Information and Database Management Systems I (CIS 4301)
(Fall 2016)

Instructor: Dr. Markus Schneider

TA: Yang Chen

Homework 1 Solutions

Name: ____________________________

UFID: ____________________________

Email Address: ____________________

Pledge (Must be signed according to UF Honor Code)

On my honor, I have neither given nor received unauthorized aid in doing this assignment.

______________________________________________
Signature

For scoring use only:

<table>
<thead>
<tr>
<th></th>
<th>Maximum</th>
<th>Received</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exercise 1</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Exercise 2</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Exercise 3</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
Exercise 1 (Knowledge Questions) [30 points]

Provide precise and concise replies for the following statements and questions.

1. What is a database? What are its implicit properties? [4 points]
   A database is a collection of related data. A database has the following implicit properties:
   - Representing some aspect of the real world (called the miniworld). Changes to this miniworld are reflected in the database.
   - A database is a logically coherent collection of data with some inherent meaning. A random assortment of data cannot correctly be referred to as a database.
   - A database is designed, built, and populated with data for a specific purpose. It has an intended group of users and some preconceived applications in which these users are interested.

2. What is a DBMS? What are the processes that a DBMS facilitates? [2 points]
   A DBMS is a collection of programs that enables users to create and maintain a database. The DBMS is a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications.

3. What are the limitations of a file-based system? [2 points]
   Repeated occurrence of the same data in different files (redundancy), lacking logical concordance of file contents (inconsistency), changes of the file structure lead to changes of the application program, extensions of the functionality of an application program lead to new requirements of the file structure and to a restructuring of files (data-program dependence), analysis of data as well as the realization of new applications is problematic (inflexibility)

4. What is a DDL? What is a DML? Briefly explain them and give one SQL example of each. [4 points]
   DDL is short for Data Definition Language or Data Description Language. It is the language to manipulate a database schema. e.g.:
   ```sql
   CREATE TABLE Employee (  
     EmplId integer PRIMARY KEY,  
     Name varchar(25) NOT NULL);  
   ```
   DML is short for Data Manipulation Language. It is the query language for inserting, deleting and updating data in a database. e.g.:
   ```sql
   INSERT INTO Employee VALUES (567, 'Meyer');  
   ```

5. Explain the concept of physical data independence, and its importance in database systems. [2 points]
   Physical data independence is the ability to modify the physical schema without making it necessary to rewrite application programs. Such modifications include changing from sequential files to random access files and adding a field to every record in a table. An
application program’s view can hide the above modifications from the program.

6. What is a key? Should a key contain only a single value? If yes, explain why. If not, give an example [2 points].
A key is a minimal set of attributes whose values uniquely characterize the associated entity among all entities of its type. A key need not to contain only one single value, it can be a set of attributes. For example: {company_brand, sequence_number} can be a key to characterize a laptop.

7. Describe the levels of abstraction in a database. [3 points]
- External/view level describe the part of the DB, which is relevant for the user
- Conceptual/logical level gives information about existing data and relationships in the DB
- Physical/internal level describes how data are physically stored

8. What is the difference between procedural and nonprocedural DMLs? [4 points].
A high-level or nonprocedural DML can be used on its own to specify complex database operations concisely. Many DBMSs allow high-level DML statements either to be entered interactively from a display monitor or terminal or to be embedded in a general-purpose programming language.

A low-level or procedural DML must be embedded in a general-purpose programming language. This type of DML typically retrieves individual records or objects from the database and processes each separately. Therefore, it needs to use programming language constructs, such as looping, to retrieve and process each record from a set of records.

9. Describe 3 tables that might be used to store information in a social networking system such as Facebook. [3 points]
A User table containing records about users, with attributes such as account name, real name, age, gender, location, and other profile information.
A Content table containing user provided content, such as text and images, associated with the user who uploaded the content.
A Friend table recording for each user which other users are connected to that user.

10. What are the types of binary relationship sets? Explain them. [4 points]
- One-to-one(1:1)-relationship: if for a binary relationship set R(E1,E2) each entity in E1 is associated with at most one entity in E2, and vice versa.
- One-to-many(1:*)-relationship: if for a binary relationship set R(E1,E2) each entity in E1 is associated with any number of entities in E2, and each entity in E2 is associated with at most one entity in E1.
- Many-to-one(*:1)-relationship: if for a binary relationship set R(E1,E2) each entity in E1 is associated with at most one entity in E2, and each entity in E2 is associated with any number of entities in E1.
- Many-to-many(*:*)-relationship: if for a binary relationship set R(E1,E2) each entity in E1 is associated with any number of entity in E2, and vice versa.
Exercise 2 (Oracle) [30 points]

Consider the following table ‘Student’ maintained by a university.

<table>
<thead>
<tr>
<th>student_id</th>
<th>student_name</th>
<th>state</th>
<th>date_of_birth</th>
<th>account_balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td>Alex</td>
<td>FL</td>
<td>10/20/1989</td>
<td>+100.00</td>
</tr>
<tr>
<td>1002</td>
<td>Bob</td>
<td>FL</td>
<td>07/23/1992</td>
<td>-68.10</td>
</tr>
<tr>
<td>1003</td>
<td>Caitlyn</td>
<td>CA</td>
<td>03/15/1990</td>
<td>-32.56</td>
</tr>
<tr>
<td>1004</td>
<td>Dave</td>
<td>CA</td>
<td>06/07/1990</td>
<td>+54.50</td>
</tr>
<tr>
<td>1005</td>
<td>Emma</td>
<td>FL</td>
<td>09/08/1989</td>
<td>+89.99</td>
</tr>
<tr>
<td>1006</td>
<td>Freddie</td>
<td>CA</td>
<td>02/01/1989</td>
<td>-110.00</td>
</tr>
<tr>
<td>1007</td>
<td>Gus</td>
<td>NM</td>
<td>05/05/1988</td>
<td>+76.25</td>
</tr>
</tbody>
</table>

(“student_id” is the primary key)

Use your CISE Oracle account to create this table and perform the operations below. Provide SQL statements for all operations. Show the screenshots of query results from Oracle.

[5 points each]

(1) Create the Student table.
CREATE TABLE Student (student_id INTEGER PRIMARY KEY, student_name VARCHAR(255), state VARCHAR(255), date_of_birth DATE, account_balance NUMERIC(7,2));

(2) Insert the data into the table.
INSERT INTO Student VALUES (1001, 'Alex', 'FL', TO_DATE('10/20/1989', 'mm/dd/yyyy'), 100.00);
INSERT INTO Student VALUES (1002, 'Bob', 'FL', TO_DATE('07/23/1992', 'mm/dd/yyyy'), 68.00);
INSERT INTO Student VALUES (1003, 'Caitlyn', 'CA', TO_DATE('03/15/1990', 'mm/dd/yyyy'), -32.56);
INSERT INTO Student VALUES (1004, 'Dave', 'CA', TO_DATE('06/07/1990', 'mm/dd/yyyy'), 54.50);
INSERT INTO Student VALUES (1005, 'Emma', 'FL', TO_DATE('09/08/1989', 'mm/dd/yyyy'), 89.99);
INSERT INTO Student VALUES (1006, 'Freddie', 'CA', TO_DATE('02/01/1989', 'mm/dd/yyyy'), -110.00);
INSERT INTO Student VALUES (1007, 'Gus', 'NM', TO_DATE('05/05/1988', 'mm/dd/yyyy'), 76.25);
(3) Express the following colloquial queries in SQL.
(a) List the names of students from Florida.
SELECT student_name FROM Student WHERE state = 'FL';

– Return –
Alex
Bob
Emma

(b) List the names and states of students whose name begins with “A.”
Hint: There is an operator named *like* in SQL that can be used here. Look this operator up in your textbook, or search the Internet for it to understand its syntax and meaning.
SELECT student_name, state FROM Student WHERE student_name LIKE 'A%';

– Return –
Alex  FL

(c) For all students with a negative balance and born after 01/01/1990, add $100 to their balance.
UPDATE Student SET account_balance = account_balance + 100 WHERE account_balance < 0 AND date_of_birth > TO_DATE('01/01/1990', 'mm/dd/yyyy');

(4) Count the number of students from Florida or California.
SELECT COUNT(*) FROM Student WHERE state = 'FL' OR state = 'CA';

– Return –
6
Exercise 3 (ER Model) [40 points]

Consider the following requirements about an online movie review system:

- Every movie has a unique ID, name, year, length, brief description, and the url to its cover.
- Movies are directed and acted by celebrities. The database keeps the name and date of birth of celebrities to identify a celebrity.
- For a celebrity that acts in a movie, the database keeps what role he/she acts. A celebrity might act several roles in one movie.
- Production corporations produce movies. A production corporation has a name, a website, and an address. The address includes street, city, state, and zip code.
- Each online user must have an email address (which is used to log in), a password, and a name. They can also provide age and gender to the system.
- Online users can write reviews for movies. A review has a title, content, a score ranging 1-5, and a timestamp.
- Online users can also make friends. They can add one another as a friend. Friend is bi-directional, meaning “A is a friend of B” implies that “B is a friend of A”.

Design an Entity-Relationship diagram that models the online movie review system and takes into account the requirements listed above. That means that you have to identify suitable entity sets, relationship sets, attributes, keys of entity sets, and so on. Further add the cardinalities (1:1, 1:m, m:1, m:n) to the relationship sets.