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

\[
\text{PI HomeTeam, WinningTeam, LosingTeam, GameID, Date, Season} \\
\text{ (GAME JOIN GameID=GameID} \\
\text{ (RHO(LosingTeam, GameID) PI_AwayTeam, GameID (SIGMA HomeTeam=WinningTeam GAME)} \\
\text{ UNION} \\
\text{ RHO(LosingTeam, GAMEID) PI_HomeTeam, GameID (SIGMA AwayTeam=WinningTeam GAME)))}
\]

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:

\[
\text{lostIn2018Away} \leftarrow \Pi_{Team.TName, Team.HostCity} \\
(\sigma_{\text{Game.AwayTeam}=Team.TName} \land \text{Game.Season}=2018 \land \text{Game.winningTeam} \neq Team.TName (TEAM \times GAME)) \\
\text{LostIn2018Home} \leftarrow \Pi_{Team.TName, Team.HostCity} \\
(\sigma_{\text{Game.HomeTeam}=Team.TName} \land \text{Game.Season}=2018 \land \text{Game.winningTeam} \neq Team.TName (TEAM \times GAME)) \\
\text{Result} \leftarrow \Pi_{Team.HostCity (Team)} - \text{LostIn2018Away} - \text{LostIn2018Home}
\]
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' AND Player.CityOfBirth=Team.HostCity
  (Player JOIN Referees_Game JOIN Game JOIN Team)
```

415/615:

```
PI_GameID,Date (  
  SIGMA_RefereePosition='Head' AND Player.CityOfBirth=Team.HostCity
  (Player JOIN Referees_Game JOIN Game JOIN Team)
  -
    (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)
\[
\text{TeamHasWonAMatch } \leftarrow \text{PI WinningTeam (GAME)} \\
\text{TeamHasNeverWonAMatch } \leftarrow \text{PI Tname (TEAM) - RHO(Tname) TeamHasWonAMatch} \\
\text{ResultPlayer } \leftarrow \text{PI Name,Birthdate (Player JOIN PlaysForTeam JOIN TeamHasNeverWonAMatch} \\
\]

(4b)
\[
\text{TeamHasWonAMatch } \leftarrow \text{PI WinningTeam (GAME)} \\
\text{TeamHasNeverWonAMatch } \leftarrow \text{PI Tname (TEAM) - RHO(Tname) TeamHasWonAMatch} \\
\text{ResultPlayer1 } \leftarrow \text{PI Name,Birthdate (Player JOIN PlaysForTeam JOIN TeamHasNeverWonAMatch} \\
\text{ResultPlayer2 } \leftarrow \text{PI Name,Birthdate (SIGMA Player.Pname=Referees_Game.Pname ^} \\
\text{Player.Pname=PlayedInGame.Pname ^} \\
\text{Referees_Game.GameID=PlayedInGame.GameID} \\
\text{(Player X Referees_Game X PlayedInGame) } \\
\text{Result } \leftarrow \text{ResultPlayer1 INTERSECT ResultPlayer2} \\
\]
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 | \exists p \in \text{PLAYER}(t[\text{name}] = p[\text{name}] \land p[\text{Bdate}] = t[\text{Bdate}] \land t[\text{sex}] = p[\text{sex}] \land t[\text{Country of Birth}] = p[\text{Country of Birth}] \\
\land \exists s \in \text{Player}(s[\text{PName}] = "WallyWinger" \land s[\text{Country of Birth}] = p[\text{Country of Birth}] \\
\land s[\text{Bdate}] = p[\text{Bdate}] \land s[\text{sex}] = p[\text{sex}] \\
\land \exists pf \in \text{Plays for}(pf[\text{Pname}] = "WallyWinger" \\
\land \exists pf2 \in \text{Plays for}(pf2[\text{Pname}] = p[\text{name}] \land pf2[\text{Tname}] = pf[\text{Tname}] \land \neg (p[\text{City of Birth}] = s[\text{City of Birth}])))}

(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 p \in \text{PLAYER}(t[\text{name}] = p[\text{name}] \land p[\text{Bdate}] = t[\text{Bdate}] \land t[\text{sex}] = p[\text{sex}] \land t[\text{Country of Birth}] = p[\text{Country of Birth}] \\
\land \exists s \in \text{Player}(s[\text{PName}] = "WallyWinger" \land s[\text{Country of Birth}] = p[\text{Country of Birth}] \\
\land s[\text{Bdate}] = p[\text{Bdate}] \land s[\text{sex}] = p[\text{sex}] \\
\land \exists pf \in \text{Plays for}(pf[\text{Pname}] = "WallyWinger" \\
\land \exists pf2 \in \text{Plays for}(pf2[\text{Pname}] = p[\text{name}] \land pf2[\text{Tname}] = pf[\text{Tname}] \\
\land \neg (\exists g \in \text{Played in game}(t[\text{PName}] = g[\text{PName}] \\
\land \exists g1 \in \text{Played in game}(g1[\text{PName}] = "WallyWinger" \land g1[\text{Game ID}] = g[\text{Game ID}])))))

\}}
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 \land \exists P2 \in PLAYER \land \exists PF1 \in PLAYS_FOR \land \exists PF2 \in PLAYS_FOR \land \exists PG2 \in PLAYED_IN_GAME \land \neg \exists PG1 \in PLAYED_IN_GAME \land P2.Pname = 'Wally Winger' \land P1.CountryOfBirth = P2.CountryOfBirth \land P1.BDate = P2.BDate \land P1.Sex = P2.Sex \land \neg P1.CityOfBirth = P2.CityOfBirth \land P1.Pname = PF1.Pname \land P2.Pname = PF2.Pname \land PF1.TName = PF2.TName \land PG2.GameID = PG1.GameID \land T.Lname = P1.Lname \land T.Birthdate = P1.Birthdate \land T.Sex = P1.Sex \land 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.

```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 ^ 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 ^ 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
```

8
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 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
JOIN
(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 = 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 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
JOIN
(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 = 2018) as Points2018v2
ON Points2017v2PName = Points2018v2.PName)
(c) **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
```

(ec) **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 Winger.

<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.</td>
<td></td>
<td></td>
<td></td>
<td>¬ .c</td>
</tr>
<tr>
<td>Wally Whinger</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>¬ .c</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></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></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>_pn</td>
<td></td>
<td>_t1</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>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</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></td>
<td>_g1</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></td>
<td>_g2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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).

<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>\neg b</td>
<td>\neg s</td>
<td></td>
<td></td>
<td>\neg c</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></td>
<td></td>
<td></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>\neg</td>
<td>.pn</td>
<td>.t</td>
<td></td>
</tr>
<tr>
<td>Wally Whinger</td>
<td>.t</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>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</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>\neg</td>
<td>.pn</td>
<td>.g</td>
<td>.g</td>
<td></td>
</tr>
<tr>
<td>Wally Whinger</td>
<td>.g</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>
(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>P_pn</td>
<td>P_bd</td>
<td></td>
<td></td>
<td>P_citb</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>_citb</td>
<td>_tn</td>
<td>_coub</td>
<td></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>_pn</td>
<td>_tn</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>
</table>

<table>
<thead>
<tr>
<th>Conditions</th>
</tr>
</thead>
</table>
Question 8 - Functional Dependencies (10 points)

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

\[
BC \rightarrow D, \ D \rightarrow E, \ CE \rightarrow A
\]

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

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

BC->B by decomposition
BC->C by decomposition
BC->D given
BC->E because BC->D and D->E
BC->A because BC->CE (by union of BC->C and BC->E above), and CE->A so transitive

BC->ABCDE so BC->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: