Tennis Game Analysis

The data represents table tennis games between different players.

You are free to use any SQL query language, be it standard SQL, T-SQL (MSSQL), MySQL, ORACLE, …

We strongly advise to work in an environment such as SQL Management studio (MSSQL) or equivalent to work on your answers.

Score

The score table contains the game, the player and score

Score contains the following columns:

[ID] int, [Game] int, [Player] varchar (1), [Score] int.

- ID is the primary key
- Player is a foreign key derived from Player Table primary key ID
- Game is a foreign key derived from Game Table primary key ID

ID	Game	Player	Score
1	1	A	11
2	1	B	7
3	2	A	15
4	2	C	13
5	3	B	11
6	3	D	9
7	4	D	11
8	4	A	5
9	5	A	11
10	6	B	11
11	6	C	2
12	6	D	5

Game

Game contains the following columns:

The Game table contains the player who is a winner in each game and date of the game

([ID] int, [Winner] varchar (1), [Date] datetime)

- ID is the primary key
- Winner is a foreign key derived from Player primary key ID
-

ID	Winner	Date
1	A	2017-01-02
2	A	2016-05-06
3	B	2017-12-15
4	D	2016-05-06

Player

The Player table contains the following columns:

([ID] varchar (1), [Name] varchar (4), [LastName] varchar (9))

- ID is the primary key

ID	Name	LastName
A	Phil	Watertank
B	Eva	Smith
C	John	Wick
D	Bill	Bull
E	Lisa	Owen

Question 1:

Use the information above to create a database that contains Score, Winner and Player table with relationships

Question 2: Show the average score of each player, even if they didn't play any games.

Expected output (Player ID, Name, Average Score)

Question 3 a: The score table is corrupted: a game can only have two players (not more, not less). Write a query that identifies and only shows the valid games and their winner.

Expected output (Game, Winner)

Question 3 b: As an additional challenge, you can also display the winner's score. The condition described above should still apply.

Expected output (Game, Winner, Winner Score)

Question 4: Show the score of player "Phil Watertank" for games that he lost.

Expected output (Game ID, Player Name, Player LastName, Score)

Question 5: The two following queries return the same result. Why and what is the difference?

Query 1:

```
Select  *  from Player
left join Score on Score.Player = Player.ID
where Score.Player is not null
```

Query 2:

```
Select  *  from Player
right join Score on Score.Player = Player.ID and Score.Player is not null
where Score.Player is not null
```

Question 6:

The two following queries return the players which have not played any games. In your opinion, which one is the best and why? Discuss.

Query 1:

```
Select Distinct Player.ID, Player.Name, Player.LastName  from Player
left join Score on Score.Player = Player.ID
where Score.Player is null
```

Query 2:

```
Select Player.ID, Player.Name, Player.LastName  from Player
where Player.ID not in (select distinct Score.Player from Score)
```

Question 7: Show the list of player combinations who have never played together.

Expected Output (Player1, Player2). Reverse duplicates are authorized (A-E and E-A for example).