: 55 marks**

**Step3-1:**

Add primary key to my\_contacts: According to ERD, I should create an ID column and add the increment value automatically and set it as primary key. Here below picture shows this step runs successfully.

A screenshot of a computer

Description automatically generated

**Step3-2:** Creating a table named 'friends' to store unique friend names and populating the 'friends' table with distinct friend names from 'my\_contacts'.

A screenshot of a computer

Description automatically generated

**Step3-3:** Creating a table named 'contact\_friends' to establish a many-to-many relationship between contacts and friends. Using foreign keys to reference the 'my\_contacts' and 'friends' tables. Populating the 'contact\_friends' table with IDs mapping contacts to friends.

A screenshot of a computer

Description automatically generated

**Step3-4:** Creating a table named 'attributes' to store unique attribute names and populating the 'attributes' table with distinct attribute names from 'my\_contacts'

A screenshot of a computer

Description automatically generated

**Step3-5:** Creating a table named 'contact\_attributes' to establish a many-to-many relationship between contacts and attributes and using foreign keys to reference the 'my\_contacts' and 'attributes' tables. Then, Populating the 'contact\_attributes' table with IDs mapping contacts to attributes.

A screenshot of a computer

Description automatically generated

Here is how it looks like in my ERD:

A screenshot of a computer

Description automatically generated

**For more details, please check my SQL. Document. The order of executing each statement CANNOT be changed!**

**Part 2:**

See below 2 invoices that have been generated using data from the Georgian Dog Hospital. Please build a SQL script that will both create the database and insert the data that would have been used to generate the invoices. After the database is complete, export an Entity Relationship Diagram illustrating the data model you have developed showing all necessary tables, fields and relationships. Include the ERD in your report along with a summary of relevant design considerations.

**Before starting the code execution part, I need to explain the logic behind part two.**

1. Firstly, since the purpose of this part is to help the hospital manage its data, I believe that all raw data should be inserted only once to avoid mistakes.
2. Secondly, raw data typically should not be changed. However, if changes occur, the staff can run my code once instead of multiple times.
3. Thirdly, the tables I designed are not only intended to complete this assignment but also to be applicable in a real workplace. For example, In the invoice\_summary table, I am including both customer\_name and postal\_code. It's to handle cases where customers with the same name live in different places. (And I believe ‘invoice\_summary table’ is more useful than 'invoice\_details\_table'…)
4. Fourthly, the database of this assignment is too small. Some errors might happen if we have a hugh database. For example, if two customers have the same name, in my case, the invoice\_rawdata table is joining the data with the customer table based on both customer\_name and postal\_code to avoid ambiguity. This ensures that customers with the same name in different locations are treated as distinct entities. ( I believe these two persons are impossible to live together…)

**Here is my ERD:**

A screenshot of a computer

Description automatically generated

**Step1:** Create database and use the database.

A screenshot of a computer

Description automatically generated

**Step 2:**

-- Create a table called invoice\_rawdata to store raw invoice data -> insert data in it.

-- Why date is not DATE data type? - Because when I insert the date into the table, this format is not standard 'date' format.

-- Since most of the data are string, I set the data type as varchar. The varchar length is based on the information length.

A screenshot of a computer

Description automatically generated

Show the table:

A screenshot of a computer

Description automatically generated

**Step 3:**

-- Create the customer table and insert customer information.

-- Set AUTO\_INCREMENT PRIMARY KEY which is used as customer\_id

A screenshot of a computer

Description automatically generated

Show the table:

A screenshot of a computer

Description automatically generated

**Step 4:**

-- Create the procedure table and insert relevant data. This table contains the price of each procedure. It could be used to manage the price of each procedure.

-- Normally this table will be used by marketing or sales staff in a company. They may also make some pricing adjustment based on the market fluctuations.

A screenshot of a computer

Description automatically generated

Show the table:

A screenshot of a computer

Description automatically generated

**Step 5**

-- Create the pet table with additional columns for foreign keys, this table will be VERY improtant!

-- Insert data into the pet table from rawdata

-- Select distinct pet data along with corresponding customer and procedure information

-- To avoid two customers have the same name, I matched both customer\_name and postal\_code in 'invoice\_rawdata’

**A screenshot of a computer

Description automatically generated**

Show the table**:**

**A screenshot of a computer

Description automatically generated**

**Step 6:**

-- Create a table to summarize invoice data

-- The 'invoice\_summary' table calculates the total amount of an invoice, including amounts before and after tax.

-- This table is useful for the finance team to check the hospital's income per day. It provides a high-level overview

-- and is not meant to contain detailed information.

-- The 'postal\_code' is included in this table to handle cases where customers with the same name live in different places.

-- This is not redundant data; it serves as a reference for financial colleagues.

A screenshot of a computer

Description automatically generated

Show the table**:**

A screenshot of a computer

Description automatically generated

**Step 7:**

-- Create a table to store detailed invoice information

-- Populate the 'invoice\_details\_table' with data from the 'pet' table. This includes pet-specific information such as pet\_id, pet\_name, customer\_id, and procedure\_id

-- Use customer\_id to retrieve customer information

- Use procedure\_id to retrieve procedure information

- Use invoice\_id to retrieve invoice summary information

A screenshot of a computer

Description automatically generated

Show the table**:**

A screenshot of a computer

Description automatically generated**For more details, please check my SQL. Document. The order of executing each statement CANNOT be changed!**