Homework 4
Due Date: April 18 (11.59pm)
On Blackboard

Submission Instructions.
For part I, submit the MySQL create table statements in a separate text file. For part II, add your MySQL queries to the same file.

PART I (E-R diagrams)

Exercise 1. E-R Diagram (30 points)
Consider the following domain description about an airport. Build an E-R diagram (20 points) to represent the information and derive the relational schema and from that diagram (5 points). In your E-R diagram, clearly indicate the primary keys and the cardinality constraints. In the relational schema, write for each table the foreign key constraints, if there are any. Make your drawing of the E-R diagram as clear as possible (I suggest you start it on a different piece of paper and put here the final result only). Write in a separate text file, the create table statements in MySQL for the relational schema that you devise and make sure it works there because we will test it there (5 points). Do not forget the foreign key constraints.

The airport serves different airline companies. Every airline is identified by its name and has an address and a telephone number. There are two types of airlines, passenger airlines (e.g., American Airlines, Southwest Airlines), and cargo airlines (e.g., FedEx, DHL).

Passenger airlines rent gates from the airport authority. Every rented gate has therefore a rent fee that the airline pays for that gate. There can be gates that are not rented. Every gate has a gate number as an identifier (e.g., K12, K13, K14) and is located in a terminal (e.g., Terminal 3). The database must keep track of which passenger airline rented which gate and what fees was associated to this rent.

Cargo airlines rent hangars from the airport authority. Hangars are identified by an identification number. In addition, every hangar has an area measured in square feet, a door size measured in feet, and a maximum number of airplanes that can fit inside that hangar. The database must keep track of what airplanes are in which hangars.

Every airline (passenger or cargo) owns one or more airplanes. All airplanes are identified by a unique identification number and have a weight (i.e. the number of tons) associated with them. In addition, airplanes are divided in two types corresponding to their use, passenger airplanes and cargo airplanes. Passenger airplanes, which are used by the passenger airlines, have a number of seats and a maximum number of people they are allowed to carry at any time. Cargo airplanes have a cargo weight that specifies the maximum weight the airplane can carry (note, this is different from the airplane weight). The database must keep track of which airlines own which airplanes, together with the airplane info.

Airplanes land in runways which are identified by a unique code and have a length attribute measured in feet associated with them. Runways too are of two types depending on their use, passenger runways, and cargo runways. Airplanes belonging to passenger airlines land in passenger runways, airplanes belonging to cargo airlines land in cargo runways. For every airplane that lands in a runway, the

database must keep track also of the time and date in which that airplane landed in that runway.

Ex.2. E-R Diagram (30 points).
Consider the following domain description about a gym. Build an E-R diagram (20 points) to represent the information and derive the relational schema from that diagram (5 points). In your E-R diagram, clearly indicate the primary keys and the cardinality constraints. In the relational schema, write for each table the foreign key constraints, if there are any. Make your drawing of the E-R diagram as clear as possible. Write, in a separate text file (the same as for the first E-R diagram), the create table statements in MySQL for the relational schema that you devise and make sure it works there because we will test it there (5). Do not forget the foreign key constraints.

Gym instructors are identified by their SSN and have a name and last name, date of birth, email, and a phone number. In addition, they can instruct on one or more types of activities in the gym.

The gym activities are identified by a code, and can be generally grouped in two categories: body building, and cardio (e.g., spinning classes, aerobics classes, running, etc). For the cardio activities, we want to store the name (e.g., spinning, running), the level (e.g., beginner, etc), possible tools used (e.g., light weights, jumping rope, etc).
In addition, we want to keep track of the schedule of the cardio activities. This information includes the day of the week, beginning and end times, and the name of the instructor. More than one activity can be held at the same time in different rooms of the gym. The same activity can be held at different days and within the same day but in different times. An instructor cannot be at two activities at the same time but one activity can have more than one instructors at the same time.

The gym members, are identified by their registration number. For every member, we know the name, date of birth, a type (e.g., student, senior, etc), and membership validity period. We want to keep track of the times every member has used the gym. Every gym use is characterized by a day, by an entry time and by an exit time. A member can use the gym more than once a day.

Gym members can permanently reserve lockers to store their clothes, shoes, protein mixes, and other such stuff. A locker is identified by a number. There are three types of locker reservations: one-year reservations, one semester reservations, and one-use reservations. The first two are characterized by the start and end date, and by a cost, while the latter is characterized by the start and end time and it is free. Note that a member can also not use a locker at all.

PART II. SQL (40 points).

Given the following MySQL Schema, write the MySQL queries to answer the questions.

```
CREATE TABLE Country (
  country_id INT NOT NULL AUTO_INCREMENT,
  country_name VARCHAR(45),
  PRIMARY KEY (country_id)
);

CREATE TABLE Genre (
  genre_id INT NOT NULL AUTO_INCREMENT,
  genre_name VARCHAR(32) NOT NULL,
  PRIMARY KEY (genre_id)
);

CREATE TABLE RecordLabel (
  label_id INT NOT NULL AUTO_INCREMENT,
  label_name NVARCHAR(32) NOT NULL,
  country_id INT NOT NULL,
  PRIMARY KEY (label_id),
  FOREIGN KEY (country_id) REFERENCES Country (country_id)
);

CREATE TABLE Album (
  album_id INT NOT NULL AUTO_INCREMENT,
  album_name NVARCHAR(64) NOT NULL,
  yr_release YEAR,
  label_id INT,
  PRIMARY KEY (album_id),
  FOREIGN KEY (label_id) REFERENCES RecordLabel (label_id)
);

CREATE TABLE User (
  user_id INT NOT NULL AUTO_INCREMENT,
  username VARCHAR(32)  NOT NULL,
  password VARCHAR(32)  NOT NULL,
  first_name VARCHAR(32),
  last_name VARCHAR(32),
  email_addr VARCHAR(64) NOT NULL,
  registration_date DATE NOT NULL,
  membership_type CHAR NOT NULL,
  country_id INT NOT NULL,
  PRIMARY KEY (user_id),
  FOREIGN KEY (country_id) REFERENCES Country (country_id),
  UNIQUE KEY (username)
);

CREATE TABLE Artist (
```

```
  artist_id INT NOT NULL AUTO_INCREMENT,
  artist_name NVARCHAR(64) NOT NULL,
  PRIMARY KEY (artist_id)
);

CREATE TABLE Song (
  song_id INT NOT NULL AUTO_INCREMENT,
  song_title NVARCHAR(64) NOT NULL,
  song_length TIME,
  genre_id INT NOT NULL,
  album_id INT NOT NULL,
  PRIMARY KEY (song_id),
  FOREIGN KEY (genre_id) REFERENCES Genre (genre_id),
  FOREIGN KEY (album_id) REFERENCES Album (album_id)
);

CREATE TABLE ArtistSong_Relationship (
  rel_id INT NOT NULL AUTO_INCREMENT,
  song_id INT NOT NULL,
  artist_id INT NOT NULL,
  PRIMARY KEY (rel_id),
  FOREIGN KEY (song_id) REFERENCES Song (song_id),
  FOREIGN KEY (artist_id) REFERENCES Artist (artist_id)
);

CREATE TABLE User_History (
  history_id INT NOT NULL AUTO_INCREMENT,
  played_at TIMESTAMP NOT NULL,
  user_id INT NOT NULL,
  song_id INT NOT NULL,
  PRIMARY KEY (history_id),
  FOREIGN KEY (user_id) REFERENCES User (user_id),
  FOREIGN KEY (song_id) REFERENCES Song (song_id)
);
```

SQL QUERIES:
Given this schema for a database: Music_Library solve the following:

1.) Generate an ER-Diagram for Music_Library

2.) Write a query that returns a list of all songs played by both user "j_park" and "minosong", organized in alphabetical order by song name.

3.) Write a query that returns a list of all users who registered after 2015, live in the United States or Canada, and have listened to at least 2 different songs that originated in a country other than Canada or the United States.

4.) Write a query that returns the name of the top most played artist for users who live in South Korea.

5.) Write a query that returns a list of all record labels the artist Wu-Tang Clan has produced albums/songs under
in reverse-alphabetical order by record label name.

6.) Write a query that returns the top ten countries, in ascending with the most registered users whose membership type is premium: 'p',
and for each country: the total number of those users, and the top user of each country (hint: the user with the largest history).

7.) Write a query that returns a list of all songs and the year they were released by the artist "Johnny Thunders".

8.) Write a query that returns a list of all songs where artist's name is like "%Bordello%" in alphabetical order.

9.) Write a query that returns the percent of songs, played by user "alex_carapetis" on March 1, 2016,
whose genre matches "Garage Rock" or "Heavy Metal"