LABORATORY MANUAL

CZ2007: Introduction to Databases

Software Lab 2 (Location: N4-01c-06)

or

Software Lab 3 (Location: N4-B1c-14)

Implementation of a Database Application

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
NANYANG TECHNOLOGICAL UNIVERSITY

1. OBJECTIVES

Upon completion of the assignment, the student should be able to:

a. Construct an entity-relationship model at a conceptual level
b. Map the model into a schema of a relational DBMS
c. Implement the given schema on a relational DBMS
d. Use a database language (SQL) for manipulating and updating data

2. LABORATORY

This is a group assignment. Each group consists of **six to seven** members from your laboratory group. You have the choice of selecting your group members. However, all the names within your group must be given to the laboratory technician(s) during your **first laboratory session**.

There are five scheduled lab sessions for this group assignment. Laboratory sessions will start from the **third week** of the semester onwards. For students whose lab sessions are scheduled on odd weeks, the first lab session is on Week 3. For students whose lab sessions are scheduled on even weeks, the first lab session is on Week 4.

You might need more than the scheduled five sessions for the actual implementation. You are encouraged to **start early** with your assignment (as soon as the topic is covered in the lectures).

Attendance is taken for all scheduled laboratory sessions. It is the responsibility of each student to sign-in at the beginning of each session. **Failing to sign-in for the first, third, or fifth lab session may result in a F grade for the respective assessment.**

3. INTRODUCTION

The assignment covers the portion of the course concerning data modelling, database design and implementation from the user's viewpoint. Thus, the assignment involves modelling as well as implementation aspects of the database course.

The overall aim of the laboratory is to develop an application based on a given data model using a given database management system. This exercise will bring you through a crucial first part of the life cycle of a database application. It is assumed that the data analysis has been performed. Note that this manual provides you with more information than is required for the first laboratory

session; e.g., not all constraints can be modelled in the beginning but are included at a later implementation stage. In contrast you might require additional information for an understanding of the application. Proceed by stating your assumptions in written form and / or ask your laboratory supervisor.

4. DESCRIPTION OF THE ASSIGNMENT

The description of the application is given in Appendix A and B. This includes background and general requirements of the application, conceptual information about the system and its users as well as a list of SQL queries that must be fulfilled as a minimum. Note that teamwork is required. Every group will submit one solution. **No individual submission will be accepted.**

4.1 First Laboratory Session: Creating an ER Diagram

Appendix A gives conceptual information about the project obtained after a partial system analysis was performed. Based on the description, construct a suitable ER diagram. Analyze the cardinality of relationships, the usage of weak entity sets, choice of entity sets, subclasses, etc. and compare them with alternative solutions from your group members. The laboratory technician will provide the necessary information at the beginning of the lab session.

You need to submit the following, latest **three working days** after the first laboratory session:

- A printed hard copy of your ER diagram and a written discussion of your solution (maximum one page) that highlights the reasons for the chosen design. Marks will be given for neat presentation of your ER diagram.

4.2 Second Laboratory Session: Finalization of the ER Diagram

There is no submission for the second laboratory session. In this lab, each group should finalize their database design based on the feedback received from their lab supervisor. Please note that the second laboratory session is a free-access session; i.e., attendance is not mandatory (but recommended in case the group has questions).

4.3 Third Laboratory Session: Generation of Normalized Database Schema

In this lab you must ensure that the database is at least in 3NF. Follow the general guidelines covered during the lectures and tutorials to produce suitable normalized relations. For each

relation, the key(s), primary key, and functional dependencies must be specified. If a relation is generated due to normalization of an original relation, then the normalization steps must be presented.

You need to submit the following, latest **three working days** after the third laboratory session:

- A printed hardcopy of the normalized database schema and FDs associated with each relation. If a relation created from the ER diagram violates 3NF, then this should be highlighted along with the decomposed normalized relations. Note that for this lab, no SQL code should be submitted. Hence, the structure of your solution shall be similar to the following example:

  R1(A, B, C, D)
  Keys: AB, AD
  Primary Key: AB
  FDs: AB → CD, A → D
  The relation is in 3NF.

4.4 Fourth Laboratory Session: Implementation of the database schema

There is no submission for the third laboratory session. In this lab, the finalized database schema must be implemented using SQL DDL commands. Your implementation should clearly incorporate the primary and foreign keys, data types, integrity constraints, value-based and tuple-based constraints. The lab provides the MS SQL Server software for your implementation. You are free to use other SQL database software for the implementation.

Please note that the fourth laboratory session is a free-access session; i.e., attendance is not mandatory (but recommended in case the group has questions).

4.5 Fifth Laboratory Session: Final demonstration

The fifth session is the final assessment of your implementation. The implementation obtained from the previous laboratory session must now be extended to include suitably populated tables to support the queries in Appendix B. Formulate your queries in SQL.

This session has two components. First, **at the beginning of the lab** a printed hardcopy of the schema implemented using the SQL DDL commands together with the populated tables and sample queries need to be submitted. Auto-generated relations are not permitted. Hence, the structure of

your solution for the database schema definition shall be similar to the following example and written by yourselves:

```
CREATE TABLE name (
        attr1 datatype NOT NULL,
        attr2 datatype,
        ...
        PRIMARY KEY (attr1),
        FOREIGN KEY (attr3) REFERENCES name(attr1)
                ON DELETE ... ON UPDATE ...,
);
```

The second component of this session involves **demonstration and Q&A** of your system. All group members are required to contribute actively during the demonstration session. Additionally, the laboratory supervisor will ask individual questions. During the demo session, the evaluation shall be based on the followings:

- Implementation and execution of additional queries on the spot
- Demonstration of understanding of design and related issues
- Demonstration of proper working of your implementation
- Additional effort in terms of implementation, etc.
- Presentation quality

Note that your group might be required to begin the presentation at any time during the fifth laboratory session; i.e., one group will be asked to present at the beginning of the session.

APPENDIX A: APPLICATION DESCRIPTION

Suppose you are asked to construct a database for Sharkee, an online shopping website. The requirements are as follows:

- The Sharkee website hosts a number of shops, each of which has a unique name. Each shop sells a number of products, each of which has a name, a category, a maker, a price, and the quantity in stock.
- One product could be sold at multiple shops at difference prices. In addition, the price of a product in a shop may change over time. We need to record the history of price changes.
- Sharkee allows users to place orders from the shops. Each user has an ID number and a name. Each order has an ID and timestamp. Each order involves one or more products, which could be from different shops. For each product involved in an order, its price and quantity are recorded. Each order has a total shipping cost and a shipping address.
- After an order is made, the user can track the status of the order on Sharkee. Initially, the status of each product in the order is shown to be "being processed". After the shop (that sells the product) ships the product, the status of the product will be changed to "shipped". Once a product is delivered to the user (as reported by the courier), the status of the product is changed to "delivered", and the delivery date is recorded. Within 30 days from the delivery date of a product, the user may return the product for a refund. Once the shop refunds the product, its status will be changed to "returned".
- After a user purchases a product, he/she is allowed to rate and comment on any product once. There are five possible ratings: 1, 2, 3, 4, and 5, with 5 being the highest. The average rating for a product, as well as the number of users that have rated the product, are shown on the web page that displays the product information to the users. In addition, a user can modify his/her ratings and comments anytime.
- Sharkee users are allowed to file complaints on any product and shop. For example, if a user does not receive a product that has been shown to be "delivered" in an order, he/she can file a complaint

to Sharkee. If he/she is not happy about a certain shop, he/she can file a complaint. After a complaint is filed, the user can check the status of his/her complaint. Initially, the status of the complaint is set to "pending". After the complaint is picked up by a Sharkee employee, the status is changed to "being handled", and the name of the employee is shown. Once the complaint is addressed, its status is changed to "addressed".

- Sharkee has a number of employees that handles complaints from users. Each employee has an ID, a name, and a monthly salary. Each complaint is handled by one employee.
- The database should support the queries listed in Appendix B.

Note that the provided information may not be complete. Some aspects of the database application's details may have been omitted. It is expected that you come up with their own solutions in case of inconsistencies or missing information. However, you have to keep track of these aspects and explain your assumptions in your submitted report. Extensions to the implementation of the basic system are encouraged.

APPENDIX B: QUERIES

1. Find the average price of "iPhone Xs" on Sharkee from 1 August 2020 to 31 August 2020.
2. Find products that received at least 100 ratings of "5" in August 2020, and order them by their average ratings.
3. For all products purchased in June 2020 that have been delivered, find the average time from the ordering date to the delivery date.
4. Let us define the "latency" of an employee by the average that he/she takes to process a complaint. Find the employee with the smallest latency.
5. Produce a list that contains (i) all products made by Samsung, and (ii) for each of them, the number of shops on Sharkee that sell the product.
6. Find shops that made the most revenue in August 2020.
7. For users that made the most amount of complaints, find the most expensive products he/she has ever purchased.
8. Find products that have never been purchased by some users, but are the top 5 most purchased products by other users in August 2020.
9. Find products that are increasingly being purchased over at least 3 months.