CS:GO Tournament Database Documentation

This is a database designed to store results from a Counter-Strike Global Offensive (CS:GO) Tournament.

All operations for this database are completed in Microsoft SQL Server Management Studio. To initialize this database, open the csgo.sql file. Then right click on ‘Databases’ from the left most windows and select New Database. Name the database whatever you would like. Then select the newly created database from the drop down window next to the execute button. After the proper database has been selected, click on Execute. Once the database has been created and the file has been executed the database is ready for use. Note the New Query button next to the Execute button. This will be used for all operations.

Adding a New Team to the Database

To add a new team to the tournament database, begin by clicking new query and confirming the proper database is selected. Due to the way the database is designed, a team name must be entered before players and team members can be populated. Enter the following into the query to add a new team:

```
INSERT INTO Team (team_name) VALUES ('team_name');
```

Where team_name is the name of the team to be entered. After filling out the query, hit execute and confirm it was run without error. If there is an error there will be red text in the console, and it will inform you what needs to be fixed.

Adding New Player to the Database

To add a new team to the tournament database, begin by clicking new query and confirming the proper database is selected. If you know the team that this player will be a member of, then enter the following into the query to add a new player:

```
INSERT INTO Player VALUES ('Username', 'First Name', 'Last Name', 'Team Name');
```

If you do not know the team that this player will be a member of, then enter the following into the query to add a new team:

```
INSERT INTO Player (in_game_name, f_name, l_name) VALUES ('Username','First Name','Last Name');
```

After populating the query with either command, execute it to enter the player.

Adding Players to a Team

To populate a team's roster, confirm the necessary players have been added to the database beforehand. Then fill out the query with the command below:

```
UPDATE Team
SET p1_ign ='UN1', p2_ign =' UN2', p3_ign =' UN3', p4_ign =' UN4', p5_ign =' UN5'
WHERE team_name = 'Team Name';
```

Be sure to fill out UN1 through UN5 with the appropriate usernames of the team members, as well as Team Name with the appropriate team you wish to populate. Once filled out, execute the command to populate the team.

Adding Match Results

Once a match has been completed, the Match results must be entered. Use to following command in a query to enter match results:

```
INSERT INTO Matches VALUES (Match_ID, 'Map Name', 'Team 1 Name', Team 1 1st Half Score, Team 1 2nd Half Score, Team 1 Final Score, 'Team 2 Name', Team 2 1st Half Score, Team 2 2nd Half Score, Team 2 Final Score, Best of Three Value);
```

Make sure to fill out the values that you wish to enter. Note that Match_ID is an incrementing integer value (i.e. for the first match, use 1, then for the next match use 2 e.t.c). Also note that team 1 is the team that starts as Counter-Terrorists and Best of Three Value is a value to determine how the match was played (i.e. 0 is best of 1 Match, 1 is the first match in a Best of Three Match, 2 is the second match in a Best of Three Match e.t.c). Execute the command once the data has been inserted into the command.

Adding Player Statistics

After a match has been completed, the players individual statistics must be entered in addition to team statistics. This must be completed for every player in the game. The following command will enter individual player statistics:

```
INSERT INTO Player_Statistics (match, in_game_name, team_name, Kills, Deaths, Assists, ADR, HS_Percentage)
VALUES (Match_ID, 'Username', ‘Team Name’, Kills, Deaths, Assists, ADR, Headshot Percentage);
```

Be sure to fill in Match_ID through Headshot Percentage. This table also stores Kills to Deaths Ratio but that is calculated by the data entered. Using the above command will populate the Kills to Deaths Ratio slot as well.

Updating a Team’s Roster

If the need to update a team’s roster arises. The same command from populating a team can be used. Simply update it to include the new roster and run the command. It will update the current listing with the most recent players.

Updating a Player’s Team

If a player is traded to a new team, the data needs to be updated. The following command will change the players team to the new one:

```
UPDATE Player
SET team_name = ‘New Team Name’
WHERE in_game_name =’Username’;
```

Alter the command so it includes you’re the new team name and the target players username and execute the command.

Deleting Data

The only case in which data should be deleted is in the case of a cheater. In the case of a cheater, their records must be expunged from the database. This requires two parts. First, update the team

record with the new roster. This new roster should include the old roster except for the cheater, and a new player taking the cheaters spot. To complete this execute the same command as you would to add players to a team, except with the new roster's usernames. Then to expunge the cheaters records run the following command:

```
EXEC ban 'Username';
```

Make sure to replace username with the username of the cheater. This will call a process to delete the player and all their statistics without having to worry about foreign key constraints.

Find Best of Three Value for a Given Match

To find out how many games it took to decide a match you can use the best_of_three stored procedure. Run the following command to find the answer for a given match

```
EXEC best_of_three 'team1name', 'team2name'
```

Be sure to replace team1name and team2name with the respective team names. It does not matter what team name is used first.

Find the MVP of a Given Game

To find the MVP of a given game you can use the MVP stored procedure. Run the following command to determine the MVP based on kills and assists.

```
EXEC MVP 'matchID'
```

Make sure to replace matchID with the match_ID from the Matches table for the target match.

Find all Overtime Matches

To find all matches that went to overtime, you can view the Overtime View. This can be viewed with the following command:

```
SELECT *
FROM Overtime;
```

This will display all the overtime matches, the teams that played in them, the score, and how many rounds of overtime was played.

To View Overall Performance

To view overall performance, you can call the Overall_Performance view. This view shows total kills and deaths throughout the tournament for a given player. Additionally, it will give the overall Kills to Deaths Ratio for the tournament. To view overall performance and sort by Kills, execute the following query:

```
SELECT *
FROM Overall_performance
ORDER BY Total_Kills DESC;
```

To view overall performance and sort by Deaths, execute the following query:

```
SELECT *
FROM Overall_performance
ORDER BY Total_Deaths DESC;
```

To view overall performance and sort by K/D Ratio, execute the following query:

```
SELECT *
FROM Overall_performance
ORDER BY KD DESC;
```

Removing the word DESC from the command, will sort in ascending order instead of descending.

To View Players in each Match

To see who played in each match you can call the Roster View. This can be viewed with the following command:

```
SELECT *
FROM Roster;
```

This view highlights who played in each game and allows you to see any roster substitutions as it is ordered in chronological order.

Built-In Functions

This database includes a function to calculate total kills or total deaths for a given player. It returns just an int value to represent the requested total. To return the value use the following line and set it equal to whatever you need it to be.

```
SELECT dbo.total('Username', 'X')
```

Fill in username with the player you wish to gather the totals from. X, is a character flag. If you want to receive the total kills for a player, replace 'X' with 'k'. For deaths replace 'X' with 'd'.