Lab 4: SQL

Part 3 - Simple Retrieval Queries

Get the Titles of all books by <Author>

```
SELECT Title
FROM Titles
WHERE Author = <Author>;
```

Get Serial numbers of all books by <Author>

```
SELECT Serial
FROM Inventory NATURAL JOIN Titles
WHERE Author = <Author>;
```

Get the Titles of all books checked out by <Patron's name>

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

Get phone number(s) of anyone with <Title> checked out

```
SELECT Phone
FROM Phones
JOIN CheckedOut ON CheckedOut.CardNum = Phones.CardNum
JOIN Inventory ON Inventory.Serial = CheckedOut.Serial
JOIN Titles ON Inventory.ISBN = Titles.ISBN
WHERE Titles.Title = <Title>;
```

Part 4 - Intermediate Retrieval Queries

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

```
SELECT Title
FROM Titles
NATURAL JOIN Inventory
ORDER BY Inventory.Serial
LIMIT <N>;
```

Find the name of the person who has checked out the most recent book. Assume the highest serial number is the newest book. Hint: the highest serial number book may not be checked out by anyone.

```
SELECT Name FROM Patrons
NATURAL JOIN CheckedOut
NATURAL JOIN Inventory
ORDER BY Inventory.Serial DESC
LIMIT 1;
```

Find the phone number(s) of anyone who has not checked out any books.

```
SELECT DISTINCT Phone
FROM Phones
WHERE Phone NOT IN (
    SELECT DISTINCT Phone
    FROM Phones
    NATURAL JOIN Patrons
    NATURAL JOIN CheckedOut
);
```

The library wants to expand the number of unique selections in its inventory, thus, it must know the ISBN and Title of all books that it owns at least one copy of. Create a query that will return the ISBN and Title of every book in the library, but will not return the same book twice.

```
SELECT DISTINCT Title, ISBN
FROM Titles
NATURAL JOIN Inventory;
```

Part 5 - Chess Queries

Find the names of any player with an Elo rating of 2850 or higher.

```
SELECT Name
FROM Players
WHERE Elo >= 2850;
```

Find the names of any player who has ever played a game as white.

```
SELECT Name FROM Players
JOIN (
    SELECT DISTINCT WhitePlayer FROM Games
) AS AllWhitePlayers ON Players.pID = AllWhitePlayers.WhitePlayer;
```

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

```
SELECT Name FROM Players
JOIN (
    SELECT DISTINCT WhitePlayer FROM Games WHERE Result = 'W'
) AS AllWhiteWinners ON Players.pID = AllWhiteWinners.WhitePlayer;
```

Find the names of any player who played any games in 2018.

```
SELECT DISTINCT Name
FROM Players
JOIN (
    SELECT BlackPlayer, WhitePlayer
    FROM Games
    NATURAL JOIN Events
    WHERE YEAR(Events.Date) = 2018
) AS Players2018
ON (Players.pID = Players2018.BlackPlayer OR Players.pID = Players2018.WhitePlayer);
```

Find the names and dates of any event in which Magnus Carlsen lost a game.

```
SELECT Name, Date
FROM Events
JOIN (
    SELECT DISTINCT eID
    FROM Games
    JOIN (
        SELECT pID
        FROM Players
        WHERE Name = 'Carlsen, Magnus'
    ) AS MagnusCarlsen ON (
        (Games.BlackPlayer = MagnusCarlsen.pID AND Games.Result = 'W')
        OR (Games.WhitePlayer = MagnusCarlsen.pID AND Games.Result = 'B')
    )
) AS LoserGame ON Events.eID = LoserGame.eID;
```

Find the names of all opponents of Magnus Carlsen. An opponent is someone who he has played a game against.

```
SELECT Name
FROM Players
JOIN (
    (
        SELECT DISTINCT pID
        FROM (
            (
                SELECT DISTINCT WhitePlayer AS pID
                FROM Games
                JOIN (
                    SELECT pID
                    FROM Players
                    WHERE Name = 'Carlsen, Magnus'
                ) AS MagnusCarlsen ON Games.BlackPlayer = MagnusCarlsen.pID
            )
            UNION
            (
                SELECT DISTINCT BlackPlayer AS pID
                FROM Games
                JOIN (
                    SELECT pID
                    FROM Players
                    WHERE Name = 'Carlsen, Magnus'
                ) AS MagnusCarlsen ON Games.WhitePlayer = MagnusCarlsen.pID
            )
        ) AS Opponents
    )
) AS Opponents ON Opponents.pID = Players.pID;
```