SQL PRT QUESTIONS

Q1)For each team, calculate the total prize money won and list the teams that have won more than $500,000 in prize money.

```
SELECT Team,
    SUM(CAST(REPLACE([Prize], ',', '') AS INT)) AS Total_Prize
FROM [dbo].[valorant_la]
GROUP BY Team
HAVING SUM(CAST(REPLACE([Prize], ',', '') AS INT)) > 500000;
```

Q2)Find the player with the highest kill contribution per round (total kills divided by rounds played). Display the player's name, kill contribution, and team.

```
SELECT TOP 1 Player,
    CAST([Kill] AS FLOAT) / [Rounds_Played] AS Kill_Per_Round,
    Team
FROM [dbo].[valorant_la]
ORDER BY Kill_Per_Round DESC;
```

Q3)Identify the nationality that has the most players in the top 10 based on KAST (%) and display the nationality and the count of players.

```
WITH Top10_KAST AS (
   SELECT TOP 10 Nationality
   FROM [dbo].[valorant_la]
   ORDER BY [KAST] DESC
)
SELECT Nationality, COUNT(Nationality) AS Player_Count
FROM Top10_KAST
GROUP BY Nationality
ORDER BY Player_Count DESC;
```

Q4)Create a table-valued function named fn_TopPlayersByKAST that returns the top N players based on their KAST (%). The function should take an integer input to specify the number of players to return.

```
CREATE FUNCTION fn_TopPlayersByKAST (@TopN INT)
RETURNS TABLE
AS
RETURN
(
   SELECT TOP (@TopN) Player, [KAST]
```

```
    FROM [dbo].[valorant_la]
    ORDER BY [KAST] DESC
);

-- Example usage:
SELECT * FROM dbo.fn_TopPlayersByKAST(5);
```

Q5)Write a stored procedure named sp_UpdatePlayerRole that takes a player's name and a new role as input, and updates the player's role in the valorant_la table.

```
CREATE PROCEDURE sp_UpdatePlayerRole
    @PlayerName NVARCHAR(100),
    @NewRole NVARCHAR(50)
AS
BEGIN
    UPDATE [dbo].[valorant_la]
    SET Role = @NewRole
    WHERE Player = @PlayerName;
END;

-- Example usage:
EXEC sp_UpdatePlayerRole 'Demon1', 'Support';
```

Q6)Write a stored procedure named sp_GetTeamStats that takes a team name as input and returns the total kills, total deaths, and average K/D ratio for the team.

```
CREATE PROCEDURE sp_GetTeamStats
    @TeamName NVARCHAR(100)
AS
BEGIN
    SELECT Team, SUM([Kill]) AS Total_Kills, SUM([Death]) AS Total_Deaths,
        AVG(CAST([Kill] AS FLOAT) / [Death]) AS Avg_KD_Ratio
    FROM [dbo].[valorant_la]
    WHERE Team = @TeamName
    GROUP BY Team;
END;

-- Example usage:
EXEC sp_GetTeamStats 'Evil Geniuses';
```