## Creating continent nodes manually:

```
CREATE (node:continent {continentCode: 'AS', continentName: 'Asia'})
CREATE (node:continent {continentCode: 'EU', continentName: 'Europe'})
CREATE (node:continent {continentCode: 'AN', continentName: 'Antarctica'})
CREATE (node:continent {continentCode: 'AF', continentName: 'Africa'})
CREATE (node:continent {continentCode: 'OC', continentName: 'Oceania'})
CREATE (node:continent {continentCode: 'NA', continentName: 'North America'})
CREATE (node:continent {continentCode: 'SA', continentName: 'South America'})
```

## Creating country nodes using .CSV file

```
LOAD CSV WITH HEADERS FROM 'file:///country-and-continent-codes-list.csv' AS row
MERGE (c:country {countryName: row.Country_Name})
ON CREATE SET c.twoLetterCode = row.Two_Letter_Country_Code, c.threeLetterCode =
row.Three_Letter_Country_Code, c.countryNumber = row.Country_Number
```

## Establishing relationship between country---PART_OF--->continent

```
LOAD CSV WITH HEADERS FROM 'file:///country-and-continent-codes-list.csv' AS row
MATCH (con:continent {continentName: row.Continent_Name})
MATCH (cou:country {countryName: row.Country_Name})
CREATE (cou)-[:PART_OF]->(con)
```

## Creating player nodes using .CSV file

```
LOAD CSV WITH HEADERS FROM 'file:///highest_earning_players.csv' AS row

MERGE (p:players {playerID: row.PlayerId})

ON CREATE SET p.firstName = row.NameFirst, p.lastName = row.NameLast, p.playerHandle =
row.CurrentHandle, p.totalPrizeWon = row.TotalUSDPrize
```

## Establishing relationship between players---FROM--->country

```
LOAD CSV WITH HEADERS FROM 'file:///highest_earning_players.csv' AS row

MATCH (c:country {twoLetterCode: TOUPPER(row.CountryCode)})

MATCH (p:players {playerID: row.PlayerId})

MERGE (p)-[:FROM]->(c)
```

## Creating games nodes manually

```
MERGE (g:games {gameName: 'Counter-Strike: Global Offensive'})

MERGE (g:games {gameName: 'Dota 2'})

MERGE (g:games {gameName: 'League of Legends'})

MERGE (g:games {gameName: 'Fortnite'})

MERGE (g:games {gameName: 'Overwatch'})

MERGE (g:games {gameName: 'Starcraft II'})

MERGE (g:games {gameName: 'Heroes of the Storm'})

MERGE (g:games {gameName: 'PUBG'})

MERGE (g:games {gameName: 'Arena of Valor'})

MERGE (g:games {gameName: 'Hearthstone'})
```

## Establishing relationship between players---PLAYS--->games

```
LOAD CSV WITH HEADERS FROM 'file:///highest_earning_players.csv' AS row
MATCH (p:players {playerID: row.PlayerId})
MATCH (g:games {gameName: row.Game})
CREATE (p)-[:PLAYS]->(g)
```

## Creating genre nodes manually

```
MERGE (g:genre {genreName: 'First-Person Shooter'})
MERGE (g:genre {genreName: 'Multiplayer Online Battle Arena'})
MERGE (g:genre {genreName: 'Battle Royale'})
MERGE (g:genre {genreName: 'Strategy'})
MERGE (g:genre {genreName: 'Collectible Card Game'})
```

## Establishing relationship between players---PLAYS_IN--->genre

```
LOAD CSV WITH HEADERS FROM 'file:///highest_earning_players.csv' AS row
MATCH (p:players {playerID: row.PlayerId})
MATCH (gen:genre {genreName: row.Genre})
CREATE (p)-[:PLAYS_IN]->(gen)
```

## Creating teams nodes using .CSV file

```
LOAD CSV WITH HEADERS FROM 'file:///highest_earning_teams.csv' AS row
MERGE (t:teams {teamID: row.TeamId})
ON CREATE SET t.teamName = row.TeamName, t.totalPrizeWon = row.TotalUSDPrize,
t.tournamentsPlayed = row.TotalTournaments
```

## Establishing relationship between teams---PLAYS--->games

```
LOAD CSV WITH HEADERS FROM 'file:///highest_earning_teams.csv' AS row
MATCH (t:teams {teamID: row.TeamId})
MATCH (g:games {gameName: row.Game})
CREATE (t)-[:PLAYS]->(g)
```

## Establishing relationship between teams---PLAYS_IN--->genre

```
LOAD CSV WITH HEADERS FROM 'file:///highest_earning_teams.csv' AS row
MATCH (t:teams {teamID: row.TeamId})
MATCH (gen:genre {genreName: row.Genre})
CREATE (t)-[:PLAYS_IN]->(gen)
```

## Establishing relationship between games---IS--->genre

```
LOAD CSV WITH HEADERS FROM 'file:///highest_earning_players.csv' AS row
MATCH (g:games {gameName: row.Game})
MATCH (gen:genre {genreName: row.Genre})
MERGE (g)-[:IS]->(gen)
```

# Aggregation functions

## Calculating avg_Income according to the country and store it into avgIncome property of the country node using AVG() function

```
MATCH (p:players)-[:FROM]->(c:country)
WITH c, AVG(toInteger(p.totalPrizeWon)) AS avg_income
SET c.avgIncome = avg_income
```

## Calculating avgContinentIncome according to the continent and store it into avgIncome property of the continent node using AVG() function

```
MATCH (c:country)-[:PART_OF]->(ct:continent)
WHERE c.avgIncome IS NOT NULL
WITH ct, AVG(c.avgIncome) AS avgContinentIncome
SET ct.avgIncome = avgContinentIncome;
```

## Listing continents according to their avgIncome in ascending order using MIN() function

```
MATCH(ct:continent)
WHERE ct.avgIncome IS NOT NULL
RETURN ct.continentName, MIN(toInteger(ct.avgIncome)) as lowestIncome
ORDER BY lowestIncome ASC
```

## Listing countries according to their avgIncome in descending order using MAX() function

```
MATCH(c:country)
WHERE c.avgIncome IS NOT NULL
RETURN c.countryName, MAX(toInteger(c.avgIncome)) as highestIncome
ORDER BY highestIncome DESC
```

## Listing continents according to their total income in ascending order using SUM() function

```
MATCH (p:players)-[:FROM]->(c:country)-[:PART_OF]->(ct:continent)
WITH ct.continentName AS ctName, SUM(toInteger(p.totalPrizeWon)) AS totalIncome
RETURN ctName, totalIncome
ORDER BY totalIncome ASC
```

## Listing continents according to the number of players who have any source of income in ascending order using COUNT() function

```
MATCH (p:players)-[:FROM]->(c:country)-[:PART_OF]->(ct:continent)
WHERE p.totalPrizeWon IS NOT NULL
RETURN ct.continentName, COUNT(DISTINCT p) AS playerCount
ORDER BY playerCount ASC
```

## Creating tournaments node manually and establishing relationships manually

```
MERGE (t1:tournaments {tournamentName: 'Overwatch_2023'})
ON CREATE SET t1.startDate = date('2023-05-01'), t1.endDate = date('2023-05-07')
WITH t1
MATCH (g:games {gameName: 'Overwatch'})
MERGE (g)-[:HAS]->(t1)
WITH t1
MATCH (c:country {countryName: 'United States of America'})
MERGE (t1)-[:HELD]->(c)

MERGE (t2:tournaments {tournamentName: 'Starcraft II_2023'})
ON CREATE SET t2.startDate = date('2023-06-15'), t2.endDate = date('2023-06-20')
WITH t1
MATCH (g:games {gameName: 'Starcraft II'})
MERGE (g)-[:HAS]->(t1)
WITH t1
MATCH (c:country {countryName: 'Japan'})
MERGE (t1)-[:HELD]->(c)

MERGE (t3:tournaments {tournamentName: 'League of Legends_2023'})
ON CREATE SET t3.startDate = date('2023-07-10'), t3.endDate = date('2023-07-15')
WITH t3
MATCH (g:games {gameName: 'League of Legends'})
MERGE (g)-[:HAS]->(t3)
WITH t3
MATCH (c:country {countryName: 'Thailand, Kingdom of'})
MERGE (t3)-[:HELD]->(c)
```

```
MERGE (t4:tournaments {tournamentName: 'Fortnite_2023'})
ON CREATE SET t4.startDate = date('2023-08-05'), t4.endDate = date('2023-08-12')
WITH t4
MATCH (g:games {gameName: 'Fortnite'})
MERGE (g)-[:HAS]->(t4)
WITH t4
MATCH (c:country {countryName: 'Korea, Republic of'})
MERGE (t4)-[:HELD]->(c)

MERGE (t5:tournaments {tournamentName: 'Counter-Strike_2023: Global Offensive'})
ON CREATE SET t5.startDate = date('2023-09-01'), t5.endDate = date('2023-09-07')
WITH t5
MATCH (g:games {gameName: 'Counter-Strike: Global Offensive'})
MERGE (g)-[:HAS]->(t5)
WITH t5
MATCH (c:country {countryName: 'Australia, Commonwealth of'})
MERGE (t5)-[:HELD]->(c)

MERGE (t6:tournaments {tournamentName: 'Dota2_2023'})
ON CREATE SET t6.startDate = date('2023-10-10'), t6.endDate = date('2023-10-15')
WITH t6
MATCH (g:games {gameName: 'Dota 2'})
MERGE (g)-[:HAS]->(t6)
WITH t6
MATCH (c:country {countryName: 'Spain, Kingdom of'})
MERGE (t6)-[:HELD]->(c)
```

```
MERGE (t7:tournaments {tournamentName: 'PUBG_2023'})
ON CREATE SET t7.startDate = date('2023-11-15'), t7.endDate = date('2023-11-22')
WITH t7
MATCH (g:games {gameName: 'PUBG'})
MERGE (g)-[:HAS]->(t7)
WITH t7
MATCH (c:country {countryName: 'Switzerland, Swiss Confederation'})
MERGE (t7)-[:HELD]->(c)

MERGE (t8:tournaments {tournamentName: 'Heroes of the Storm_2023'})
ON CREATE SET t8.startDate = date('2023-12-03'), t8.endDate = date('2023-12-09')
WITH t8
MATCH (g:games {gameName: 'Heroes of the Storm'})
MERGE (g)-[:HAS]->(t8)
WITH t8
MATCH (c:country {countryName: 'Brazil, Federative Republic of'})
MERGE (t8)-[:HELD]->(c)

MERGE (t9:tournaments {tournamentName: 'Hearthstone Tournament9_2023'})
ON CREATE SET t9.startDate = date('2024-01-10'), t9.endDate = date('2024-01-17')
WITH t9
MATCH (g:games {gameName: 'Hearthstone'})
MERGE (g)-[:HAS]->(t9)
WITH t9
MATCH (c:country {countryName: 'Germany, Federal Republic of'})
MERGE (t9)-[:HELD]->(c)
```

```
MERGE (t10:tournaments {tournamentName: 'Arena of Valor Tournament10_2023'})
ON CREATE SET t10.startDate = date('2024-02-20'), t10.endDate = date('2024-02-27')
WITH t10
MATCH (g:games {gameName: 'Arena of Valor'})
MERGE (g)-[:HAS]->(t10)
WITH t10
MATCH (c:country {countryName: 'Italy, Italian Republic'})
MERGE (t10)-[:HELD]->(c)
```

## Creating platform nodes manually

```
MERGE (p1:platform {platformName: 'PC'})
MERGE (p2:platform {platformName: 'Mobile'})
MERGE (p3:platform {platformName: 'Console'})
```

## Establishing relationships between tournaments---[PLAYED_ON]--->platform

```
MATCH (t:tournaments {tournamentName: 'Overwatch_2023'}), (p:platform {platformName: 'PC'})
MERGE (t)-[:PLAYED_ON]->(p)

MATCH (t:tournaments {tournamentName: 'Starcraft II_2023'}), (p:platform {platformName: 'PC'})
MERGE (t)-[:PLAYED_ON]->(p)

MATCH (t:tournaments {tournamentName: 'League of Legends_2023'}), (p:platform {platformName: 'PC'})
MERGE (t)-[:PLAYED_ON]->(p)

MATCH (t:tournaments {tournamentName: 'Dota2_2023'}), (p:platform {platformName: 'PC'})
```

```
MERGE (t)-[:PLAYED_ON]->(p)

MATCH (t:tournaments {tournamentName: 'Heroes of the Storm_2023'}), (p:platform {platformName: 'PC'})

MERGE (t)-[:PLAYED_ON]->(p)


MATCH (t:tournaments {tournamentName: 'Fortnite_2023'}), (p:platform {platformName: 'Console'})

MERGE (t)-[:PLAYED_ON]->(p)


MATCH (t:tournaments {tournamentName: 'Counter-Strike_2023: Global Offensive'}), (p:platform {platformName: 'Console'})

MERGE (t)-[:PLAYED_ON]->(p)


MATCH (t:tournaments {tournamentName: 'PUBG_2023'}), (p:platform {platformName: 'Mobile'})

MERGE (t)-[:PLAYED_ON]->(p)


MATCH (t:tournaments {tournamentName: 'Hearthstone Tournament9_2023'}), (p:platform {platformName: 'Mobile'})

MERGE (t)-[:PLAYED_ON]->(p)


MATCH (t:tournaments {tournamentName: 'Arena of Valor Tournament10_2023'}), (p:platform {platformName: 'Mobile'})

MERGE (t)-[:PLAYED_ON]->(p)
```

## Creating data model

```
MATCH (c:country)-[:PART_OF]->(ct:continent {continentName: 'North America'})
MATCH (p:players)-[:FROM]->(c)
MATCH (p)-[:PLAYS]->(g:games)
MATCH (p)-[:PLAYS_IN]->(gen:genre)
MATCH (t:teams)-[:PLAYS]->(g)
MATCH (t:teams)-[:PLAYS_IN]->(gen)
MATCH (g)-[:IS]->(gen)
MATCH (g)-[:HAS]->(tour:tournaments)
MATCH (tour)-[:HELD]->(c)
MATCH (tour)-[:PLAYED_ON]->(plat:platform)
RETURN ct, c, p, t, g, gen, tour, plat
```

## Deleting null data nodes

### Deleting country nodes and their relationships with continent node that has null value in avgIncome (no players)

```
MATCH (c:country)-[r:PART_OF]->(ct:continent)
WHERE c.avgIncome IS NULL
DELETE r, c
```

### Deleting continent nodes with null value as avgIncome (no players)

```
MATCH (c:continent)
WHERE c.avgIncome IS NULL
DELETE c
```

# Querying the database

## Get all the players of a specific game

```
MATCH (p:players)-[:PLAYS]->(g:games {gameName: 'Counter-Strike: Global Offensive'})
RETURN p, g
```

## Get all the teams of a specific genre

```
MATCH (t:teams)-[:PLAYS_IN]->(g:genre {genreName: 'First-Person Shooter'})
RETURN t, g
```

## Get all the players, teams, games, tournaments, tournament_platform and tournament_location of a specific genre

```
MATCH (p:players)-[:PLAYS_IN]->(gen:genre {genreName: 'First-Person Shooter'})<-[:IS]-(g:games)-[:HAS]->(tour:tournaments)-[:PLAYED_ON]->(plat:platform)
MATCH (tour)-[:HELD]->(c:country)
MATCH (g)<-[:PLAYS]-(t:teams)-[:PLAYS_IN]->(gen)
RETURN p, gen, g, tour, plat, c, t
```

## Get all the players, teams, tournaments, tournament_platform and tournament_location for a specific game

```
MATCH (p:players)-[:PLAYS]->(g:games {gameName: 'Counter-Strike: Global Offensive'})-[:HAS]->(tour:tournaments)-[PLAYED_ON]->(plat:platform)
MATCH (tour)-[:HELD]->(c:country)
MATCH (t:teams)-[:PLAYS]->(g)
RETURN p, g, tour, plat, c, t
```

## Get all the players who play games whose tournaments are going to held in a specific country

```
MATCH (c:country {countryName: 'United States of America'})<-[:FROM]-(p:players)-[:PLAYS]->(g:games)
MATCH (c)<-[:HELD]-(t:tournaments)<-[:HAS]-(g)
RETURN c, p, t, g
```