

Name: *SAMPLE SOLUTION*

Section (315/415/615):

## MIDTERM EXAM - 601.315/415/615 - Databases

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)

- (5 points) Create a View WINNER using the *relational algebra* that lists the HomeTeam (team name), WinningTeam (team name), LosingTeam (team name), GameID, Date and Season of all games in the database.

You can use this view WINNER for all other relational algebra questions on this exam.

Create VIEW Winner AS

```
PI HomeTeam,WinningTeam,LosingTeam,GameID,Date,Season
  (GAME JOIN_GameID=GameID
    (RHO(LosingTeam,GameID) PI_AwayTeam,GameID (SIGMA HomeTeam=WinningTeam GAME)
      UNION
      RHO(LosingTeam,GameID) PI_HomeTeam,GameID (SIGMA AwayTeam=WinningTeam GAME)))
```

### Question 2 - Relational Algebra (5 points)

Express the following query in the relational algebra:

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

$lostIn2018Away \leftarrow \Pi_{Team.TName,Team.HostCity}$

$(\sigma_{Game.AwayTeam=Team.TName \wedge Game.Season=2018 \wedge Game.winningTeam \neq Team.TName}(TEAM \times GAME))$

$LostIn2018Home \leftarrow \Pi_{Team.TName,Team.HostCity}$

$(\sigma_{Game.HomeTeam=Team.TName \wedge Game.Season=2018 \wedge Game.winningTeam \neq Team.TName}(TEAM \times GAME))$

$Result \leftarrow \Pi_{TName,HostCity}(Team) - LostIn2018Away - LostIn2018Home$

**601.415/615:**

$lostIn2018Away \leftarrow \Pi_{Team.TName, Team.HostCity}$

$(\sigma_{Game.AwayTeam=Team.TName \wedge Game.Season=2018 \wedge Game.winningTeam \neq Team.TName}(TEAM \times GAME))$

$LostIn2018Home \leftarrow \Pi_{Team.TName, Team.HostCity}$

$(\sigma_{Game.HomeTeam=Team.TName \wedge Game.Season=2018 \wedge Game.winningTeam \neq Team.TName}(TEAM \times GAME))$

$WinIn2017 \leftarrow \Pi_{Team.TName, Team.HostCity}$

$(\sigma_{Game.HomeTeam=Team.TName \wedge Game.Season=2017 \wedge Game.winningTeam=Team.TName}(TEAM \times GAME))$

$Result \leftarrow \Pi_{TName, HostCity}(Team) - LostIn2018Away - LostIn2018Home - WinIn2017$

### Question 3 - Relational Algebra (5 points)

- (a) (5 points) **601.315 only:** List the GameID and Date of games where the CityofBirth of the “Head” referee is the same as the HostCity of the winning team.
- (b) (5 points) **601.415/615 only:** List the GameID and Date of games where the CityofBirth of the “Head” referee is the same as the HostCity of the winning team **and** where the CountryofBirth of the “Head” referee is **not** the same as the CountryofBirth of any of the players on the

315:

$\Pi_{GameID, Date}$  (

$SIGMA_{RefereePosition='Head' \wedge Player.CityOfBirth=Team.HostCity}$   
(Player JOIN Referees\_Game JOIN Game JOIN Team)

415/615:

$\Pi_{GameID, Date}$  (

$SIGMA_{RefereePosition='Head' \wedge Player.CityOfBirth=Team.HostCity}$   
(Player JOIN Referees\_Game JOIN Game JOIN Team)

-

$SIGMA_{RefereePosition='Head' \wedge Player.CountryOfBirth \neq Player2.CountryofBirth}$   
(Player JOIN Referees\_Game JOIN Game JOIN RHO\_Player2(Player) JOIN PlayedInGame)

#### Question 4 - Relational Algebra (5 points)

Express the following query in the relational algebra.

- (a) (5 points) **601.315 only:** List the name and birthdate for players who have played for at least one team which has never won a match.
- (b) (5 points) **601.415/615 only:** List the name and birthdate for players who have played for at least one team which has never won a match, but has refereed for the same game that they have also played in.

(4a)

```
TeamHasWonAMatch <- PI_WinningTeam (GAME)
TeamHasNeverWonAMatch <- PI_Tname (TEAM) - RHO(Tname)_TeamHasWonAMatch
ResultPlayer <- PI_Name,Birthdate (Player JOIN PlaysForTeam JOIN TeamHasNeverWonAMatch
```

(4b)

```
TeamHasWonAMatch <- PI_WinningTeam (GAME)

TeamHasNeverWonAMatch <- PI_Tname (TEAM) - RHO(Tname)_TeamHasWonAMatch

ResultPlayer1 <- PI_Name,Birthdate (Player JOIN PlaysForTeam JOIN TeamHasNeverWonAMatch

ResultPlayer2 <- PI_Name,Birthdate (SIGMA Player.Pname=Referees_Game.Pname ^
                                     Player.Pname=PlayedInGame.Pname ^
                                     Referees_Game.GameID=PlayedInGame.GameID
                                     (Player X Referees_Game X PlayedInGame)

Result <-ResultPlayer1 INTERSECT ResultPlayer2
```

### 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.

$$\{ t \mid \exists p \in \text{PLAYER}(t[\text{name}] = p[\text{name}] \wedge p[\text{Bdate}] = t[\text{Bdate}] \wedge t[\text{sex}] = p[\text{sex}] \wedge t[\text{CountryofBirth}] = p[\text{CountryofBirth}] \wedge \exists s \in \text{Player}(s[\text{PName}] = \text{"WallyWinger"} \wedge s[\text{CountryofBirth}] = p[\text{CountryofBirth}] \wedge s[\text{Bdate}] = p[\text{Bdate}] \wedge s[\text{sex}] = p[\text{sex}] \wedge \exists pf \in \text{Plays\_for}(pf[\text{Pname}] = \text{"WallyWinger"} \wedge \exists pf2 \in \text{Plays\_for}(pf2[\text{Pname}] = p[\text{name}] \wedge pf2[\text{Tname}] = pf[\text{Tname}] \wedge \neg(p[\text{CityofBirth}] = s[\text{CityofBirth}]))))\}$$

- (aa) (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).

$$\{ t \mid \exists p \in \text{PLAYER}(t[\text{name}] = p[\text{name}] \wedge p[\text{Bdate}] = t[\text{Bdate}] \wedge t[\text{sex}] = p[\text{sex}] \wedge t[\text{CountryofBirth}] = p[\text{CountryofBirth}] \wedge \exists s \in \text{Player}(s[\text{PName}] = \text{"WallyWinger"} \wedge s[\text{CountryofBirth}] = p[\text{CountryofBirth}] \wedge s[\text{Bdate}] = p[\text{Bdate}] \wedge s[\text{sex}] = p[\text{sex}] \wedge \exists pf \in \text{Plays\_for}(pf[\text{Pname}] = \text{"WallyWinger"} \wedge \exists pf2 \in \text{Plays\_for}(pf2[\text{Pname}] = p[\text{name}] \wedge pf2[\text{Tname}] = pf[\text{Tname}] \wedge \neg(\exists g \in \text{Played\_in\_game}(t[\text{PName}] = g[\text{PName}] \wedge \exists g1 \in \text{Played\_in\_game}(g1[\text{PName}] = \text{"WallyWinger"} \wedge g1[\text{GameID}] = g[\text{GameID}]))))))))\}$$

**Question 5 - Tuple Relational Calculus (6 points) - ALTERNATIVE SOLUTION**

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.
- (aa) (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| EXISTS P1 IN PLAYER           ^
  EXISTS P2 IN PLAYER           ^
  EXISTS PF1 IN PLAYS_FOR       ^
  EXISTS PF2 IN PLAYS_FOR       ^
  EXISTS PG2 IN PLAYED_IN_GAME  ^   # omit for 315
  NEXISTS PG1 IN PLAYED_IN_GAME ^   # omit for 315

  P2.Pname = 'Wally Winger'     ^
  P1.CountryOfBirth = P2.CountryOfBirth ^
  P1.BDate = P2.BDate          ^
  P1.Sex = P2.Sex              ^

  NOT P1.CityOfBirth = P2.CityOfBirth ^

  P1.Pname = PF1.Pname         ^
  P2.Pname = PF2.Pname         ^
  PF1.TName = PF2.TName       ^

  PG2.GameID = PG1.GameID      ^   # omit for 315

  T.Lname = P1.Lname           ^
  T.Birthdate = P1.Birthdate   ^
  T.Sex = P1.Sex               ^
  T.CountryOfBirth = P1.CountryOfBirth } } } } } } } }
```

### 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.

```
SELECT p.PName, p.CountryofBirth
FROM PLAYER p, TEAM t, PLAYS_FOR f
WHERE p.PName=f.PName ^ f.season = 2018 ^ f.TName=t.TName
      ^ t.HostCountry!=p.CountryofBirth ^ p.CountryofBirth!='USA'
```

- (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.

```
SELECT p.PName, p.CountryofBirth
FROM PLAYER p, TEAM t, PLAYS_FOR pf
WHERE p.PName=pf.PName ^ pf.TName=t.TName ^ 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.

```
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 ^ 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.

```
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 ^ 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.

```

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.

```

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).

```

SELECT T.PName
FROM
  (SELECT Points2017.PName as PName, (Points2018.PS - Points2017.PS) as diff
FROM
  (SELECT FIG.PName as PName, SUM(FIG.PointsScored) as PS
FROM Played_In_Game FIG
JOIN Player P ON FIG.PName = P.PName
JOIN Game G ON FIG.GameID = G.GameID
GROUP BY FIG.PName
WHERE P.Sex = 'Female'
AND G.Season = 2017) as Points2017
JOIN
  (SELECT FIG.PName as PName, SUM(FIG.PointsScored) as PS
FROM Played_In_Game FIG
JOIN Player P ON FIG.PName = P.PName
JOIN Game G ON FIG.GameID = G.GameID
GROUP BY FIG.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 FIG.PName as PName, SUM(FIG.PointsScored) as PS
FROM Played_In_Game FIG
JOIN Player P ON FIG.PName = P.PName
JOIN Game G ON FIG.GameID = G.GameID
GROUP BY FIG.PName
WHERE P.Sex = 'Female'
AND G.Season = 2017) as Points2017v2
JOIN
  (SELECT FIG.PName as PName, SUM(FIG.PointsScored) as PS
FROM Played_In_Game FIG
JOIN Player P ON FIG.PName = P.PName
JOIN Game G ON FIG.GameID = G.GameID
GROUP BY FIG.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.

PLAYER	<u>PName</u>	BDate	Sex	CityOfBirth	CountryofBirth
	<b>P.<sub>pn</sub></b> <b>Wally Whinger</b>	<b>P.</b>			<b>_c</b> <b>≠ _c</b>

TEAM	HostCity	<u>TName</u>	HostCountry

PLAYS_FOR	<u>PName</u>	<u>TName</u>	<u>Season</u>
	<b>-pn</b>	<b>-t1</b>	
	<b>-p2</b>	<b>-t2</b>	
	<b>Wally Whinger</b>	<b>-t3</b>	

GAME	<u>GameID</u>	HomeTeam	AwayTeam	Date	Season	WinningTeam	HomeScore	AwayScore

PLAYED_IN_GAME	<u>PName</u>	<u>GameID</u>	Position	PointsScored
	<b>-pn</b>	<b>-g1</b>		
	<b>-p2</b>	<b>-g1</b>		
	<b>-p2</b>	<b>-g2</b>		
	<b>Wally Whinger</b>	<b>-g2</b>		

Conditions

- (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).

PLAYER	<u>PName</u>	BDate	Sex	CityOfBirth	CountryofBirth
	<b>P._pn</b> Wally Whinger	<b>P._b</b> ¬_b	<b>p._s</b> ¬_s		<b>P._c</b> ¬_c

TEAM	HostCity	<u>TName</u>	HostCountry

PLAYS_FOR	<u>PName</u>	<u>TName</u>	<u>Season</u>
¬	<b>_pn</b> Wally Whinger	<b>_t</b> _t	

GAME	<u>GameID</u>	HomeTeam	AwayTeam	Date	Season	WinningTeam	HomeScore	AwayScore

PLAYED_IN_GAME	<u>PName</u>	<u>GameID</u>	Position	PointsScored
¬	<b>_pn</b> Wally Whinger	<b>_g</b> _g		

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.

PLAYER	<u>PName</u>	BDate	Sex	CityOfBirth	CountryofBirth
	<b>P._pn</b>	<b>P._bd</b>		<b>P._citb</b>	<b>_coub</b>

TEAM	HostCity	<u>TName</u>	HostCountry
	<b>_citb</b>	<b>_tn</b>	<b>_coub</b>

PLAYS_FOR	<u>PName</u>	<u>TName</u>	<u>Season</u>
	<b>_pn</b>	<b>_tn</b>	

GAME	<u>GameID</u>	HomeTeam	AwayTeam	Date	Season	WinningTeam	HomeScore	AwayScore

PLAYED_IN_GAME	<u>PName</u>	<u>GameID</u>	Position	PointsScored

Conditions

**Question 8 - Functional Dependencies (10 points)**

(a) Consider the relation  $r(A,B,C,D,E)$  with functional dependencies:

$$BC \rightarrow D, D \rightarrow E, CE \rightarrow A$$

	Circle One	If yes, briefly show your derivation
Does $D \rightarrow A$ ?	<b>Yes/No</b>	
Does $CD \rightarrow E$ ?	<b>Yes/No</b>	$CD \rightarrow D$ (Decomposition) and $D \rightarrow E$ transitivity
Does $CDE \rightarrow A$ ?	<b>Yes/No</b>	$CDE \rightarrow CE$ (Decomposition) and $CE \rightarrow A$ Transitivity
Does $BCD \rightarrow A$ ?	<b>Yes/No</b>	$BCD \rightarrow D$ and $D \rightarrow E$ so $BCD \rightarrow E$ , $BCD \rightarrow C$ , $BCD \rightarrow CE$ , $CE \rightarrow A$ , $BCD \rightarrow A$

(b) List at least one candidate key for  $r$  (informally show your work):

$BC \rightarrow B$  by decomposition

$BC \rightarrow C$  by decomposition

$BC \rightarrow D$  given

$BC \rightarrow E$  because  $BC \rightarrow D$  and  $D \rightarrow E$

$BC \rightarrow A$  because  $BC \rightarrow CE$  (by union of  $BC \rightarrow C$  and  $BC \rightarrow E$  above), and  $CE \rightarrow A$  so transitive

$BC \rightarrow ABCDE$  so  $BC \rightarrow r$  so  $BC$  is a superkey

$B$  is not a superkey because doesn't derive  $A$  by itself

$C$  is not a superkey because doesn't derive  $A$  by itself

So  $BC$  is a minimal superkey, thus  $BC$  is a candidate key.

## **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:**