

DNA Project

Team 20

November 2023

1 Team 20-Data Dudes

- Sahil Patel(2022101046)
- Kevin Thakkar(2022101064)
- Tanishq Agarwal(2022101060)
- Gopal Garg(2022101079)

2 Mapping ER Model to Relational Model

2.1 Changes Made to Model

- Team was converted from a Weak Entity to a Strong Entity by adding a team_id attribute to the entity.
- Wins added in players - To find Winrate of the Player efficiently.
- Added win or lose in plays table - To obtain statistics of player in a particular match in particular team efficiently.
- Added score in plays Table - To find top players and Analyse their performance in a particular match.

2.2 Explanation of General Steps taken to Convert to Relational Model

- All Entities were converted into relations with their primary key and the entity's attributes.
- Each attribute of an entity becomes a column in the corresponding table.
- Relationships become foreign keys in the related tables. If there's a one-to-many relationship, the table on the "many" side gets a foreign key referencing the primary key of the table on the "one" side.
- We Determined whether the relationships are one-to-one, one-to-many, or many-to-many.

2.3 Explanation of Steps taken to Convert to Relational Model

- **Plays** relationship was converted into **Plays** relationship in relational model which have (matchid + playerid).
- **PartOf** relationship was converted into **MemberOf** relationship in relational model which have (playerid + clanid).
- **Kills** 4-ary relationship was converted into **Kills** relationship in relational model which have (playerid1 + playerid2 + matchid + weaponid).
- Weak entities were converted into a relation by adding the primary key of the entity that identifies them and listing the rest of their attributes such as **Inventory items** was converted into a relation which was identified by using **playerid**
- 1. Plays Relation - It contains matchid, playerid, winorlose and score which is connecting player to match player played and the result of match.
- 2. Kills - It contains Playerid1, playerid2, weaponid and matchid which is connecting the 2 players in recursive relationship with the match id and the weapon used to kill, where playerid1 refers to the killer.
- 3. MemberOf - It contains the Playerid and Clanid which is to connect Clan to a Player.
- 4. MatchDescription - It contains the playerid, matchid and teamid which connects players to team and teams to matches.

2.4 Relational Model

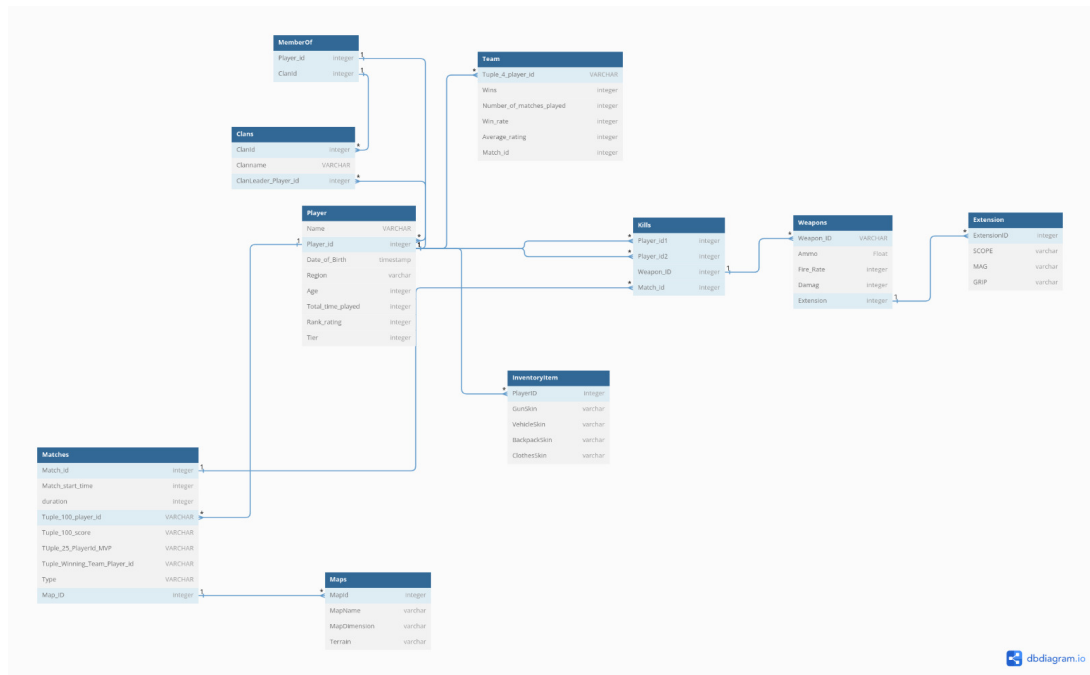


Figure 1: Relational Model

- We have removed all the functional dependencies in our previous relational model. Tuple of 100 player-id has been broken to two tables' plays.
- We have removed the Tuple 4 player id and converted team from weak entity to strong entity and made a table **match-description** for it.

3 First Normal Form

- By converting the ER model to a relational model, we've converted it into first normal form from the steps mentioned above which has all attributes as atomic/simple/indivisible, i.e. no composite or multi-valued attributes.

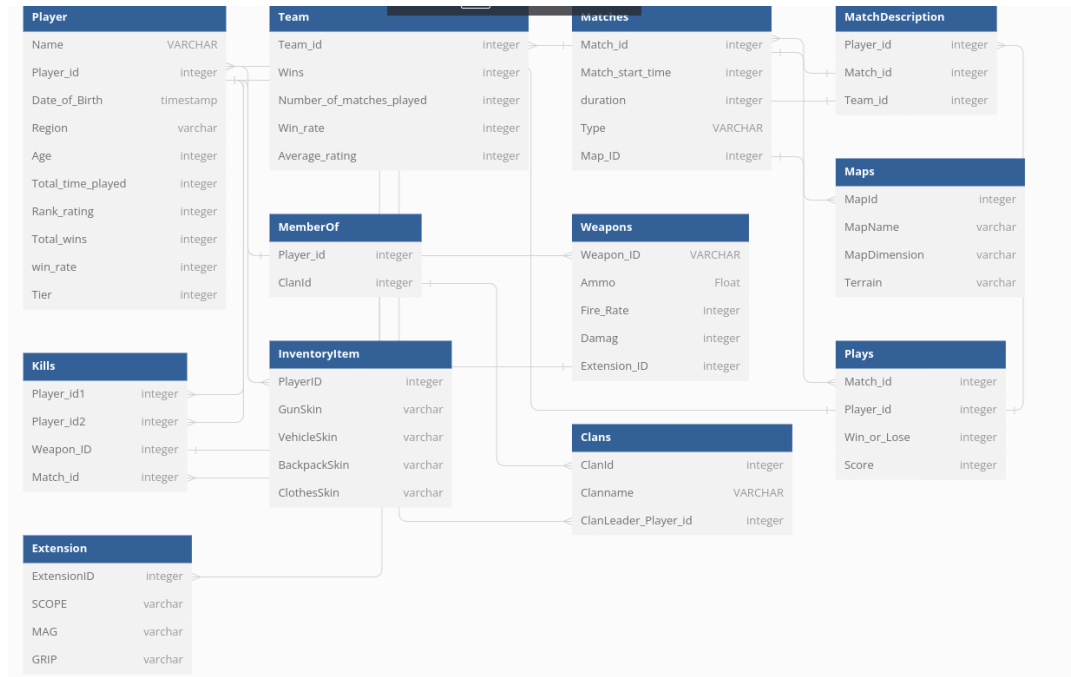


Figure 2: normalised relational model

4 Second Normal Form

- After converting the ER model by aforementioned steps , we see that for every relation in the relational model, every non-prime attribute is fully functionally dependent on the primary key of the entity. Hence, the relational model is already in second normal form as well.

5 Third Normal Form

- After converting the ER model by aforementioned steps, we see that for every relation in the relational model, every non-prime attribute of every relation R in the relational model is not transitively dependent on the primary key of that relation. Hence, the relational model is already in third normal form as well.