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

FROM player LEFT JOIN clubplayer

ON player.playerID = clubplayer.PlayerID

NATURAL JOIN club

WHERE player.PlayerID in (

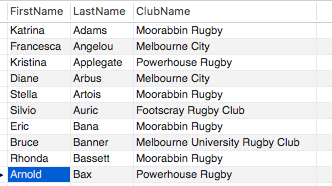
SELECT player.PlayerID

FROM player NATURAL JOIN playerteam

GROUP BY player.PlayerID

) AND clubplayer.ToDate IS NULL

ORDER BY player.LastName ASC, player.FirstName ASC;



142 rows

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 FullName

FROM player

WHERE playerID in

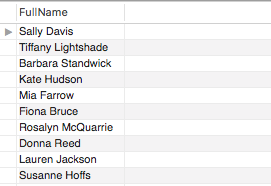
(SELECT PlayerID

FROM playerteam

GROUP BY PlayerID

HAVING COUNT(TeamID) > 1) AND

Sex = 'F';



71 row(s) returned

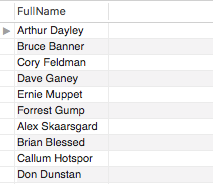
Q3. List the full name (presented as a single string) of players who have never played in a 'mixed’ competition.

SELECT CONCAT(player.FirstName, ' ', player.LastNAME) AS FullName

FROM player NATURAL JOIN playerteam NATURAL JOIN game NATURAL JOIN season NATURAL JOIN competition

WHERE NOT(competition.CompetitionType = 'Mixed')

GROUP BY player.PlayerID



130 row(s) 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(season.CompetitionID) As CancelGameNum

FROM season

WHERE SeasonID in (

SELECT SeasonID

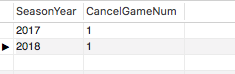
FROM game

WHERE T1Score IS NULL and T2Score IS NULL

)

GROUP BY season.SeasonYear

ORDER BY COUNT(season.CompetitionID) Desc;



2 row(s) 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, MaleTable.MaleNum, FemaleTable.FemaleNum, ABS(MaleTable.MaleNum - FemaleTable.FemaleNum) As GenderGap

FROM club NATURAL JOIN

(SELECT club.ClubID, Count(player.Sex) As MaleNum

FROM player NATURAL JOIN clubplayer NATURAL JOIN club

WHERE clubplayer.ToDate IS NULL AND player.Sex = 'M'

GROUP BY club.ClubID) As MaleTable

NATURAL JOIN

(SELECT club.ClubID, Count(player.Sex) As FemaleNum

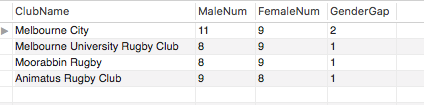
FROM player NATURAL JOIN clubplayer NATURAL JOIN club

WHERE clubplayer.ToDate IS NULL AND player.Sex = 'F'

GROUP BY club.ClubID) As FemaleTable

WHERE MaleTable.MaleNum != FemaleTable.FemaleNum

ORDER BY GenderGap DESC



4 row(s) 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, GameTable2017.GamePlayed2017, GameTable2018.GamePlayed2018

FROM player NATURAL JOIN playerteam NATURAL JOIN

(SELECT playerteam.PlayerID, COUNT(playerteam.GAMEID) AS GamePlayed2017

FROM playerteam NATURAL JOIN game NATURAL JOIN season

WHERE season.SeasonYear = '2017'

GROUP BY playerteam.PlayerID) AS GameTable2017 NATURAL JOIN

(SELECT playerteam.PlayerID, COUNT(playerteam.GAMEID) AS GamePlayed2018

FROM playerteam NATURAL JOIN game NATURAL JOIN season

WHERE season.SeasonYear = '2018'

GROUP BY playerteam.PlayerID) AS GameTable2018

WHERE GameTable2017.GamePlayed2017 > GameTable2018.GamePlayed2018

GROUP BY playerteam.PlayerID



49 row(s) 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, ScoreTable.TotalScore

FROM team NATURAL JOIN (

SELECT team.TeamName,

CASE WHEN Team1Table.T1Total IS NULL AND Team2Table.T2Total > 100 THEN

Team2Table.T2Total

WHEN Team1Table.T1Total > 100 AND Team2Table.T2Total IS NULL THEN

Team1Table.T1Total

WHEN (Team1Table.T1Total + Team2Table.T2Total) > 100 THEN

Team1Table.T1Total + Team2Table.T2Total

END AS TotalScore

FROM team LEFT JOIN (

SELECT Team1, SUM(T1Score) AS T1Total

FROM game

WHERE game.SeasonID IN

(SELECT SeasonID

FROM competition NATURAL JOIN season

Where CompetitionName = 'Bingham Trophy' And season.SeasonYear =

'2017')

GROUP BY Team1) AS

Team1Table ON

team.TeamID = Team1Table.Team1 LEFT JOIN(

SELECT Team2, SUM(T2Score) AS T2Total

FROM game

WHERE game.SeasonID IN

(SELECT SeasonID

FROM competition NATURAL JOIN season

Where CompetitionName = 'Bingham Trophy' And season.SeasonYear = '2017' )

GROUP BY Team2) AS

Team2Table ON

team.TeamID = Team2Table.Team2

WHERE NOT(Team1Table.T1Total IS NULL AND Team2Table.T2Total IS NULL)) AS

ScoreTable

WHERE ScoreTable.TotalScore IS NOT NULL

ORDER BY TotalScore DESC;



5 row(s) 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 FullName,

CASE WHEN clubplayer.ToDate IS NULL THEN

DATEDIFF('2020-04-30', clubplayer.FromDate)

ELSE

DATEDIFF(clubplayer.ToDate, clubplayer.FromDate)

END As ServingDay

FROM player NATURAL JOIN clubplayer NATURAL JOIN club

WHERE club.ClubName = 'Melbourne City'

ORDER BY ServingDay ASC

LIMIT 1



1 row(s) 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 FirstName, LastName, COUNT(PlayerID) As ForeignGameNum

FROM player NATURAL JOIN

(SELECT PlayerID, ClubID As Belonged\_To, Played.Played\_For

FROM clubplayer NATURAL JOIN

(SELECT PlayerID, TeamID AS Played\_For

FROM playerteam

GROUP BY PlayerID, TeamID

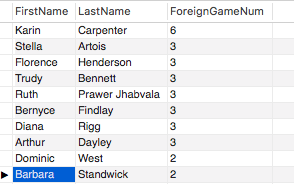
ORDER BY playerID) AS Played

WHERE ClubID <> Played.Played\_For

ORDER BY PlayerID) AS ForeignTable

GROUP BY PlayerID

ORDER BY COUNT(PlayerID) DESC



140 row(s) 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 team.TeamName, WalkoverTable.WalkoverNum\_Team1 As WalkoverNum, WalkoverTable.SeasonYear

FROM team RIGHT JOIN

(SELECT game.Team1, COUNT(game.Team1) As WalkoverNum\_Team1, season.SeasonYear

FROM game NATURAL JOIN season

WHERE (game.T1Score = 0 AND game.T2Score = 28)

GROUP BY game.Team1, season.SeasonYear

UNION

SELECT game.Team2, COUNT(game.Team2) As WalkoverNum\_Team2, season.SeasonYear

FROM game NATURAL JOIN season

WHERE (game.T1Score = 28 AND game.T2Score = 0)

GROUP BY game.Team2, season.SeasonYear) As

WalkoverTable ON

team.TeamID = WalkoverTable.Team1

WHERE WalkoverTable.WalkoverNum\_Team1 >= 2;



3 row(s) returned