***Final Project: Final Draft***

***Introduction:***

SIMPLE DESCRIPTION AND MOTIVATION:

I created a database model to represent the professional chess world, specifically online chess on Chess.com. Using real-world data to complete the tables, I used queries to identify any noteworthy anomalies in the data as well as investigate other items of interest.

I chose this topic for two reasons. One of my friends during my undergraduate program convinced me to begin playing chess online during the COVID-19 pandemic; online chess quickly became one of my favorite hobbies, making chess a natural topic for my project.

The second reason is the prevalence of drama surrounding cheating in chess lately. With allegations against players being levied frequently and suspicions of the problem being more pervasive than previously known, delving into the relevant data has never been more important.

BACKGROUND INFORMATION:

The most popular platform that players use to play chess online is Chess.com. There are several different time controls available on Chess.com: the players are each given two minutes or less in ‘bullet’ games, three to five minutes for blitz games, and ten or more minutes for rapid games. Daily games are also played, with players being given a day or more to make each move.

Players are also assigned ratings to assess their skill level. These ratings differ by time class. For example, an account may have a 2500 rating in bullet chess, but only a 2200 in blitz. These ratings change over time according to players’ performances on their accounts.

Chess openings are known sequences of moves that frequently appear at the beginning of games. Openings are assigned an ECO (Encyclopedia of Chess Openings) code to facilitate identification, although each has an assigned name too. Players will typically have what is called a ‘repertoire,’ or a set of openings that they regularly play.

FIDE is the International Chess Federation and the governing body for international chess competitions. FIDE assigns their own ratings to players using a similar rating system to Chess.com.

Titled chess players are ones who achieved a certain amount of success in FIDE-rated tournaments. The highest official title is Grandmaster (GM), followed by International Master (IM), FIDE Master (FM), and Candidate Master (CM). There are also women-specific titles (e.g., WGM, WIM, WFM, and WCM), which have slightly different requirements than their corresponding open titles.

Titled Tuesday is a blitz tournament played online on Chess.com. It is run two times each Tuesday, with the two tournaments commonly being referred to as “Early Titled Tuesday” and “Late Titled Tuesday” due to their times during the day. Titled Tuesday has been the subject of many of the cheating discussions regarding online chess, as some elite players claim that up to 50 percent of the players participating in the event cheat.

Cheating in online chess is commonly done by referencing a chess engine (e.g., Stockfish, Torch) to receive the best move in a position. The top chess engines are significantly stronger than the best human players. Therefore, receiving assistance from an engine is prohibited. However, due to easy online access and a lack of monitoring during such events, cheaters can typically only be caught using anti-cheating algorithms that can detect computer-level play.

PLANS:

My pessimistic plan was to construct a complex ER diagram and then complete the tables with some real-world data when possible. The data was to be normalized as well. The table count would be at least seven, with entities containing account information on Chess.com, Chess.com ratings, Chess.com game results, cheater information, opening names and ECO codes, opening use by account, and Titled Tuesday results.

My optimistic plan was to make each table thoroughly filled with entirely real-world data and then perform queries to identify outliers. For example, I could find standout players who achieve particularly strong results in late-round games on Titled Tuesday, as those are the games that often determine cash prizes and thereby incentivize cheating. Performing significance testing would be one way of accomplishing this. Moreover, I could investigate any other interesting ideas, such as which players have increased their rating the most in the past year; large rating increases can be suspicious.

***Exploring:***

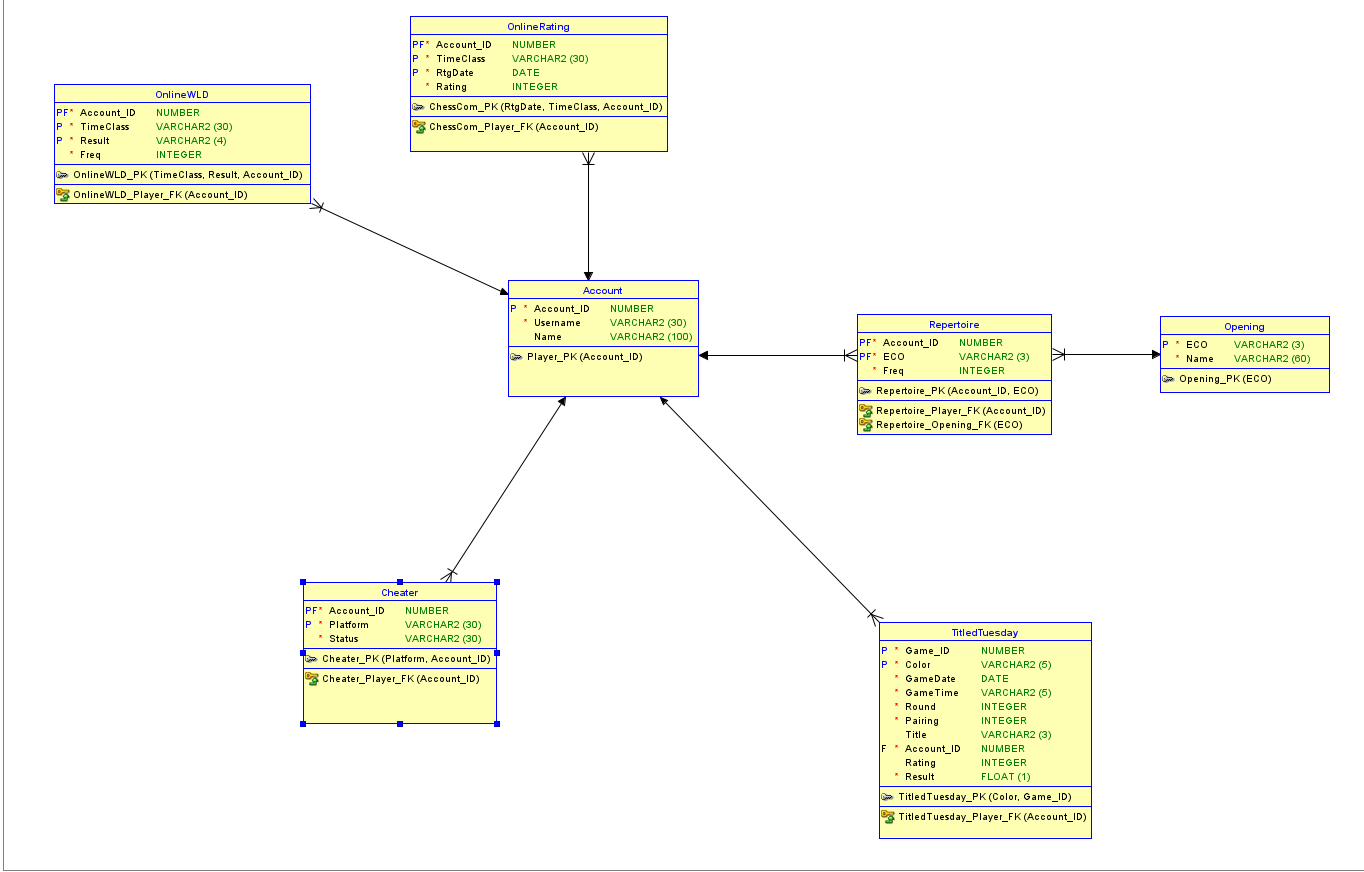
While I had some prior knowledge of SQL, I had very minimal experience working with databases. Therefore, most of my abilities pertaining to this project are ones I have developed during this course. For instance, while I had some familiarity with ER diagrams, I had never made one before, let alone used them to generate actual entities and relationships via DDL (Data Definition Language).

I was always set on making a database model for my final project. The two topics I considered were basketball and chess. However, I settled on the latter due to the abundance of resources in the area that I have recently worked with. My quality understanding of the various sources that I will list below in the following section was the deciding factor.

***Building:***

To build the database, I first considered all the different entities I wanted to include. Of course, I would need to have an account table that would be a foundation for most of the others. This would include an account ID, username, and (if possible) displayed name for each Chess.com account in the Titled Tuesday dataset I acquired. I also determined that a rating table showing the progression of an account’s rating in each time class over time would be necessary. Moreover, a table that contains win, draw, and loss frequencies in each time class is also a valuable inclusion. A repertoire table contains data on the favored openings of each player, and an opening table gives more information about each opening. A Titled Tuesday table contains information specific to the biweekly online tournament by the same name, showing game results, the players involved, the date each game was played on, and more. Finally, a cheater table includes players that have been accused of, widely suspected of, and/or admitted to cheating.

From here, I used Data Modeler to create a logical model, which I then forward engineered into a relational model.



My data sources include:

* [Chess.com](https://www.chess.com/home) for online ratings and statistics, accessed via API through the [chessR R package](https://jaseziv.github.io/chessR/articles/using_chessR_package.html)
* [This dataset](https://www.kaggle.com/datasets/garyongguanjie/chess-com-titled-tuesday-dataset/) from Kaggle that contains Titled Tuesday results from 2022 into October 2023
* [365Chess.com](https://www.365chess.com/eco.php) for opening names and ECO codes

All data was imported into SQL Developer via CSV file. All data collection and preparation were performed in RStudio, with the data frames then being written to CSV files.

For the account table, I took the list of usernames in the Titled Tuesday dataset and assigned each an account ID. I then looped through each username to find their displayed name on Chess.com. I did this by using the rvest package and XPath queries to obtain the correct element from each user’s profile page. Information on how to do this was found [here](https://www.opencodez.com/how-to-guide/how-to-use-xpath-for-web-scraping-with-r.htm).

For the onlinewld, onlinerating, and repertoire tables, I used the aforementioned chessR package to obtain information on every game played on Chess.com by a sample of the accounts in the accounts table. The information provided included the date of each game, the accounts’ ratings at the time of each game, the ECO code for each game, the result of each game, and more. I used the dplyr and stringr packages to clean and process this data as necessary to prepare it for entry into SQL Developer.

For the Titled Tuesday table, I used the Kaggle dataset linked above. I split the “White” and “Black” columns as necessary to obtain the title, username, and rating for each player. I also split the “Result” column to obtain separate results for white and black. Subsequently, I cleaved the data frame into two halves, one with the information about those who played with the white pieces and the other for those who played with the black pieces. After adding a color column to each half describing which color of pieces the player used, I recombined the data frames in such a fashion that each player in a game had their own row. Finally, I replaced each username with its corresponding account ID.

The cheater table was entirely constructed from my own memory of online discussions that have taken place. Some posts online, such as [this one on Reddit](https://www.reddit.com/r/chess/comments/xbvkqn/according_to_ukranian_fm_expert_on_cheating/), name players who have had suspicious account closures on Chess.com and other platforms. Such closures may indicate that they were caught cheating by the platform.

Examples of queries that I made possible include:

* Calculating and selecting the average result in the Titled Tuesday table by account for accounts with at least 100 games in the sample and an average score above 0.7
* Selecting the number of games and average result in the Titled Tuesday sample for accounts of players accused, suspected, or confirmed to have cheated in the past
* Selecting the number of games played, average result, and standard deviation of result for accounts with at least 30 games played before round 9 and at least 30 played in round 9 and later in the Titled Tuesday sample, with the numbers separated by round group
* Calculating and selecting the win rates of accounts in the Titled Tuesday table
* Selecting the number of total games played, games played with the ECO code A00, and rate of games played with the ECO code A00 for accounts in the repertoire table
* Selecting the most and least played openings by the accounts in the repertoire table
* Selecting the number of games played by time class per account
* Calculating the average result by time class per account
* Calculating rating changes over the past year by player and time class

Each of these examples can be found in the FP\_630\_queries.sql file in my OneDrive.

Notably, none of the accounts in the Titled Tuesday table had suspiciously high win rates in later rounds (9, 10, 11) compared to earlier rounds (1-8). However, as will be discussed further in the following section, the number of accounts with a sizable sample of games from later rounds was small, with just six having more than 30 such games in the dataset.

Of my dataset, the account with username JimDiGrease experienced the largest increase in rating in any time class over the past year. This account went from 400 to 2469 rating in bullet. However, due to time constraints it is quite difficult to cheat in bullet chess, so this was likely done legitimately. The account with username lamomiajunior is more noteworthy, as it went from a 354 rating to 2298 rating in rapid. There are many plausible explanations for this increase in rating, however, which do not include cheating.

***Discovering:***

One key lesson that I learned was the importance of caution when naming entities and columns. There were several occasions early on where I was unable to successfully insert data due to the relevant column or table identifiers being invalid. This in turn made dropping the affected tables more difficult.

Another crucial concept is constraints. Constraints are an excellent way to ensure the data entered is all correct, as hand-typed data often contains errors. Adding more constraints to tables via ALTER TABLE statements can make the data insertion process more efficient.

This project advanced my knowledge in several areas. For one, my comfort level with scraping web data in R went up significantly; this was my first time using XPath queries and the rvest package. I also became much more accustomed to the distinctive features in Data Modeler, as I needed to ensure my DDL statements would be suitable for the data I had gathered in RStudio. Before this class, I had never even constructed my own ER diagram before, let alone used Data Modeler successfully.

Moreover, my ability to perform complex queries in SQL Developer was also bolstered by this project, as evidenced by my example queries given in the FP\_630\_queries.sql file referenced above. Prior to this class, I had some experience constructing queries in SQLite, but nothing to the rigorous extent of what I performed in this project. Finally, I grew my confidence when importing data into SQL Developer via CSV files. Previously, I had only worked with CSV files in RStudio and Jupyter Notebook; being able to add SQL Developer to my repertoire should prove to be quite useful.

Several parts of my project did not go as planned. One notable component was the Titled Tuesday dataset I downloaded from Kaggle. Unfortunately, this dataset appears to be incomplete. Grandmaster Hikaru Nakamura is shown to have played zero games during and after round nine and 101 before round nine in this dataset. It is unfathomable that Nakamura, who has been the face of Titled Tuesday for years, never completed an event. This makes the queries I performed with this table rather unimportant. With unlimited time, I would gather all the Titled Tuesday data myself.

Another unfortunate development was when I encountered the rate limits on the Chess.com API. I was able to collect the complete game data for less than 10 percent of the accounts in the Titled Tuesday dataset after approximately five hours of runtime. With unlimited time, I would gather all the data.

Finally, the accounts have display names that are in many different alphabets. Some names, therefore, do not get parsed into SQL Developer properly when importing the CSV file.

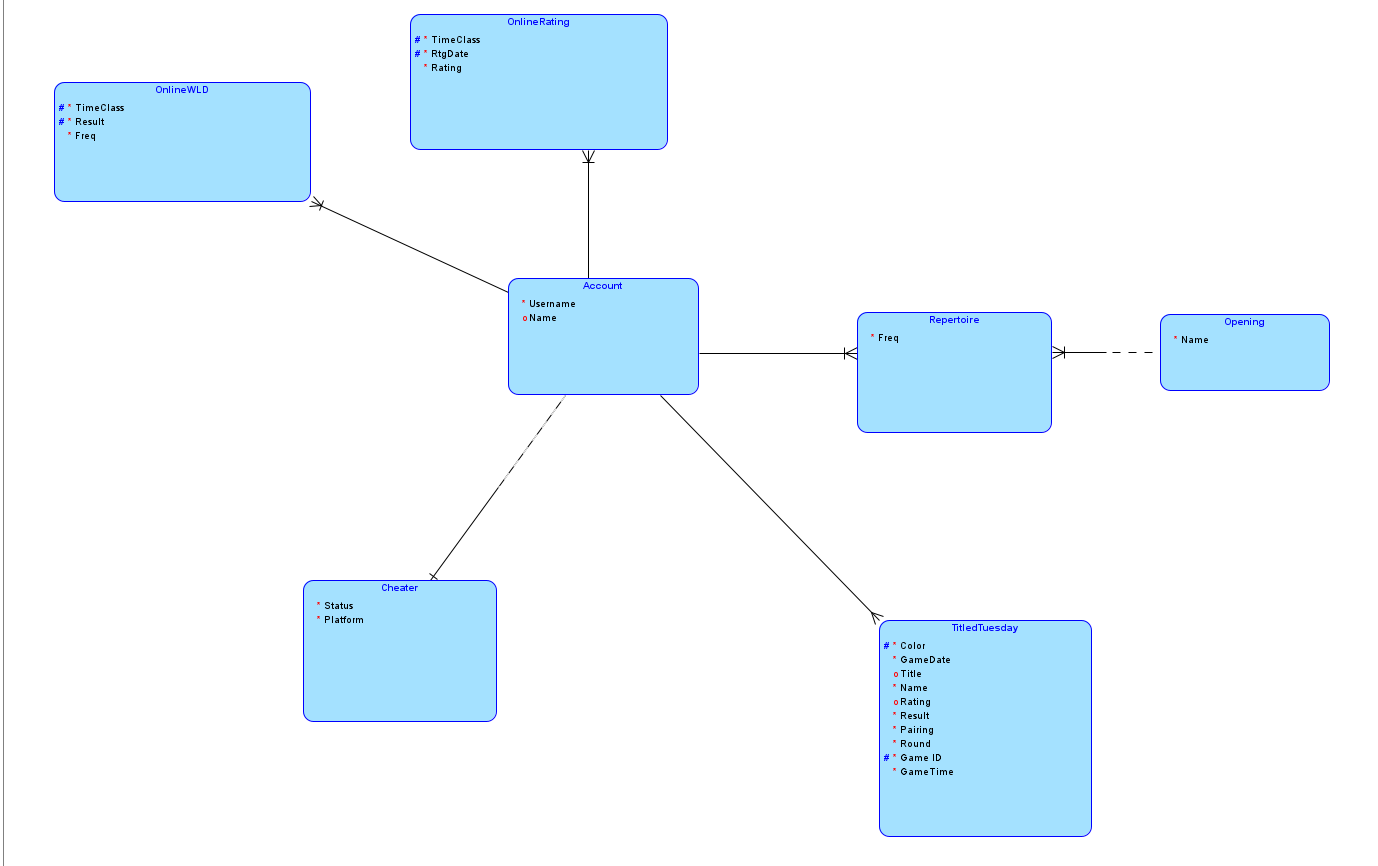
With more time, I also planned to add entities to the database that contained data pertaining to over-the-board performance. However, as the primary focus of my project related to online chess, this was not a priority.

***Five Topics from Class:***

TOPIC #1: CONSTRUCTING ER DIAGRAMS IN DATA MODELER:

As described and illustrated above, I produced a relational model for my database in Data Modeler before constructing the DDL statements for SQL Developer. As I already knew which entities I had the data to fill, their relationships, and their necessary constraints, constructing this model was simple.

I also reverse-engineered a logical model from this relational model. This can be seen here:



I also used Data Modeler to generate DDL statements to construct my database. These will be discussed further in Topic #4.

TOPIC #2: ADVANCED QUERY FORMULATION:

As described in the above Building section, I performed many advanced queries with my database. All these queries can be found in the FP\_630\_queries.sql file in my OneDrive.

Most of the queries I performed required multiple joins and/or nested subqueries. For example, the longest query I conducted calculated the rating change by account over the past year in each time class.

This required me to first select the most recent game date for each account at each time control they have played, excluding such dates within the past year, and joining that result with the onlinerating table on the date, account ID, and time class to select the rating that account had at that time control on that date. I then selected the account ID, relevant date, time class, and rating from that result.

I then select the most recent game date for each account at each time control they have played, *including* such dates within the past year, and joining that result with the onlinerating table, again on the date, account ID, and time class, to select the rating that account had at that time control on that date. I then again selected the account ID, relevant date, time class, and rating from that result.

Finally, I joined these two results on the account ID and time class, and then selected the account ID, time class, old rating, current rating, and rating change as the difference between the current and old ratings. When these steps are put together into one query, it results in many joins and nested subqueries.

TOPIC #3: NORMALIZATION:

The primary normalization topic I sought to address was multi-valued facts. In our class survey, there were different columns provided for a handful of different skills we did or did not possess. We were then tasked on an assignment with normalizing that data by making sure each person-skill combination had its own row, as in the table’s original form, it would be exceedingly difficult to group by the skills.

I ensured that my onlinerating table met the same standard. Instead of having one row per account with columns for each time class, I constructed the dataset to have one row for each account-time class combination, with the relevant date and ratings given in the final two columns. This facilitated the grouping on the different time classes.

Similarly, for my onlinewld table, I made sure to have a result column instead of having three separate win, loss, and draw columns that contained the frequencies. Again, this made the grouping process simpler when grouping on results.

TOPIC #4: DATABASE CONSTRUCTION WITH DDL STATEMENTS:

To create my database, I generated a DDL file from my model in Data Modeler.

This file includes:

* CREATE TABLE statements for each of the seven entities in the model, complete with each column, data type, and NOT NULL constraint if applicable
* 14 different ALTER TABLE statements, including
  + Seven that add the proper primary key constraint to each table
  + Six that add the proper foreign key constraints to the applicable tables
  + One that removes the mandatory constraint on the title column of the titledtuesday table; this was deemed necessary after further exploration of my dataset revealed inconsistencies with the data relating to six different accounts
* Two CREATE SEQUENCE statements, each paired with a CREATE OR REPLACE TRIGGER statement. These produced the account\_id and eco primary, non-foreign keys used in the account and opening tables, respectively. However, as my CSV files already contained the relevant keys, this was superfluous.

The entire file can be found in my OneDrive as FP\_630\_DDL.ddl.

TOPIC #5: PLAIN TEXT SERIALIZATION:

In RStudio, I used the built in read.csv() function to read the data from the Titled Tuesday dataset from Kaggle. Likewise, I used the write.csv() function to write each of the seven data frames I planned to import into my database to CSV files.

To import this data, I right-clicked on the relevant table name in my connections tab on SQL Developer. I then went through a series of prompts where I chose the correct file, selected my import method of Insert, selected the columns I wished to include in my table, assigned the columns in the CSV to the columns of the table, and finished.

***Conclusion:***

My database contains data relevant to several areas of online chess. With the inclusion of accounts’ ratings over time; win, loss, and draw frequencies; and Titled Tuesday results, important investigations can be made into the legitimacy of their play. Moreover, the opening and repertoire tables provide interesting insights into players’ habits.

This project certainly furthered my understanding of databases and of SQL. The data collection process, model construction in Data Modeler, data importation in SQL Developer, and query formulation all enhanced my understanding of the class topics.