

Name: Dada Joshua

Title: Data Analytics SQL Review

Table of Figures

Figure 1 Player Table.....	4
Figure 2 Game Table	5
Figure 3 Score Table.....	5
Figure 4 Solution for question 2	6
Figure 5 Solution 3a	6
Figure 6 Solution 3b	7
Figure 7 Solution 4	8
Figure 8 Solution 7	9

Solution: Tennis Game Analysis

Question 1:

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

solution

```
CREATE TABLE PLAYER(ID VARCHAR(1) PRIMARY KEY,  
  
NAME VARCHAR(4) NOT NULL,  
  
LASTNAME VARCHAR(9) NOT  
NULL);
```

```
CREATE TABLE GAME(ID INT PRIMARY KEY,  
  
WINNER VARCHAR(1) NOT NULL, Date  
DATETIME NOT NULL,  
  
CONSTRAINT FK_GAME_PLAYER_ID  
FOREIGN KEY (WINNER) REFERENCES PLAYER  
(ID));
```

```
CREATE TABLE SCORE(ID INT PRIMARY KEY,  
  
GAME INT NOT NULL,  
  
PLAYER VARCHAR(1) NOT NULL,  
  
SCORE int NOT NULL,  
  
CONSTRAINT FK_SCORE_PLAYER_ID  
FOREIGN KEY (PLAYER) REFERENCES  
PLAYER (ID),  
  
CONSTRAINT FK_SCORE_GAME_ID  
FOREIGN KEY (GAME) REFERENCES GAME  
(ID))  
INSERT INTO PLAYER  
VALUES ('A','Phil','Watertank'),  
  
('B','Eva','Smith'),  
('C','John','Wick'),  
('D','Bill','Bull'),
```

('E','Lisa','Owen');

```
INSERT INTO GAME  
VALUES (1,'A','2017-01-02'),
```

```
(2,'A','2016-05-06'),  
(3,'B','2017-12-15'),  
(4,'D','2016-05-06');
```

---Disabling Foreign key

```
ALTER TABLE SCORE NOCHECK CONSTRAINT ALL;
```

```
INSERT INTO SCORE  
VALUES (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);
```

--- Enabling Foreign Key

```
ALTER TABLE SCORE CHECK CONSTRAINT ALL;
```

Tables

Services C
Agent (Ag
er

Results		Messages	
	ID	Name	LastName
1	A	Phil	Watertank
2	B	Eva	Smith
3	C	John	Wick
4	D	Bill	Bull
5	E	Lisa	Owen

Figure 1 Player Table

Results Messages			
	ID	Winner	Date
1	1	A	2017-01-02 00:00:00.000
2	2	A	2016-05-06 00:00:00.000
3	3	B	2017-12-15 00:00:00.000
4	4	D	2016-05-06 00:00:00.000

Figure 2 Game Table

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

Figure 3 Score Table

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

Expected output (Player ID, Name, Average Score)

Solution

```
SELECT PLAYER,
       NAME,
       AVG(SCORE) "Average Score"
FROM PLAYER P
LEFT JOIN SCORE S ON P.ID = S.PLAYER
GROUP BY PLAYER,
       NAME;
```

Results		Messages	
	Player	Name	Average Score
1	D	Bill	8
2	B	Eva	9
3	C	John	7
4	NULL	Lisa	NULL
5	A	Phil	10

Figure 4 Solution for question 2

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)

Solution

```
WITH NEW_SCORE AS
  (SELECT GAME,
    COUNT(PPLAYER) AS COUNT
    FROM SCORE
    GROUP BY GAME
    HAVING COUNT(PPLAYER) = 2)
SELECT GAME,
  WINNER
FROM NEW_SCORE N
JOIN GAME G ON G.ID = N.GAME ;
```

Results		Messages	
	game	winner	
1	1	A	
2	2	A	
3	3	B	
4	4	D	

Figure 5 Solution 3a

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

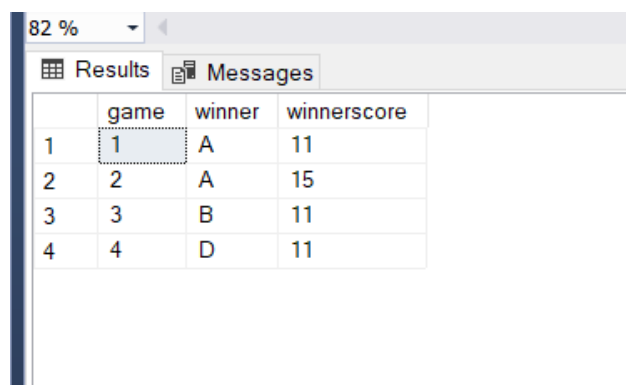
Expected output (Game, Winner, Winner Score)

solution

```

WITH NEW_SCORE AS
  (SELECT GAME,
    MAX(SCORE) AS WINNERSCORE,
    COUNT(PLOYER) AS COUNT
    FROM SCORE
    GROUP BY GAME
    HAVING COUNT(PLOYER) = 2)
SELECT GAME,
  WINNER,
  WINNERSCORE
FROM NEW_SCORE N
JOIN GAME G ON G.ID = N.GAME ;

```



	game	winner	winnerscore
1	1	A	11
2	2	A	15
3	3	B	11
4	4	D	11

Figure 6 Solution 3b

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

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

Solution

```

SELECT GAME " Game ID",
  NAME "PLayer Name",
  LASTNAME "Player LastName",
  SCORE
FROM SCORE S
JOIN PLAYER P ON P.ID = S.PLAYER
WHERE NAME = 'Phil'
  AND LASTNAME = 'Watertank'
  AND SCORE =
    (SELECT MIN(SCORE)
     FROM SCORE S
     JOIN PLAYER P ON P.ID = S.PLAYER
     WHERE NAME = 'Phil'
       AND LASTNAME = 'Watertank')

```

Results		Messages		
	Game ID	Player Name	Player LastName	Score
1	4	Phil	Watertank	5

Figure 7 Solution 4

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.

Solution

Answer: Query 1

A LEFT [OUTER] JOIN is faster than an equivalent subquery because the server is able to optimize it better.

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

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).

Solution

```
SELECT P1.ID AS PLAYER1,  
       P2.ID AS PLAYER2  
FROM PLAYER P1,  
     PLAYER P2  
WHERE P1.ID != P2.ID  
EXCEPT  
SELECT P1.PLAYER AS PLAYER1,  
       P2.PLAYER AS PLAYER2  
FROM SCORE P1,  
     SCORE P2
```

	PLAYER1	PLAYER2
1	E	A
2	E	B
3	E	C
4	E	D
5	A	E
6	B	E
7	C	E
8	D	E

Figure 8 Solution 7