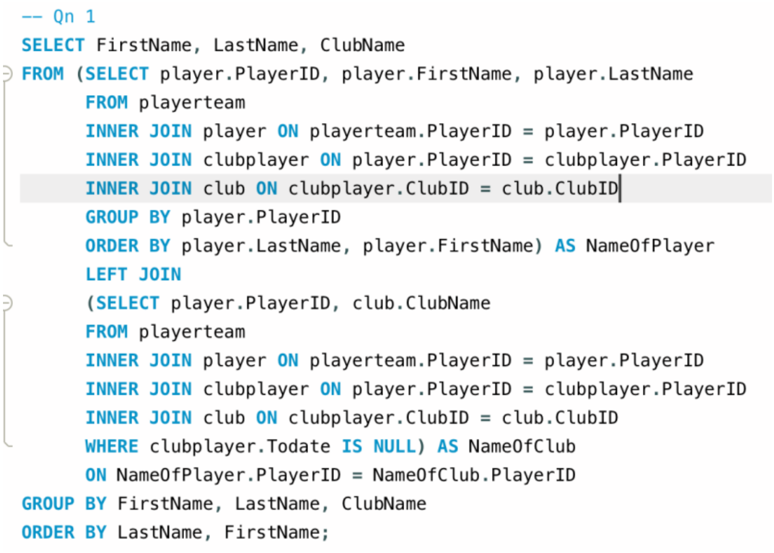
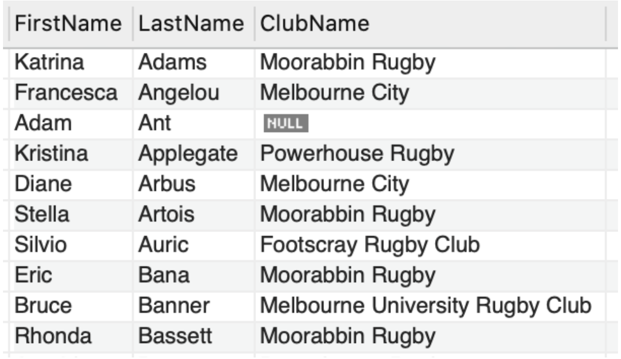
Q 1. List the first name and last name of all players who have ever played in the league and their current club if applicable. List the players in ascending alphabetical order, primarily on last name and secondarily on first name. A game is still considered to have been 'played' even if it was cancelled or a walkover. (1 mark)

```
SELECT FirstName, LastName, ClubName
FROM (SELECT player.PlayerID, player.FirstName, player.LastName
          FROM playerteam
    INNER JOIN player ON playerteam.PlayerID = player.PlayerID
    INNER JOIN clubplayer ON player.PlayerID = clubplayer.PlayerID
    INNER JOIN club ON clubplayer.ClubID = club.ClubID
    GROUP BY player.PlayerID
    ORDER BY player.LastName, player.FirstName) AS NameOfPlayer
    LEFT JOIN
    (SELECT player.PlayerID, club.ClubName
    FROM playerteam
    INNER JOIN player ON playerteam.PlayerID = player.PlayerID
    INNER JOIN clubplayer ON player.PlayerID = clubplayer.PlayerID
    INNER JOIN club ON clubplayer.ClubID = club.ClubID
    WHERE clubplayer.Todate IS NULL) AS NameOfClub
    ON NameOfPlayer.PlayerID = NameOfClub.PlayerID
GROUP BY FirstName, LastName, ClubName
ORDER BY LastName, FirstName;
```



```
-- Qn 1
SELECT FirstName, LastName, ClubName
FROM (SELECT player.PlayerID, player.FirstName, player.LastName
      FROM playerteam
      INNER JOIN player ON playerteam.PlayerID = player.PlayerID
      INNER JOIN clubplayer ON player.PlayerID = clubplayer.PlayerID
      INNER JOIN club ON clubplayer.ClubID = club.ClubID
      GROUP BY player.PlayerID
      ORDER BY player.LastName, player.FirstName) AS NameOfPlayer
      LEFT JOIN
      (SELECT player.PlayerID, club.ClubName
      FROM playerteam
      INNER JOIN player ON playerteam.PlayerID = player.PlayerID
      INNER JOIN clubplayer ON player.PlayerID = clubplayer.PlayerID
      INNER JOIN club ON clubplayer.ClubID = club.ClubID
      WHERE clubplayer.Todate IS NULL) AS NameOfClub
      ON NameOfPlayer.PlayerID = NameOfClub.PlayerID
GROUP BY FirstName, LastName, ClubName
ORDER BY LastName, FirstName;
```

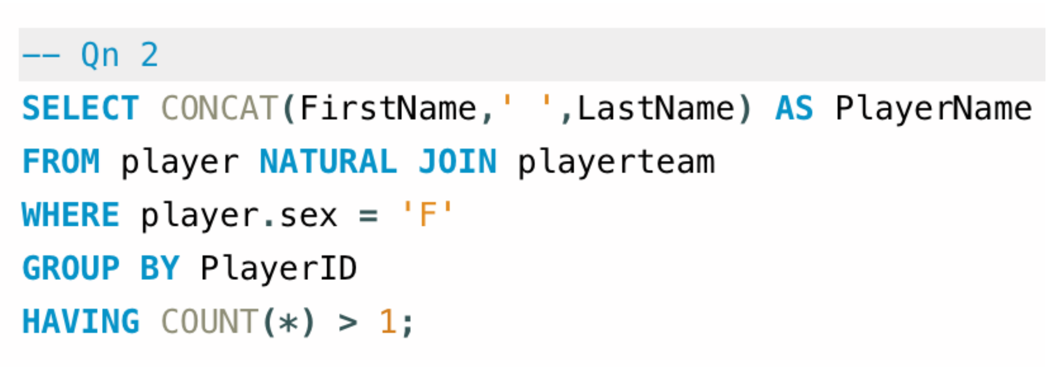


FirstName	LastName	ClubName
Katrina	Adams	Moorabbin Rugby
Francesca	Angelou	Melbourne City
Adam	Ant	NULL
Kristina	Applegate	Powerhouse Rugby
Diane	Arbus	Melbourne City
Stella	Artois	Moorabbin Rugby
Silvio	Auric	Footscray Rugby Club
Eric	Bana	Moorabbin Rugby
Bruce	Banner	Melbourne University Rugby Club
Rhonda	Bassett	Moorabbin Rugby

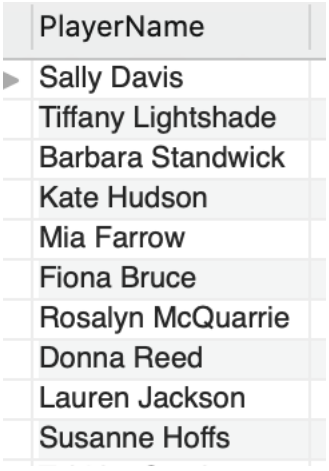
143 rows returned

Q2. List the full name (presented as a single string) of female players who have played for more than one team. (1 mark)

SELECT CONCAT(FirstName,' ',LastName) AS PlayerName
FROM player NATURAL JOIN playerteam
WHERE player.sex = 'F'
GROUP BY PlayerID
HAVING COUNT(*) > 1;



```
-- Qn 2
SELECT CONCAT(FirstName,' ',LastName) AS PlayerName
FROM player NATURAL JOIN playerteam
WHERE player.sex = 'F'
GROUP BY PlayerID
HAVING COUNT(*) > 1;
```

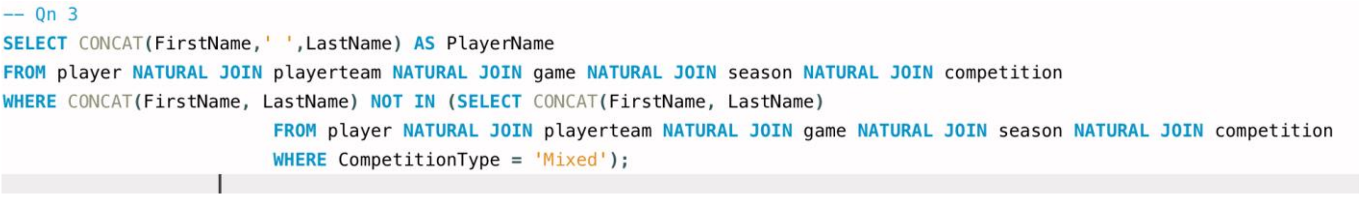


PlayerName
Sally Davis
Tiffany Lightshade
Barbara Standwick
Kate Hudson
Mia Farrow
Fiona Bruce
Rosalyn McQuarrie
Donna Reed
Lauren Jackson
Susanne Hoffs

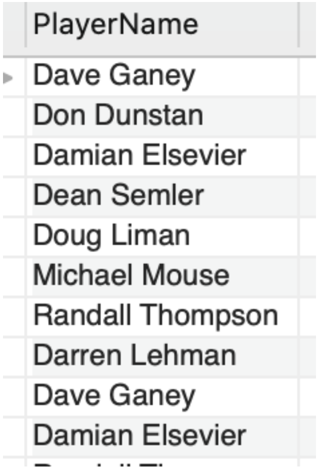
71 rows returned

Q3. List the full name (presented as a single string) of players who have never played in a 'mixed' competition. (1 mark)

```
SELECT CONCAT(FirstName,' ',LastName) AS PlayerName
FROM player NATURAL JOIN playerteam NATURAL JOIN game NATURAL JOIN season
NATURAL JOIN competition
WHERE CONCAT(FirstName, LastName) NOT IN (SELECT CONCAT(FirstName, LastName)
                                            FROM player NATURAL JOIN playerteam
NATURAL JOIN game NATURAL JOIN season NATURAL JOIN competition
            WHERE CompetitionType = 'Mixed');
```



```
-- Qn 3
SELECT CONCAT(FirstName,' ',LastName) AS PlayerName
FROM player NATURAL JOIN playerteam NATURAL JOIN game NATURAL JOIN season NATURAL JOIN competition
WHERE CONCAT(FirstName, LastName) NOT IN (SELECT CONCAT(FirstName, LastName)
                          FROM player NATURAL JOIN playerteam NATURAL JOIN game NATURAL JOIN season NATURAL JOIN competition
                          WHERE CompetitionType = 'Mixed');
```

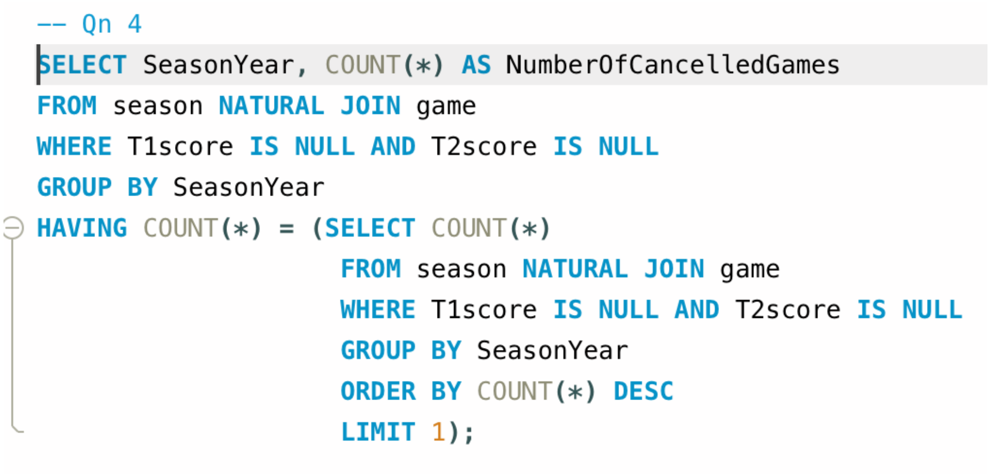


PlayerName
Dave Ganey
Don Dunstan
Damian Elsevier
Dean Semler
Doug Liman
Michael Mouse
Randall Thompson
Darren Lehman
Dave Ganey
Damian Elsevier

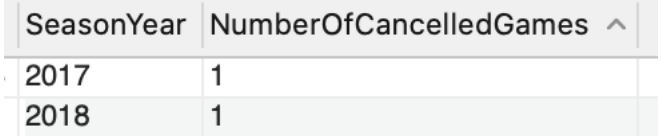
109 rows returned

Q4. List the season year(s) with the highest number of cancelled games, and the corresponding number of cancelled games in the year(s). (2 marks)

```
SELECT SeasonYear, COUNT(*) AS NumberOfCancelledGames
FROM season NATURAL JOIN game
WHERE T1score IS NULL AND T2score IS NULL
GROUP BY SeasonYear
HAVING COUNT(*) = (SELECT COUNT(*)
                                        FROM season NATURAL JOIN game
                                              WHERE T1score IS NULL AND T2score IS NULL
              GROUP BY SeasonYear
              ORDER BY COUNT(*) DESC
              LIMIT 1);
```



```
-- Qn 4
SELECT SeasonYear, COUNT(*) AS NumberOfCancelledGames
FROM season NATURAL JOIN game
WHERE T1score IS NULL AND T2score IS NULL
GROUP BY SeasonYear
HAVING COUNT(*) = (SELECT COUNT(*)
                    FROM season NATURAL JOIN game
                    WHERE T1score IS NULL AND T2score IS NULL
                    GROUP BY SeasonYear
                    ORDER BY COUNT(*) DESC
                    LIMIT 1);
```

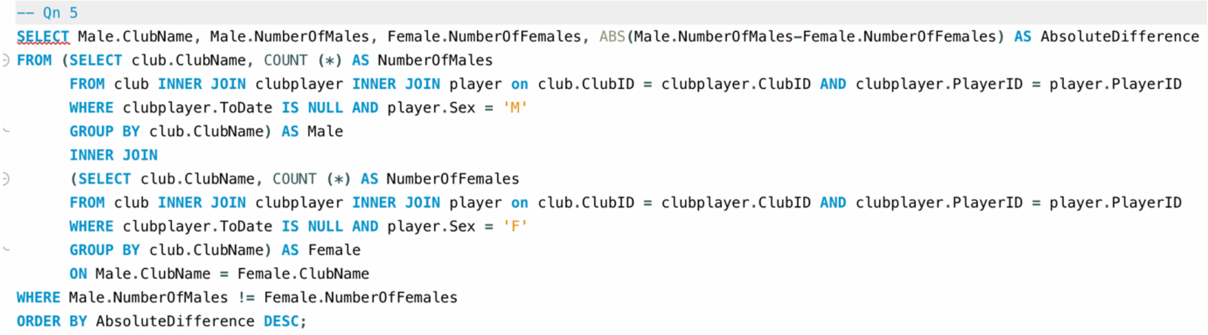


SeasonYear	NumberOfCancelledGames
2017	1
2018	1

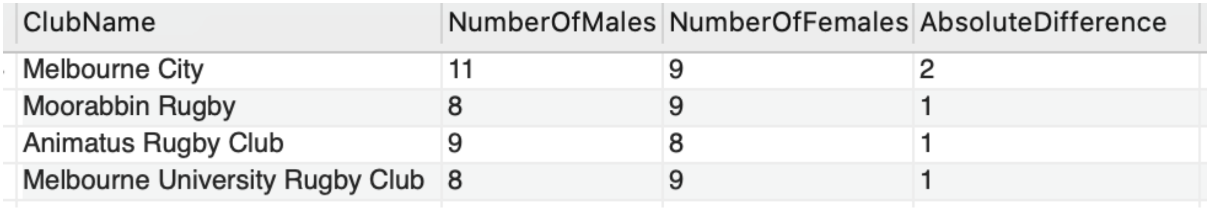
2 rows returned

Q5. List the name of the clubs where the number of current male members is not the same as the number of current female members. For these clubs, present the number of current male and female players, and the absolute difference between the number of current males and females in the club. Order the results by the absolute difference between current males and females in the club, with the club having the highest difference listed first. (2 marks)

```
SELECT Male.ClubName, Male.NumberOfMales, Female.NumberOfFemales,
ABS(Male.NumberOfMales-Female.NumberOfFemales) AS AbsoluteDifference
FROM (SELECT club.ClubName, COUNT (*) AS NumberOfMales
            FROM club INNER JOIN clubplayer INNER JOIN player on club.ClubID =
clubplayer.ClubID AND clubplayer.PlayerID = player.PlayerID
            WHERE clubplayer.ToDate IS NULL AND player.Sex = 'M'
            GROUP BY club.ClubName) AS Male
    INNER JOIN
    (SELECT club.ClubName, COUNT (*) AS NumberOfFemales
            FROM club INNER JOIN clubplayer INNER JOIN player on club.ClubID =
clubplayer.ClubID AND clubplayer.PlayerID = player.PlayerID
            WHERE clubplayer.ToDate IS NULL AND player.Sex = 'F'
            GROUP BY club.ClubName) AS Female
    ON Male.ClubName = Female.ClubName
WHERE Male.NumberOfMales != Female.NumberOfFemales
ORDER BY AbsoluteDifference DESC;
```



```
-- Qn 5
SELECT Male.ClubName, Male.NumberOfMales, Female.NumberOfFemales, ABS(Male.NumberOfMales-Female.NumberOfFemales) AS AbsoluteDifference
FROM (SELECT club.ClubName, COUNT (*) AS NumberOfMales
      FROM club INNER JOIN clubplayer INNER JOIN player on club.ClubID = clubplayer.ClubID AND clubplayer.PlayerID = player.PlayerID
      WHERE clubplayer.ToDate IS NULL AND player.Sex = 'M'
      GROUP BY club.ClubName) AS Male
      INNER JOIN
      (SELECT club.ClubName, COUNT (*) AS NumberOfFemales
      FROM club INNER JOIN clubplayer INNER JOIN player on club.ClubID = clubplayer.ClubID AND clubplayer.PlayerID = player.PlayerID
      WHERE clubplayer.ToDate IS NULL AND player.Sex = 'F'
      GROUP BY club.ClubName) AS Female
      ON Male.ClubName = Female.ClubName
WHERE Male.NumberOfMales != Female.NumberOfFemales
ORDER BY AbsoluteDifference DESC;
```

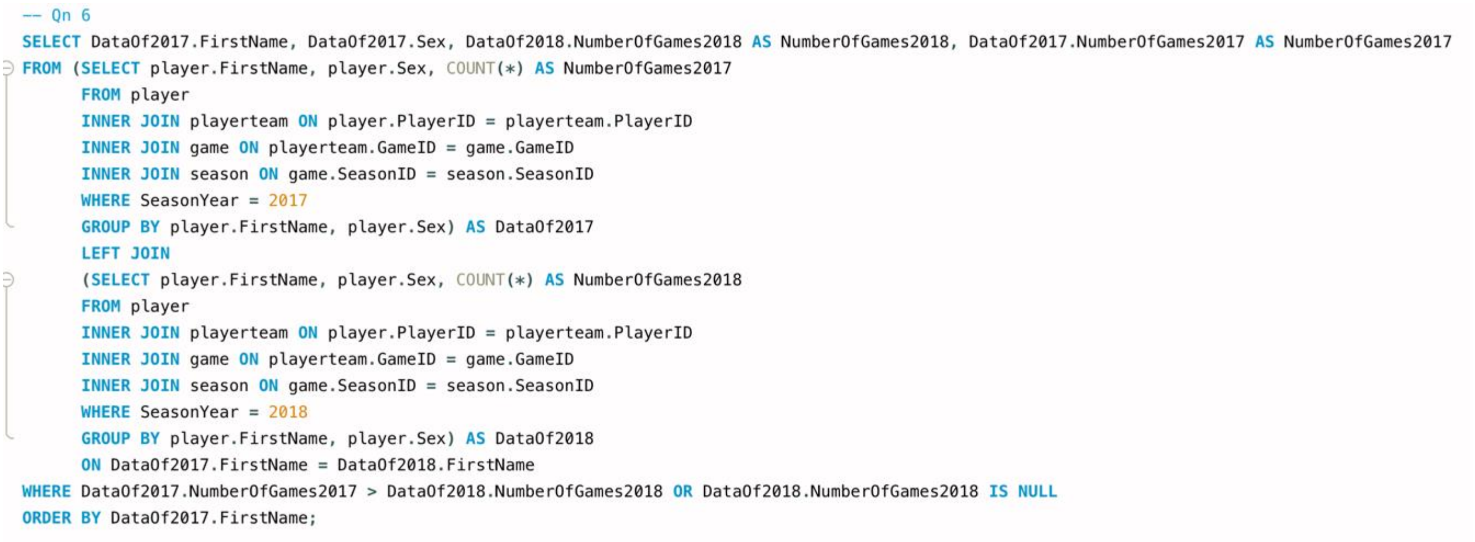


ClubName	NumberOfMales	NumberOfFemales	AbsoluteDifference
Melbourne City	11	9	2
Moorabbin Rugby	8	9	1
Animatus Rugby Club	9	8	1
Melbourne University Rugby Club	8	9	1

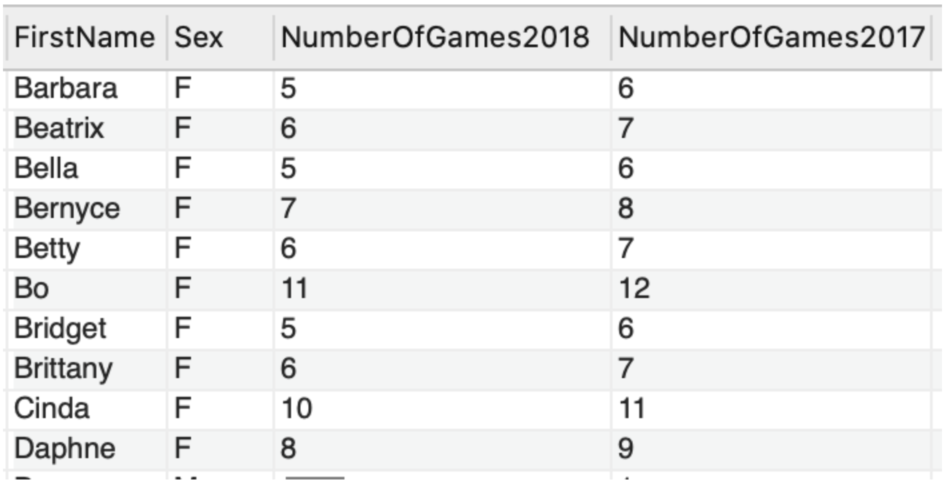
4 rows returned

Q6. List the first name, sex, and number of games played in 2018 and 2017 of players who played more games in 2017 than in 2018. There might be players that didn’t play in 2018 but did in 2017. You should take these cases in account as well. A game is still considered to have been 'played' even if it was cancelled or a walkover. (2 marks)

```
SELECT DataOf2017.FirstName, DataOf2017.Sex, DataOf2018.NumberOfGames2018 AS
NumberOfGames2018, DataOf2017.NumberOfGames2017 AS NumberOfGames2017
FROM (SELECT player.FirstName, player.Sex, COUNT(*) AS NumberOfGames2017
          FROM player
    INNER JOIN playerteam ON player.PlayerID = playerteam.PlayerID
    INNER JOIN game ON playerteam.GameID = game.GameID
    INNER JOIN season ON game.SeasonID = season.SeasonID
    WHERE SeasonYear = 2017
    GROUP BY player.FirstName, player.Sex) AS DataOf2017
    LEFT JOIN
    (SELECT player.FirstName, player.Sex, COUNT(*) AS NumberOfGames2018
          FROM player
    INNER JOIN playerteam ON player.PlayerID = playerteam.PlayerID
    INNER JOIN game ON playerteam.GameID = game.GameID
    INNER JOIN season ON game.SeasonID = season.SeasonID
    WHERE SeasonYear = 2018
    GROUP BY player.FirstName, player.Sex) AS DataOf2018
    ON DataOf2017.FirstName = DataOf2018.FirstName
WHERE DataOf2017.NumberOfGames2017 > DataOf2018.NumberOfGames2018 OR
DataOf2018.NumberOfGames2018 IS NULL
ORDER BY DataOf2017.FirstName;
```



```
-- Qn 6
SELECT DataOf2017.FirstName, DataOf2017.Sex, DataOf2018.NumberOfGames2018 AS NumberOfGames2018, DataOf2017.NumberOfGames2017 AS NumberOfGames2017
FROM (SELECT player.FirstName, player.Sex, COUNT(*) AS NumberOfGames2017
      FROM player
      INNER JOIN playerteam ON player.PlayerID = playerteam.PlayerID
      INNER JOIN game ON playerteam.GameID = game.GameID
      INNER JOIN season ON game.SeasonID = season.SeasonID
      WHERE SeasonYear = 2017
      GROUP BY player.FirstName, player.Sex) AS DataOf2017
      LEFT JOIN
      (SELECT player.FirstName, player.Sex, COUNT(*) AS NumberOfGames2018
      FROM player
      INNER JOIN playerteam ON player.PlayerID = playerteam.PlayerID
      INNER JOIN game ON playerteam.GameID = game.GameID
      INNER JOIN season ON game.SeasonID = season.SeasonID
      WHERE SeasonYear = 2018
      GROUP BY player.FirstName, player.Sex) AS DataOf2018
      ON DataOf2017.FirstName = DataOf2018.FirstName
WHERE DataOf2017.NumberOfGames2017 > DataOf2018.NumberOfGames2018 OR DataOf2018.NumberOfGames2018 IS NULL
ORDER BY DataOf2017.FirstName;
```

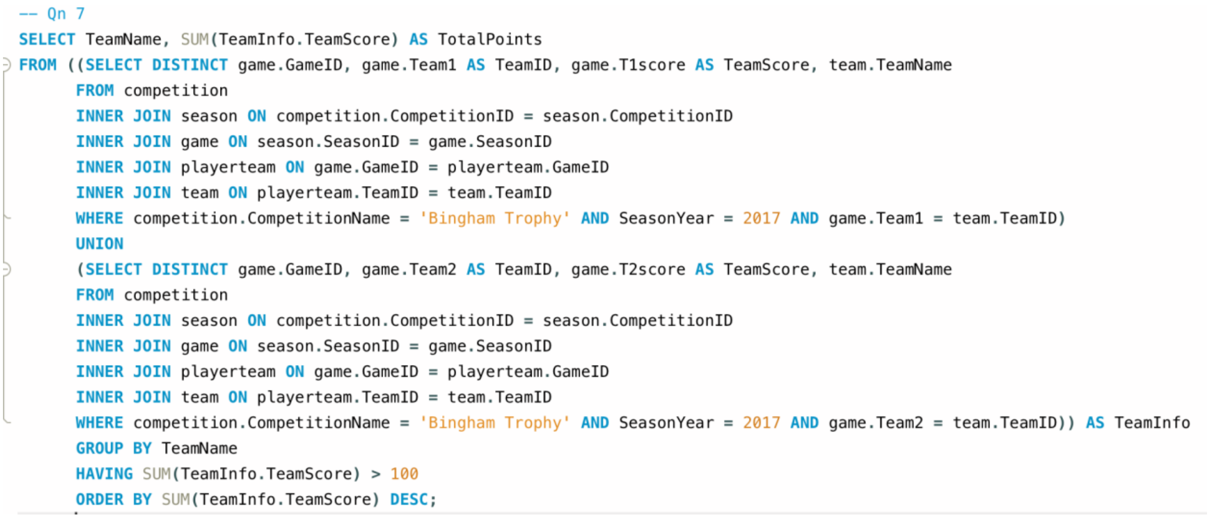


FirstName	Sex	NumberOfGames2018	NumberOfGames2017
Barbara	F	5	6
Beatrix	F	6	7
Bella	F	5	6
Bernyce	F	7	8
Betty	F	6	7
Bo	F	11	12
Bridget	F	5	6
Brittany	F	6	7
Cinda	F	10	11
Daphne	F	8	9

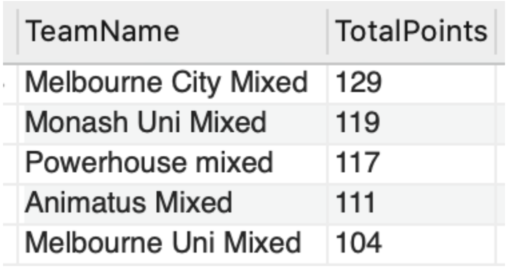
49 rows returned

Q7. List the name of all teams that scored more than 100 points in the Bingham Trophy in 2017, along with the total points they scored. Order the result according to the points the team scored, highest scoring team first. (2 marks)

```
SELECT TeamName, SUM(TeamInfo.TeamScore) AS TotalPoints
FROM ((SELECT DISTINCT game.GameID, game.Team1 AS TeamID, game.T1score AS
TeamScore, team.TeamName
            FROM competition
    INNER JOIN season ON competition.CompetitionID = season.CompetitionID
    INNER JOIN game ON season.SeasonID = game.SeasonID
    INNER JOIN playerteam ON game.GameID = playerteam.GameID
    INNER JOIN team ON playerteam.TeamID = team.TeamID
    WHERE competition.CompetitionName = 'Bingham Trophy' AND SeasonYear = 2017 AND
game.Team1 = team.TeamID)
    UNION
    (SELECT DISTINCT game.GameID, game.Team2 AS TeamID, game.T2score AS TeamScore,
team.TeamName
    FROM competition
    INNER JOIN season ON competition.CompetitionID = season.CompetitionID
    INNER JOIN game ON season.SeasonID = game.SeasonID
    INNER JOIN playerteam ON game.GameID = playerteam.GameID
    INNER JOIN team ON playerteam.TeamID = team.TeamID
    WHERE competition.CompetitionName = 'Bingham Trophy' AND SeasonYear = 2017 AND
game.Team2 = team.TeamID)) AS TeamInfo
    GROUP BY TeamName
    HAVING SUM(TeamInfo.TeamScore) > 100
    ORDER BY SUM(TeamInfo.TeamScore) DESC;
```



```
-- Qn 7
SELECT TeamName, SUM(TeamInfo.TeamScore) AS TotalPoints
FROM ((SELECT DISTINCT game.GameID, game.Team1 AS TeamID, game.T1score AS TeamScore, team.TeamName
      FROM competition
      INNER JOIN season ON competition.CompetitionID = season.CompetitionID
      INNER JOIN game ON season.SeasonID = game.SeasonID
      INNER JOIN playerteam ON game.GameID = playerteam.GameID
      INNER JOIN team ON playerteam.TeamID = team.TeamID
      WHERE competition.CompetitionName = 'Bingham Trophy' AND SeasonYear = 2017 AND game.Team1 = team.TeamID)
      UNION
      (SELECT DISTINCT game.GameID, game.Team2 AS TeamID, game.T2score AS TeamScore, team.TeamName
      FROM competition
      INNER JOIN season ON competition.CompetitionID = season.CompetitionID
      INNER JOIN game ON season.SeasonID = game.SeasonID
      INNER JOIN playerteam ON game.GameID = playerteam.GameID
      INNER JOIN team ON playerteam.TeamID = team.TeamID
      WHERE competition.CompetitionName = 'Bingham Trophy' AND SeasonYear = 2017 AND game.Team2 = team.TeamID)) AS TeamInfo
      GROUP BY TeamName
      HAVING SUM(TeamInfo.TeamScore) > 100
      ORDER BY SUM(TeamInfo.TeamScore) DESC;
```

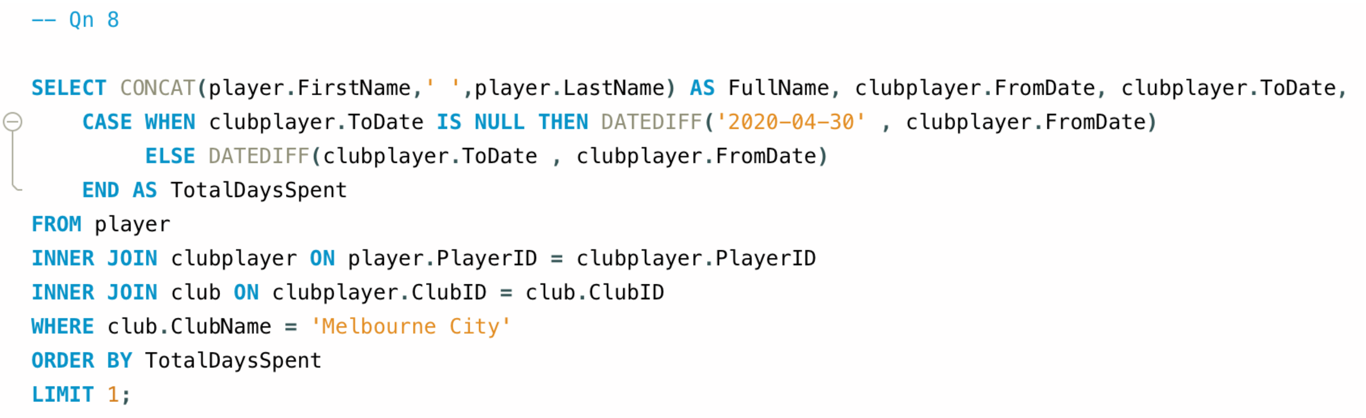


TeamName	TotalPoints
Melbourne City Mixed	129
Monash Uni Mixed	119
Powerhouse mixed	117
Animatus Mixed	111
Melbourne Uni Mixed	104

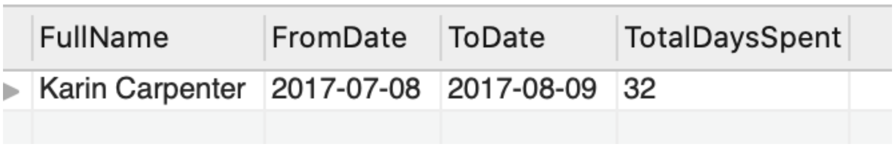
5 rows returned

Q8. For the player who has spent the fewest number of days as a member of ‘Melbourne City’ club as of April 30th 2020, give the full name (presented as a single string) and the total number of days spent as a member. Only consider players who have been a member of the club at some point (i.e. do not include players who have never been signed up with Melbourne City). Assume there are no joint shortest serving players. (3 marks)

```
SELECT CONCAT(player.FirstName,' ',player.LastName) AS FullName, clubplayer.FromDate,
clubplayer.ToDate,
        CASE WHEN clubplayer.ToDate IS NULL THEN DATEDIFF('2020-04-30' ,
clubplayer.FromDate)
                ELSE DATEDIFF(clubplayer.ToDate , clubplayer.FromDate)
        END AS TotalDaysSpent
FROM player
INNER JOIN clubplayer ON player.PlayerID = clubplayer.PlayerID
INNER JOIN club ON clubplayer.ClubID = club.ClubID
WHERE club.ClubName = 'Melbourne City'
ORDER BY TotalDaysSpent
LIMIT 1;
```



```
-- Qn 8

SELECT CONCAT(player.FirstName,' ',player.LastName) AS FullName, clubplayer.FromDate, clubplayer.ToDate,
    CASE WHEN clubplayer.ToDate IS NULL THEN DATEDIFF('2020-04-30' , clubplayer.FromDate)
         ELSE DATEDIFF(clubplayer.ToDate , clubplayer.FromDate)
    END AS TotalDaysSpent
FROM player
INNER JOIN clubplayer ON player.PlayerID = clubplayer.PlayerID
INNER JOIN club ON clubplayer.ClubID = club.ClubID
WHERE club.ClubName = 'Melbourne City'
ORDER BY TotalDaysSpent
LIMIT 1;
```

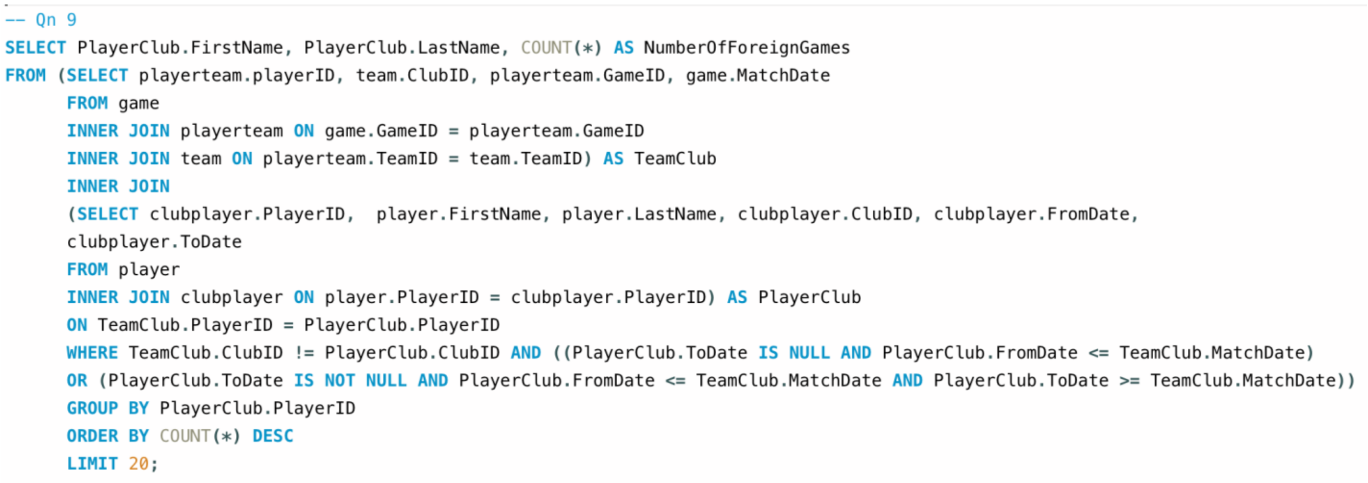


FullName	FromDate	ToDate	TotalDaysSpent
Karin Carpenter	2017-07-08	2017-08-09	32

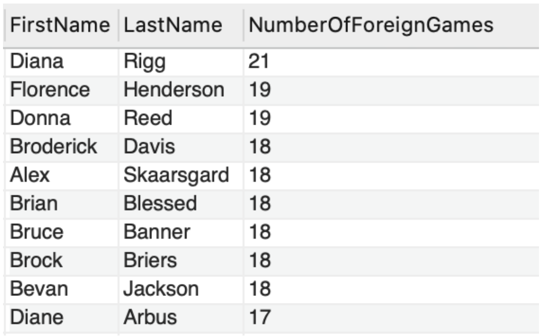
1 row returned

Q9. A ‘foreign game’ is when a player has played a game in a team which was not part of the club they were a member of at the time. Find the first and last names and the number of ‘foreign games’ played for the 20 players who have played the most ‘foreign games’. List them from highest number of foreign games to lowest. (3 marks)

```
SELECT PlayerClub.FirstName, PlayerClub.LastName, COUNT(*) AS NumberOfForeignGames
FROM (SELECT playerteam.playerID, team.ClubID, playerteam.GameID, game.MatchDate
          FROM game
    INNER JOIN playerteam ON game.GameID = playerteam.GameID
    INNER JOIN team ON playerteam.TeamID = team.TeamID) AS TeamClub
    INNER JOIN
    (SELECT clubplayer.PlayerID,  player.FirstName, player.LastName, clubplayer.ClubID,
clubplayer.FromDate,
    clubplayer.ToDate
    FROM player
    INNER JOIN clubplayer ON player.PlayerID = clubplayer.PlayerID) AS PlayerClub
    ON TeamClub.PlayerID = PlayerClub.PlayerID
    WHERE TeamClub.ClubID != PlayerClub.ClubID AND ((PlayerClub.ToDate IS NULL AND
PlayerClub.FromDate <= TeamClub.MatchDate)
    OR (PlayerClub.ToDate IS NOT NULL AND PlayerClub.FromDate <= TeamClub.MatchDate
AND PlayerClub.ToDate >= TeamClub.MatchDate))
    GROUP BY PlayerClub.PlayerID
    ORDER BY COUNT(*) DESC
    LIMIT 20;
```



```
-- Qn 9
SELECT PlayerClub.FirstName, PlayerClub.LastName, COUNT(*) AS NumberOfForeignGames
FROM (SELECT playerteam.playerID, team.ClubID, playerteam.GameID, game.MatchDate
      FROM game
      INNER JOIN playerteam ON game.GameID = playerteam.GameID
      INNER JOIN team ON playerteam.TeamID = team.TeamID) AS TeamClub
      INNER JOIN
      (SELECT clubplayer.PlayerID,  player.FirstName, player.LastName, clubplayer.ClubID, clubplayer.FromDate,
      clubplayer.ToDate
      FROM player
      INNER JOIN clubplayer ON player.PlayerID = clubplayer.PlayerID) AS PlayerClub
      ON TeamClub.PlayerID = PlayerClub.PlayerID
      WHERE TeamClub.ClubID != PlayerClub.ClubID AND ((PlayerClub.ToDate IS NULL AND PlayerClub.FromDate <= TeamClub.MatchDate)
      OR (PlayerClub.ToDate IS NOT NULL AND PlayerClub.FromDate <= TeamClub.MatchDate AND PlayerClub.ToDate >= TeamClub.MatchDate))
      GROUP BY PlayerClub.PlayerID
      ORDER BY COUNT(*) DESC
      LIMIT 20;
```



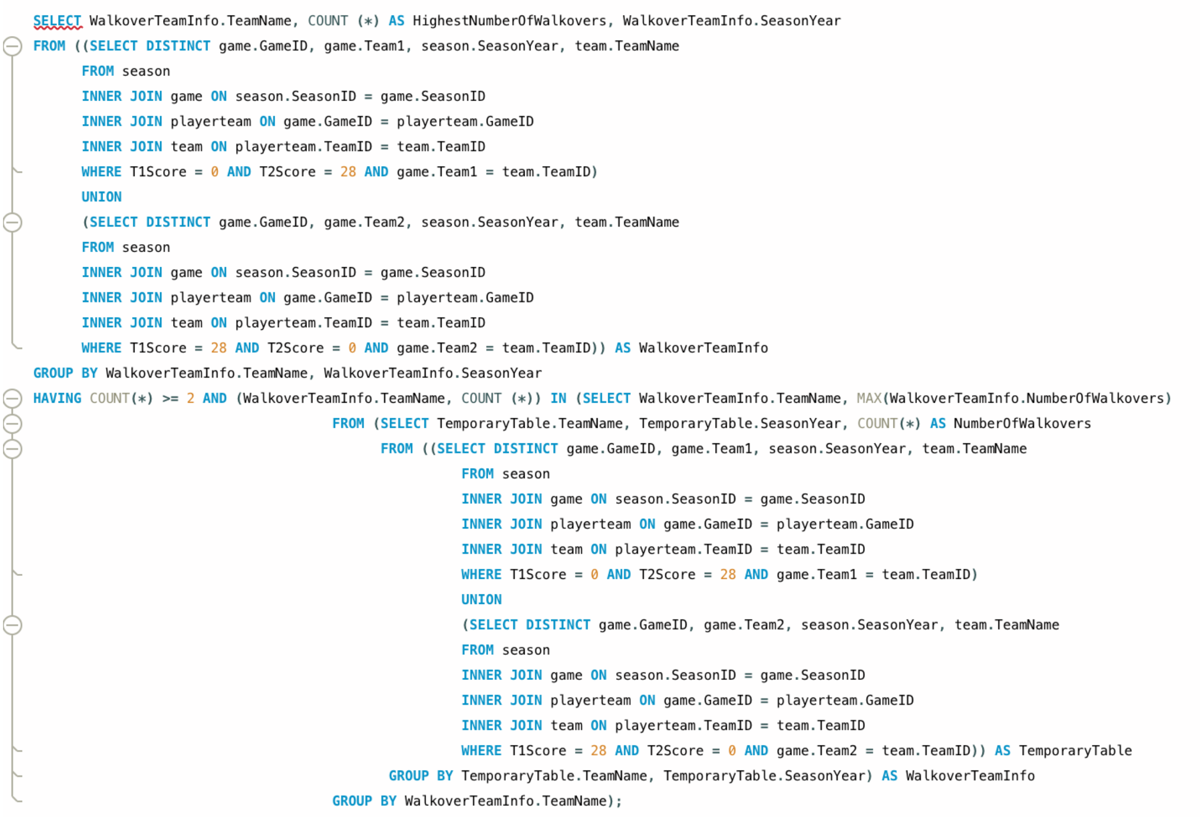
FirstName	LastName	NumberOfForeignGames
Diana	Rigg	21
Florence	Henderson	19
Donna	Reed	19
Broderick	Davis	18
Alex	Skaarsgard	18
Brian	Blessed	18
Bruce	Banner	18
Brock	Briers	18
Bevan	Jackson	18
Diane	Arbus	17

20 rows returned

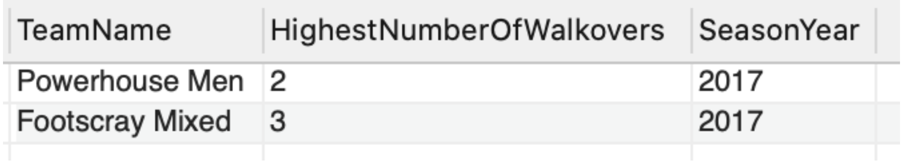
Q10. Return the team name, the highest number of walkovers (forfeits) the team has ever given away in a season, and the season year(s) during which this occurred for teams who have given away 2 or more walkovers in the same season. Assume that any game with one team scoring 28 and the other scoring 0 implies that the game was a walkover. (3 marks)

```
SELECT WalkoverTeamInfo.TeamName, COUNT (*) AS HighestNumberOfWalkovers,
WalkoverTeamInfo.SeasonYear
FROM ((SELECT DISTINCT game.GameID, game.Team1, season.SeasonYear, team.TeamName
            FROM season
    INNER JOIN game ON season.SeasonID = game.SeasonID
    INNER JOIN playerteam ON game.GameID = playerteam.GameID
    INNER JOIN team ON playerteam.TeamID = team.TeamID
    WHERE T1Score = 0 AND T2Score = 28 AND game.Team1 = team.TeamID)
    UNION
    (SELECT DISTINCT game.GameID, game.Team2, season.SeasonYear, team.TeamName
    FROM season
    INNER JOIN game ON season.SeasonID = game.SeasonID
    INNER JOIN playerteam ON game.GameID = playerteam.GameID
    INNER JOIN team ON playerteam.TeamID = team.TeamID
    WHERE T1Score = 28 AND T2Score = 0 AND game.Team2 = team.TeamID)) AS
WalkoverTeamInfo
GROUP BY WalkoverTeamInfo.TeamName, WalkoverTeamInfo.SeasonYear
HAVING COUNT(*) >= 2 AND (WalkoverTeamInfo.TeamName, COUNT (*)) IN (SELECT
WalkoverTeamInfo.TeamName, MAX(WalkoverTeamInfo.NumberOfWalkovers)
                                                                                    FROM (SELECT
TemporaryTable.TeamName, TemporaryTable.SeasonYear, COUNT(*) AS NumberOfWalkovers
                                                                                               FROM ((SELECT
DISTINCT game.GameID, game.Team1, season.SeasonYear, team.TeamName

FROM season

INNER JOIN game ON season.SeasonID = game.SeasonID

INNER JOIN playerteam ON game.GameID = playerteam.GameID

INNER JOIN team ON playerteam.TeamID = team.TeamID

WHERE T1Score = 0 AND T2Score = 28 AND game.Team1 = team.TeamID)

UNION

(SELECT DISTINCT game.GameID, game.Team2, season.SeasonYear, team.TeamName

FROM season

INNER JOIN game ON season.SeasonID = game.SeasonID

INNER JOIN playerteam ON game.GameID = playerteam.GameID

INNER JOIN team ON playerteam.TeamID = team.TeamID

WHERE T1Score = 28 AND T2Score = 0 AND game.Team2 = team.TeamID)) AS TemporaryTable
                                                                                                      GROUP
BY TemporaryTable.TeamName, TemporaryTable.SeasonYear) AS WalkoverTeamInfo
```

GROUP BY WalkoverTeamInfo.TeamName);



```
SELECT WalkoverTeamInfo.TeamName, COUNT (*) AS HighestNumberOfWalkovers, WalkoverTeamInfo.SeasonYear
FROM ((SELECT DISTINCT game.GameID, game.Team1, season.SeasonYear, team.TeamName
      FROM season
      INNER JOIN game ON season.SeasonID = game.SeasonID
      INNER JOIN playerteam ON game.GameID = playerteam.GameID
      INNER JOIN team ON playerteam.TeamID = team.TeamID
      WHERE T1Score = 0 AND T2Score = 28 AND game.Team1 = team.TeamID)
      UNION
      (SELECT DISTINCT game.GameID, game.Team2, season.SeasonYear, team.TeamName
      FROM season
      INNER JOIN game ON season.SeasonID = game.SeasonID
      INNER JOIN playerteam ON game.GameID = playerteam.GameID
      INNER JOIN team ON playerteam.TeamID = team.TeamID
      WHERE T1Score = 28 AND T2Score = 0 AND game.Team2 = team.TeamID)) AS WalkoverTeamInfo
GROUP BY WalkoverTeamInfo.TeamName, WalkoverTeamInfo.SeasonYear
HAVING COUNT(*) >= 2 AND (WalkoverTeamInfo.TeamName, COUNT (*)) IN (SELECT WalkoverTeamInfo.TeamName, MAX(WalkoverTeamInfo.NumberOfWalkovers)
                                     FROM (SELECT TemporaryTable.TeamName, TemporaryTable.SeasonYear, COUNT(*) AS NumberOfWalkovers
                                           FROM ((SELECT DISTINCT game.GameID, game.Team1, season.SeasonYear, team.TeamName
                                                  FROM season
                                                  INNER JOIN game ON season.SeasonID = game.SeasonID
                                                  INNER JOIN playerteam ON game.GameID = playerteam.GameID
                                                  INNER JOIN team ON playerteam.TeamID = team.TeamID
                                                  WHERE T1Score = 0 AND T2Score = 28 AND game.Team1 = team.TeamID)
                                                  UNION
                                                  (SELECT DISTINCT game.GameID, game.Team2, season.SeasonYear, team.TeamName
                                                  FROM season
                                                  INNER JOIN game ON season.SeasonID = game.SeasonID
                                                  INNER JOIN playerteam ON game.GameID = playerteam.GameID
                                                  INNER JOIN team ON playerteam.TeamID = team.TeamID
                                                  WHERE T1Score = 28 AND T2Score = 0 AND game.Team2 = team.TeamID)) AS TemporaryTable
                                           GROUP BY TemporaryTable.TeamName, TemporaryTable.SeasonYear) AS WalkoverTeamInfo
                                     GROUP BY WalkoverTeamInfo.TeamName);
```



TeamName	HighestNumberOfWalkovers	SeasonYear
Powerhouse Men	2	2017
Footscray Mixed	3	2017

2 rows returned