

Database Systems Assignment2, 2020

Name: Aoi Fujii

Q1. 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.

```
SELECT player.FirstName, player.LastName, club.ClubName AS CurrentClub
FROM player
    INNER JOIN clubplayer ON player.PlayerID = clubplayer.PlayerID
    LEFT JOIN club ON clubplayer.ClubID = club.ClubID AND ToDate IS NULL
WHERE player.PlayerID IN (SELECT DISTINCT PlayerID
                        FROM playerteam
                        WHERE GameID IS NOT NULL)
-- get the most recent data about the club where the player belongs to
-- the row with player who was in a club listed but currently in a different club is removed
-- players with current club name
AND (ToDate IS NULL
    -- not currently in any club but was in the club before
    OR (ToDate IS NOT NULL
        -- players not in the list of players with the current club team
        AND player.PlayerID NOT IN (SELECT PlayerID
                                    FROM clubplayer
                                    LEFT JOIN club ON clubplayer.ClubID = club.ClubID
                                    WHERE ToDate IS NULL)))
ORDER BY player.LastName, player.FirstName ASC;
```

	FirstName	LastName	CurrentClub
►	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.

```
SELECT CONCAT(player.FirstName, ' ', player.LastName) AS FemalePlayer
```

```
FROM player
```

```
WHERE Sex = 'F'
```

```
AND PlayerID IN (SELECT PlayerID
```

```
FROM playerteam
```

```
GROUP BY PlayerID
```

```
HAVING COUNT(DISTINCT (TeamID)) > 1);
```

	FemalePlayer
▶	Sally Davis
	Tiffany Lightshade
	Barbara Standwick
	Kate Hudson
	Fiona Bruce
	Rosalyn McQuarrie
	Donna Reed
	Susanne Hoffs
	Tabitha Stephens
	Betty Davis

59 rows returned

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

```
SELECT CONCAT(FirstName, ' ', LastName) AS PlayerName
FROM player
-- find players not in the list of players who have played mixed competition
WHERE PlayerID NOT IN (SELECT PlayerID
                        FROM playerteam
                        INNER JOIN game ON playerteam.GameID = game.GameID
                        INNER JOIN season ON game.SeasonID = season.SeasonID
                        -- get the CompetitionID of the 'mixed' competition
                        WHERE season.CompetitionID IN (SELECT CompetitionID
                                                    FROM competition
                                                    WHERE CompetitionType = 'Mixed'));
```

	PlayerName
▶	Dave Ganey
	Mia Farrow
	Don Dunstan
	Doug Liman
	Melanie Griffith
	Michelle Pfeiffer
	Damian Elsevier
	Randall Thompson
	Michael Mouse
	Melinda Gates

15 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).

```
SELECT season.SeasonYear, COUNT(*) AS CancelledGames
FROM season
    INNER JOIN game ON season.SeasonID = game.SeasonID
-- choose the cancelled games
WHERE game.T1Score IS NULL AND game.T2Score IS NULL
GROUP BY season.SeasonYear
-- pick the season year with the highest cancelled games
HAVING cancelledGames = (SELECT MAX(cancelled.CancelledGames)
    -- table of season year with the number of cancelled games
    FROM (SELECT SeasonYear, COUNT(*) AS CancelledGames
        FROM season
            INNER JOIN game ON season.SeasonID = game.SeasonID
        WHERE game.T1Score IS NULL AND game.T2Score IS NULL
        GROUP BY season.SeasonYear) AS cancelled);
```

	SeasonYear	CancelledGames
▶	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.

```
SELECT club.ClubName, total.TotalPlayers - female.FemalePlayers AS MalePlayers, female.FemalePlayers,
       ABS(2*female.FemalePlayers - total.TotalPlayers) AS AbsoluteDifference
FROM
  -- count the total number of players for each club
  (SELECT clubplayer.ClubID, COUNT(*) AS TotalPlayers
   FROM player
    INNER JOIN clubplayer ON player.PlayerID = clubplayer.PlayerID
   WHERE ToDate IS NULL
   GROUP BY clubplayer.ClubID) AS total

  NATURAL JOIN

  -- count the number of female players for each club
  (SELECT clubplayer.ClubID, COUNT(*) AS FemalePlayers
   FROM player
    INNER JOIN clubplayer ON player.PlayerID = clubplayer.PlayerID
   WHERE ToDate IS NULL
    AND player.Sex = 'F'
   GROUP BY clubplayer.ClubID) AS female

  NATURAL JOIN club

-- include the club that have same number of male and female players
WHERE ABS(2*female.FemalePlayers - total.TotalPlayers) != 0
ORDER BY AbsoluteDifference DESC;
```

	ClubName	MalePlayers	FemalePlayers	AbsoluteDifference
►	Melbourne City	11	9	2
	Melbourne University Rugby Club	8	9	1
	Moorabbin Rugby	8	9	1
	Animatus Rugby Club	9	8	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.

```
SELECT player.FirstName, player.Sex, Year2017.Games2017 AS Game2017,
       IF(Year2018.Games2018 IS NOT NULL, Year2018.Games2018, 0) AS Game2018
FROM player
      INNER JOIN
      -- count the total number of games played in 2017 for each player
      (SELECT playerteam.PlayerID, COUNT(*) AS Games2017
      FROM playerteam
            INNER JOIN game ON playerteam.GameID = game.GameID
            INNER JOIN season ON game.SeasonID = season.SeasonID
      WHERE SeasonYear = 2017
      GROUP BY playerteam.PlayerID) AS Year2017 ON player.PlayerID = Year2017.PlayerID

      LEFT JOIN
      -- count the total number of games played in 2018 for each player
      (SELECT playerteam.PlayerID, COUNT(*) AS Games2018
      FROM playerteam
            INNER JOIN game ON playerteam.GameID = game.GameID
            INNER JOIN season ON game.SeasonID = season.SeasonID
      WHERE SeasonYear = 2018
      GROUP BY playerteam.PlayerID) AS Year2018 ON Year2017.PlayerID = Year2018.PlayerID

-- find the players who played more games in 2017 than in 2018 including those who didn't play in 2018
WHERE Year2017.Games2017 > Year2018.Games2018
```

```
OR (Year2018.Games2018 IS NULL AND Year2017.Games2017 IS NOT NULL);
```

	FirstName	Sex	Game2017	Game2018
▶	Garret	M	8	5
	Darren	M	1	0
	Bernyce	F	8	7
	Florence	F	10	9
	Ruth	F	9	8
	Diana	F	11	10
	Barbara	F	6	5
	Kate	F	5	4
	Mia	F	5	4
	Fiona	F	9	8

50 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.

```
SELECT team.TeamName, T1S + T2S AS TotalScore
```

```
FROM team
```

```
    INNER JOIN
```

```
-- games of team 1 in Bindham Trophy in 2017
```

```
(SELECT game.Team1, count(*), SUM(game.T1Score) AS T1S
```

```
FROM game
```

```
    INNER JOIN season ON game.SeasonID = season.SeasonID
```

```
    INNER JOIN competition ON season.CompetitionID = competition.CompetitionID
```

```
WHERE competition.CompetitionName = 'Bingham Trophy'
```

```
    AND season.SeasonYear = 2017
```

```
GROUP BY game.Team1) AS T1 ON team.TeamID = T1.Team1
```

```
    INNER JOIN
```

```
-- games of team 2 in Bindham Trophy in 2017
```

```
(SELECT game.Team2, count(*), SUM(game.T2Score) AS T2S
```

```
FROM game
```

```
    INNER JOIN season ON game.SeasonID = season.SeasonID
```

```
    INNER JOIN competition ON season.CompetitionID = competition.CompetitionID
```

```
WHERE competition.CompetitionName = 'Bingham Trophy'
```

```
    AND season.SeasonYear = 2017
```

```
GROUP BY game.Team2) AS T2 ON team.TeamID = T2.Team2
```

```
-- list the name of all teams that scored more than 100 points
```

```
WHERE T1S + T2S > 100
```

```
ORDER BY TotalScore DESC;
```

	TeamName	TotalScore
►	Melbourne City Mixed	129
	Monash Uni Mixed	119
	Melbourne Uni Mixed	104

3 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.

```
SELECT CONCAT(player.FirstName, ' ', player.LastName) AS PlayerName,  
        -- find the number of days the player has spent as the club member as of April 30th 2020  
        -- To date is [null] or [not null and before 2020-04-30 or after]  
        SUM(IF(clubplayer.ToDate IS NOT NULL  
                AND DATEDIFF('2020-04-30', clubplayer.ToDate) < 0,  
                DATEDIFF(clubplayer.ToDate, clubplayer.FromDate),  
                DATEDIFF('2020-04-30', clubplayer.FromDate))) AS MembershipDays  
FROM player  
        INNER JOIN clubplayer ON player.PlayerID = clubplayer.PlayerID  
        INNER JOIN club ON clubplayer.ClubID = club.ClubID  
WHERE club.ClubName = 'Melbourne City'  
-- sum up the membership days if the player joined the same club for more than twice  
GROUP BY player.PlayerID, club.ClubID  
ORDER BY MembershipDays ASC  
limit 1;
```

	PlayerName	MembershipDays
▶	Karin Carpenter	1027

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.

```
SELECT player.FirstName, player.LastName, COUNT(*) AS TotalForeignGames
```

```
FROM clubplayer
```

```
    INNER JOIN player ON clubplayer.PlayerID = player.PlayerID
```

```
    INNER JOIN playerteam ON player.PlayerID = playerteam.PlayerID
```

```
    INNER JOIN team ON playerteam.TeamID = team.TeamID
```

```
    INNER JOIN game ON playerteam.GameID = game.GameID
```

```
-- make sure the club of the player is the one in the match date
```

```
-- i.e. the match date is inbetween the date the player joined and left/currently in the club
```

```
WHERE (DATEDIFF(clubplayer.ToDate, game.MatchDate) > 0 OR clubplayer.ToDate IS NULL)
```

```
    AND DATEDIFF(game.MatchDate, clubplayer.FromDate) > 0
```

```
-- list the foreign game
```

```
    AND clubplayer.ClubID != team.ClubID
```

```
GROUP BY clubplayer.PlayerID
```

```
ORDER BY TotalForeignGames DESC
```

```
LIMIT 20;
```

	FirstName	LastName	TotalForeignGames
▶	Diana	Rigg	21
	Donna	Reed	19
	Florence	Henderson	19
	Broderick	Davis	18
	Bruce	Banner	18
	Bevan	Jackson	18
	Alex	Skaarsgard	18
	Brian	Blessed	18
	Brock	Briers	18
	Fiona	Bruce	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.

```

SELECT records.TeamName, records.Forfeits, records.SeasonYear
FROM
    -- find the highest number of forfeits for each team which have given at least one forfeit
    (SELECT t.TeamID, MAX(t.Forfeits) AS HighestForfeits
    FROM
        -- count the number of forfeits for each season
        (SELECT team.TeamID, game.SeasonID, COUNT(*) AS Forfeits
        FROM season
            INNER JOIN game ON season.SeasonID = game.SeasonID
            INNER JOIN team ON team.TeamID = (CASE
                -- Find the team that gave away a walkover
                WHEN T1Score = 0 THEN Team1
                ELSE Team2
                END)
        -- assuming that any game with the score 28 and 0 implies the game was a walkover
        WHERE (T1Score = 0 AND T2Score = 28) OR (T1Score = 28 AND T2Score = 0)
        GROUP BY game.SeasonID, team.TeamID
        ) t
    GROUP BY t.TeamID
    ) AS maximum

RIGHT JOIN

    -- number of forfeits of teams that give away at least one forfeit in each season
    (SELECT team.TeamID, team.TeamName, game.SeasonID, COUNT(*) AS Forfeits,
        season.SeasonYear
    FROM season
        INNER JOIN game ON season.SeasonID = game.SeasonID
        INNER JOIN team ON team.TeamID = (CASE
            WHEN T1Score = 0 THEN Team1
            ELSE Team2
            END)
    WHERE (T1Score = 0 AND T2Score = 28) OR (T1Score = 28 AND T2Score = 0)
    GROUP BY game.SeasonID, team.TeamID
    ) AS records

ON maximum.TeamID = records.TeamID

-- list the records with the highest number of forfeits of the team
-- who have given away 2 or more walkovers
WHERE maximum.HighestForfeits = records.Forfeits
    AND maximum.HighestForfeits >= 2;

```

	TeamName	Forfeits	SeasonYear
▶	Powerhouse Men	2	2017
	Powerhouse Men	2	2018
	Footscray Mixed	3	2017

3 rows returned