Database Management Systems (COP 5725)

(Spring2018)

Instructor: Dr. Markus Schneider

TA: Matin Kheirkhahan

Homework 1

Name:	Jinansh Rupesh Patel
UFID:	94318155
Email Address:	jinansh.patel@ufl.edu

Pledge (Must be signed according to UF Honor Code)

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

Jinansh Rupesh Patel_____
Signature

For scoring use only:

	Maximum	Received
Exercise 1	30	
Exercise 2	30	
Exercise 3	20	
Exercise 4	20	
Total	100	

1**. What is a database? What is a DBMS? What is the relationship of database and DBMS? [4 points]**

- Database is an 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. It can also be described as a systematic collection of data or collection of related data. Databases support storage and manipulation of data.

- DBMS or Data Base Management System is an 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 is a collection of programs that enables users to create and maintain a database.
  It is a set of programs for the management of and access to the data in the DB.
  It is the software level between physical database and user.

- A database is a collection of organized data and the system that manages a collection of databases is called a Database Management System. The database holds the records, fields and cells of data. The DBMS is the tool used to manipulate the data inside the database. DBMS and Database together forms the database systems

2. **Briefly state three advantages of using database systems? [3 points]**

Advantages of database systems are:

1. **Efficient data access**

- Multitude of sophisticated techniques for the efficient storage of and efficient access to persistent data
- Application of index structures

2. **Integrity of data**

- Correctness and completeness of data (semantical aspect)
- formulation of integrity constraints or integrity rules
- DBMS checks constraints for each insertion, change and deletion of data

3. **Data security**

- Protection of the database against unauthorized access (view on data)
- access control with authentication and encoding as possible protection mechanisms

3. **What is the data normalization? Give an example where controlled redundancy is helpful. [6 points]**

Normalization is a process of reducing redundancies of data in a database. It is a technique that is used when designing and redesigning a database. It can also be defined as a process or set of guidelines used to optimally design a database to reduce redundant data. Data redundancy can lead to inconsistency in the database unless controlled.
Controlled Redundancy is a technique which helps us to use redundant fields in a physical database in order to speed up reading database access and to also help recover data. An example where controlled redundancy is useful is in duplication of data in 2 different places in 2 different ways. To initiate a transaction, we use a Sale which then auto-generates an Invoice at an applicable stage in the sales process. The Invoice data is copied

from the Sale and SaleItem tables to the Invoice and InvoiceItem tables, there being reciprocal FOREIGN KEYs between them of their respective PRIMARY KEYs on a 1:1 basis. In case of a failure/corruption of the Invoice table, this model facilitates this table's rebuilding in two probable ways:
1) Copying all the Sale data for those which have been invoiced using the PRIMARY KEY-FOREIGN KEY 1:1 relationship as unique identifiers, and
2) Summarising all of the requisite data from the underlying InvoiceItems using the Item FOREIGN KEYs on a 1-to-many basis.

4. **What is a Data Model? What are the most common data models that a DBMS has? [4 points]**

Data Model is a mathematical formalism consisting of a notation for describing the data of interest and of a set of operations for manipulating these data. It can also be described as a description of the structure of a database (data types, relationships, conditions).

The 2 most common data models that a DBMS has:
- Physical data models for the storage-oriented representation of data.
- Logical data models for the user-oriented representation of data.

5. **DBMS offers data abstraction. Explain it briefly. [6 points]**

Database systems are made-up of complex data structures. To ease the user interaction with database, the developers hide internal irrelevant details from users. The process of hiding these irrelevant details from user is called data abstraction. Masking the unwanted data from the users happens at different levels in the database. This masking of data is called data abstraction. There are 3 most common data abstraction levels:

1. external/view levels
2. conceptual/logical level
3. physical/internal level

6. **Explain 3 most common DBS abstraction levels. [3 points]**

The 3 most common DBMS abstraction levels are:
- E**xternal/view level:** describes the part of the DB, which is relevant for the user.
- C**onceptual/logical level:** gives information about existing data and relationships in the DB.
- P**hysical/internal level:** describes how data are physically stored

7. **When do you think one should not use a DBMS? [4 points**]

Some of the scenarios where one should not use DBMS are listed below:

1. When multiple accesses to the data is not required.
2. The database and application are simple and well defined and not expected to change.
3. If unnecessary overhead costs are incurred compared to traditional file system.
4. When it is a high initial investment for software, hardware and training with no long time returns.

Exercise 2 (Oracle) [30 points]

MOVIE_ID	TITLE	GENRE	RUNTIME	YR	BUDGET	SELL	DIRECTOR
1	Jurassic Park	SciFi	127	1993	63	1029	Steven Spielberg
2	Guardians of the Galaxy	SciFi	136	2017	200	863	James Gunn
3	Schindlers List	Drama	195	1993	22	221	Steven Spielberg
4	Saving Private Ryan	Drama	169	1998	70	480	Steven Spielberg
5	Fight Club	Drama	151	1999	63	100	David Fincher
6	The Social Network	Drama	120	2010	40	225	David Fincher
7	This is the end	Comedy	107	2013	32	126	Seth Rogen
8	The Interview	Comedy	112	2015	44	12	Seth Rogen

(1) Create the MOVIE table.

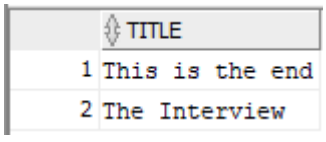
```
CREATE TABLE MOVIE(
MOVIE_ID integer primary key,
TITLE varchar2(60),
GENRE varchar2(25),
RUNTIME integer,
YR integer,
BUDGET integer,
SELL integer,
DIRECTOR varchar2(60));
```

(2) Insert all the records into the MOVIE table.

```
insert into MOVIE values(1,'Jurassic Park', 'SciFi', 127, 1993, 63, 1029, 'Steven Spielberg');
insert into MOVIE values(2,'Guardians of the Galaxy', 'SciFi', 136, 2017, 200, 863, 'James Gunn');
insert into MOVIE values(3,'Schindlers List', 'Drama', 195, 1993, 22, 221, 'Steven Spielberg');
insert into MOVIE values(4,'Saving Private Ryan', 'Drama', 169, 1998, 70, 480, 'Steven Spielberg');
insert into MOVIE values(5,'Fight Club', 'Drama', 151, 1999, 63, 100, 'David Fincher');
insert into MOVIE values(6,'The Social Network', 'Drama', 120, 2010, 40, 225, 'David Fincher');
insert into MOVIE values(7,'This is the end', 'Comedy', 107, 2013, 32, 126, 'Seth Rogen');
insert into MOVIE values(8,'The Interview', 'Comedy', 112, 2015, 44, 12, 'Seth Rogen');
```

(3) Find the list of movie titles which are shorter than 2 hours (120 minutes).

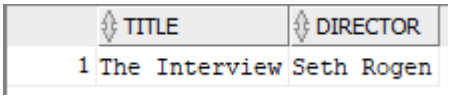
select TITLE from MOVIE where RUNTIME < 120;



	TITLE
1	This is the end
2	The Interview

(4) Find the list of movie titles and their directors that are NOT profitable (budget > sell).

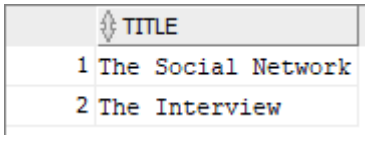
select TITLE, DIRECTOR from MOVIE where budget > sell;



	TITLE	DIRECTOR
1	The Interview	Seth Rogen

(5) Find the list of movie titles that start with 'The'.

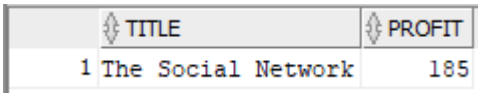
select TITLE from MOVIE where TITLE like 'The%';



	TITLE
1	The Social Network
2	The Interview

(6) Find the list of drama movie titles that are released after 2000 and their profits under attribute PROFIT.

select TITLE, SELL-BUDGET as PROFIT from MOVIE where YR>2000 and GENRE = 'Drama';



	TITLE	PROFIT
1	The Social Network	185

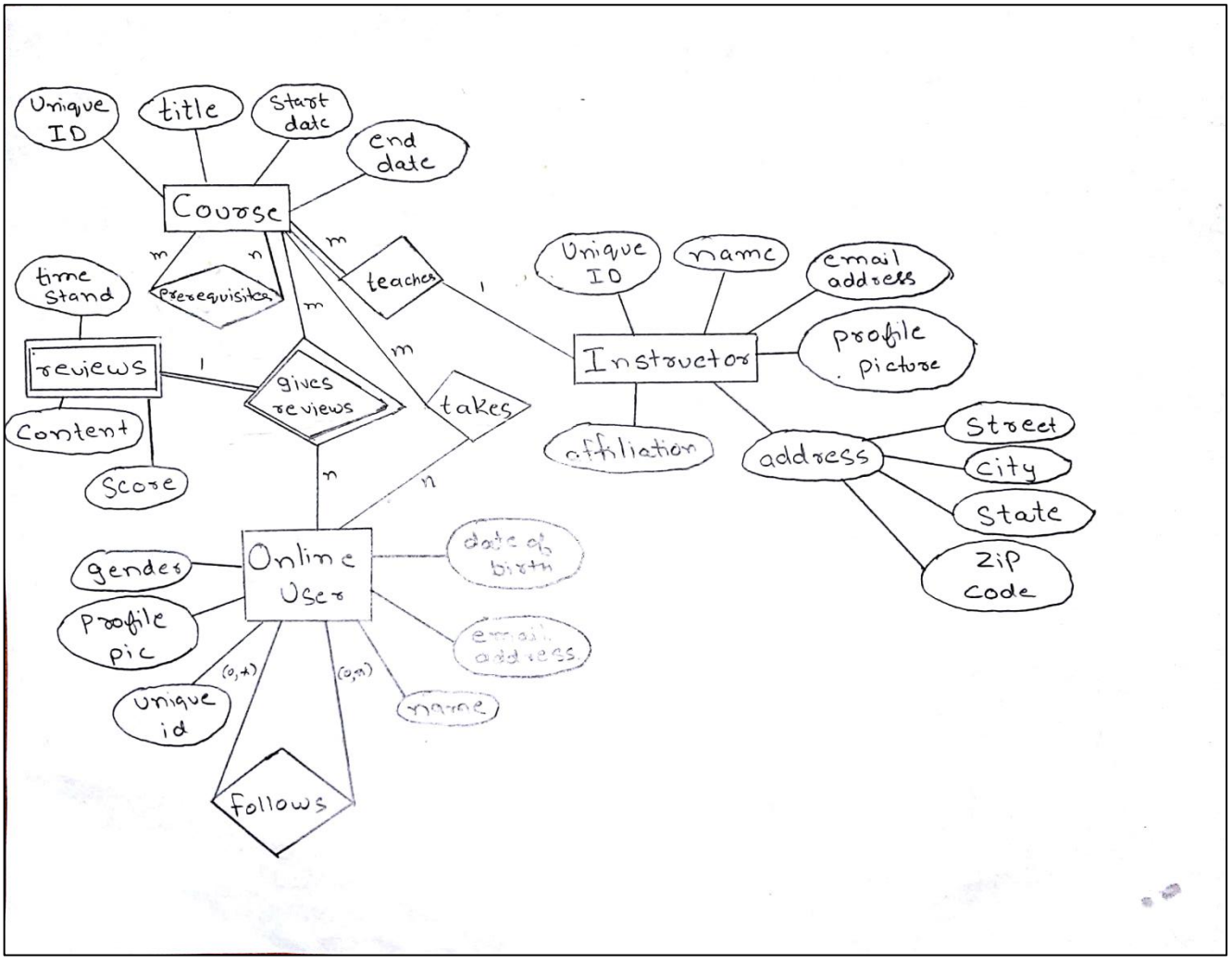
Exercise 3 (ER Model) [20 points]

Consider the following requirements about a Massive Open Online Course (MOOC) system:

- Every course has a unique ID, title, start and end dates.
- Each course might have prerequisites.
- Each course is offered by one instructor.
- Each instructor has a unique ID, name, email address, profile picture, address (street, city, state, and zip code), and affiliation (university of institute).
- Online users can take multiple courses.
- Each online user has a unique ID, name, email address, profile picture, date of birth, and gender.
- An online user might follow other users. Following is directional. For example, if A follows B, that does NOT entail B follows A.

 An online user can write reviews for courses. A review has content, a score (ranging from 1 to 5) and a timestamp.

Design an Entity-Relationship diagram that models the MOOC 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.



Exercise 4 (ER Model) [20 points]

Suppose you are designing a recipe-sharing website. This website has the following entities and requirements:
 Every recipe has a unique ID, name, time-to-cook, picture, and calorie.
 Each recipe has multiple ingredients. Ingredient amounts are specified.
 An ingredient entity has a unique ID, name, unit (pound, gallon, etc.), and unit price.
 Every recipe is prepared by a chef.
 A chef has a unique ID, name, and email address.
 Users can write reviews for recipes. A review has content, timestamp, and a score (ranging from 1 to 10).

 Every user has a unique ID, name, gender, age, and email address.
 Users can follow chefs.

Design an Entity-Relationship diagram that models the recipe-sharing 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.