2.

Firstly, find all the candidate key:

Initial schema R = (Game_ID, Rated, Start_time, End_time, Number_of_Turns, Game_Status, Winner, Time_increment, White_Player_ID, White_Player_Rating, Black_Player_ID, Black_Player_Rating, moves, Opening_Eco, Opening_Name, Opening_Ply).

```
mysql> Select count(*) From initial_chess;
+----------+
| count(*) |
+----------+
|    20058 |
+----------+
1 row in set (0.14 sec)
```

```
mysql> Select count(DISTINCT Game_ID) From initial_chess;
+------------------------+
| count(DISTINCT Game_ID) |
+------------------------+
|                  19113 |
+------------------------+
1 row in set (0.05 sec)
```

```
mysql> Select count(DISTINCT Start_time) From initial_chess;
+---------------------------+
| count(DISTINCT Start_time) |
+---------------------------+
|                     13151 |
+---------------------------+
1 row in set (0.04 sec)
```

```
mysql> Select count(DISTINCT End_time) From initial_chess;
+-------------------------+
| count(DISTINCT End_time) |
+-------------------------+
|                   13186 |
+-------------------------+
1 row in set (0.03 sec)
```

```
mysql> Select count(DISTINCT Number_of_Turns) From initial_chess;
+--------------------------------+
| count(DISTINCT Number_of_Turns) |
+--------------------------------+
|                            211 |
+--------------------------------+
1 row in set (0.01 sec)

mysql> Select count(DISTINCT Game_Status) From initial_chess;
+-----------------------------+
| count(DISTINCT Game_Status) |
+-----------------------------+
|                           4 |
+-----------------------------+
1 row in set (0.02 sec)

mysql> Select count(DISTINCT Winner) From initial_chess;
+------------------------+
| count(DISTINCT Winner) |
+------------------------+
|                      3 |
+------------------------+
1 row in set (0.01 sec)

mysql> Select count(DISTINCT Time_Increment) From initial_chess;
+--------------------------------+
| count(DISTINCT Time_Increment) |
+--------------------------------+
|                            400 |
+--------------------------------+
1 row in set (0.02 sec)
```

```
mysql> Select count(DISTINCT White_Player_ID) From initial_chess;
+--------------------------------+
| count(DISTINCT White_Player_ID) |
+--------------------------------+
|                           9438 |
+--------------------------------+
1 row in set (0.02 sec)

mysql> Select count(DISTINCT White_Player_Rating) From initial_chess;
+-----------------------------------+
| count(DISTINCT White_Player_Rating) |
+-----------------------------------+
|                              1516 |
+-----------------------------------+
1 row in set (0.01 sec)
```

```
mysql> Select count(DISTINCT Black_Player_ID) From initial_chess;
+--------------------------------+
| count(DISTINCT Black_Player_ID) |
+--------------------------------+
|                           9331 |
+--------------------------------+
1 row in set (0.02 sec)

mysql> Select count(DISTINCT Black_Player_Rating) From initial_chess;
+-----------------------------------+
| count(DISTINCT Black_Player_Rating) |
+-----------------------------------+
|                              1521 |
+-----------------------------------+
1 row in set (0.01 sec)
```

```
mysql> Select count(DISTINCT moves) From initial_chess;
+----------------------+
| count(DISTINCT moves) |
+----------------------+
|                18920 |
+----------------------+
1 row in set (0.26 sec)
```

```
mysql> Select count(DISTINCT Opening_Eco) From initial_chess;
+----------------------------+
| count(DISTINCT Opening_Eco) |
+----------------------------+
|                        365 |
+----------------------------+
1 row in set (0.02 sec)
```

```
mysql> Select count(DISTINCT Opening_Name) From initial_chess;
+-----------------------------+
| count(DISTINCT Opening_Name) |
+-----------------------------+
|                        1477 |
+-----------------------------+
1 row in set (0.04 sec)

mysql> Select count(DISTINCT Opening_Ply) From initial_chess;
+----------------------------+
| count(DISTINCT Opening_Ply) |
+----------------------------+
|                         23 |
+----------------------------+
1 row in set (0.01 sec)
```

```
mysql> Select count(DISTINCT Game_ID, Start_time, End_time, Number_of_Turns, Gam
e_Status,Winner,Time_Increment,White_Player_ID,White_Player_Rating,Black_Player_
ID,Black_Player_Rating,moves,Opening_ECO,Opening_name,Opening_Ply) From initial_
chess;
+----------------------------------------------------------------------------
----------------------------------------------------------------------------
-----------------------------------------------------+
| count(DISTINCT Game_ID, Start_time, End_time, Number_of_Turns, Game_Status,Win
ner,Time_Increment,White_Player_ID,White_Player_Rating,Black_Player_ID,Black_Pla
yer_Rating,moves,Opening_ECO,Opening_name,Opening_Ply) |
+----------------------------------------------------------------------------
----------------------------------------------------------------------------
-----------------------------------------------------+
|
                                                 19629 |
+----------------------------------------------------------------------------
----------------------------------------------------------------------------
-----------------------------------------------------+
1 row in set (0.36 sec)
```

From this result, we can see that there are some duplicate rows in the given dataset.

To figure the primary key, we need to remove the duplicate rows in the given dataset

```
mysql> source /Users/mengtianao/Documents/ece656/MengTianao/A2/initial_load_csv.sql;
Query OK, 19629 rows affected (0.39 sec)
Records: 19629  Deleted: 0  Skipped: 0  Warnings: 0
```

```
SELECT COUNT(DISTINCT Game_ID, Start_time) FROM Initial_chess;
```

```
+-------------------------------------+
| COUNT(DISTINCT Game_ID, Start_time) |
+-------------------------------------+
|                               19629 |
+-------------------------------------+
1 row in set (0.02 sec)
```

Thus, we can get that that Game_ID and Start_time is our primary key; To make the table in BCNF, we need atomic attributes; However, the move attribute in this table is not atomic.

Seen in py file.

õ