Database Management Systems (COP 5725)
(Spring 2017)

Instructor: Dr. Markus Schneider

TAs: Keke Zhai, Man Mohan Devineni

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]

Please provide concise but precise answers.

1. What are the problems of early data processing? [6 points]
   + Redundancy: 1) repeated occurrence of the same data in different files; 2) waste of external memory, increased management and processing costs
   + Inconsistency: 1) lacking logical concordance of file contents; 2) especially caused due to changes
   + Data-program dependence: 1) data are directly created and accessed by an application program; 2) changes of the file structure lead to changes of the application program; 3) extensions of the functionality of an application program lead to new requirements of the file structure and to a restructuring of files
   + Inflexibility: analysis of data as well as the realization of new applications is problematic; 2) data from several files can only be combined with very high costs

2. List 3 advantages of database systems compared to file systems? [6 points]
   + Data independence: independence of application programs from the details of data representation and data storage
   + Efficient data access: 1) multitude of sophisticated techniques for the efficient storage of and efficient access to persistent data; 2) application of index structures
   + Common data basis: common data basis for all current and future application programs
   + Concurrent data access: 1) simultaneous access to the same data by different users; 2) each user gets the impression of exclusively accessing data; 3) concept of transaction for the synchronization of concurrent data accesses
   + Lacking or controlled redundancy: 1) avoiding copies of the same data by an integrated view on data; 2) controlled redundancy for improving performance
   + Consistency of data: 1) caused by lacking redundancy; 2) DBMS must ensure consistency of data for controlled redundancy
   + Integrity of data: 1) correctness and completeness of data (semantical aspect); 2) formulation of integrity constraints or integrity rules; 3) DBMS checks constraints for each insertion, change and deletion of data
   + Data security: 1) protection of the database against unauthorized access (view on data); 2) access control with authentication and encoding as possible protection mechanisms
   + Backup and recovery: protection against the consequences of system errors
   (Any 3 of the above advantages)

3. What is a database? What is a DBMS? What is the relationship of database and DBMS? [4 points]
A database is a integrated and structured repository of large collections of persistent data, which serves for all users of an application area as a common and reliable basis of up-to-date information.

A DBMS is a all-purpose software system, which supports the user in the definition, construction and manipulation of databases for different applications in an application-neutral and efficient manner.

Relationship: The DBMS is a set of programs for the management of and access to the data in the DB. It’s at software level between physical database and user.

4. What is the 3-level model of DBS? Explain it briefly. [3 points]
   +External/view levels 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.

5. Explain the concept of data independence, and give two examples that use this concept in DBS according to the question above and explain. [4 points]
   +Data independence denotes the property that higher levels of the model are not influenced by changes of lower levels.
   +Two examples: logical data independence and physical data independence
   +Logical data independence: changes of the conceptual schema (e.g., information about new types of entities, further information about existing entities) do not have impact on external schemas (e.g., existing application programs)
   +Physical data independence: changes of the physical schema (e.g., change of an access structure to a more efficient one, use of other data structures, exchange of algorithms) do not have impact on the conceptual schema and thus also not to external schemas

6. What is the difference between DDL and DML? [4 points]
   +Data definition language (DDL): 1) a language to manipulate a database schema 2) (meta) data for the description of a schema (data dictionary, system catalog) 3) permits the specification of implementation details
   +Data manipulation language (DML): 1) query language for the retrieval of data objects in a database 2) “actual” data manipulation language for the change of stored data objects, for the insertion of new data, and for the deletion of stored data

7. Describe 3 tables that might be used to store information in an online shopping website. [3 points]
   + A user table containing records about users, with attributes such as account name, password, recently bought stuff, address, and other profile information.
+ A item table containing records about item information, with attributes such as item name, item id, category, price, picture and other profile information.
+ A shopping cart table containing information the user is going to buy, such as account name, item id, amount.

**Exercise 2 (Oracle) [30 points]**

Consider the following table ‘CLASS’ maintained by a class database.

<table>
<thead>
<tr>
<th>ClassID</th>
<th>Title</th>
<th>Time</th>
<th>Location</th>
<th>Room</th>
<th>Department</th>
<th>Professor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td>Embedded System</td>
<td>3:00-3:50</td>
<td>MCCC</td>
<td>100</td>
<td>CSE</td>
<td>Mish</td>
</tr>
<tr>
<td>1002</td>
<td>Machine Learning</td>
<td>3:00-4:55</td>
<td>CSE</td>
<td>E121</td>
<td>CSE</td>
<td>Baner</td>
</tr>
<tr>
<td>1003</td>
<td>Software Engineering</td>
<td>9:35-10:25</td>
<td>CSE</td>
<td>E121</td>
<td>CSE</td>
<td>Judy</td>
</tr>
<tr>
<td>1004</td>
<td>Graphics</td>
<td>10:40-12:35</td>
<td>CSE</td>
<td>E121</td>
<td>CSE</td>
<td>Michael</td>
</tr>
<tr>
<td>1005</td>
<td>Electronic Circuits 1</td>
<td>8:30-10:25</td>
<td>LAR</td>
<td>310</td>
<td>ECE</td>
<td>Yoon</td>
</tr>
<tr>
<td>1006</td>
<td>Bioelectrical Systems</td>
<td>1:55-2:45</td>
<td>LAR</td>
<td>239</td>
<td>ECE</td>
<td>Judy</td>
</tr>
<tr>
<td>1007</td>
<td>Acoustics</td>
<td>1:55-2:45</td>
<td>BEN</td>
<td>328</td>
<td>ECE</td>
<td>Mark</td>
</tr>
<tr>
<td>1008</td>
<td>Circuits</td>
<td>7:25-8:15</td>
<td>ECE</td>
<td>101</td>
<td>ECE</td>
<td>James</td>
</tr>
</tbody>
</table>

ClassID 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 outputs of all results as *screen snapshots in Oracle*. [5 points each]

1. Create the CLASS table.

```sql
create table CLASS(
    ClassID integer,
    Title varchar(100),
    Time varchar(20),
    Location varchar(20),
    Room varchar(10),
    Department varchar(10),
    Professor varchar(20),
    PRIMARY KEY(ClassID));
```
(2) Insert all the records into the table.

```
insert into CLASS values (1001, 'Embedded System', '3:00-3:50', 'MCCC', '100', 'CSE', 'Mish');
insert into CLASS values (1002, 'Machine Learning', '3:00-4:55', 'CSE', 'E121', 'CSE', 'Baner');
insert into CLASS values (1003, 'Software Engineering', '9:35-10:25', 'CSE', 'E121', 'CSE', 'Judy');
insert into CLASS values (1004, 'Graphics', '10:40-12:35', 'CSE', 'E121', 'CSE', 'Micheal');
insert into CLASS values (1005, 'Electronic Circuits 1', '8:30-10:25', 'LAR', '310', 'ECE', 'Yoon');
insert into CLASS values (1006, 'Bioelectrical Systems', '1:55-2:45', 'LAR', '329', 'ECE', 'Judy');
insert into CLASS values (1007, 'Acoustics', '1:55-2:45', 'BEN', '328', 'ECE', 'Mark');
insert into CLASS values (1008, 'Circuits', '7:25-8:15', 'ECE', '101', 'ECE', 'James');
```

(3) Find the list of class titles held in the time period from 1:55-2:45.

```
select Title from CLASS
where time = '1:55-2:45';
```

```
<table>
<thead>
<tr>
<th>CLASSID</th>
<th>TITLE</th>
<th>TIME</th>
<th>LOCATION</th>
<th>ROOM</th>
<th>DEPARTMENT</th>
<th>PROFESSOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td>Embedded System</td>
<td>3:00-3:50</td>
<td>MCCC</td>
<td>100</td>
<td>CSE</td>
<td>Mish</td>
</tr>
<tr>
<td>1002</td>
<td>Machine Learning</td>
<td>3:00-4:55</td>
<td>CSE</td>
<td>E121</td>
<td>CSE</td>
<td>Baner</td>
</tr>
<tr>
<td>1003</td>
<td>Software Engineering</td>
<td>9:35-10:25</td>
<td>CSE</td>
<td>E121</td>
<td>CSE</td>
<td>Judy</td>
</tr>
<tr>
<td>1004</td>
<td>Graphics</td>
<td>10:40-12:35</td>
<td>CSE</td>
<td>E121</td>
<td>CSE</td>
<td>Micheal</td>
</tr>
<tr>
<td>1005</td>
<td>Electronic Circuits 1</td>
<td>8:30-10:25</td>
<td>LAR</td>
<td>310</td>
<td>ECE</td>
<td>Yoon</td>
</tr>
<tr>
<td>1006</td>
<td>Bioelectrical Systems</td>
<td>1:55-2:45</td>
<td>LAR</td>
<td>329</td>
<td>ECE</td>
<td>Judy</td>
</tr>
<tr>
<td>1007</td>
<td>Acoustics</td>
<td>1:55-2:45</td>
<td>BEN</td>
<td>328</td>
<td>ECE</td>
<td>Mark</td>
</tr>
<tr>
<td>1008</td>
<td>Circuits</td>
<td>7:25-8:15</td>
<td>ECE</td>
<td>101</td>
<td>ECE</td>
<td>James</td>
</tr>
</tbody>
</table>
```
(4) Find the list of class title that department and location is not the same, and the room number is 239.

```sql
select Title from CLASS
where Department != Location and Room = '239';
```

<table>
<thead>
<tr>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Biocellieal Systems</td>
</tr>
</tbody>
</table>

(5) Find the list of class ids whose professor is Judy and whose titles have 'engineering' in their names (case insensitive).

```sql
select ClassID, Title from CLASS
where professor = 'Judy' and lower(Title) like '%engineering%';
```

<table>
<thead>
<tr>
<th>ClassID</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0003 Software Engineering</td>
</tr>
</tbody>
</table>

(6) How many classes are in the ‘CLASS’ table? Store the result under the attribute Total.

```sql
select count(*) as Total from CLASS;
```

<table>
<thead>
<tr>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
</tr>
</tbody>
</table>
Exercise 3 (ER Model) [40 points]

Consider the following requirements about an online restaurant review system:

- Every restaurant has a unique ID, name, location (includes street, city, state, and zip code), distance to the user, brief description, and the url to the restaurant website.
- Restaurant sells dishes. Each dishes has a unique ID, name, picture and price. Assume that no restaurant sells the same dishes.
- Dishes are cooked by cooks. Each cook has a unique employee id and name. Each restaurant employs multiple cooks to cook dishes.
- 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 restaurants. A review has a title, content, a score ranging 1-5, and a timestamp which identifies each review.
- 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 restaurant 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.