Lab4: SQL

Tailang CAO u1480633

Part 3

1. **Get the ISBNs of all books by <Author>**

```
SELECT ISBN
FROM Titles
WHERE Author = '<Author>';
```

2. **Get Serial numbers (descending order) of all books by <ISBN>:**

```
SELECT Serial
FROM Inventory
WHERE ISBN = '<ISBN>' ORDER BY Serial DESC;
```

3. **Get the Serial numbers and ISBNs of all books checked out by <Patron's name>**

```
SELECT Inventory.Serial, Inventory.ISBN
FROM CheckedOut
JOIN Inventory ON CheckedOut.Serial = Inventory.Serial JOIN Patrons ON
CheckedOut.CardNum = Patrons.CardNum WHERE Patrons.Name = '<Patron’s
name>';
```

4. **Get phone number(s) and Name of anyone with <ISBN> checked out**

```
SELECT Patrons.Phone, Patrons.Name
FROM CheckedOut
JOIN Inventory ON CheckedOut.Serial = Inventory.Serial JOIN Patrons ON
CheckedOut.CardNum = Patrons.CardNum WHERE Inventory.ISBN = '<ISBN>';
```

Part 4

1. **Find the Authors of the library's oldest <N> books. Assume the lowest serial number is the oldest book.**

   ```
   SELECT t.Author
   FROM Titles t
   JOIN Inventory i ON t.ISBN = i.ISBN ORDER BY i.Serial ASC
   LIMIT <N>;
   ```

2. **Find the name and phone number of the person who has checked out the most recent book. Assume higher serial numbers are newer. Note that this query is not concerned with the absolute highest serial number, it is concerned with the highest one that has been checked out.**

   ```
   SELECT p.Name, p.Phone
   FROM CheckedOut c
   JOIN Patrons p ON c.CardNum = p.CardNum ORDER BY c.Serial DESC
   LIMIT 1;
   ```

3. **Find the phone number(s) and name of anyone who has checked out any book.**

   ```
   SELECT p.Name, ph.Phone
   FROM CheckedOut c
   JOIN Patrons p ON c.CardNum = p.CardNum JOIN Phones ph ON p.CardNum =
   ph.CardNum ORDER BY c.Serial DESC
   LIMIT 1;
   ```

4. **Find the Authors and Titles of the books who have NOT been checked out by anyone. The query should not return duplicates.**

   ```
   SELECT DISTINCT t.Author, t.Title
   FROM Titles t
   JOIN Inventory i ON t.ISBN = i.ISBN
   LEFT JOIN CheckedOut c ON i.Serial = c.Serial
   WHERE c.Serial IS NULL;
   ```

Part 5

1. **Find the names and IDs of any player with the 10 highest Elo ratings.**

```
SELECT Name, pID, Elo FROM Players
ORDER BY Elo DESC LIMIT 10;
```

2. **Find the names and Elo ratings of any player who has ever played a game as black.**

```
SELECT DISTINCT p.Name, p.Elo
FROM Players p
JOIN Games g ON p.pID = g.BlackPlayer;
```

3. **Find the names of any player who has ever won a game as white.**

```
SELECT DISTINCT p.Name
FROM Players p
JOIN Games g ON p.pID = g.WhitePlayer
WHERE g.Result = 'W';
```

4. **Find the names of any player who played any games between 2014 and 2018 in Budapest HUN .**

```
SELECT p.Name
FROM Players p
JOIN Games g ON p.pID = g.WhitePlayer OR p.pID = g.BlackPlayer
JOIN Events e ON p.Name = e.Name
WHERE e.Site = ‘Budapest HUN’ AND e.Date BETWEEN ‘2014-01-01' AND
'2018-12-31';
```

5. **Find the Sites and dates of any event in which Garry Kasparov won a game.**

```
SELECT DISTINCT e.Site, e.Date
FROM Games g
JOIN Events e ON g.eID = e.eID
JOIN Players p ON g.WhitePlayer = p.pID
WHERE p.Name = 'Kasparov, Garry' AND g.Result = 'W' UNION
SELECT DISTINCT e.Site, e.Date
FROM Games g
JOIN Events e ON g.eID = e.eID
JOIN Players p ON g.BlackPlayer = p.pID
WHERE p.Name = 'Kasparov, Garry' AND g.Result = 'B';
```

6. **Find the names of all opponents of Magnus Carlsen. An opponent is someone who he has played a game against. Hint: Both Magnus and his opponents could play as white or black.**

```
SELECT DISTINCT p2.Name
FROM Games g
JOIN Players p1 ON g.WhitePlayer = p1.pID OR g.BlackPlayer = p1.pID
JOIN Players p2 ON (g.WhitePlayer = p2.pID OR g.BlackPlayer = p2.pID) AND
p2.pID != p1.pID
WHERE p1.Name = 'Carlsen, Magnus';
```