1. Find the names (Last name and First Name) and State of
   a. All persons that were in a Movie in 1997

   \[
   \text{Movies1997}(\text{ShowID}) = \pi_{\text{ShowID}} (\sigma_{\text{Year}=1997}(\text{Movies}))
   \]

   \[
   \text{MovieActors}(\text{PersonID}) = \pi_{\text{PersonID}} (\sigma_{\text{Movies1997.ShowID}=\text{MovieRoles.ShowID}} (\text{Movies1997 x MovieRoles}))
   \]

   1a. Answer = \[
   \pi_{\text{Lname}, \text{Fname}, \text{State}} (\sigma_{\text{Person.PersonID} = \text{AllActors.PersonID}} (\text{Person x MovieActors}))
   \]

   b. All persons that were in a TV show in 1987.

   \[
   \text{TVShows1987}(\text{ShowID}) = \pi_{\text{ShowID}} (\sigma_{\text{StartYear}=1987}(\text{TVShows}))
   \]

   \[
   \text{TVActors}(\text{PersonID}) = \pi_{\text{PersonID}} (\sigma_{\text{TVShows1987.ShowID}=\text{TVRoles.ShowID}} (\text{TVShows1987 x TVRoles}))
   \]

   1b. Answer = \[
   \pi_{\text{Lname}, \text{Fname}, \text{State}} (\sigma_{\text{Person.PersonID} = \text{AllActors.PersonID}} (\text{Person x TVActors}))
   \]

   c. All persons that were in either a Movie in 1997 or a TV show in 1987

   \[
   \text{AllActors}(\text{PersonID}) = 1a.\text{Answer} \cup 1b.\text{Answer}
   \]

   d. All persons that were in both a Movie in 1997 or a TV show in 1987

   \[
   \text{AllActors}(\text{PersonID}) = 1a.\text{Answer} \cap 1b.\text{Answer}
   \]

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).

   \[
   \text{MovieIDs}(\text{ShowID}) = \pi_{\text{Movies.ShowID}} (\sigma_{\text{MovieName} = \text{ShowName}} (\text{Movies x TVShows}))
   \]

   \[
   \text{TVShowIDs}(\text{ShowID}) = \pi_{\text{TVShows.ShowID}} (\sigma_{\text{MovieName} = \text{ShowName}} (\text{Movies x TVShows}))
   \]

   \[
   \text{MovieDirs}(\text{PersonID}) = \pi_{\text{PersonID}} (\sigma_{\text{Movie.ShowID} = \text{MovieDirectors.ShowID}} (\text{MovieIDs x MovieDirectors}))
   \]

   \[
   \text{TVDirs}(\text{PersonID}) = \pi_{\text{PersonID}} (\sigma_{\text{TV.ShowID} = \text{TVDirectors.ShowID}} (\text{TVShowIDs x TVDirectors}))
   \]

   Answer = \[
   \pi_{\text{Lname}, \text{Fname}} (\sigma_{\text{Person.PersonID} = \text{AllActors.PersonID}} (\text{Person x (MovieDirs \cup TVDirs)}))
   \]

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

   \[
   \text{Friends}(\text{ShowID}) = \pi_{\text{ShowID}} (\sigma_{\text{ShowName}=\text{Friends}} (\text{TV Shows}))
   \]

   \[
   \text{ActorsandRoles}(\text{PersonID}, \text{RoleID}) =
   \pi_{\text{PersonID}, \text{RoleID}} (\sigma_{\text{TVRoles.ShowID}=\text{Friends.ShowID}}(\text{and}(\text{EpisodeID}>10\text{andEpisodeID}<26)) (\text{TVRoles x Friends}))
   \]

   \[
   \text{RoleNames}(\text{PersonID}, \text{RLName}, \text{RFName}) =
   \pi_{\text{PersonID}, \text{RLName}, \text{RFName}} (\sigma_{\text{ActorsandRoles.RoleID}=\text{Roles.RoleID}} (\text{ActorsandRoles x Roles}))
   \]

   Answer = \[
   \pi_{\text{Lname}, \text{Fname}, \text{RLName}, \text{RFName}} (\sigma_{\text{Person.PersonID} = \text{RoleNames.PersonID}} (\text{Person x RoleNames}))
   \]

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.

   \[
   \text{TVwithEmmy} = \pi_{\text{PersonID}, \text{RLName}, \text{RFName}} (\sigma_{\text{EmmyFlag}=\text{True}} (\text{TVRoles}) \ast \text{RoleID,ShowID Roles})
   \]

   \[
   \text{MovieNoOscar} = \pi_{\text{PersonID}, \text{RLName}, \text{RFName}} (\sigma_{\text{OscarFlag}=\text{False}} (\text{MovieRoles}) \ast \text{RoleID,ShowID Roles})
   \]

   Answer = \[
   \pi_{\text{Lname}, \text{Fname}} \text{ Person} \ast \text{PersonID} (\text{TVwithEmmy (natural join) MovieNoOscar})
   \]

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

   Answer = \[
   \pi_{\text{ShowName}} (\sigma_{\text{NumEmmy}>0} (\text{TVShows})) \cup \pi_{\text{MovieName}} (\sigma_{\text{NumOscar}>0} (\text{Movies}))
   \]
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.

```
WHERE Chinook.Playlist.PlaylistId = Chinook.PlaylistTrack.PlaylistId
AND Chinook.Track.AlbumId = Chinook.Album.AlbumId
AND Chinook.Album.ArtistId = Chinook.Artist.ArtistId
AND Chinook.Genre.Name='Pop'
GROUP BY Chinook.PlaylistTrack.PlaylistId HAVING COUNT(Chinook.PlaylistTrack.PlaylistId) =
(SELECT MAX(A.CNT) FROM (SELECT COUNT(Chinook.PlaylistTrack.PlaylistId) AS CNT from Chinook.Playlist,
WHERE Chinook.Playlist.PlaylistId = Chinook.PlaylistTrack.PlaylistId
AND Chinook.Track.AlbumId = Chinook.Album.AlbumId
AND Chinook.Album.ArtistId = Chinook.Artist.ArtistId
AND Chinook.Genre.Name='Pop'
GROUP BY Chinook.Playlist.playlistId) AS A)
```

```
SELECT Chinook.Playlist.Name, Chinook.Playlist.PlaylistId, COUNT(*)
WHERE Chinook.Playlist.PlaylistId = Chinook.PlaylistTrack.PlaylistId
AND Chinook.Track.AlbumId = Chinook.Album.AlbumId
AND Chinook.Album.ArtistId = Chinook.Artist.ArtistId
AND Chinook.Genre.Name='Pop'
GROUP BY Chinook.Playlist.playlistId, Chinook.Playlist.Name
HAVING COUNT(*) = (SELECT MAX(plist.count) FROM
(SELECT COUNT(*) AS count FROM Chinook.Playlist, Chinook.PlaylistTrack,
Chinook.Track, Chinook.Album, Chinook.Artist, Chinook.Genre
WHERE Chinook.Playlist.PlaylistId = Chinook.PlaylistTrack.PlaylistId
AND Chinook.Track.AlbumId = Chinook.Album.AlbumId
AND Chinook.Album.ArtistId = Chinook.Artist.ArtistId
AND Chinook.Genre.Name='Pop'
GROUP BY Chinook.Playlist.playlistId) plist)
```
CSE4701 Homework 2 Solution

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.

```sql
SELECT Chinook.Artist.Name, Chinook.Genre.Name, COUNT(*)
FROM Chinook.Playlist, Chinook.PlaylistTrack,
Chinook.Track, Chinook.Album, Chinook.Artist, Chinook.Genre
WHERE Chinook.Playlist.PlaylistId = Chinook.PlaylistTrack.PlaylistId
AND Chinook.PlaylistTrack.TrackId = Chinook.Track.TrackId
AND Chinook.Track.AlbumId = Chinook.Album.AlbumId
AND Chinook.Album.ArtistId = Chinook.Artist.ArtistId
AND (Chinook.Genre.Name='Pop' OR Chinook.Genre.Name='Rock'
OR Chinook.Genre.Name='Jazz' OR Chinook.Genre.Name='Rock And Roll'
OR Chinook.Genre.Name='Classical')
AND Chinook.Genre.GenreID = Chinook.Track.GenreID
GROUP BY Chinook.Genre.Name, Chinook.Artist.Name
```

```sql
SELECT MAX(y.numtracks)
FROM (SELECT COUNT(*) AS numtracks
FROM Chinook.Playlist, Chinook.PlaylistTrack,
Chinook.Track, Chinook.Album, Chinook.Artist, Chinook.Genre
WHERE Chinook.Playlist.PlaylistId = Chinook.PlaylistTrack.PlaylistId
AND Chinook.PlaylistTrack.TrackId = Chinook.Track.TrackId
AND Chinook.Track.AlbumId = Chinook.Album.AlbumId
AND Chinook.Album.ArtistId = Chinook.Artist.ArtistId
AND (Chinook.Genre.Name='Pop' OR Chinook.Genre.Name='Rock'
OR Chinook.Genre.Name='Jazz' OR Chinook.Genre.Name='Rock And Roll'
OR Chinook.Genre.Name='Classical')
AND Chinook.Genre.GenreID = Chinook.Track.GenreID
GROUP BY Chinook.Genre.Name, Chinook.Artist.Name) y
```

SEE OTHER FILE

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.

```sql
SELECT Chinook.Artist.Name, Chinook.Genre.Name, COUNT(*)
FROM Chinook.Playlist, Chinook.PlaylistTrack,
Chinook.Track, Chinook.Album, Chinook.Artist, Chinook.Genre
WHERE Chinook.Playlist.PlaylistId = Chinook.PlaylistTrack.PlaylistId
AND Chinook.PlaylistTrack.TrackId = Chinook.Track.TrackId
AND Chinook.Track.AlbumId = Chinook.Album.AlbumId
AND Chinook.Album.ArtistId = Chinook.Artist.ArtistId
AND (Chinook.Genre.Name='Pop' OR Chinook.Genre.Name='Rock'
OR Chinook.Genre.Name='Jazz' OR Chinook.Genre.Name='Rock And Roll'
OR Chinook.Genre.Name='Classical')
AND Chinook.Genre.GenreID = Chinook.Track.GenreID
GROUP BY Chinook.Genre.Name, Chinook.Artist.Name
```
CSE4701 Homework 2 Solution

```
SELECT Chinook.Artist.Name, Count(*)
FROM Track, Album, Artist
WHERE Chinook.Album.AlbumId = Chinook.Track.AlbumId
    AND Chinook.Artist.ArtistId = Chinook.Album.ArtistId
GROUP BY Chinook.Artist.ArtistId, Chinook.Artist.Name
HAVING COUNT(*) > 5;
```

<table>
<thead>
<tr>
<th>Artist</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC/DC</td>
<td>18</td>
</tr>
<tr>
<td>Aerosmith</td>
<td>15</td>
</tr>
<tr>
<td>Alanis Morissette</td>
<td>13</td>
</tr>
<tr>
<td>Antônio Carlos Jobim</td>
<td>31</td>
</tr>
<tr>
<td>Alice In Chains</td>
<td>12</td>
</tr>
<tr>
<td>Apocalyptica</td>
<td>8</td>
</tr>
<tr>
<td>Artist</td>
<td>Tracks</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>Audioslave</td>
<td>40</td>
</tr>
<tr>
<td>BackBeat</td>
<td>12</td>
</tr>
<tr>
<td>Billy Cobham</td>
<td>8</td>
</tr>
<tr>
<td>Black Label Society</td>
<td>18</td>
</tr>
<tr>
<td>Black Sabbath</td>
<td>17</td>
</tr>
<tr>
<td>Body Count</td>
<td>17</td>
</tr>
<tr>
<td>Bruce Dickinson</td>
<td>11</td>
</tr>
<tr>
<td>Buddy Guy</td>
<td>11</td>
</tr>
<tr>
<td>Caetano Veloso</td>
<td>21</td>
</tr>
<tr>
<td>Chico Buarque</td>
<td>34</td>
</tr>
<tr>
<td>Chico Science &amp; Nação Zumbi</td>
<td>36</td>
</tr>
<tr>
<td>Cidade Negra</td>
<td>31</td>
</tr>
<tr>
<td>Cláudio Zoli</td>
<td>10</td>
</tr>
<tr>
<td>Various Artists</td>
<td>56</td>
</tr>
<tr>
<td>Led Zeppelin</td>
<td>114</td>
</tr>
<tr>
<td>Frank Zappa &amp; Captain Beefheart</td>
<td>9</td>
</tr>
<tr>
<td>Marcos Valle</td>
<td>17</td>
</tr>
<tr>
<td>Gilberto Gil</td>
<td>32</td>
</tr>
<tr>
<td>O Rappa</td>
<td>17</td>
</tr>
<tr>
<td>Ed Motta</td>
<td>14</td>
</tr>
<tr>
<td>Elis Regina</td>
<td>14</td>
</tr>
<tr>
<td>Milton Nascimento</td>
<td>26</td>
</tr>
<tr>
<td>Jorge Ben</td>
<td>14</td>
</tr>
<tr>
<td>Metallica</td>
<td>112</td>
</tr>
<tr>
<td>Queen</td>
<td>45</td>
</tr>
<tr>
<td>Kiss</td>
<td>35</td>
</tr>
<tr>
<td>Spyro Gyra</td>
<td>21</td>
</tr>
<tr>
<td>Green Day</td>
<td>34</td>
</tr>
<tr>
<td>David Coverdale</td>
<td>12</td>
</tr>
<tr>
<td>Gonzaguinha</td>
<td>14</td>
</tr>
<tr>
<td>Os Mutantes</td>
<td>14</td>
</tr>
<tr>
<td>Deep Purple</td>
<td>92</td>
</tr>
<tr>
<td>Santana</td>
<td>27</td>
</tr>
<tr>
<td>Miles Davis</td>
<td>37</td>
</tr>
<tr>
<td>Gene Krupa</td>
<td>22</td>
</tr>
</tbody>
</table>
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.
SELECT Chinook.Playlist.PlaylistId, Chinook.Playlist.Name
FROM Chinook.Playlist, Chinook.PlaylistTrack,
     Chinook.Track, Chinook.Album, Chinook.Artist
WHERE Chinook.Playlist.PlaylistId = Chinook.PlaylistTrack.PlaylistId
  AND Chinook.PlaylistTrack.TrackId = Chinook.Track.TrackId
  AND Chinook.Track.AlbumId = Chinook.Album.AlbumId
  AND Chinook.Album.ArtistId = Chinook.Artist.ArtistId
  AND Chinook.Artist.Name = 'AC/DC'
GROUP BY Chinook.Playlist.PlaylistId, Chinook.Playlist.Name
HAVING COUNT(*) =
    (SELECT MAX(plist.count) FROM
      (SELECT COUNT(*) AS count
       FROM Chinook.Playlist, Chinook.PlaylistTrack,
            Chinook.Track, Chinook.Album, Chinook.Artist
       WHERE Chinook.Playlist.PlaylistId = Chinook.PlaylistTrack.PlaylistId
         AND Chinook.PlaylistTrack.TrackId = Chinook.Track.TrackId
         AND Chinook.Track.AlbumId = Chinook.Album.AlbumId
         AND Chinook.Album.ArtistId = Chinook.Artist.ArtistId
         AND Chinook.Artist.Name = 'AC/DC'
       GROUP BY Chinook.Playlist.playlistId) plist)

1 Music
8 Music
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.

```
SELECT northwind.suppliers.CompanyName, northwind.suppliers.country
FROM northwind.suppliers
WHERE northwind.suppliers.Country="UK"
ORDER BY northwind.suppliers.CompanyName ASC
```

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

```
SELECT northwind.customers.country, COUNT(*)
FROM northwind.customers
WHERE northwind.customers.country IS NOT NULL
GROUP BY northwind.customers.country;
```

<table>
<thead>
<tr>
<th>Country</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Argentina</td>
<td>3</td>
</tr>
<tr>
<td>Austria</td>
<td>2</td>
</tr>
<tr>
<td>Belgium</td>
<td>2</td>
</tr>
<tr>
<td>Brazil</td>
<td>9</td>
</tr>
<tr>
<td>Canada</td>
<td>3</td>
</tr>
<tr>
<td>Denmark</td>
<td>2</td>
</tr>
<tr>
<td>Finland</td>
<td>2</td>
</tr>
<tr>
<td>France</td>
<td>11</td>
</tr>
<tr>
<td>Germany</td>
<td>11</td>
</tr>
<tr>
<td>Ireland</td>
<td>1</td>
</tr>
<tr>
<td>Italy</td>
<td>3</td>
</tr>
<tr>
<td>Mexico</td>
<td>5</td>
</tr>
<tr>
<td>Norway</td>
<td>1</td>
</tr>
<tr>
<td>Poland</td>
<td>1</td>
</tr>
<tr>
<td>Portugal</td>
<td>2</td>
</tr>
<tr>
<td>Spain</td>
<td>5</td>
</tr>
<tr>
<td>Sweden</td>
<td>2</td>
</tr>
<tr>
<td>Switzerland</td>
<td>2</td>
</tr>
<tr>
<td>UK</td>
<td>7</td>
</tr>
<tr>
<td>USA</td>
<td>13</td>
</tr>
<tr>
<td>Venezuela</td>
<td>4</td>
</tr>
</tbody>
</table>

Note: The table above lists the countries and their respective counts.

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. *Hint: Only list products that that have ShortageAmount > 0.*

```
SELECT northwind.products.productid, northwind.products.productname,
       northwind.products.UnitsinStock, northwind.products.reoderlevel,
       northwind.products.ReorderLevel - northwind.products.UnitsInStock
FROM northwind.products
```
WHERE (northwind.products.ReorderLevel - northwind.products.UnitsInStock) > 0
ORDER BY (northwind.products.ReorderLevel - northwind.products.UnitsInStock);

<table>
<thead>
<tr>
<th>Product Name</th>
<th>UnitsInStock</th>
<th>UnitPrice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gorgonzola Telino</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>Louisiana Hot Spiced Okra</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>Mascarpone Fabioli</td>
<td>9</td>
<td>25</td>
</tr>
<tr>
<td>Outback Lager</td>
<td>15</td>
<td>30</td>
</tr>
<tr>
<td>Gravad lax</td>
<td>11</td>
<td>25</td>
</tr>
<tr>
<td>Aniseed Syrup</td>
<td>13</td>
<td>25</td>
</tr>
<tr>
<td>Rogede sild</td>
<td>5</td>
<td>15</td>
</tr>
<tr>
<td>Chocolade</td>
<td>15</td>
<td>25</td>
</tr>
<tr>
<td>Scottish Longbreads</td>
<td>6</td>
<td>15</td>
</tr>
</tbody>
</table>

Gnocchi di nonna Alice 21 30 9
Wimmers gute Semmelkndel 22 30 8
Chang 17 25 8
Ipoh Coffee 17 25 8
Queso Cabrales 22 30 8
Maxilaku 10 15 5
Nord-Ost Matjeshering 10 15 5
Sir Rodney's Scones 3 5 2
Longlife Tofu 4 5 1

---

d. Find the Product Name, UnitPrice, and UnitsInStock and Sorted by ProductName for all products where the product name starts with “Ch”.

SELECT northwind.products.productid, northwind.products.productname, northwind.products.UnitsInStock
FROM northwind.products
WHERE northwind.products.productname LIKE 'CH%';

<table>
<thead>
<tr>
<th>Product Name</th>
<th>UnitsInStock</th>
<th>UnitPrice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chai</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>Chang</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>Chartreuse verte</td>
<td>69</td>
<td></td>
</tr>
<tr>
<td>Chef Anton's Cajun Seasoning</td>
<td>53</td>
<td></td>
</tr>
<tr>
<td>Chef Anton's Gumbo Mix</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Chocolade</td>
<td>15</td>
<td></td>
</tr>
</tbody>
</table>
Grading: 40 pts total
4pts each for each entity Student, Dept, Course, Section+OfferedAs
4pts each for each Relationship: GradeRep, Planned Schedule, Offers, Major/Minors, Plan of Study, Prereq
Take off ½ pt for each missing label on relationships (1, N)
Take off ½ pt for missing attributes
Step 1: Regular Entities (2pts each – 8 pts total)
- Person(Person#, Name, Street, City, Email)
- DrugPrescriber (DEA#)
- DrugCompany(Name, Address, Phone#, Website)
- Drug(DrugCode, Name, Price, Status, Expiration)

Step 2: No Weak Entities
Step 3: No 1:1 Relationships
Step 6: Multi-valued Entities (2pts)
- PhoneContacts(Person#, PhoneNumber)

Step 8: Subclasses (2pts each – 4 pts total)
- Patient(Person#, Insurance#)
- Physician(Person#, DEA#, Specialty)

Step 4: 1:N Relationships (4pts)
- Patient(Person#, Insurance#, PrimaryPhysicianPerson#, PrimaryPhysicianDEA#)

Step 5: M:N Relationships (4pts each – 8 pts total)
- PurchasingContract(DrugCompanyName, DrugDrugCode, State_Date, End_Date)
- SoldBy(DrugCompanyName, DrugDrugCode)

Step 7: Higher Order Relationships (4pts)
- Prescribes(PatientPerson#, PhysicianPerson#, DEA#, DrugCode, Dosage, FillRequirement, Pattern, Refills, Date)

Grading: 30 pts total