CS6131 Database Design

Project Final Report Submission

By Yap Yuan Xi

Submission Instructions

* You will need to submit the following files in your final project submission: * Your Jupyter Notebook report. Name the report `ProjectFinalReport.ipynb`. * All relevant image files to be displayed in this report (make sure you use relative file referencing and the image will display in another computer). * Attached each file one by one and upload on Coursemology. * Please print a copy of the final report to OneNote Individual Notebook space > Project. Double check on the image resolution. If the resolution is poor, please copy and paste the ORIGINAL clear image into the OneNote page (paste at the side of the printed image). * Any submission that fails to comply to the above instructions will result in upto 5% penalty. * You may wish to refer to the following reference to help organize and "beautify" your final report here. https://thecodingbot.com/markdown-in-jupyter-ipython-notebook-cheatsheet/

Section A: Overview & Business Rules

Overview

Complete your writeup of the project overview here.

Chess has been gaining popularity since the start of the pandemic. With the game of chess becoming more mainstream, someone needs to provide the service of playing chess online. A chess website is one of the ways of providing such services, with several purposes such as:

- **Online play**: Chess websites provide an easy and convenient way for people to play chess against other players from around the world.
- **Learning and Improvement**: Chess websites offer resources such as puzzles, and tactics trainers to help users improve their skills.
- **Community**: Chess websites provide a community for chess players to interact and connect with others who share their interest in the game.
- **Competition**: Chess websites offer organized tournaments and events for players to compete against each other and test their skills.
- **Accessibility**: Chess websites make the game of chess accessible to people who may not have easy access to physical chess sets or opponents.

When making a chess website with the above purposes, there is a lot of data to store. Tournament information, Player information, Player history, Played Games, Puzzles, Tactics. These are all data that the website must handle. All these information are related to one another, like which players are playing a chess game and which players are participating in the tournament. With a Relational DataBase Management System (RDBMS), the data and all its relationships can be maintained and managed easily and efficiently, allowing for a smooth experience when playing chess online.

Business Rules

Complete your writeup of the final business rules here.

A **Player** has a unique ***username***, with some **rating history** and some **puzzle history**.
Each **rating history** or **puzzle history** consists of a certain unique ***datetime*** and an elo ***rating***. ***rating*** is derived from the rated games played by the **Player**.
Additionally, each player has a ***email***, ***birthday***, ***about***, ***avatar***, ***account_create_date*** and ***password*** for login purposes

Games must be *played* by 2 **Players**. The 2 **Player**s will *play* as white or black. The **Game** has a unique ***game_id***, ***time control***, ***datetime*** of when it is played, ***pgn*** that is a string representing the game, is either ***rated*** or not, and is either ***public*** or not. ***time control*** is made up of ***initial time*** and ***increment*** per move.

A **Team** has a unique ***name***, ***creation date***, ***about*** and ***icon***. **Players** can *belong to* at most 1 **Team**.
On joining, the ***join date*** of the **Player**s are recorded. **Team**s are moderated by some **Player**s in the **Team**, and their ***start date*** is recorded.
A **Team** must have at least 1 **Player** moderating and the creator is automatically a moderator of the team.
Teams also have a ***member_count***, the number of players in the **Team**.
If there are no moderators, the oldest **Player** will become a moderator.
If a **Team** has no **Player**s, then the **Team** is deleted.

A **Player** can create an **Application**, containing a short ***message*** and an ***id*** unique for each player.
The **Application** is to join an existing **Team**.
Only a **Team**'s moderator can approve the application.

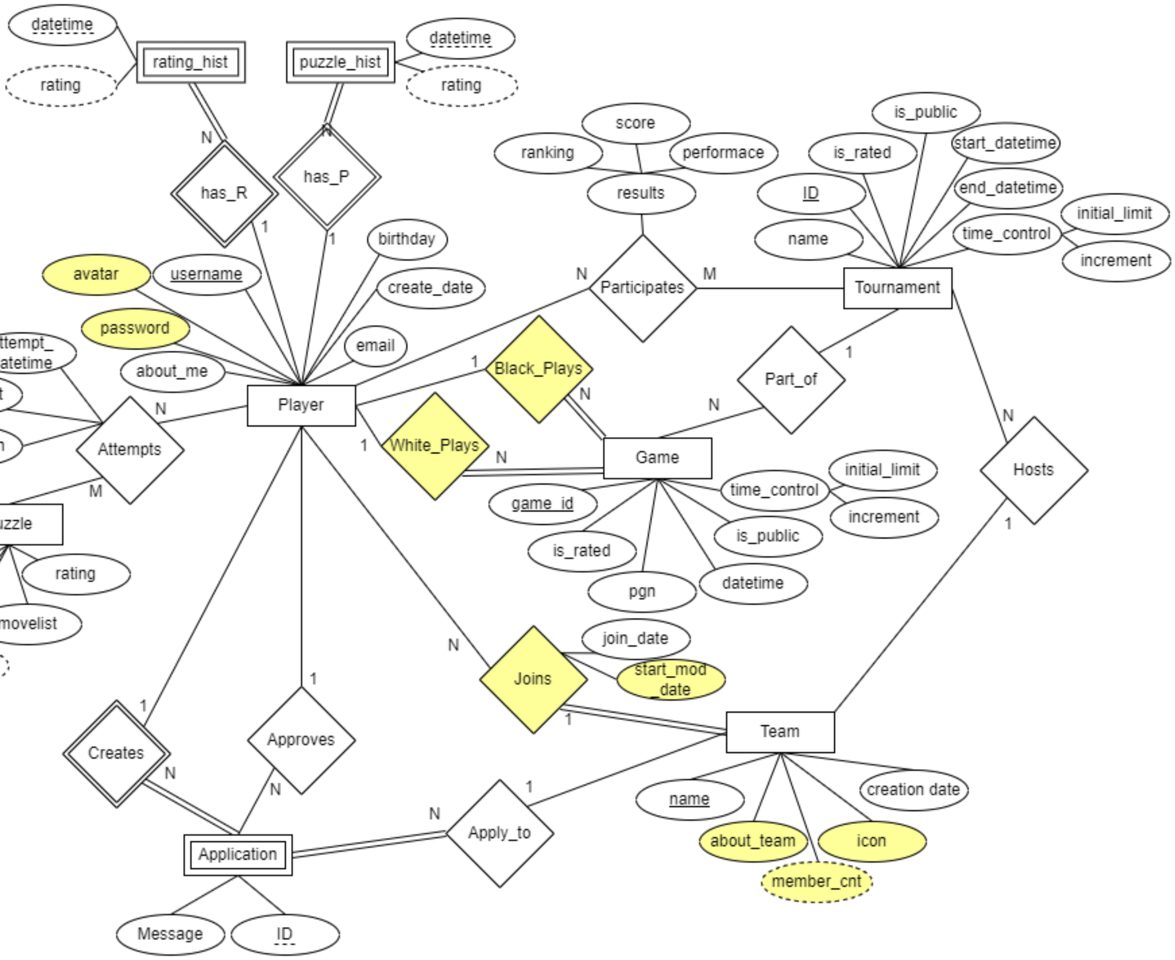
Each **Tournament** hosted will have a unique ***id***, ***name***, ***start datetime***, ***end datetime***, ***time control***, whether it ***is rated*** and whether it **is public**.
Tournaments can *consists* of **Game**s, and may be hosted by one **Team**.
A **Player** can *participate* in **Tournament**s, and will play **Game**s in the **Tournament**.
When **Player**s *participate*, their results, made up of their ***ranking***, raw ***score*** and ***performace***, will be saved.
Tournaments which are not ***public*** must be hosted by a **Team** and only **Player**s in the **Team** can *participate* in it.

Puzzles can be *attempted* by **Player**, and the **Player *is either correct*** or not, taking some ***time***. ***time*** is a factor in the rating change of the player.
For each *attempt*, ***attempt datetime*** is also recorded. </br> **Puzzle**s has a unique ***id***, ***rating***, ***movelist*** of puzzle solution, ***fen*** of position and ***solve count***, which is derived from the number of correct attempts.

Section B: ER Model

Attached the image of your FINAL ER Model here.

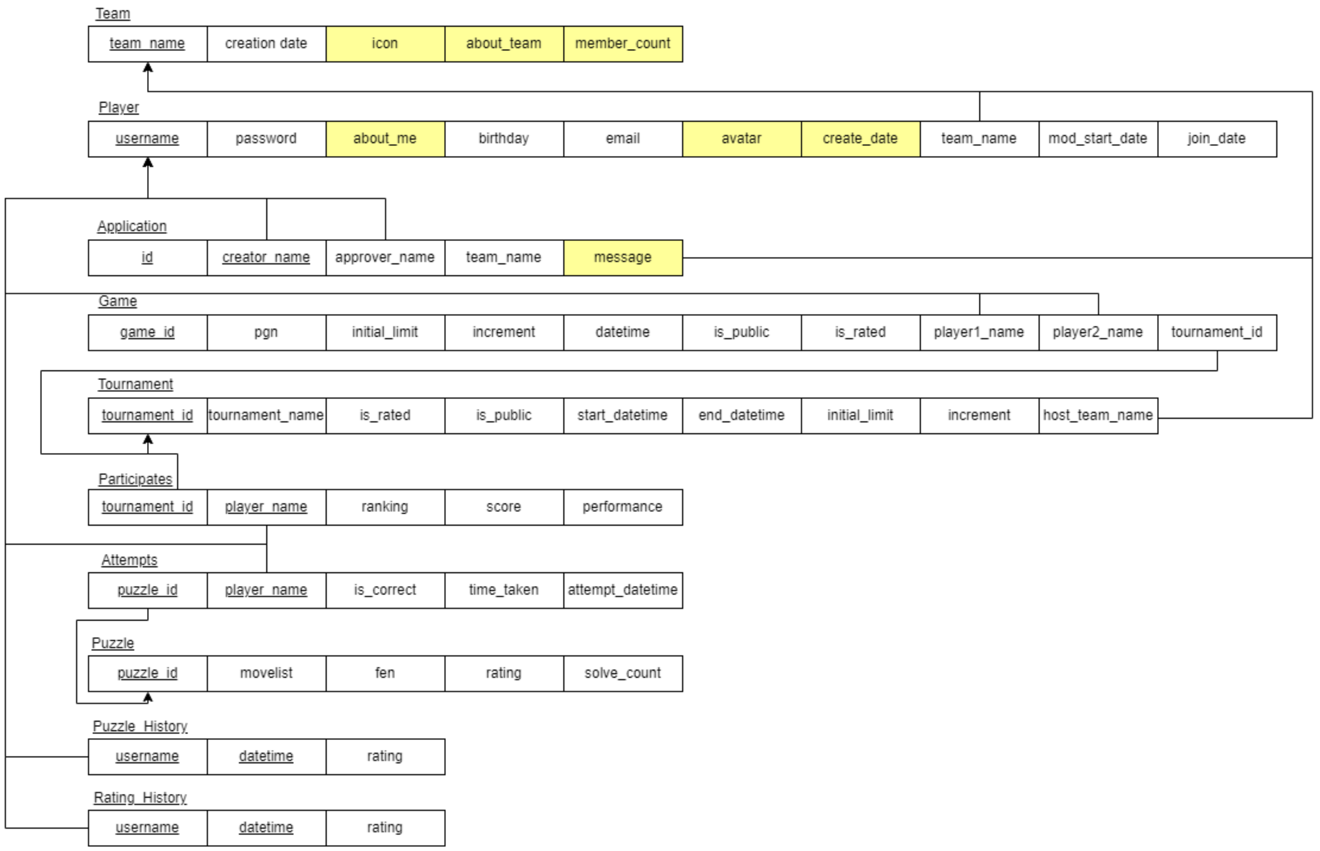
Double click here to insert image. Any further changes from previous deliverables should be highlighted in yellow.



Section C: Relational Model

Attached the image of your FINAL Relational Model here.

Double click here to insert image. Any further changes from previous deliverables should be highlighted in yellow.



Team

team_name	creation date	icon	about_team	member_count

Player

username	password	about_me	birthday	email	avatar	create_date	team_name	mod_start_date	join_date

Application

id	creator_name	approver_name	team_name	message

Game

game_id	pgn	initial_limit	increment	datetime	is_public	is_rated	player1_name	player2_name	tournament_id

Tournament

tournament_id	tournament_name	is_rated	is_public	start_datetime	end_datetime	initial_limit	increment	host_team_name

Participates

tournament_id	player_name	ranking	score	performance

Attempts

puzzle_id	player_name	is_correct	time_taken	attempt_datetime

Puzzle

puzzle_id	movelist	fen	rating	solve_count

Puzzle_History

username	datetime	rating

Rating_History

username	datetime	rating

> Justify your mapping strategy from ER to relational, particularly if the approach deviates from the norm, or you have inheritance in your ER model (i.e. which strategy is adopted for inheritance mapping and why).

Relationship Mapping:

- one-to-many: Following the norm, we will store the foreign key in the entity with cardinality many. This reduces number of tables, and the number of joins when performing a query.
  The relationships are in this category are:
  - `Creates`
  - `Approves`
  - `Apply_to`
  - `Hosts`
  - `Part_of`
  - `has_R`
  - `has_P`
  - `Belong_to`

- many-to-many: Following the norm, we will store the relationship in a new table. This is the only way to adequately represent the relationship.
  The relationship is in this category are:
  - `Attempts`
  - `Participates`

- one-to-one: There are 2 one-to-one relationships, `White_Plays` and `Black_Plays`. Following the norm, we will store the relationship in Game, which participates fully in the relationship.

> If the relational schema mapped from the ER is not in 3NF, propose relevant normalization to make all relations in 3NF. You may leave this part blank if no further normalization is required.

Proposed normalization, if any

Section D: DDL Schema

> Fill in the relevant code required to create the relations from your database.
> Your code should be end to end (i.e. I should be able to execute on my computer without much problem).

In [1]:

```
%load_ext sql
```

In [2]:

```
%sql mysql+pymysql://root:admin@localhost/  --make sure user is root and password is adm
```

In [3]:

```
%%sql
DROP DATABASE IF EXISTS CHESSIBLE;
CREATE DATABASE CHESSIBLE;
USE CHESSIBLE;

CREATE TABLE Teams (
    team_id       INT AUTO_INCREMENT,
    team_name     VARCHAR(40),
    creation_date DATE,
    about_team    TEXT,
    icon          BLOB,
    member_count  INT DEFAULT 0,

    PRIMARY KEY(team_id),
    UNIQUE(team_name)
);

CREATE TABLE Players (
    username        VARCHAR(40),
    password        VARCHAR(120),
    birthday        DATE,
    email           VARCHAR(300),
    create_date     DATETIME,
    team_id         INT,
    mod_start_date  DATETIME,
    join_date       DATETIME,
    about_me        TEXT,
    avatar          BLOB,

    PRIMARY KEY(username),
    FOREIGN KEY(team_id) REFERENCES Teams(team_id) ON UPDATE cascade ON DELETE cascade
);

CREATE TABLE Puzzles (
    puzzle_id   INT AUTO_INCREMENT,
    movelist    VARCHAR(90),
    fen         VARCHAR(90),
    rating      INT,
    solve_count INT,

    PRIMARY KEY(puzzle_id)
);

CREATE TABLE Tournaments (
    tournament_id   INT AUTO_INCREMENT,
    tournament_name VARCHAR(40),
    is_rated        BIT,
    is_public       BIT,
    start_datetime  DATETIME,
    end_datetime    DATETIME,
    initial_limit   INT,
    increment       INT,
    host_team_id    INT,

    PRIMARY KEY(tournament_id),
    UNIQUE(tournament_name),
    FOREIGN KEY(host_team_id) REFERENCES Teams(team_id) ON UPDATE cascade ON DELETE casc
);

CREATE TABLE Applications (
    id              INT AUTO_INCREMENT,
    message         TEXT,
    creator_name    VARCHAR(40),
    approver_name   VARCHAR(40),
    team_id         INT,

    PRIMARY KEY(id, creator_name),
    UNIQUE (creator_name, team_id),
    FOREIGN KEY(creator_name)  REFERENCES Players(username),
    FOREIGN KEY(approver_name) REFERENCES Players(username),
    FOREIGN KEY(team_id) REFERENCES Teams(team_id) ON UPDATE cascade ON DELETE cascade
);

CREATE TABLE Games (
    game_id       INT AUTO_INCREMENT,
    pgn           VARCHAR(2000),
    is_public     BIT,
    is_rated      BIT,
    initial_limit INT,
    increment     INT,
    datetime      DATETIME,
    white_name    VARCHAR(40),
    black_name    VARCHAR(40),
    tournament_id INT,

    PRIMARY KEY(game_id),
    FOREIGN KEY(white_name)    REFERENCES Players(username),
    FOREIGN KEY(black_name)    REFERENCES Players(username),
    FOREIGN KEY(tournament_id) REFERENCES Tournaments(tournament_id)
);

CREATE TABLE Participates (
    tournament_id  INT,
    player_name    VARCHAR(40),
    ranking        INT,
    score          INT,
    performance    INT,

    FOREIGN KEY(tournament_id) REFERENCES Tournaments(tournament_id),
    FOREIGN KEY(player_name) REFERENCES Players(username)
);

CREATE TABLE Attempts (
    puzzle_id        INT,
    player_name      VARCHAR(40),
    is_correct       BIT,
    time_taken       INT,
    attempt_datetime DATETIME,

    FOREIGN KEY(puzzle_id) REFERENCES Puzzles(puzzle_id),
    FOREIGN KEY(player_name) REFERENCES Players(username)
);


CREATE TABLE Puzzle_History (
    username VARCHAR(40),
    datetime DATETIME,
    rating   INT,

    PRIMARY KEY(username, datetime),
    FOREIGN KEY(username) REFERENCES Players(username)
);

CREATE TABLE Rating_History (
    username VARCHAR(40),
    datetime DATETIME,
    rating   INT,

    PRIMARY KEY(username, datetime),
    FOREIGN KEY(username) REFERENCES Players(username)
);
```

```
 * mysql+pymysql://root:***@localhost/
10 rows affected.
1 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
```

Out[3]:

```
[]
```

Section δ: Triggers

I am putting triggers here, so that my data population is easier.

ELO update trigger

When a game is uploaded and is rated, we need to calculate the new rating for each person.
The details for the formula can be found on wikipedia.
We use $k = 32$ as it is common practice, ensuring some significant penalty but not drastic decrements in their rating.

In [4]:

```
%%sql
DROP TRIGGER IF EXISTS update_elo;
DROP TRIGGER IF EXISTS update_elo_puzzle;

CREATE TRIGGER update_elo AFTER INSERT ON Games
```

```
FOR EACH ROW
BEGIN
    DECLARE white_rating INT;
    DECLARE black_rating INT;
    DECLARE white_expected FLOAT;
    DECLARE black_expected FLOAT;
    DECLARE white_new_rating FLOAT;
    DECLARE black_new_rating FLOAT;

    -- get the ratings of the players
    SELECT rating INTO white_rating FROM Rating_History WHERE username = NEW.white_name
    SELECT rating INTO black_rating FROM Rating_History WHERE username = NEW.black_name

    -- calculate the expected scores
    SET white_expected = 1 / (1 + POWER(10, (black_rating - white_rating) / 400));
    SET black_expected = 1 / (1 + POWER(10, (white_rating - black_rating) / 400));

    -- calculate the new ratings
    IF NEW.pgn LIKE "%1-0" THEN
        SET white_new_rating = white_rating + 32 * (1 - white_expected);
        SET black_new_rating = black_rating + 32 * (0 - black_expected);
    ELSEIF NEW.pgn LIKE "%0-1" THEN
        SET white_new_rating = white_rating + 32 * (0 - white_expected);
        SET black_new_rating = black_rating + 32 * (1 - black_expected);
    ELSE
        SET white_new_rating = white_rating + 32 * (0.5 - white_expected);
        SET black_new_rating = black_rating + 32 * (0.5 - black_expected);
    END IF;

    -- insert the new ratings into the Rating_History table
    IF NEW.is_rated THEN
        INSERT INTO Rating_History (username, rating, datetime) VALUES (NEW.white_name,
        INSERT INTO Rating_History (username, rating, datetime) VALUES (NEW.black_name,
    END IF;
END ;

CREATE TRIGGER update_elo_puzzle AFTER INSERT ON Attempts
FOR EACH ROW
BEGIN
    DECLARE player_rating INT;
    DECLARE puzzle_rating INT;
    DECLARE player_expected FLOAT;
    DECLARE puzzle_expected FLOAT;
    DECLARE player_new_rating FLOAT;
    DECLARE puzzle_new_rating FLOAT;

    -- get the ratings of the player and puzzle
    SELECT rating INTO player_rating FROM Puzzle_History WHERE username = NEW.player_nam
    SELECT rating INTO puzzle_rating FROM Puzzles WHERE puzzle_id = NEW.puzzle_id;

    -- calculate the expected score (treated as elo match between puzzle and player)
    SET player_expected = 1 / (1 + POWER(10, (puzzle_rating - player_rating) / 400));

    -- calculate the new ratings
    IF NEW.is_correct THEN
        SET player_new_rating = player_rating + 32 * (1 - player_expected);
    ELSE
        SET player_new_rating = player_rating + 32 * (0 - player_expected);
    END IF;

    -- insert the new ratings into the Rating_History table
    INSERT INTO Puzzle_History (username, rating, datetime) VALUES (NEW.player_name, ROU
END ;
```

```
 * mysql+pymysql://root:***@localhost/
0 rows affected.
```

```
0 rows affected.
0 rows affected.
0 rows affected.
```

Out[4]:

```
[]
```

Member Count update trigger

When a player leaves a team/joins a team, the member count of the team must be updated.

In [5]:

```
%%sql
DROP TRIGGER IF EXISTS update_member_count_insert;
DROP TRIGGER IF EXISTS update_member_count_update;
DROP TRIGGER IF EXISTS update_member_count_delete;

CREATE TRIGGER update_member_count_insert AFTER INSERT ON Players
    FOR EACH ROW
    BEGIN
        IF NEW.team_id IS NOT NULL THEN
            UPDATE Teams
            SET member_count = (
                SELECT COUNT(*)
                FROM Players
                WHERE team_id = NEW.team_id
            )
            WHERE team_id = NEW.team_id;
        END IF;
    END;

CREATE TRIGGER update_member_count_update AFTER UPDATE ON Players
    FOR EACH ROW
    BEGIN
        IF NEW.team_id IS NOT NULL THEN
            UPDATE Teams
            SET member_count = (
                SELECT COUNT(*)
                FROM Players
                WHERE team_id = NEW.team_id
            )
            WHERE team_id = NEW.team_id;
        END IF;

        IF OLD.team_id IS NOT NULL THEN
            UPDATE Teams
            SET member_count = (
                SELECT COUNT(*)
                FROM Players
                WHERE team_id = OLD.team_id
            )
            WHERE team_id = OLD.team_id;
        END IF;
    END;

CREATE TRIGGER update_member_count_delete AFTER DELETE ON Players
    FOR EACH ROW
    BEGIN
        IF OLD.team_id IS NOT NULL THEN
            UPDATE Teams
            SET member_count = (
                SELECT COUNT(*)
                FROM Players
                WHERE team_id = OLD.team_id
            )
            WHERE team_id = OLD.team_id;
        END IF;
    END;
```

```
 * mysql+pymysql://root:***@localhost/
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
```

Out[5]:

```
[]
```

Player Team update trigger

When an application is approved, automatically update the player's team status

In [6]:

```
%%sql
DROP TRIGGER IF EXISTS team_status_insert;
DROP TRIGGER IF EXISTS team_status_update;

CREATE TRIGGER team_status_insert BEFORE INSERT ON Applications
    FOR EACH ROW
tr: BEGIN
        -- Application not approved, cannot update user
        IF NEW.approver_name IS NULL THEN
            LEAVE tr;
        END IF;

        -- Check if approver is actually moderator
        IF (SELECT COUNT(*) FROM Players WHERE username = NEW.approver_name AND NEW.team
            SET NEW.approver_name = NULL;
            LEAVE tr;
        END IF;

        -- Ok finally time to update this
        UPDATE Players
        SET team_id = NEW.team_id, join_date = NOW(), mod_start_date = NULL
        WHERE username = NEW.creator_name;
    END;

CREATE TRIGGER team_status_update BEFORE UPDATE ON Applications
    FOR EACH ROW
tr: BEGIN
        -- Application not approved, cannot update user
        IF NEW.approver_name IS NULL THEN
            LEAVE tr;
        END IF;

        -- Check if approver is actually moderator
        IF (SELECT COUNT(*) FROM Players WHERE username = NEW.approver_name AND NEW.team
            SET NEW.approver_name = NULL;
            LEAVE tr;
        END IF;

        -- Ok finally time to update this
        UPDATE Players
        SET team_id = NEW.team_id, join_date = NOW(), mod_start_date = NULL
        WHERE username = NEW.creator_name;
    END;
```

```
 * mysql+pymysql://root:***@localhost/
0 rows affected.
0 rows affected.
```

```
0 rows affected.
0 rows affected.
```

Out[6]:

```
[]
```

Puzzle Solve Count trigger

When a puzzle is attempted, update the solve count.

In [7]:

```
%%sql
DROP TRIGGER IF EXISTS puzzle_solve_count;

CREATE TRIGGER puzzle_solve_count AFTER INSERT ON Attempts
    FOR EACH ROW
    BEGIN
        UPDATE Puzzles SET solve_count = (
            SELECT COUNT(distinct player_name)
            FROM Attempts
            WHERE puzzle_id = NEW.puzzle_id AND is_correct
        )
        WHERE puzzle_id = NEW.puzzle_id;
    END;
```

```
 * mysql+pymysql://root:***@localhost/
0 rows affected.
0 rows affected.
```

Out[7]:

```
[]
```

Player Initial Rating trigger

When a player registers a new account, the rating history must initalise with 1200 rating points for each type

In [8]:

```
%%sql
DROP TRIGGER IF EXISTS player_join_date_init;
DROP TRIGGER IF EXISTS player_rating_init;
DROP TRIGGER IF EXISTS player_puzzle_init;

CREATE TRIGGER player_join_date_init BEFORE INSERT ON Players
    FOR EACH ROW
    BEGIN
        IF NEW.create_date IS NULL THEN
            SET NEW.create_date = NOW();
        END IF;
    END;

CREATE TRIGGER player_rating_init AFTER INSERT ON Players
    FOR EACH ROW
    BEGIN
        INSERT INTO Rating_History VALUES
            (NEW.username, NEW.create_date, 1200);
        INSERT INTO Puzzle_History VALUES
            (NEW.username, NEW.create_date, 1200);
    END;
```

```
 * mysql+pymysql://root:***@localhost/
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
0 rows affected.
```

Out[8]:

```
[]
```

Section E: Data Population Script

Fill in relevant code to populate data into your database. It is sufficient to have 20-50 records per table (some may have more, some less). They should be logically related and realistic. Please do not overpopulate data. Note that you should use INSERT commands. If you are are using other means to populate your database, please ensure I can run the scripts easily.

In [9]:

```
%%sql
INSERT INTO Teams (team_name, creation_date, about_team) VALUES
    ('NUSH Black Knights', '2022-02-15', 'The real dark horses of NUS High!'),
    ('Pawn Stars', '2022-01-05', 'We are a group of beginners who are passionate about c
    ('Team Alpha', '2022-01-01', 'We are a group of chess enthusiasts who love playing t
    ('The Bishops', '2022-02-01', 'We are a team of players who focus on developing crea
    ('The Kings', '2022-03-01', 'We are a team of players who are determined to dominate
    ('The Chess Masters', '2022-01-15', 'We are a team of highly skilled and experienced
    ('The Chessaholics', '2022-03-01', 'We are a team of players who are addicted to the
    ('The Chess Wizards', '2022-01-25', 'We are a team of players who believe in the mag
    ('The Black and Whites', '2022-02-05', 'We are a team of players who believe in the
    ('The Checkmates', '2022-01-10', 'We are a team of players who enjoy the thrill of c
    ('The Rooks', '2022-02-20', 'We are a team of competitive players who aim to be the
    ('The Rook-ies', '2022-02-25', 'We are a team of beginners who are eager to learn an
    ('The Knights of Chess', '2022-02-10', 'We are a team of players who aim to be versa
    ('Chess Wizards', '2022-03-05', 'We are a group of experienced players who enjoy sha
    ('The Knights Templar', '2022-03-15', 'We are a team of players who are committed to
    ('The Grandmasters', '2021-12-01', 'We are a team of highly skilled players who have
    ('The Chess Titans', '2022-01-20', 'We are a team of players who aspire to be the be
    ('Queen\'s Gambits', '2022-03-10', 'We are a group of female chess players who suppo
    ('The Queens', '2022-02-05', 'We are a team of female players who are passionate abo
    ('Knight Movers', '2021-12-15', 'We are a team of experienced players who enjoy chal
```

```
 * mysql+pymysql://root:***@localhost/
20 rows affected.
```

Out[9]:

```
[]
```

In [10]:

```
%%sql
INSERT INTO Players(username, password, birthday, email, team_id, mod_start_date, join_d
    ("ethantan2509", "pbkdf2:sha256:260000$8QLpI0htlIEWohuQ$55a50ce1a38d45f282f50b7612e6
    ("KH-Brawl", "pbkdf2:sha256:260000$8QLpI0htlIEWohuQ$55a50ce1a38d45f282f50b7612e672c2
    ("zachary_song", "pbkdf2:sha256:260000$8QLpI0htlIEWohuQ$55a50ce1a38d45f282f50b7612e6
    ("hnnggrrrrr", "pbkdf2:sha256:260000$8QLpI0htlIEWohuQ$55a50ce1a38d45f282f50b7612e672
    ("drewmtr", "pbkdf2:sha256:260000$8QLpI0htlIEWohuQ$55a50ce1a38d45f282f50b7612e672c24
    ("oxidyZe", "pbkdf2:sha256:260000$8QLpI0htlIEWohuQ$55a50ce1a38d45f282f50b7612e672c24
    ("Passwordcringe", "pbkdf2:sha256:260000$8QLpI0htlIEWohuQ$55a50ce1a38d45f282f50b7612
    ("coolingTundra", "pbkdf2:sha256:260000$8QLpI0htlIEWohuQ$55a50ce1a38d45f282f50b7612e
    ("zetaNegativeOne", "pbkdf2:sha256:260000$8QLpI0htlIEWohuQ$55a50ce1a38d45f282f50b761
    ("LoverKent", "pbkdf2:sha256:260000$8QLpI0htlIEWohuQ$55a50ce1a38d45f282f50b7612e672c
    ("meower37", "pbkdf2:sha256:260000$8QLpI0htlIEWohuQ$55a50ce1a38d45f282f50b7612e672c2
    ("Haloen", "pbkdf2:sha256:260000$8QLpI0htlIEWohuQ$55a50ce1a38d45f282f50b7612e672c24d
    ("DillyWeed", "pbkdf2:sha256:260000$8QLpI0htlIEWohuQ$55a50ce1a38d45f282f50b7612e672c
    ("PICKLEPICKLEPICKLE", "pbkdf2:sha256:260000$8QLpI0htlIEWohuQ$55a50ce1a38d45f282f50b
    ("jsck413", "pbkdf2:sha256:260000$8QLpI0htlIEWohuQ$55a50ce1a38d45f282f50b7612e672c24
    ("WisdomGuy", "pbkdf2:sha256:260000$8QLpI0htlIEWohuQ$55a50ce1a38d45f282f50b7612e672c
    ("jane_doe", "pbkdf2:sha256:260000$8QLpI0htlIEWohuQ$55a50ce1a38d45f282f50b7612e672c2
    ("jim_smith", "pbkdf2:sha256:260000$8QLpI0htlIEWohuQ$55a50ce1a38d45f282f50b7612e672c
    ("jessica_brown", "pbkdf2:sha256:260000$8QLpI0htlIEWohuQ$55a50ce1a38d45f282f50b7612e
    ("bob_johnson", "pbkdf2:sha256:260000$8QLpI0htlIEWohuQ$55a50ce1a38d45f282f50b7612e67
    ("emily_williams", "pbkdf2:sha256:260000$8QLpI0htlIEWohuQ$55a50ce1a38d45f282f50b7612
    ("jason_brown", "pbkdf2:sha256:260000$8QLpI0htlIEWohuQ$55a50ce1a38d45f282f50b7612e67
    ("maria_garcia", "pbkdf2:sha256:260000$8QLpI0htlIEWohuQ$55a50ce1a38d45f282f50b7612e6
    ("sarah_taylor", "pbkdf2:sha256:260000$8QLpI0htlIEWohuQ$55a50ce1a38d45f282f50b7612e6
    ("daniel_kim", "pbkdf2:sha256:260000$8QLpI0htlIEWohuQ$55a50ce1a38d45f282f50b7612e672
    ("julie_lee", "pbkdf2:sha256:260000$8QLpI0htlIEWohuQ$55a50ce1a38d45f282f50b7612e672c

-- All passwords are "password"
-- ethantan2509, KH-Brawl, and zachary_song are the only accounts without a team
```

```
 * mysql+pymysql://root:***@localhost/
26 rows affected.
0 rows affected.
```

Out[10]:
```
[]
```

In [11]:
```
player_list = ["ethantan2509","KH-Brawl","zachary_song","hnngggrrrr","drewmtr","oxidyZe"
```

In [12]:
```
%%sql
INSERT INTO Tournaments (tournament_name, is_rated, is_public, start_datetime, end_datet
    ('Spring Blitz Tournament', 1, 1, '2023-04-22 12:00:00', '2023-04-22 16:00:00', 5, 2
    ('Summer Rapid Tournament', 0, 1, '2023-07-15 14:00:00', '2023-07-15 18:00:00', 6, 3
    ('Fall Classic Tournament', 1, 0, '2023-09-23 10:00:00', '2023-09-24 17:00:00', 10,
    ('Winter Warm-Up Tournament', 0, 1, '2023-12-02 13:00:00', '2023-12-02 17:00:00', 8,
    ('Beginner''s Cup', 0, 1, '2023-04-30 11:00:00', '2023-04-30 13:00:00', 3, 1, 5),
    ('Intermediate Invitational', 1, 1, '2023-05-12 15:00:00', '2023-05-12 19:00:00', 8,
    ('Advanced Arena', 1, 0, '2023-08-08 12:00:00', '2023-08-08 18:00:00', 12, 4, 11),
    ('International Open', 1, 1, '2023-10-14 10:00:00', '2023-10-15 17:00:00', 20, 5, 6)
    ('Queen''s Gambit Cup', 1, 0, '2023-11-25 14:00:00', '2023-11-25 20:00:00', 16, 3, 2
    ('King''s Cup', 1, 1, '2023-12-31 20:00:00', '2024-01-01 01:00:00', 10, 2, 10),
    ('Chess Tournament 11', 1, 0, '2023-05-01 14:00:00', '2023-05-01 18:00:00', 10, 2, N
    ('Friendly match', 0, 1, '2023-06-10 15:00:00', '2023-06-10 17:00:00', 8, 1, NULL),
    ('International Blitz', 1, 1, '2023-07-15 10:00:00', '2023-07-15 18:00:00', 16, 3, 3
    ('Grand Prix 2023', 1, 0, '2023-08-01 10:00:00', '2023-08-15 10:00:00', 30, 5, 2),
    ('Rapid Open 2023', 1, 1, '2023-09-01 11:00:00', '2023-09-01 16:00:00', 20, 2, NULL)
    ('Blitz Championship', 1, 0, '2023-10-15 12:00:00', '2023-10-15 16:00:00', 12, 1, 4)
    ('Team Cup', 0, 1, '2023-11-01 09:00:00', '2023-11-01 18:00:00', 32, 3, NULL),
    ('King of the Hill', 1, 1, '2024-01-01 14:00:00', '2024-01-01 16:00:00', 8, 1, NULL)
    ('World Cup', 1, 0, '2024-02-01 10:00:00', '2024-03-01 10:00:00', 64, 5, 1),
    ('Chess Olympiad 2024', 0, 1, '2024-07-01 10:00:00', '2024-07-15 10:00:00', 40, 3, N
```

```
 * mysql+pymysql://root:***@localhost/
20 rows affected.
```

Out[12]:
```
[]
```

In [13]:
```
%%sql
INSERT INTO Participates (tournament_id, player_name, ranking, score, performance)
VALUES
    (4, "zachary_song", 1, 3, 1700),
    (4, "DillyWeed",    2, 2, 1600),
    (4, "ethantan2509", 3, 1, 1400),
    (4, "Haloen",    4, 0, 1100),
    (4, "emily_williams", 5, 0, 1100),
    (4, "oxidyZe", 5, 0, 1100),
    (1, "Haloen", 1, 3, 1900),
    (1, "PICKLEPICKLEPICKLE", 2, 0, 1400),
    (1, "WisdomGuy", 2, 1, 1700),
    (5, "oxidyZe", NULL, NULL, NULL),
    (5, "coolingTundra", NULL, NULL, NULL),
    (5, "jane_doe", NULL, NULL, NULL),
    (5, "jim_smith", NULL, NULL, NULL),
    (5, "jessica_brown", NULL, NULL, NULL),
    (5, "bob_johnson", NULL, NULL, NULL),
    (5, "emily_williams", NULL, NULL, NULL),
    (5, "jason_brown", NULL, NULL, NULL),
    (5, "maria_garcia", NULL, NULL, NULL),
    (5, "sarah_taylor", NULL, NULL, NULL),
    (5, "daniel_kim", NULL, NULL, NULL),
    (5, "julie_lee", NULL, NULL, NULL),
    (5, "PICKLEPICKLEPICKLE", NULL, NULL, NULL),
    (5, "coolingTundra", NULL, NULL, NULL),
    (5, "WisdomGuy", NULL, NULL, NULL),
    (5, "DillyWeed", NULL, NULL, NULL),
    (5, "meower37", NULL, NULL, NULL);
```

```
 * mysql+pymysql://root:***@localhost/
26 rows affected.
```

Out[13]:

```
[]
```

In [14]:

```
%%sql
INSERT INTO Games(is_rated, is_public, datetime, initial_limit, increment, white_name, b
VALUES
    (1, 1, "2023-03-11 09:00:00", 10, 15, "zachary_song", "DillyWeed", 4, "1. e4 e5 2. N
    (1, 1, "2023-03-12 09:00:00", 10, 15, "DillyWeed", "ethantan2509", 4, "1. d4 d5 2. N
    (1, 0, "2023-03-14 12:30:00", 60, 30, "Haloen", "WisdomGuy", 1, "1. e4 c5 2. Nf3 d6 
    (1, 0, "2023-03-15 12:30:00", 60, 30, "Haloen", "PICKLEPICKLEPICKLE", 1, "1. d4 d5 2
```

```
 * mysql+pymysql://root:***@localhost/
4 rows affected.
```

Out[14]:

```
[]
```

In [15]:

```
# More Game Data
import mysql.connector
import random

# Not enough game data, importing them is very troublesome for both me and you.
# This set of 6 games, 2 draw, 2 wins, 2 losses will demostrate all I need to demo
possible_games = [
    "1. e4 e5 2. Nf3 Nc6 3. Bc4 h6 4. O-O Nf6 5. Nc3 d6 6. d3 a6 7. a4 b5 8. axb5 Nd4 9.
    "1. d4 d5 2. Nc3 Nf6 3. e4 dxe4 4. d5 c6 5. f3 exf3 6. gxf3 cxd5 7. Bb5+ Bd7 8. Nxd5
    "1. d4 Nf6 2. c4 c6 3. g3 d5 4. Nc3 e6 5. Nf3 Nbd7 6. c5 b6 7. b4 b5 8. Bg2 a5 9. Ba
    "1. e4 c5 2. Nf3 d6 3. Nc3 Nf6 4. Bc4 g6 5. a3 Bg7 6. d3 Nc6 7. Be3 O-O 8. Qe2 Bg4 9
    "1. e4 e5 2. Nf3 Nc6 3. Bc4 d6 4. O-O Nf6 5. Ng5 d5 6. exd5 Nxd5 7. Qf3 Qxg5 8. Qxd5
    "1. e4 e5 2. Nf3 Nc6 3. Bc4 d6 4. d3 Nf6 5. Be3 Be7 6. Ng5 O-O 7. Nc3 Ng4 8. O-O Nxe
    "1. e4 e5 2. Nf3 Nc6 3. Nc3 Nf6 4. d4 exd4 5. Nxd4 Nxd4 6. Qxd4 Qe7 7. Bg5 h6 8. Bxf
]

db = mysql.connector.connect(
    host="localhost",
    user="root",
    password="admin",
    database="chessible"
)

cur = db.cursor()

query = """
INSERT INTO Games(is_rated, is_public, datetime, initial_limit, increment, white_name, b
(1, 1, "2023-03-11 13:30:00", 10, 15, "ethantan2509", "zachary_song", 4, "1. d4 Nf6 2. c
"""
params = []

for j in range(2):
    random.shuffle(player_list)
    for i in range(10):
        query += f", (%s, %s, '2023-02-11 16:02:{11 + j}', 1, 2, %s, %s, NULL, %s)"
        params.append(random.randint(0, 1))
        params.append(random.randint(0, 1))
        params.append(player_list[i])
        params.append(player_list[-1-i])
        params.append(possible_games[random.randint(0, 5)])

cur.execute(query, tuple(params))
db.commit()
```

In [16]:

```
import pandas as pd
data = pd.read_csv('lichess_db_puzzle.csv', header=None, names=["PuzzleId","fen","moveli
cur.executemany(
    """INSERT INTO Puzzles(fen, movelist, rating) VALUES
    (%s, %s, %s)""",
    data.loc[:50, ['fen', 'movelist', 'rating']].values.tolist()
)
db.commit()
```

In [17]:

```
%%sql
INSERT INTO Applications(id, creator_name, approver_name, team_id, message)
VALUES
    (1, "KH-Brawl", NULL, 6, "Hello, Please Add Me"),
    (2, "KH-Brawl", NULL, 3, "Hello, Please Add Me"),
    (3, "KH-Brawl", NULL, 1, "Hello, Please Add Me"),
    (4, "KH-Brawl", NULL, 2, "Hello, Please Add Me"),
    (5, "KH-Brawl", NULL, 5, "Hello, Please Add Me"),
    (6, "KH-Brawl", NULL, 4, "Hello, Please Add Me"),
    (1, "ethantan2509", NULL, 6, "Hello, Please Add Me"),
    (2, "ethantan2509", NULL, 3, "Hello, Please Add Me"),
    (3, "ethantan2509", NULL, 4, "Hello, Please Add Me"),
    (4, "ethantan2509", NULL, 1, "Hello, Please Add Me"),
    (5, "ethantan2509", NULL, 5, "Hello, Please Add Me"),
    (6, "ethantan2509", NULL, 2, "Hello, Please Add Me"),
    (1, "zachary_song", NULL, 6, "Hello, Please Add Me"),
    (2, "zachary_song", NULL, 2, "Hello, Please Add Me"),
    (3, "zachary_song", NULL, 5, "Hello, Please Add Me"),
    (4, "zachary_song", NULL, 4, "Hello, Please Add Me"),
    (5, "zachary_song", NULL, 3, "Hello, Please Add Me"),
    (6, "zachary_song", NULL, 1, "Hello, Please Add Me");

-- Only players not in a team can apply
-- This is a very simple table, so I inserted lesser tables, the complexity is in the pr
```

```
 * mysql+pymysql://root:***@localhost/
18 rows affected.
0 rows affected.
```

Out[17]:

```
[]
```

In [18]:

```
%%sql
INSERT INTO Attempts
VALUES
    (1, "Haloen", 1, 60, "2023-11-22 18:26:54"),
    (1, "Haloen", 0, 32, "2023-11-22 18:44:15"),
    (4, "Haloen", 1, 24, "2023-11-22 18:01:13"),
    (7, "Haloen", 1, 18, "2023-11-22 18:10:08"),
    (8, "Haloen", 1, 46, "2023-11-22 18:33:25"),
    (1, "DillyWeed", 0, 32, "2023-11-22 18:44:15"),
    (7, "DillyWeed", 1, 18, "2023-11-22 18:10:08"),
    (1, "DillyWeed", 1, 60, "2023-11-22 18:26:54"),
    (8, "DillyWeed", 1, 46, "2023-11-22 18:33:25"),
    (2, "PICKLEPICKLEPICKLE", 1, 60, "2023-11-22 18:26:54"),
    (3, "PICKLEPICKLEPICKLE", 1, 46, "2023-11-22 18:33:25"),
    (4, "PICKLEPICKLEPICKLE", 0, 32, "2023-11-22 18:44:15"),
    (1, "coolingTundra", 1, 60, "2023-11-22 18:26:54"),
    (1, "coolingTundra", 0, 32, "2023-11-22 18:44:15"),
    (4, "coolingTundra", 1, 24, "2023-11-22 18:01:13"),
    (2, "coolingTundra", 0, 8, "2023-11-22 18:24:53"),
    (2, "DillyWeed", 0, 5, "2023-11-22 18:02:48"),
    (2, "meower37", 0, 102, "2023-11-22 18:49:14"),
    (4, "ethantan2509", 1, 16, "2023-11-22 18:05:21"),
    (4, "oxidyZe", 0, 1, "2023-11-22 18:04:57"),
    (6, "oxidyZe", 1, 94, "2023-11-22 18:34:35"),
    (8, "oxidyZe", 1, 37, "2023-11-22 18:19:50");
```

```
 * mysql+pymysql://root:***@localhost/
22 rows affected.
```

Out[18]:

```
[]
```

In [19]:

```
%%sql
-- Rating_History will be automaticlly updated with the triggers
-- Puzzle_History will be automaticlly updated with the triggers
```

```
 * mysql+pymysql://root:***@localhost/
0 rows affected.
```

Out[19]:

```
[]
```

Add in relevant select statements to show that your data is populated correctly FOR EACH relation, one cell each relation.

In [20]:

```
%%sql
SELECT * FROM Teams
```

```
 * mysql+pymysql://root:***@localhost/
20 rows affected.
```

Out[20]:

team_id	team_name	creation_date	about_team	icon	member_count
1	NUSH Black Knights	2022-02-15	The real dark horses of NUS High!	None	2
2	Pawn Stars	2022-01-05	We are a group of beginners who are passionate about chess and want to improve our skills together.	None	5
3	Team Alpha	2022-01-01	We are a group of chess enthusiasts who love playing the game and improving our skills.	None	4
4	The Bishops	2022-02-01	We are a team of players who focus on developing creative and unorthodox strategies in the game.	None	7
5	The Kings	2022-03-01	We are a team of players who are determined to dominate the game and be the undisputed champions.	None	4
6	The Chess Masters	2022-01-15	We are a team of highly skilled and experienced players who are constantly pushing the boundaries of the game.	None	1
7	The Chessaholics	2022-03-01	We are a team of players who are addicted to the game and can never get enough of it.	None	0
8	The Chess Wizards	2022-01-25	We are a team of players who believe in the magic of the game and enjoy exploring its infinite possibilities.	None	0
9	The Black and Whites	2022-02-05	We are a team of players who believe in the power of simplicity and efficiency in the game.	None	0
10	The Checkmates	2022-01-10	We are a team of players who enjoy the thrill of checkmating our opponents and ending the game in style.	None	0
11	The Rooks	2022-02-20	We are a team of competitive players who aim to be the best in the game.	None	0
12	The Rook-ies	2022-02-25	We are a team of beginners who are eager to learn and improve our skills in the game.	None	0
13	The Knights of Chess	2022-02-10	We are a team of players who aim to be versatile and adaptable in the game, capable of handling any situation.	None	0

14	Chess Wizards	2022-03-05	We are a group of experienced players who enjoy sharing our knowledge and skills with others.	None	0
15	The Knights Templar	2022-03-15	We are a team of players who are committed to playing the game with honor and integrity.	None	0
16	The Grandmasters	2021-12-01	We are a team of highly skilled players who have achieved the highest level of proficiency in the game.	None	0
17	The Chess Titans	2022-01-20	We are a team of players who aspire to be the best in the game and are constantly pushing ourselves to improve.	None	0
18	Queen's Gambits	2022-03-10	We are a group of female chess players who support and encourage each other in our pursuit of the game.	None	0
19	The Queens	2022-02-05	We are a team of female players who are passionate about chess and want to inspire more women to join the game.	None	0
20	Knight Movers	2021-12-15	We are a team of experienced players who enjoy challenging each other and learning new strategies.	None	0

In [21]:

```
%%sql
SELECT username, about_me, team_id, mod_start_date, join_date FROM Players
ORDER BY team_id, mod_start_date DESC;
```

```
 * mysql+pymysql://root:***@localhost/
26 rows affected.
```

Out[21]:

username	about_me	team_id	mod_start_date	join_date
ethantan2509	I love Chess	None	None	None
KH-Brawl	I love Chess	None	None	None
zachary_song	I love Chess	None	None	None
hnngggrrrr	I love Chess	1	2022-12-31 00:00:00	2019-01-01 00:00:00
jane_doe	Chess is my passion!	1	2021-01-01 00:00:00	2021-01-03 00:00:00
jessica_brown	Let's play some chess!	2	2021-01-01 00:00:00	2021-01-05 00:00:00
Passwordcringe	I love Chess	2	2018-06-25 00:00:00	2017-01-01 00:00:00
coolingTundra	I love Chess	2	None	2018-01-01 00:00:00
jim_smith	Looking for like-minded players!	2	None	2021-01-04 00:00:00
zetaNegativeOne	I love Chess	2	None	2021-01-01 00:00:00
LoverKent	I love Chess	3	2022-12-31 00:00:00	2018-01-01 00:00:00
bob_johnson	I'm a chess coach!	3	2021-01-01 00:00:00	2021-01-06 00:00:00
emily_williams	I'm new to chess, looking for guidance	3	None	2021-01-07 00:00:00
meower37	I love Chess	3	None	2023-01-01 00:00:00
DillyWeed	I love Chess	4	2023-01-01 00:00:00	2017-01-01 00:00:00
Haloen	I love Chess	4	2021-12-31 00:00:00	2020-01-01 00:00:00
jason_brown	I'm a chess veteran!	4	2021-01-01 00:00:00	2021-01-08 00:00:00
maria_garcia	I'm a fan of positional chess!	4	2021-01-01 00:00:00	2021-01-09 00:00:00
jsck413	I love Chess	4	None	2020-01-01 00:00:00
PICKLEPICKLEPICKLE	I love Chess	4	None	2019-01-01 00:00:00

WisdomGuy	I love Chess	4	None	2022-01-01 00:00:00
drewmtr	I love Chess	5	2022-12-31 00:00:00	2019-01-01 00:00:00
sarah_taylor	I love to play blitz chess!	5	2021-01-01 00:00:00	2021-01-10 00:00:00
daniel_kim	I'm an up-and-coming chess player!	5	None	2021-01-11 00:00:00
oxidyZe	I love Chess	5	None	2019-01-01 00:00:00
julie_lee	Let's play some chess!	6	2021-01-01 00:00:00	2021-01-12 00:00:00

In [22]:

```
%%sql
SELECT tournament_id ID, tournament_name Name, IF(is_rated, "TRUE", "FALSE") "rated?", I
FROM Tournaments
```

```
 * mysql+pymysql://root:***@localhost/
20 rows affected.
```

Out[22]:

ID	Name	rated?	public?	start_datetime	end_datetime	initial_limit	increment	host_team_id
1	Spring Blitz Tournament	TRUE	TRUE	2023-04-22 12:00:00	2023-04-22 16:00:00	5	2	8
2	Summer Rapid Tournament	FALSE	TRUE	2023-07-15 14:00:00	2023-07-15 18:00:00	6	3	4
3	Fall Classic Tournament	TRUE	FALSE	2023-09-23 10:00:00	2023-09-24 17:00:00	10	5	9
4	Winter Warm-Up Tournament	FALSE	TRUE	2023-12-02 13:00:00	2023-12-02 17:00:00	8	2	1
5	Beginner's Cup	FALSE	TRUE	2023-04-30 11:00:00	2023-04-30 13:00:00	3	1	5
6	Intermediate Invitational	TRUE	TRUE	2023-05-12 15:00:00	2023-05-12 19:00:00	8	3	3
7	Advanced Arena	TRUE	FALSE	2023-08-08 12:00:00	2023-08-08 18:00:00	12	4	11
8	International Open	TRUE	TRUE	2023-10-14 10:00:00	2023-10-15 17:00:00	20	5	6
9	Queen's Gambit Cup	TRUE	FALSE	2023-11-25 14:00:00	2023-11-25 20:00:00	16	3	2
10	King's Cup	TRUE	TRUE	2023-12-31 20:00:00	2024-01-01 01:00:00	10	2	10
11	Chess Tournament 11	TRUE	FALSE	2023-05-01 14:00:00	2023-05-01 18:00:00	10	2	None
12	Friendly match	FALSE	TRUE	2023-06-10 15:00:00	2023-06-10 17:00:00	8	1	None
13	International Blitz	TRUE	TRUE	2023-07-15 10:00:00	2023-07-15 18:00:00	16	3	3
14	Grand Prix 2023	TRUE	FALSE	2023-08-01 10:00:00	2023-08-15 10:00:00	30	5	2
15	Rapid Open 2023	TRUE	TRUE	2023-09-01 11:00:00	2023-09-01 16:00:00	20	2	None
16	Blitz Championship	TRUE	FALSE	2023-10-15 12:00:00	2023-10-15 16:00:00	12	1	4
17	Team Cup	FALSE	TRUE	2023-11-01 09:00:00	2023-11-01 18:00:00	32	3	None

18	King of the Hill	TRUE	TRUE	2024-01-01 14:00:00	2024-01-01 16:00:00	8	1	None
19	World Cup	TRUE	FALSE	2024-02-01 10:00:00	2024-03-01 10:00:00	64	5	1
20	Chess Olympiad 2024	FALSE	TRUE	2024-07-01 10:00:00	2024-07-15 10:00:00	40	3	None

In [23]:

```
%%sql
SELECT * FROM Puzzles
```

```
 * mysql+pymysql://root:***@localhost/
51 rows affected.
```

Out[23]:

puzzle_id	movelist	fen	rating	solve_count
1	f2g3 e6e7 b2b1 b3c1 b1c1 h6c1	r6k/pp2r2p/4Rp1Q/3p4/8/1N1P2R1/PqP2bPP/7K b - - 0 24	1789	3
2	d3d6 f8d8 d6d8 f6d8	5rk1/1p3ppp/pq3b2/8/8/1P1Q1N2/P4PPP/3R2K1 w - - 2 27	1482	1
3	b6c5 e2g4 h3g4 d1g4	r2qr1k1/b1p2ppp/pp4n1/P1P1p3/4P1n1/B2P2Pb/3NBP1P/RN1QR1K1 b - - 1 16	1172	1
4	g4h5 g6h5 f4f5 e6e5 f5f6 e5f6	8/8/4k1p1/2KpP2p/5PP1/8/8/8 w - - 0 53	1565	3
5	e5f6 e8e1 g1f2 e1f1	4r3/1k6/pp3r2/1b2P2p/3R1p2/P1R2P2/1P4PP/6K1 w - - 0 35	1533	None
6	g5e7 a5c3 b2c3 c6e7	r4rk1/pp3ppp/2n1b3/q1pp2B1/8/P1Q2NP1/1PP1PP1P/2KR3R w - - 0 15	1430	1
7	e8f7 e2e6 f7f8 e6f7	r1bqk2r/pp1nbNp1/2p1p2p/8/2BP4/1PN3P1/P3QP1P/3R1RK1 b kq - 0 19	1554	2
8	e3g3 f7f4 e5f4 f8f4	5r1k/5rp1/p7/1b2B2p/1P1P1Pq1/2R1Q3/P3p1P1/2R3K1 w - - 0 41	1961	3
9	a6a5 e5c7 a5b4 c7d8	3R4/8/K7/pB2b3/1p6/1P2k3/3p4/8 w - - 4 58	1039	None
10	d4b6 f6e4 h1g1 e4f2	4r3/5pk1/1p3np1/3p3p/2qQ4/P4N1P/1P3RP1/7K w - - 6 34	2051	None
11	d8f6 d1h5 h7h6 h5c5	r2q1rk1/5ppp/1np5/p1b5/2p1B3/P7/1P3PPP/R1BQ1RK1 b - - 1 17	1800	None
12	b7b6 d4a1 a7a5 f1c1	2kr3r/pp3p2/4p2p/1N1p2p1/3Q4/1P1P4/2q2PPP/5RK1 b - - 1 20	2578	None
13	e8e5 d1d8 e5e8 d8e8	4r1k1/5ppp/r1p5/p1n1RP2/8/2P2N1P/2P3P1/3R2K1 b - - 0 21	1128	None
14	b8c7 b1b7 c7b7 e4f6 e7f6 g2b7	1qr2rk1/pb2bppp/8/8/2p1N3/P1Bn2P1/2Q2PBP/1R3RK1 b - - 3 23	1648	None
15	c7a8 a5c3	r6r/1pNk1ppp/2np4/b3p3/4P1b1/N1Q5/P4PPP/R3KB1R w KQ - 3 18	885	None
16	c4e2 h4h7 c7h7 c1c8 g8g7 c8c7	2r3k1/2r4p/4p1p1/1p1q1pP1/p1bP1P1Q/P6R/5B2/2R3K1 b - - 5 34	1869	None
17	g4h4 c5f2 g2g3 f2g3	5r1k/pp4pp/5p2/1BbQp1r1/6K1/7P/1PP3P1/3R3R w - - 2 26	991	None
18	c7b6 a6c8	2r3k1/p1q2pp1/Q3p2p/b1Np4/2nP1P2/4P1P1/5K1P/2B1N3 b - - 3 33	2175	None

	g8h7 c8b7			
19	f8f7 c2h7 g8h7 g7g8q	1rb2rk1/q5P1/4p2p/3p3p/3P1P2/2P5/2QK3P/3R2R1 b - - 0 29	1362	None
20	h2g3 g6h5	r3k2r/pb1p1ppp/1b4q1/1Q2P3/8/2NP1Pn1/PP4PP/R1B2R1K w kq - 1 17	1339	None
21	f2f3 d6c5 g1h1 c5c4	r4rk1/p3ppbp/Pp1q1np1/3PpbB1/2B5/2N5/1PPQ1PPP/3RR1K1 w - - 4 18	954	None
22	d3a6 b6c5 a6c8 c5c4	k1r1b3/p1r1nppp/1p1qpn2/2Np4/1P1P4/PQRBPN2/5PPP/2R3K1 w - - 0 19	1870	None
23	g7f6 e7f7 f6e5 f7f4	8/4R1k1/p5pp/3B4/5q2/8/5P1P/6K1 b - - 5 40	1186	None
24	b1c3 f5g4 h5g4 d7g4	r3kb1r/pppqpn1p/5p2/3p1bpQ/2PP4/4P1B1/PP3PPP/RN2KB1R w KQkq - 1 11	1535	None
25	f7g7 e4e5 f6g6 g2g6	r7/2p2r1k/p2p1q1p/Pp1P4/1P2P3/2PQ4/6R1/R5K1 b - - 2 28	1162	None
26	c6a4 g5d2	8/8/kpq5/p4pQp/P7/7P/3r2P1/4R2K b - - 10 48	974	None
27	c6e5 d4e5 d6e7 e5f6	r3brk1/5pp1/p1nqpn1p/P2pN3/2pP4/2P1PN2/5PPP/RB1QK2R b KQ - 4 16	1230	None
28	f8e7 f3e5 c7e5 e2g4	r3kb1r/ppq2ppp/4pn2/2Ppn3/1P4bP/2P2N2/P3BPP1/RNBQ1RK1 b kq - 2 10	1618	None
29	d7e5 f5f6 e5f3 f1f3	3r1rk1/1b1n1pp1/3p4/p4PPQ/4P3/3q1BN1/8/2R2RK1 b - - 1 28	2489	None
30	f5g4 e5g4 f6g4 d1g4	r3qrk1/2p2pp1/p2bpn1p/2ppNb2/3P1P2/1PP1P1B1/P2N2PP/R2Q1RK1 b - - 0 14	1077	None
31	f1b5 e7b4 b1c3 b4b2	r3kbnr/ppp1qppp/2n5/3pP3/5B2/4PQ2/PPP2PPP/RN2KB1R w KQkq - 1 7	1575	None
32	g1h1 f6f4 d1f3 f4f3	r4rk1/pp3ppp/3p1q2/P1P1p3/2B5/2B2n2/2P2P1P/R2Q1RK1 w - - 0 16	1751	None
33	e6d5 d4f6 g5f5 c6d5	8/6p1/2B1bn2/6k1/3B4/6K1/4P3/8 b - - 4 44	1432	None
34	d7e6 e2f4 c5d4 f4g6 f7g6 h5h6	r6k/q1pb1p1p/1b3Pr1/p1ppP2Q/3P2p1/4B3/PP2NRPP/3R2K1 b - - 1 25	1688	None
35	d3f5 e6f5 e1e7 c6e7	r2r2k1/1p2qppp/2n1p3/5n2/p2P4/P2Q1N2/BP3PPP/2R1R1K1 w - - 4 20	1615	None
36	g6h4 c5b5 h4g2 b5a6 g2e3 a6b7	2nk4/8/2PBp1n1/1pK1P1p1/1P4P1/8/8/8 b - - 2 42	2074	None
37	d4a4 b8b1 e1f2 b1h1 a4a7 e7f8	1r5r/p3kp2/4p2p/4P3/3R1Pp1/6P1/P1P4P/4K2R w K - 1 25	1156	None
38	f4f5 e4g3	8/3pk3/R7/1R2Pp1p/2PPnKr1/8/8/8 w - - 4 43	1686	None
39	d4a7 e7g5 c1b1 g5h6	6k1/3bqr1p/2rpp1pR/p7/Pp1QP3/1B3P2/1PP3P1/2KR4 w - - 6 22	1041	None
40	e1e4 f3d2 b1a1 c8c1	2r5/pR5p/5p1k/4p3/4r3/B4nPP/PP3P2/1K2R3 w - - 0 27	1601	None
41	d8f6 d4e5 d6e5 c1g5 f6d6 f3e5 d6d1 f1d1	r1bqk1nr/1pp2ppp/p1pb4/4p3/3PP3/5N2/PPP2PPP/RNBQ1RK1 b kq - 0 6	1729	None
42	h7g8 h1h8	8/r1b1q2k/2p3p1/2Pp4/1P2p1n1/2B1P3/NQ6/2K4R b - - 1 36	1096	None

	g8f7 h8h7 f7e8 h7e7			
43	f3f2 e5d3 f2e3 d3f4 h1h5 f4h5	rn3rk1/p5pp/3N4/4np1q/5Q2/1P3K2/PB1P2P1/2R4R w - - 0 25	1098	None
44	c6c5 d4a4 a6b5 a4b5	r3kb1r/p4ppp/b1p1p3/3q4/3Q4/4BN2/PPP2PPP/R3K2R b KQkq - 0 11	1001	None
45	b8h8 d6f7 e5e4 f7h8	1R6/1p2k2p/p2n2p1/4K3/8/6P1/P6P/8 w - - 10 37	1008	None
46	f1e1 g3e4 e1e4 d5e4 d1h5 h4h5 b3f7 h5f7	r1b1k2N/ppp3pp/2n5/2bp4/7q/1B4n1/PPPP1P1P/RNBQ1RK1 w q - 0 10	2152	None
47	f1g1 f5g3 f2g3 g8g3	6rk/pp6/2n5/3ppn1p/3p4/2P2P1q/PP3QNB/R4R1K w - - 2 29	1307	None
48	g1f3 d8a5 d1d2 a5b5	r1bqk2r/pp1nbppp/3p4/1B1p4/3P1B2/8/PPP2PPP/R2QK1NR w KQkq - 2 9	1076	None
49	d5e6 b4d3 f2g1 d3f4	r2qr1k1/ppp2ppp/4b3/3P4/1nP2Q2/2N2N1P/PP3KP1/R4R2 w - - 1 15	1111	None
50	a1f1 f6d6 f4d6 f8f1	2r2rk1/6pp/3Q1q2/8/3N1B2/6P1/PP1K3P/R4b2 w - - 0 24	1973	None
51	d6e5 d2d8 b6d8 d1d8	3r2k1/4nppp/pq1p1b2/1p2P3/2r2P2/2P1NR2/PP1Q2BP/3R2K1 b - - 0 24	640	None

In [24]:

```
%%sql
SELECT puzzle_id, player_name, IF(is_correct, "TRUE", "FALSE") "correct?", time_taken
FROM Attempts
```

```
 * mysql+pymysql://root:***@localhost/
22 rows affected.
```

Out[24]:

puzzle_id	player_name	correct?	time_taken
1	Haloen	TRUE	60
1	Haloen	FALSE	32
4	Haloen	TRUE	24
7	Haloen	TRUE	18
8	Haloen	TRUE	46
1	DillyWeed	FALSE	32
7	DillyWeed	TRUE	18
1	DillyWeed	TRUE	60
8	DillyWeed	TRUE	46
2	PICKLEPICKLEPICKLE	TRUE	60
3	PICKLEPICKLEPICKLE	TRUE	46
4	PICKLEPICKLEPICKLE	FALSE	32
1	coolingTundra	TRUE	60
1	coolingTundra	FALSE	32
4	coolingTundra	TRUE	24
2	coolingTundra	FALSE	8

2	DillyWeed	FALSE	5
2	meower37	FALSE	102
4	ethantan2509	TRUE	16
4	oxidyZe	FALSE	1
6	oxidyZe	TRUE	94
8	oxidyZe	TRUE	37

In [25]:

```
%%sql
-- Hid the PGN because its super big
SELECT game_id, IF(is_rated, "TRUE", "FALSE") "rated?", IF(is_public, "TRUE", "FALSE") "
FROM Games;
```

```
 * mysql+pymysql://root:***@localhost/
25 rows affected.
```

Out[25]:

game_id	rated?	public?	datetime	initial_limit	increment	white_name	black_name	tournament_i
1	TRUE	TRUE	2023-03-11 09:00:00	10	15	zachary_song	DillyWeed	
2	TRUE	TRUE	2023-03-12 09:00:00	10	15	DillyWeed	ethantan2509	
3	TRUE	FALSE	2023-03-14 12:30:00	60	30	Haloen	WisdomGuy	
4	TRUE	FALSE	2023-03-15 12:30:00	60	30	Haloen	PICKLEPICKLEPICKLE	
5	TRUE	TRUE	2023-03-11 13:30:00	10	15	ethantan2509	zachary_song	
6	FALSE	TRUE	2023-02-11 16:02:11	1	2	drewmtr	KH-Brawl	Non
7	FALSE	FALSE	2023-02-11 16:02:11	1	2	sarah_taylor	jessica_brown	Non
8	TRUE	FALSE	2023-02-11 16:02:11	1	2	LoverKent	ethantan2509	Non
9	FALSE	FALSE	2023-02-11 16:02:11	1	2	jim_smith	jane_doe	Non
10	TRUE	TRUE	2023-02-11 16:02:11	1	2	bob_johnson	oxidyZe	Non
11	FALSE	TRUE	2023-02-11 16:02:11	1	2	Haloen	jason_brown	Non
12	TRUE	FALSE	2023-02-11 16:02:11	1	2	zetaNegativeOne	julie_lee	Non

13	TRUE	TRUE	2023-02-11 16:02:11	1	2	WisdomGuy	DillyWeed	Non
14	TRUE	TRUE	2023-02-11 16:02:11	1	2	emily_williams	coolingTundra	Non
15	TRUE	TRUE	2023-02-11 16:02:11	1	2	PICKLEPICKLEPICKLE	maria_garcia	Non
16	TRUE	FALSE	2023-02-11 16:02:12	1	2	jsck413	drewmtr	Non
17	TRUE	FALSE	2023-02-11 16:02:12	1	2	KH-Brawl	PICKLEPICKLEPICKLE	Non
18	TRUE	FALSE	2023-02-11 16:02:12	1	2	daniel_kim	Haloen	Non
19	TRUE	TRUE	2023-02-11 16:02:12	1	2	DillyWeed	Passwordcringe	Non
20	FALSE	FALSE	2023-02-11 16:02:12	1	2	jim_smith	julie_lee	Non
21	FALSE	TRUE	2023-02-11 16:02:12	1	2	LoverKent	oxidyZe	Non
22	FALSE	FALSE	2023-02-11 16:02:12	1	2	jason_brown	coolingTundra	Non
23	FALSE	TRUE	2023-02-11 16:02:12	1	2	WisdomGuy	emily_williams	Non
24	TRUE	TRUE	2023-02-11 16:02:12	1	2	maria_garcia	bob_johnson	Non
25	TRUE	FALSE	2023-02-11 16:02:12	1	2	zetaNegativeOne	jessica_brown	Non

In [26]:

```
%%sql
SELECT * FROM Puzzle_History
ORDER BY username, datetime;
```

```
 * mysql+pymysql://root:***@localhost/
48 rows affected.
```

Out[26]:

username	datetime	rating
bob_johnson	2016-01-01 00:00:00	1200
coolingTundra	2016-01-01 00:00:00	1200
coolingTundra	2023-11-22 18:01:13	1258
coolingTundra	2023-11-22 18:24:53	1224
coolingTundra	2023-11-22 18:26:54	1231

coolingTundra	2023-11-22 18:44:15	1230
daniel_kim	2016-01-01 00:00:00	1200
DillyWeed	2016-01-01 00:00:00	1200
DillyWeed	2023-11-22 18:02:48	1194
DillyWeed	2023-11-22 18:10:08	1227
DillyWeed	2023-11-22 18:26:54	1230
DillyWeed	2023-11-22 18:33:25	1231
DillyWeed	2023-11-22 18:44:15	1199
drewmtr	2016-01-01 00:00:00	1200
emily_williams	2016-01-01 00:00:00	1200
ethantan2509	2016-01-01 00:00:00	1200
ethantan2509	2023-11-22 18:05:21	1229
Haloen	2016-01-01 00:00:00	1200
Haloen	2023-11-22 18:01:13	1258
Haloen	2023-11-22 18:10:08	1258
Haloen	2023-11-22 18:26:54	1231
Haloen	2023-11-22 18:33:25	1262
Haloen	2023-11-22 18:44:15	1230
hnngggrrrr	2016-01-01 00:00:00	1200
jane_doe	2016-01-01 00:00:00	1200
jason_brown	2016-01-01 00:00:00	1200
jessica_brown	2016-01-01 00:00:00	1200
jim_smith	2016-01-01 00:00:00	1200
jsck413	2016-01-01 00:00:00	1200
julie_lee	2016-01-01 00:00:00	1200
KH-Brawl	2016-01-01 00:00:00	1200
LoverKent	2016-01-01 00:00:00	1200
maria_garcia	2016-01-01 00:00:00	1200
meower37	2016-01-01 00:00:00	1200
meower37	2023-11-22 18:49:14	1195
oxidyZe	2016-01-01 00:00:00	1200
oxidyZe	2023-11-22 18:04:57	1197
oxidyZe	2023-11-22 18:19:50	1254
oxidyZe	2023-11-22 18:34:35	1222
Passwordcringe	2016-01-01 00:00:00	1200
PICKLEPICKLEPICKLE	2016-01-01 00:00:00	1200
PICKLEPICKLEPICKLE	2023-11-22 18:26:54	1227

PICKLEPICKLEPICKLE	2023-11-22 18:33:25	1240
PICKLEPICKLEPICKLE	2023-11-22 18:44:15	1236
sarah_taylor	2016-01-01 00:00:00	1200
WisdomGuy	2016-01-01 00:00:00	1200
zachary_song	2016-01-01 00:00:00	1200
zetaNegativeOne	2016-01-01 00:00:00	1200

In [27]:

```
%%sql
SELECT * FROM Rating_History
ORDER BY datetime;
```

```
 * mysql+pymysql://root:***@localhost/
60 rows affected.
```

Out[27]:

username	datetime	rating
bob_johnson	2016-01-01 00:00:00	1200
jane_doe	2016-01-01 00:00:00	1200
zachary_song	2016-01-01 00:00:00	1200
coolingTundra	2016-01-01 00:00:00	1200
jason_brown	2016-01-01 00:00:00	1200
daniel_kim	2016-01-01 00:00:00	1200
WisdomGuy	2016-01-01 00:00:00	1200
DillyWeed	2016-01-01 00:00:00	1200
jessica_brown	2016-01-01 00:00:00	1200
sarah_taylor	2016-01-01 00:00:00	1200
PICKLEPICKLEPICKLE	2016-01-01 00:00:00	1200
Passwordcringe	2016-01-01 00:00:00	1200
drewmtr	2016-01-01 00:00:00	1200
oxidyZe	2016-01-01 00:00:00	1200
emily_williams	2016-01-01 00:00:00	1200
julie_lee	2016-01-01 00:00:00	1200
ethantan2509	2016-01-01 00:00:00	1200
jim_smith	2016-01-01 00:00:00	1200
meower37	2016-01-01 00:00:00	1200
maria_garcia	2016-01-01 00:00:00	1200
Haloen	2016-01-01 00:00:00	1200
LoverKent	2016-01-01 00:00:00	1200
KH-Brawl	2016-01-01 00:00:00	1200
jsck413	2016-01-01 00:00:00	1200
hnngggrrrr	2016-01-01 00:00:00	1200
zetaNegativeOne	2016-01-01 00:00:00	1200
coolingTundra	2023-02-11 16:02:11	1200

DillyWeed	2023-02-11 16:02:11	1186
emily_williams	2023-02-11 16:02:11	1200
ethantan2509	2023-02-11 16:02:11	1199
LoverKent	2023-02-11 16:02:11	1200
julie_lee	2023-02-11 16:02:11	1184
maria_garcia	2023-02-11 16:02:11	1215
oxidyZe	2023-02-11 16:02:11	1184
WisdomGuy	2023-02-11 16:02:11	1199
PICKLEPICKLEPICKLE	2023-02-11 16:02:11	1169
bob_johnson	2023-02-11 16:02:11	1216
zetaNegativeOne	2023-02-11 16:02:11	1216
jessica_brown	2023-02-11 16:02:12	1217
maria_garcia	2023-02-11 16:02:12	1215
drewmtr	2023-02-11 16:02:12	1200
jsck413	2023-02-11 16:02:12	1200
KH-Brawl	2023-02-11 16:02:12	1199
PICKLEPICKLEPICKLE	2023-02-11 16:02:12	1185
DillyWeed	2023-02-11 16:02:12	1170
daniel_kim	2023-02-11 16:02:12	1201
Passwordcringe	2023-02-11 16:02:12	1215
bob_johnson	2023-02-11 16:02:12	1216
Haloen	2023-02-11 16:02:12	1215
zetaNegativeOne	2023-02-11 16:02:12	1199
DillyWeed	2023-03-11 09:00:00	1184
zachary_song	2023-03-11 09:00:00	1216
ethantan2509	2023-03-11 13:30:00	1200
zachary_song	2023-03-11 13:30:00	1215
ethantan2509	2023-03-12 09:00:00	1199
DillyWeed	2023-03-12 09:00:00	1185
Haloen	2023-03-14 12:30:00	1200
WisdomGuy	2023-03-14 12:30:00	1200
PICKLEPICKLEPICKLE	2023-03-15 12:30:00	1184
Haloen	2023-03-15 12:30:00	1216

In [28]:

```
%%sql
SELECT * FROM Applications;
```

```
 * mysql+pymysql://root:***@localhost/
18 rows affected.
```

Out[28]:

id	message	creator_name	approver_name	team_id

1	Hello, Please Add Me	ethantan2509	None	6
1	Hello, Please Add Me	KH-Brawl	None	6
1	Hello, Please Add Me	zachary_song	None	6
2	Hello, Please Add Me	ethantan2509	None	3
2	Hello, Please Add Me	KH-Brawl	None	3
2	Hello, Please Add Me	zachary_song	None	2
3	Hello, Please Add Me	ethantan2509	None	4
3	Hello, Please Add Me	KH-Brawl	None	1
3	Hello, Please Add Me	zachary_song	None	5
4	Hello, Please Add Me	ethantan2509	None	1
4	Hello, Please Add Me	KH-Brawl	None	2
4	Hello, Please Add Me	zachary_song	None	4
5	Hello, Please Add Me	ethantan2509	None	5
5	Hello, Please Add Me	KH-Brawl	None	5
5	Hello, Please Add Me	zachary_song	None	3
6	Hello, Please Add Me	ethantan2509	None	2
6	Hello, Please Add Me	KH-Brawl	None	4
6	Hello, Please Add Me	zachary_song	None	1

Section F: Stored Program & Queries Script

Query 1

Find the rating range of the participants for each tournament.
(Rating is found by taking the latest rating in the history of each participant)

In [29]:

```
%%sql
SELECT Tournaments.tournament_name Name, min(rating), max(rating)
FROM (
        SELECT username, MAX(datetime) datetime
        FROM Rating_history
        GROUP BY username
    ) latest_date
    NATURAL JOIN Rating_history AS Rating
    JOIN Participates ON Participates.player_name = Rating.username
    RIGHT JOIN Tournaments ON Participates.tournament_id = Tournaments.tournament_id
GROUP BY Tournaments.tournament_id, Tournaments.tournament_name;
```

```
 * mysql+pymysql://root:***@localhost/
20 rows affected.
```

Out[29]:

Name	min(rating)	max(rating)
Advanced Arena	None	None
Beginner's Cup	1184	1217
Blitz Championship	None	None
Chess Olympiad 2024	None	None

Chess Tournament 11	None	None
Fall Classic Tournament	None	None
Friendly match	None	None
Grand Prix 2023	None	None
Intermediate Invitational	None	None
International Blitz	None	None
International Open	None	None
King of the Hill	None	None
King's Cup	None	None
Queen's Gambit Cup	None	None
Rapid Open 2023	None	None
Spring Blitz Tournament	1184	1216
Summer Rapid Tournament	None	None
Team Cup	None	None
Winter Warm-Up Tournament	1184	1216
World Cup	None	None

Query 2

For each tournament, count the number of wins in the tournament the top player has so far.

In [30]:

```
%%sql
SELECT tournament_name, player_name, (
    SELECT COUNT(*) FROM Games
    WHERE top.tournament_id = Games.tournament_id AND (player_name = white_name AND pgn
) "Win Count"
FROM (
        SELECT Tournaments.tournament_id, tournament_name, player_name
        FROM Tournaments LEFT JOIN Participates ON Tournaments.tournament_id = Participa
        WHERE ranking = 1
    ) top
```

```
 * mysql+pymysql://root:***@localhost/
2 rows affected.
```

Out[30]:

tournament_name	player_name	Win Count
Winter Warm-Up Tournament	zachary_song	1
Spring Blitz Tournament	Haloen	1

Query 3

For each player, find the highest rating among all the opponents the player has beaten.
The rating used is the rating right before the game (ie the rating during the game).
In the case of a tie, use the latest game which they won.

In [31]:

```
%%sql
SELECT username, max(loser_rating) Best_Win
FROM (
        SELECT p1.username winner, p2.username loser,
        (SELECT rating FROM Rating_History hist WHERE hist.username = p2.username AND hi

        FROM Players p1, Players p2, Games
        WHERE
            (p1.username = white_name AND p2.username = black_name AND pgn LIKE "%1-0")
            (p2.username = white_name AND p1.username = black_name AND pgn LIKE "%0-1")

    ) won_games RIGHT JOIN Players ON won_games.winner = Players.username
GROUP BY username
ORDER BY Best_Win DESC;
```

```
 * mysql+pymysql://root:***@localhost/
26 rows affected.
```

Out[31]:

username	Best_Win
jessica_brown	1216
bob_johnson	1200
jason_brown	1200
KH-Brawl	1200
maria_garcia	1200
zetaNegativeOne	1200
emily_williams	1199
Passwordcringe	1186
Haloen	1185
zachary_song	1170
coolingTundra	None
daniel_kim	None
DillyWeed	None
drewmtr	None
ethantan2509	None
hnngggrrrr	None
jane_doe	None
jim_smith	None
jsck413	None
julie_lee	None
LoverKent	None
meower37	None
oxidyZe	None
PICKLEPICKLEPICKLE	None
sarah_taylor	None
WisdomGuy	None

Section G: Web UI

Instructions:
For this deliverable you will write a web application that interacts with your database to manage your domain. This application must allow the user to extract specific information from the database, through a user-friendly interface. Additional marks will be given for good webpage design (in terms of navigation, organization and functionality), and aesthetically pleasing webpage. Your web interface should allow you to demonstrate the CRUD operations:
* **User Info Page:**
- Allow user to register for a new account - Allow registered user to login - Allow logged in users to view and edit Profile / Account information
Other notes: - Use sessions to ensure only logged in user may access relevant information of to their account. - Relevant data validation should be done. * **Search & Browse page (i.e. Read function):**
Upon login, users can search and browse the “data”. Searching is likely the most typical action for a user. The user should be presented with a form to specify their search criteria, and based on the results of the underlying database query, will be presented either a list of matching records or a single matching record if only one was found.
It is not necessary to allow user to search for all tables (and they shouldn't be allowed to!). Select a few tables where the search & browse function make sense. You are advised to implement the queries shortlisted in Section F where applicable. * **Pages to demo Create, Update & Delete functions:**
Users should be able to insert, edit and delete their entries! Recommended to just focus on 2 each. Note that you will need to upload ALL source code for the Web UI for this section. You do not need to screen capture every page, but it should demonstrate that you have done all CRUD functions. Note that Login, Register and Profile Edit is NOT sufficient to demonstrate CRUD as it has been guided in ISSL. You should demonstrate CRUD on other tables based on shortlisted purpose of your webfrontend.

CRUDS implemented

Create: `INSERT` statements	Allow players to create an account (Register) Allow players to create Teams Allow players to create Applications to Teams
Read: `SELECT` statements	Search for a Player Search for a Team Show all applications for a team Show all applications for a player Show a Player's Profile
Update: `UPDATE` statements	Players can update their about (Profile Edit) Players can edit their team info Players can update other Players to be moderators
Delete: `DELETE` statements	Users can delete their account Users can delete applications (rejecting or withdrawing) Users can delete their team (by leaving it with no one, doesn't count originally empty teams) Players can kick other Players from the team

Attached relevant image of your FINAL web interface below.

Read_Profile.png Read_Team.png Read_Teams.png Update_Application_Team.png Update_Mod_Status.png Update_Profile.png

Section H: Project Reflection

Write a 1 page reflection here. You may reflect on the following points:

* What insights have you gained after completing this project? * How has completing the project affected your view of database systems? * How do you think this project experience would be useful to you in future? * How do you think you have managed your time for this project? Has the help provided in class been sufficient?

Through this project, I was able to realise how time consuming these kinds of projects are. When I first started web development, I was breezing through all the projects that required me to work with website development, but that was only because I was working solely on front end development. This has allowed me to gain a deeper understanding of the full-stack development workflow, and how tedious it can be just to make a decent website. Since this is my first experience working with a backend database server, I did not foresee how quickly the project will grow. Due to that and my previous web development experience, I did not allocate as much time as I should on the Chess App, especially considering my scope to include networking elements for online multiplayer. Creating a web app with a database can be a complex and time-consuming task, so it's important to allocate sufficient time for planning, design, development, and testing. The help provided in class can be useful for getting started and troubleshooting issues, but ultimately it's up to the developer to manage their time and work efficiently.

Through this project, I also realised I was allocating time in the wrong locations in time management skills. During class, I worked on tiny useless features such as notifications, dark-light theme changing, and web responsiveness. All these little features snowballed, and made my unable to fully complete the project on time. I should have focused on the bigger more pertinent features such as tournament management and user management. In the future, I will try to keep a TODO list so that I can ensure that I will prioritise the more important features instead of getting distracted by the "side quests".

I have also learnt how useful SQL is, especially triggers, which allowed me to completely ignore certain tables as they are handled by the SQL triggers. One example is the Rating Table, which will insert an initial rating, and consequently update for every other game played. These SQL triggers all work in perfect harmony to keep the interactions between the database (to retrieve information necessary to make decisions), and speed up the development cycle of the app.

However, I also realise the downsides of SQL through this project. Whenever I realised I needed additional attributes, I need to look through the DDL, add it, then add for every data row that I have. This is extremely tedious, and especially since I am hosting the website, it would be even harder, as the database is remote and stores way more data. While there are functions to ease this process, it is still ultimately tedious.

Overall, seeing my website work, to a certain extent, is very fulfilling, but it is also disappointing that I still have to submit a half working project late.