All solutions must be word-processed in MS Word and uploaded by the deadline to HuskyCT under the Homework assignment. Using the TV and Movie Relational Database Schema (see next two pages) that has been provided to you, please write the following relational algebra queries. Follow the instructions carefully in terms of using the allowed relational algebra operations. Feel free to do the queries in multiple steps by creating relations to hold your intermediate results.

For the first 3 problems, use only union, intersection, Cartesian product, selection, and projection. You cannot use join for these first three problems.

1. Find the names (Last name and First Name) and State of
   a. All persons that were in a Movie in 1997
   b. All persons that were in a TV show that started in 1987.
   c. All persons that were in either a Movie in 1997 or a TV show in 1987
   d. All persons that were in both a Movie in 1997 and a TV show that started in 1987

2. Find the last names (Last name and First Name) of the directors for all Movies and TV Shows that have the same name (TV Show and Movie have the same names).

3. Find the names (Last name and First Name) of all actors and their roles for “Friends” for Episodes 11 to 25.

For the remaining 2 problems, use union, intersection, join, selection, and projection. YOU CANNOT USE CARTESIAN PRODUCT.

4. Find the names (Last name and First Name) that played the same Role in a TV and a Movie who won an Emmy for the TV role but did not win an Oscar for the Movie role.

5. Find the names of all Shows or Movies that have won an Emmy (TV Show) or Oscar (Movie).
TV and Movie Relational Database Schema

In order to cater to the remote control-wielding, retro, couch-potato US society, the relational schema below represents the various information that can be tracked on actors, directors, TV shows, and movies. The relational schema is:

**Person** (LName, FName, NumYears, State, PersonID);
**TVShows** (ShowName, StartYear, NumSeasons, ShowID, NumEmmy);
**Episodes** (ShowID, EpisodeID, EName, EDescr);
**Movies** (MovieName, Year, Cost, Gross, ShowID, NumOscar);
**Roles** (RLName, RFName, ShowID, RoleID, RoleType);
**TVRoles** (PersonID, RoleID, ShowID, EpisodeID, EmmyFlag);
**MovieRoles** (PersonID, RoleID, ShowID, OscarFlag);
**TVDirectors** (PersonID, ShowID, EpisodeID, EmmyFlag);
**MovieDirectors** (PersonID, ShowID, OscarFlag);

**Person Table**: A person can be a Director or an Actor. LName, FName combinations are always unique (screen actors guild requires unique names), and this unique combination is assigned a unique PersonID (not part of the key of Person, but always with a unique value). NumYears is the number of years that the person has been involved in the entertainment field. State is the state in which they primarily reside.

**TVShows Table**: ShowName in combination with StartYear forms the key. ShowName is not unique. For example, there have been two Wheel of Fortunes, one in the early days (1950s/1960s) and the current version. NumSeasons represents the number of years the show was on. ShowID is a unique ID (not part of the key) that identifies each show. ShowID is unique across all Shows and Movies. There is also a NumEmmy for each Show that indicates how many Emmy Awards the show won for all of its years.

**Episodes Table**: For each TV Show, the Episodes for its entire run (all years in sequential order) are tracked; thus ShowID, EpisodeID combination uniquely identifies each Episode. In addition, each Episode has a name (EName which may not be unique) and a description (EDescr).

**Movies Table**: MovieName in combination with Year forms the key. MovieName is not unique, since there are many remakes of movies. Cost represents how much the movie cost, Gross represents the amount of money the movie grossed. There is also a NumOscar for each movie that indicates how many Oscars the movie won in total (all categories). ShowID is the common unique identifier between Shows and Movies.

**Roles Table**: Roles are tracked by Show (Movie or TV). RLName, RFName, ShowName combinations are always unique, since the same role (as identified by name) may be on more than one show. This combination is assigned a unique RoleID. RoleID is not part of the key of Roles, but it always has a unique value. RoleTypes have values of Star, Co-Star, and Guest-Star.

**TVRoles and MovieRoles Tables**: Actors are identified by the roles that they play in a given show. The combination of PersonId, RoleID, ShowID is needed as a key to identify movie roles, with an EpisodeID added to identify TV roles. There are boolean flags that indicate whether the actor won an Emmy or Oscar for their TV or Movie role in a given year. Different actors can play the same role in different years (and on different episodes).

**TVDirector and MovieDirectors Tables**: Directors are identified by the movies and TV shows that they direct in a given year. There are boolean flags that indicate whether the director won an Emmy or Oscar for their TV show or movie in a given year. For TVDirectors, we track the Episode that is being directed. For TVDirectors, there is a tracking of the individual episodes that are directed.

**Values**: 
PersonID (P1, P2, P3, …); RoleID (R1, R2, R3, … ); ShowID(S1, S2, S3, …); EpisodeID (E1, E2, E3, … ); DirID (are PersonIDs); NumYears, NumSeasons, NumOscars, NumEmmy: Integer; StartYear and Year (19XX or 20XX - where XX is the year); EmmyFlag, OscarFlag: Boolean; Cost, Gross: Large Integer; RoleType: (Star, Co-Star, Guest-Star); LName, FName, ShowName, MovieName, RLName, RFName, EName, EDescr: String;
In order to demonstrate the content of these tables, consider below various tuples of the tables:

Person ( Lname, FName, NumYears, State, PersonID );
   Eastwood Clint  40        CA    P1
   Ball     Lucy  50        CA    P2
   Martin   Steve  30      CA    P3
   Howard Ron  35    CA   P4

TVShows (ShowName, StartYear, NumSeasons, ShowID, NumEmmy);
   I Love Lucy  1951    12    S1    10
   Happy Days 1982 10    S2    8
   Andy Griff. 1958   8     S3    12

Episodes ( ShowID, EpisodeID, EName, EDescr);
   S1   E156    Making Wine . . .
   S1   E199    Candy Making . . .
   S3   E1     Opie Lies . . .
   S2   E111    Richie Marries . . .

Movies ( MovieName, Year, Cost, Gross, ShowID, NumOscar);
   Enforcer   1982 20M   50M    S47     0
   The Jerk   1986 25M   62M    S22     0

Roles ( RLName, RFName, ShowID, RoleID, RoleType);
   Taylor Opie  S3 R1  Co-Star
   Taylor Andy  S3 R2  Star
   Ricardo Lucy S1 R3  Star
   Callahan Harry S47 R4  Star
   Cunningham Richie S2 R5  Star

TVRoles ( PersonID, RoleID, ShowID, EpisodeID, EmmyFlag);
   P2  R3   S1  E156  True
   P4  R1   S1  E1   False
   P4  R5   S3  E111 False

MovieRoles (PersonID, RoleID, ShowID, OscarFlag);
   P1  R4    S47 False
   P3  R11   S22 False

TVDirectors (PersonID, ShowID, EpisodeID, EmmyFlag);
   P75   S3   E12   False
   P75   S3   E14   False
   P14   S1   E112 False

MovieDirectors (PersonID, ShowID, OscarFlag);
   P1    S47 False
   P24   S22 True
These problems require the usage of the MySQL Workbench platform to provide you experience with loading databases from .sql files, and making SQL queries. The queries will involve the Chinook and Northwind databases in order to answer each group of problems. For each problem provide both the query and the csv file that can be created from the query result.

Homework Problem 2.1: For the Chinook Database Schema

   a. Find the one or more playlist(s) that contain the largest number of pop tracks. For each playlist that satisfies this condition, print out the name of the playlist and the number of tracks.

   b. Find all artist(s) that have the most tracks classified as “Rock”, “Jazz”, “Rock And Roll”, “Pop”, and “Classical”, sorted by Genre Name and then by Artist name, and return the artist name, genre, and amount of tracks. Note that more than one artist may have the same number of most tracks.

   c. Find the name of the artist and the name of each track for all artists with more than 5 tracks. Note that an artist can have multiple tracks without having an album.

   d. For the artist “AC/DC”, find the playlist(s) which contains the most tracks. Return the artist and Name of the Playlist. There may be more than one playlist with the most tracks.

Homework Problem 2.2: For the Northwind Database Schema

   a. Find the list of all customers from United Kingdom sorted by Company Name in ascending order and return the company Name and Country.

   b. Find the number of customers that are located in each country. Your query should return just a number representing the number of customers who are located in each country. Countries include: Germany, Mexico, UK, Sweden, France, Spain, Canada, Argentina, Switzerland, Brazil, Austria, Italy, Portugal, USA, Venezuela, Ireland, Belgium, Norway, Denmark, Finland, Poland

   c. Find the product ID, Product Name, UnitsInStock, ReorderLevel, and the shortage amount (sorted by Shortage amount in Descending order) where the ShortageAmount is a calculated column and it is defined as ShortageAmount: ReorderLevel - UnitsInStock

   d. Find the Product Name, UnitPrice, and UnitsInStock and Sorted by ProductName for all products where the product name starts with “Ch”.
Homework 3 Part A: Design and ER Diagram

Construct an ER diagram (no inheritance/specialization/generalization) that would be capable of modeling information that is kept in Peoplesoft for students. Model an ER diagram that is capable of keeping track of students’ class schedule, shopping cart, plan of study, and transcripts. In order to support this, consider the following set of requirements (may not be complete) that are adapted from Exercise 7.16 in textbook.

a. The university keeps track of each student’s name, peoplesoft number, netID, Social Security number, current address and phone number, permanent address and phone number, preferred email, secondary email, birth date, sex, class (freshman, sophomore, ..., graduate), major department, minor department, department (if any), and degree program (B.A., B.S., ..., Ph.D.). Some UConn applications need to refer to the city, state, and ZIP Code of the student’s permanent address and to the student’s last name. Both Social Security number and peoplesoft number have unique values for each student.

b. Each department is described by a name, department code (U Box number at UConn), office number, office phone number, address, and college (e.g., engineering, business, nursing, etc). Both name and code have unique values for each department.

c. Each course has a course department (CSE, ECE, MATH, etc.), course number, course name, course description, credits, level (undergrad, grad, etc.), an a list of zero or more prerequisite courses. The value of the course number is unique for each course.

d. Each section of a course has an instructor, semester, year, course, and section number. The section number distinguishes sections of the same course that are taught during the same semester/year; its values are 1, 2, 3, ..., up to the number of sections taught during each semester.

e. A grade report has a student, section, letter grade, and numeric grade (0, 1, 2, 3, or 4). Note that you need to consider how to combine c, d, and e into a transcript.

f. Ability to store a planned schedule (via a shopping cart) that would eventually transition to entities that store the transcript after course registration (e.g., once you register for Spring 2016 this fall, it shows up in your generated transcript as planned courses). Since the schedule could stay in the cart for period of time, you need to store that info in the DB until registration occurs. Think about the entities and relationships that would be required to support this.

g. Ability to support the registration process – specifically, what is the possible approach to supporting the wait list? Think about the entities and relationships that would be required to support this.

h. Ability to support the plan of study that had to record not only courses that have been taken (already completed or in progress) but choices for the remaining requirements to be completed. Think about the entities and relationships that would be required to support this.

Design an ER schema for this application, and draw an ER diagram for the schema. Specify key attributes of each entity type, and structural constraints on each relationship type. Note any unspecified requirements, and make appropriate assumptions to make the specification complete.

While MySQL Workbench does allow the design of ER diagrams, there is no symbol for a relationship (diamond in and ER diagram) and it is difficult to label the relationship with cardinalities (1, n, m, etc.). It is recommended that you utilize the PPT for Part B (next page). This image is an embedded PPT that you can edit by Right Click – Presentation Object – Open – that will open powerpoint so you can save as a PPT to modify and change. For Part A – upload a PPT titled: PartA_lastname_firstname.pptx.
Homework 3 Part B: EER to relational Conversion Algorithm

Convert the following EER Diagram to a set of relational tables utilizing the conversion algorithm from Chapter 9. A description of the content of the diagram for the ABC Pharmacy includes:

- Tracking information on Patients (name, address, SSN, etc.), Physicians (name, specialty, DEA#, etc.), and Drug Companies (name, phone number, web site). May have multiple phones. Address composite.
- Tracking information on each Drug (name, price, status (generic, brand), drug companies that sell the drug, etc.).
- Each patient has a primary physician, and also each patient have a collection of physicians that can prescribe them drugs.
- Physician’s prescribe drugs for patients, can prescribe multiple drugs per patient, can prescribe to multiple patients, etc. Likewise, multiple physicians can prescribe the same or different drugs to the same patient. Each Prescription has a date, refills, DEA#, dosage (assume milligrams), pattern (1perday, 2perday, etc.), and fill requirement (brand or generic).
- The ABC Pharmacy has different purchasing contracts with each Drug Company. These contracts allow ABC to purchase the same drugs from different Companies. The contract has a start date, end date, and a quantity (in number of bottles).

Clearly identify each Step in the algorithm process (see the Chapter 9 PPTs). For Part B – upload a word doc titled: PartB_lastname_firstname.docx.