Name: SAMPLE SOLUTION

Section (315/415/615):

MIDTERM EXAM - 601.315/415/615 - Databases

Date: Thursday, November 1, 2018, 3-4:15 PM

The total number of points in this exam is 75 for both 601.315 and 601.415/615 students (although the questions are mostly different). If you work at approximately 1 minute per point, you should finish on time.

Question 1 - Relational Algebra (5 points)

Express the following query in the Relational Algebra. The tables that are used in this (and following) questions are found on your supplementary handout.

• (5 points) List the GameID, date and opponent of all games won by the BasketCases in 2018.

\[ BCGame \leftarrow \sigma_{\text{Season}="2018" \land \text{WinningTeam}="BasketCases"}(\text{Game}) \]
\[ OA \leftarrow \Pi_{\text{GameID, AwayTeam, Date}}(\sigma_{\text{HomeTeam}="BasketCases"}(BCGame)) \]
\[ OH \leftarrow \Pi_{\text{GameID, HomeTeam, Date}}(\sigma_{\text{AwayTeam}="BasketCases"}(BCGame)) \]
\[ Result \leftarrow OA \cup OH \]

Question 2 - Relational Algebra (5 points)

Express the following query in the relational algebra:

(a) (5 points) 601.315 only: List the name and host city of all teams who won every game they played in 2018.

(b) (5 points) 601.415/615 only: List the name and host city of all teams who won every game they played in 2018, but did not win any game in 2017.

601.315:

\[ \text{lostIn2018Away} \leftarrow \Pi_{\text{Team.TName, Team.HostCity}}(\sigma_{\text{Game.AwayTeam}=\text{Team.TName} \land \text{Game.Season}=2018 \land \text{Game.winningTeam} \neq \text{Team.TName}}(\text{TEAM} \times \text{GAME})) \]
\[ \text{LostIn2018Home} \leftarrow \Pi_{\text{Team.TName, Team.HostCity}}(\sigma_{\text{Game.HomeTeam}=\text{Team.TName} \land \text{Game.Season}=2018 \land \text{Game.winningTeam} \neq \text{Team.TName}}(\text{TEAM} \times \text{GAME})) \]
\[ Result \leftarrow \Pi_{\text{TName, HostCity}}(\text{Team}) - \text{LostIn2018Away} - \text{LostIn2018Home} \]
601.415/615:

\[
\text{lostIn2018Away} \leftarrow \Pi_{\text{Team.TName, Team.HostCity}} (\sigma_{\text{Game.AwayTeam=Team.TName} \land \text{Game.Season=2018} \land \text{Game.winningTeam}\neq \text{Team.TName}} (\text{TEAM} \times \text{GAME})) \\
\text{lostIn2018Home} \leftarrow \Pi_{\text{Team.TName, Team.HostCity}} (\sigma_{\text{Game.HomeTeam=Team.TName} \land \text{Game.Season=2018} \land \text{Game.winningTeam}\neq \text{Team.TName}} (\text{TEAM} \times \text{GAME})) \\
\text{WinIn2017} \leftarrow \Pi_{\text{Team.TName, Team.HostCity}} (\sigma_{\text{Game.HomeTeam=Team.TName} \land \text{Game.Season=2017} \land \text{Game.winningTeam}=\text{Team.TName}} (\text{TEAM} \times \text{GAME})) \\
\text{Result} \leftarrow \Pi_{\text{Name, HostCity(Team)}} (\text{lostIn2018Away} - \text{lostIn2018Home} - \text{WinIn2017})
\]

Question 3 - Relational Algebra (5 points)

(a) (5 points) **601.315 only**: List the names all players who have scored more points than Wally Whinger in the same game (e.g. if Colin Creep scores 106 points in the same game that Wally scores 103 points then he qualifies).

(b) (5 points) **601.415/615 only**: List the names all players who have scored more points than Wally Whinger in the same game but has never scored more points than Steven Simpleton in the same game (e.g. if Colin Creep scores 106 points in the same game that Wally scores 103 points then he satisfies the the first clause).

601.315:

\[
\text{WGames} \leftarrow \text{PG1} \leftarrow \text{PLAYED IN GAME} \\
\Pi_{\text{GameID, PointsScored}} (\sigma_{\text{Name=WallyWhinger}} (\text{PG1})) \\
\text{OtherGames} \leftarrow \sigma_{\text{Name}\neq \text{WallyWhinger}} (\text{PG1}) \\
\text{WGames}' \leftarrow \rho_{\text{WallyWhinger}/\text{PointsScored}} (\text{WGames}) \\
\text{OtherGames}' \leftarrow \rho_{\text{OtherNames}/\text{PointsScored}} (\text{OtherGames}) \\
\text{MoreThanWally} \leftarrow \Pi_{\text{Name}} (\sigma_{\text{OtherNames}>\text{WallyWhinger}} (\text{WGames}' \land \text{OtherGames}'))
\]

601.415/615:

\[
\text{PG1} \leftarrow \text{PLAYED IN GAME} \\
\text{PG2} \leftarrow \text{PLAYED IN GAME} \\
\text{PG3} \leftarrow \text{PLAYED IN GAME} \\
\text{Result} \leftarrow \Pi_{\text{PG1.Name}} (\sigma_{\text{GameID=PG1.GameID} \land \text{GameID}=\text{PG2.GameID} \land \text{PG2.Name}=\text{"WallyWhinger"}}) \\
\land \Pi_{\text{PG1.PointScored}>\text{PG2.PointScored}} \\
\land \Pi_{\text{GameID=PG3.GameID} \land \text{PG3.Name}=\text{"StevenSimpleton"}} \\
\land \Pi_{\text{PointsScored}>\text{PG3.PointsScored} (\text{PG1} \times \text{PG2} \times \text{PG3})}
\]
Question 4 - Relational Algebra (5 points)

Express the following query in the relational algebra.

(a) (5 points) **601.315 only:** List the names, birth city and birth country of all players who have never played a game in the city of their birth (you can assume a game is always played in the host city of the home team).

(b) (5 points) **601.415/615 only:** List the names, birth city and birth country of all players who have never played a game in the city of their birth and have never played against a team whose host city is the city of their birth. (you can assume a game is always played in the host city of the home team).

**601.315:**

\[
T \leftarrow PlayedInGame \bowtie Game \\
T_2 \leftarrow T \bowtie_{HomeTeam=T.Name} Team \\
T_3 \leftarrow T_2 \bowtie_{HostCity=CityOfBirth, P.Name=P.Name} Player \\
PlayedInHomeCity \leftarrow \Pi_{P.Name}(T_3) \\
Others \leftarrow \Pi_{P.Name}(Player) - PlayedInHomeCity \\
Result \leftarrow \Pi_{P.Name, CityOfBirth, Country}(Others \bowtie Player)
\]

**601.415/615:**

\[
Host \leftarrow (\sigma_{cityofbirth \neq HostCity}(PLAYER \bowtie PLAYS FOR \bowtie TEAM \bowtie_{HomeTeam=T.Name} GAME)) \\
T \leftarrow TEAM \\
Against \leftarrow (\sigma_{cityofBirth \neq T.HostCity}(PLAYER \bowtie PLYS FOR \bowtie_{AwayTeam=T.Name} GAME \bowtie_{HomeTeam=T.T.Name} T)) \\
Result \leftarrow \Pi_{P.Name, CityOfBirth, CountryOfBirth}(Host \bigcap Against)
\]
Question 5 - Tuple Relational Calculus (6 points)

Express the following query in Tuple Relational Calculus:

(a) (6 points) **601.315 only**: List the name, birthdate, sex and birthcountry of all players who were born in the same country as Wally Winger and were born on the same exact date as Wally Whinger and have the same sex as Wally Whinger and has played for a team that Wally Whinger has played for in some year, but were not born in the same city as Wally Whinger.

\[
\{ t \mid \exists p \in PLAYE R(t[name] = p[name] \land p[Date] = t[Date] \land t[Sex] = p[Sex] \land t[Country of Birth] = p[Country of Birth] \\
\land \exists s \in Player(s[PName] = "WallyWinger" \land s[Country of Birth] = p[Country of Birth] \land s[Date] = p[Date] \land s[Sex] = p[Sex] \\
\land \exists p \in Plays_for(pf[PName] = "WallyWinger" \\
\land \exists p[2] \in Plays_for(pf2[PName] = p[name] \land pf2[TName] = pf[TName] \\
\land \neg \exists g \in Played_in_\text{game}(t[PName] = g[PName] \\
\land \exists g \in Played_in_\text{game}(g[1[PName] = "WallyWinger" \land g[1[GameID] = g[GameID]])))\} \}
\]

(a) (6 points) **601.415/615 only**: List the name, birthdate, sex and birthcountry of all players who were born in the same country as Wally Winger and were born on the same exact date as Wally Whinger and have the same sex as Wally Whinger and has played for a team that Wally Whinger has played for in some year, but have never played in the same game as Wally Whinger (on the same or opposing team).

\[
\{ t \mid \exists p \in PLAYE R(t[name] = p[name] \land p[Date] = t[Date] \land t[Sex] = p[Sex] \land t[Country of Birth] = p[Country of Birth] \\
\land \exists s \in Player(s[PName] = "WallyWinger" \land s[Country of Birth] = p[Country of Birth] \land s[Date] = p[Date] \land s[Sex] = p[Sex] \\
\land \exists p \in Plays_for(pf[PName] = "WallyWinger" \\
\land \exists p[2] \in Plays_for(pf2[PName] = p[name] \land pf2[TName] = pf[TName] \\
\land \neg \exists g \in Played_in_\text{game}(t[PName] = g[PName] \\
\land \exists g \in Played_in_\text{game}(g[1[PName] = "WallyWinger" \land g[1[GameID] = g[GameID]]))))\} \}
\]
Question 6 - SQL (25 points)

Express the following queries in SQL:

(a) 601.315 only: List the names and birth country of players who were played for a US team in 2018 but were born in a different country.

```sql
SELECT p.PName, p.CountryofBirth
FROM PLAYER p, TEAM t, PLAYS_FOR f
WHERE p.PName=f.PName AND f.season = 2018 AND f.TName=t.TName
```

(aa) 601.415/615 only: List the names and birth country of players who have never played for a team whose host country is the country where they were born.

```sql
SELECT p.PName, p.CountryofBirth
FROM PLAYER p, TEAM t, PLAYS_FOR pf
WHERE p.PName=pf.PName AND pf.TName=t.TName AND t.HostCountry!=p.CountryofBirth;
```
(b) **601.315 only**: List the names and birthdate of all players born after 1/1/84 who have played for more than two teams.

```sql
SELECT Player.PName, Player.Bdate
FROM
  (SELECT PName, COUNT(DISTINCT TName) as numTeams
   FROM PLAYS_FOR
   GROUP BY PName) as C
JOIN
  PLAYER ON C.PName = Player.PName
WHERE
  C.numTeams > 2 AND Player.BDate > 1/1/84
```

(bb) **601.415/615 only**: List the names and birthdate of all players born after 1/1/84 who have played for more than two teams, and also include the total number of teams that the player has ever played for.

```sql
SELECT Player.PName, Player.Bdate, C.numTeams
FROM
  (SELECT PName, COUNT(DISTINCT TName) as numTeams
   FROM PLAYS_FOR
   GROUP BY PName) as C
JOIN
  PLAYER ON C.PName = Player.PName
WHERE
  C.numTeams > 2 AND Player.BDate > 1/1/84
```
(c) **315 and 415/615**: For each team in the league, list the total number of points scored by Colette Cretin when playing against that team.

```sql
SELECT T.team, SUM(T.PS)
FROM
  (SELECT G.HomeTeam as team, SUM(PIG.PointsScored) as PS
   FROM Played_In_Game as PIG
   JOIN Game as G ON PIG.GameID = G.GameID
   JOIN Plays_For CPF ON PIG.PName = CPF.PName
   GROUP BY G.HomeTeam
   WHERE PIG.PName = 'Colette Cretin'
   AND G.HomeTeam <> CPF.TName)
UNION

(SELECT G.AwayTeam as team, SUM(PIG.PointsScored) as PS
 FROM Played_In_Game as PIG
 JOIN Game as G ON PIG.GameID = G.GameID
 JOIN Plays_For CPF ON PIG.PName = CPF.PName
 GROUP BY G.AwayTeam
 WHERE PIG.PName = 'Colette Cretin'
 AND G.AwayTeam <> CPF.TName)
) as T
GROUP BY T.team
```

(d) **601.315 only**: Name the female player with the highest total points scored in 2017, and include that total.

```sql
SELECT Points2017.PName
FROM
  (SELECT PIG.PName as PName, SUM(PIG.PointsScored) as PS
   FROM Played_In_Game PIG
   JOIN Player P ON PIG.PName = P.PName
   JOIN Game G ON PIG.GameID = G.GameID
   GROUP BY PIG.PName
   WHERE P.Sex = 'Female'
   AND G.Season = 2017) as Points2017
WHERE Points2017.PS =
SELECT MAX(Points2017v2.PS)
FROM
  (SELECT PIG.PName as PName, SUM(PIG.PointsScored) as PS
   FROM Played_In_Game PIG
   JOIN Player P ON PIG.PName = P.PName
   JOIN Game G ON PIG.GameID = G.GameID
   GROUP BY PIG.PName
   WHERE P.Sex = 'Female'
   AND G.Season = 2017) as Points2017v2
```
(dd) **601.415/615 only:** Name the female player who is most improved between 2017 and 2018 (the increase in her total points from 2017 to 2018 is greatest).

```sql
SELECT T.PName
FROM
(SELECT Points2017.PName as PName, (Points2018.PS - Points2017.PS) as diff
FROM
(SELECT PIG.PName as PName, SUM(PIG.PointsScored) as PS
FROM Played_In_Game PIG
JOIN Player P ON P.PName = P.PName
JOIN Game G ON PIG.GameID = G.GameID
GROUP BY PIG.PName
WHERE P.Sex = 'Female'
AND G.Season = 2017) as Points2017
JOIN
(SELECT PIG.PName as PName, SUM(PIG.PointsScored) as PS
FROM Played_In_Game PIG
JOIN Player P ON P.PName = P.PName
JOIN Game G ON PIG.GameID = G.GameID
GROUP BY PIG.PName
WHERE P.Sex = 'Female'
AND G.Season = 2018) as Points2018
ON Points2017.PName = Points2018.PName) as T
WHERE T.diff =
(SELECT MAX(Points2018v2.PS - Points2017v2.PS) as diff
FROM
(SELECT PIG.PName as PName, SUM(PIG.PointsScored) as PS
FROM Played_In_Game PIG
JOIN Player P ON P.PName = P.PName
JOIN Game G ON PIG.GameID = G.GameID
GROUP BY PIG.PName
WHERE P.Sex = 'Female'
AND G.Season = 2017) as Points2017v2
JOIN
(SELECT PIG.PName as PName, SUM(PIG.PointsScored) as PS
FROM Played_In_Game PIG
JOIN Player P ON P.PName = P.PName
JOIN Game G ON PIG.GameID = G.GameID
GROUP BY PIG.PName
WHERE P.Sex = 'Female'
AND G.Season = 2018) as Points2018v2
ON Points2017v2.PName = Points2018v2.PName)
```
(e) **601.315 only**: List the names and birthdates of all players who have never played for more than one team.

```
SELECT P.PName, P.BDate
FROM Player P, (SELECT PName, Count(TName) NumTeam
                FROM Plays_for
                GROUP BY PName ) T1
WHERE P.Pname=T1.Pname AND T1.NumTeam=1
```

(ee) **601.415/615 only**: How many players in the database have played for more than two teams in the same host country in their entire careers.

```
SELECT  count(*)
FROM    SELECT T.PName
        FROM Team T,( SELECT DISTINCT PName, TName
                        FROM PLAYS_FOR) PF
WHERE T.TName=PF.TName
GROUP BY T.PName, T.HostCountry
HAVING Count(T.TName)>2
```
**Question 7 - QBE (15 points)**

Express the following queries in QBE. To simplify your work, table shells have been provided. Just fill in the appropriate cells with variables/values.

(a) (5 points) List the name and birthdate of all players who have played in a game against someone who played in a game against Wally Whinger, but were not born in the same country as Wally Whinger.

<table>
<thead>
<tr>
<th>PLAYER</th>
<th>PName</th>
<th>BDate</th>
<th>Sex</th>
<th>CityOfBirth</th>
<th>CountryOfBirth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wally Whinger</td>
<td>P.</td>
<td></td>
<td></td>
<td></td>
<td>_c \neg _c</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TEAM</th>
<th>HostCity</th>
<th>TName</th>
<th>HostCountry</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>PLAYS_FOR</th>
<th>PName</th>
<th>TName</th>
<th>Season</th>
</tr>
</thead>
<tbody>
<tr>
<td>_pn</td>
<td>t1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>_p2</td>
<td>t2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wally Whinger</td>
<td>t3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GAME</th>
<th>GameID</th>
<th>HomeTeam</th>
<th>AwayTeam</th>
<th>Date</th>
<th>Season</th>
<th>WinningTeam</th>
<th>HomeScore</th>
<th>AwayScore</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>PLAYED_IN_GAME</th>
<th>PName</th>
<th>GameID</th>
<th>Position</th>
<th>PointsScored</th>
</tr>
</thead>
<tbody>
<tr>
<td>_pn</td>
<td>g1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>_p2</td>
<td>g1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>_p2</td>
<td>g2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wally Whinger</td>
<td>g2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Conditions</th>
</tr>
</thead>
</table>
(b) (5 points) List the name, birthdate, sex and birthcountry of all players who were not born in the same country as Wally Winger and were not born on the same exact date as Wally Whinger and do not have the same sex as Wally Whinger and have never played for a team that Wally Whinger has ever played for, and has never played in the same game as Wally Whinger (on the same or opposing team).

<table>
<thead>
<tr>
<th>PLAYER</th>
<th>PName</th>
<th>BDate</th>
<th>Sex</th>
<th>CityOfBirth</th>
<th>CountryofBirth</th>
</tr>
</thead>
<tbody>
<tr>
<td>P..pn</td>
<td>p..b</td>
<td>p..s</td>
<td></td>
<td></td>
<td>P..c</td>
</tr>
<tr>
<td>Wally Whinger</td>
<td>-b</td>
<td>-s</td>
<td></td>
<td>-c</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TEAM</th>
<th>HostCity</th>
<th>TName</th>
<th>HostCountry</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>PLAYS_FOR</th>
<th>PName</th>
<th>TName</th>
<th>Season</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>Wally Whinger</td>
<td>t</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GAME</th>
<th>GameID</th>
<th>HomeTeam</th>
<th>AwayTeam</th>
<th>Date</th>
<th>Season</th>
<th>WinningTeam</th>
<th>HomeScore</th>
<th>AwayScore</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>PLAYED_IN_GAME</th>
<th>PName</th>
<th>GameID</th>
<th>Position</th>
<th>PointsScored</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>Wally Whinger</td>
<td>-g</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Conditions
(c) List the name, birthdate and birth city of all players who have played for a team with a host city and country that is the same as the city/country where the player was born.

<table>
<thead>
<tr>
<th>PLAYER</th>
<th>PName</th>
<th>BDate</th>
<th>Sex</th>
<th>CityOfBirth</th>
<th>CountryOfBirth</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P_pn</td>
<td>P_bd</td>
<td></td>
<td>P_cityb</td>
<td>_coub</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TEAM</th>
<th>HostCity</th>
<th>TName</th>
<th>HostCountry</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>_cityb</td>
<td>_tn</td>
<td>_coub</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PLAYS_FOR</th>
<th>PName</th>
<th>TName</th>
<th>Season</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>_p_n</td>
<td>_tn</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GAME</th>
<th>GameID</th>
<th>HomeTeam</th>
<th>AwayTeam</th>
<th>Date</th>
<th>Season</th>
<th>WinningTeam</th>
<th>HomeScore</th>
<th>AwayScore</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>PLAYED_IN_GAME</th>
<th>PName</th>
<th>GameID</th>
<th>Position</th>
<th>PointsScored</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Conditions</th>
</tr>
</thead>
</table>
Question 8 - Functional Dependencies (9 points)

Consider the instance of the relation \( r(A,B,C) \):

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>fever</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>cold</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>fever</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>healthy</td>
</tr>
</tbody>
</table>

State whether the following Functional Dependencies are satisfied by the relation above (circle yes or no).

<table>
<thead>
<tr>
<th>FD</th>
<th>Satisfied?</th>
</tr>
</thead>
<tbody>
<tr>
<td>A ( \rightarrow ) B</td>
<td>Yes / No</td>
</tr>
<tr>
<td>B ( \rightarrow ) A</td>
<td>Yes / No</td>
</tr>
<tr>
<td>C ( \rightarrow ) B</td>
<td>Yes / No</td>
</tr>
<tr>
<td>C ( \rightarrow ) A</td>
<td>Yes / No</td>
</tr>
<tr>
<td>BC ( \rightarrow ) A</td>
<td>Yes / No</td>
</tr>
<tr>
<td>A ( \rightarrow ) ABC</td>
<td>Yes / No</td>
</tr>
<tr>
<td>B ( \rightarrow ) ABC</td>
<td>Yes / No</td>
</tr>
</tbody>
</table>

List a possible Candidate Key for \( r \), given the values in the instance above: \( \underline{A} \)
CERTIFICATION PAGE

By signing below, I promise that my answers on this exam are entirely my own work. I have not looked at the answers written by others and I have not allowed others to look at my answers. I also have not consulted any books or notes while taking this exam besides the 2 single sided pages allowed.

My Signature: