Introduction to Data Modelling – CA

# Online chess game platform

**Maximiliano Herrera: 20103212**

Table of Contents

[Online chess game platform 1](#_Toc58493956)

[Area of interest chosen 2](#_Toc58493957)

[Online chess game platform 2](#_Toc58493958)

[Technologies used 2](#_Toc58493959)

[MySQL Workbench 2](#_Toc58493960)

[Notepad++ 2](#_Toc58493961)

[Draw.io 2](#_Toc58493962)

[C# .NET 2](#_Toc58493963)

[Assumptions 2](#_Toc58493964)

[Tables 2](#_Toc58493965)

[Player 2](#_Toc58493966)

[Membership 3](#_Toc58493967)

[MembershipType 3](#_Toc58493968)

[PlayerRole 3](#_Toc58493969)

[MatchGame 3](#_Toc58493970)

[TypeOfMatch 3](#_Toc58493971)

[ResultType 3](#_Toc58493972)

[Tournament 4](#_Toc58493973)

[TournamentMatch 4](#_Toc58493974)

[TournamentPlayer 4](#_Toc58493975)

[Move 4](#_Toc58493976)

[Country 4](#_Toc58493977)

[Extra 4](#_Toc58493978)

[Export PGN into table Move 4](#_Toc58493979)

[Steps 5](#_Toc58493980)

[Input 5](#_Toc58493981)

[Output 5](#_Toc58493982)

[Chess vocabulary 5](#_Toc58493983)

# Area of interest chosen

## Online chess game platform

The database for this platform is intended to store chess matches played online as well as historic tournaments or matches. More importantly the database support storing data for calculating the company’s profit through the entities Membership and Tournament.

Main tables:

* Player
* Membership
* Tournament

# Technologies used

## MySQL Workbench

Development environment used for creating database and tables, populate tables and create queries.

## Notepad++

I used Notepad++ for formatting queries (manually) and because I like how Notepad++ highlights SQL syntax.

## Draw.io

I used Draw.io for creating the Entity Relationship Diagram.

## C# .NET

I used C# for building an application console which converts PGN to SQL Insert statements in order to populate the table Move. This is an extra functionality that I wanted to implement, which is described in Extra section.

# Assumptions

* Ids are not auto incremental because I have chosen ORM approach which handles the auto incremental at application level. Also, for making easy the population of tables.
* Most of candidate keys from ERD are not used as PK because performance considerations and to make easy the population of tables.

# Tables

## Player

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column name | Data type | Primary Key | Foreign Key | Not null | Reference to | Constraint |
| id | int | X |  | X |  |  |
| firstName | varchar(50) |  |  | X |  |  |
| lastName | varchar(50) |  |  | X |  |  |
| email | varchar(50) |  |  |  |  | Unique |
| elo | int |  |  | X |  |  |
| countryId | int |  | X | X | Country (id) |  |
| disabled | tinyint(1) |  |  |  |  |  |

## Membership

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column name | Data type | Primary Key | Foreign Key | Not null | Reference to | Constraint |
| id | int | X |  | X |  |  |
| playerId | int |  | X | X | Player(id) |  |
| membershipTypeId | int |  | X | X | MembershipType(id) |  |
| startDate | date |  |  | X |  |  |
| expirationDate | date |  |  | X |  |  |
| disabled | tinyint(1) |  |  |  |  |  |

## MembershipType

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column name | Data type | Primary Key | Foreign Key | Not null | Reference to | Constraint |
| id | int | X |  | X |  |  |
| name | varchar(50) |  |  | X |  |  |
| description | varchar(250) |  |  | X |  |  |
| price | decimal(10,2) |  |  | X |  |  |

## PlayerRole

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column name | Data type | Primary Key | Foreign Key | Not null | Reference to | Constraint |
| playerId | int | X | X | X | Player(id) |  |
| matchGameId | int | X | X | X | MatchGame(Id) |  |
| white | tinyint(1) | X |  | X |  |  |
| resultTypeId | int |  | X | X | ResultType(Id) |  |

## MatchGame

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column name | Data type | Primary Key | Foreign Key | Not null | Reference to | Constraint |
| id | int | X |  | X |  |  |
| typeOfMatchId | int |  | X | X | TypeOfMatch(Id) |  |
| datePlayed | date |  |  | X |  |  |

## TypeOfMatch

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column name | Data type | Primary Key | Foreign Key | Not null | Reference to | Constraint |
| id | int | X |  | X |  |  |
| name | varchar(50) |  |  | X |  |  |
| label | varchar(50) |  |  |  |  |  |

## ResultType

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column name | Data type | Primary Key | Foreign Key | Not null | Reference to | Constraint |
| id | int | X |  | X |  |  |
| name | varchar(50) |  |  |  |  |  |
| label | varchar(50) |  |  |  |  |  |
| points | float |  |  | X |  |  |

## Tournament

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column name | Data type | Primary Key | Foreign Key | Not null | Reference to | Constraint |
| id | int | X |  | X |  |  |
| name | varchar(100) |  |  | X |  |  |
| year | year |  |  | X |  |  |
| startDate | date |  |  |  |  |  |
| endDate | date |  |  |  |  |  |
| countryId | int |  | X |  | Country(Id) |  |
| onWebPlatform | tinyint(1) |  |  |  |  |  |
| entryFee | decimal(10,2) |  |  |  |  |  |
| prize | decimal(10,2) |  |  |  |  |  |

## TournamentMatch

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column name | Data type | Primary Key | Foreign Key | Not null | Reference to | Constraint |
| tournamentId | int | X | X | X | Tournament(id) |  |
| matchId | int | X | X | X | MatchGame(Id) |  |

## TournamentPlayer

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column name | Data type | Primary Key | Foreign Key | Not null | Reference to | Constraint |
| tournamentId | int | X | X | X | Tournament(id) |  |
| playerId | int | X | X | X | Player(id) |  |
| points | float |  |  | X |  |  |
| ranking | int |  |  | X |  |  |

## Move

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column name | Data type | Primary Key | Foreign Key | Not null | Reference to | Constraint |
| id | int | X |  | X |  |  |
| matchGameId | int |  | X | X | MatchGame(id) |  |
| moveOrder | int |  |  | X |  | Greater than zero |
| whiteMove | varchar(10) |  |  | X |  |  |
| blackMove | varchar(10) |  |  |  |  |  |
| whiteMoveTime | time |  |  |  |  |  |
| blackMoveTIme | time |  |  |  |  |  |

## Country

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column name | Data type | Primary Key | Foreign Key | Not null | Reference to | Constraint |
| id | int | X |  | X |  |  |
| name | varchar(50) |  |  | X |  | Unique |

# 

# Queries

Queries are defined on section 3 of SQL script under the section name “3. Queries”. Please copy the section name and search it on file “Script - Db creation, data population and queries.sql”.

# Extra

## Export PGN into table Move

Chess games can be exported to PGN format which is a string for storing chess games. The benefit of PGN is that can be read by humans and is supported by most of chess software.

<https://en.wikipedia.org/wiki/Portable_Game_Notation>

My goal was to take a PGN and persist it into my database schema, but for doing that I had to normalize the PGN and insert it into table Move.

### Steps

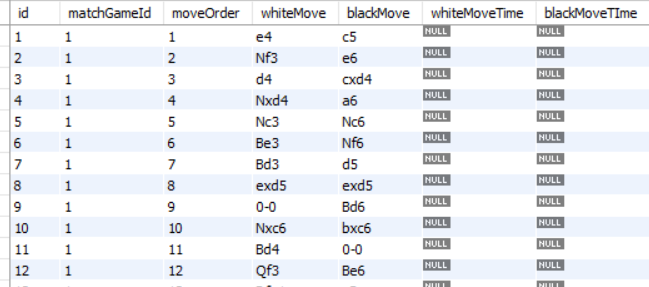
1. Take a PGN and Normalize it following Move table structure.
2. Create SQL insert statements for data normalized.
3. Execute the inserts into table Move.

### Input

PGN: <https://www.chessgames.com/perl/chessgame?gid=1008424>

1.e4 c5 2.Nf3 e6 3.d4 cxd4 4.Nxd4 a6 5.Nc3 Nc6 6.Be3 Nf6 7.Bd3 d5 8.exd5 exd5 9.0-0 Bd6 10.Nxc6 bxc6 11.Bd4 0-0 12.Qf3 Be6 13.Rfe1 c5 14.Bxf6 Qxf6 15.Qxf6 gxf6 16.Rad1 Rfd8 17.Be2 Rab8 18.b3 c4 19.Nxd5 Bxd5 20.Rxd5 Bxh2+ 21.Kxh2 Rxd5 22.Bxc4 Rd2 23.Bxa6 Rxc2 24.Re2 Rxe2 25.Bxe2 Rd8 26.a4 Rd2 27.Bc4 Ra2 28.Kg3 Kf8 29.Kf3 Ke7 30.g4 f5 31.gxf5 f6 32.Bg8 h6 33.Kg3 Kd6 34.Kf3 Ra1 35.Kg2 Ke5 36.Be6 Kf4 37.Bd7 Rb1 38.Be6 Rb2 39.Bc4 Ra2 40.Be6 h5 41.Bd7

### Output



C# program for generating SQL inserts from a PGN: <https://github.com/herreramaxi/NCIRL-INTRO-TO-DB/blob/main/DBHelpers/DBHelpers/Program.cs>

# Chess vocabulary

Elo: Method for calculating the relative skill levels of a chess player. Professionals in chess could have around 2500 or 3000 and beginners around 800.

PGN: Portable Game Notation (PGN) is a standard plain text format for recording chess games (both the moves and related data), which can be read by humans and is also supported by most chess software. (Wikipedia)