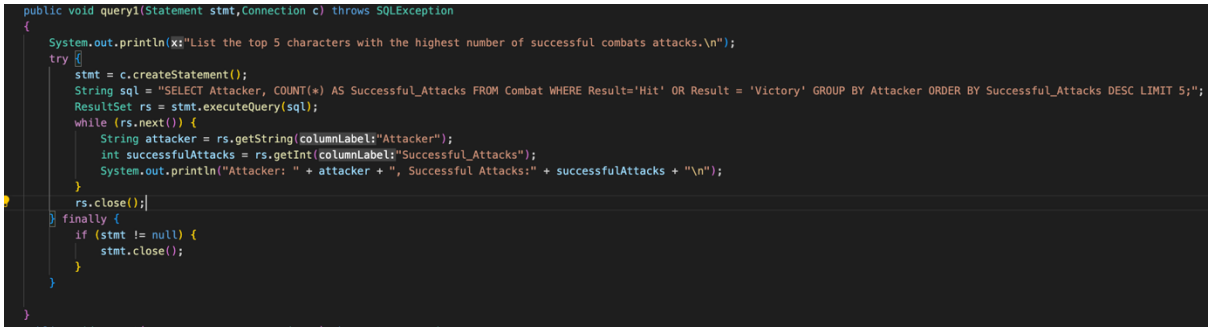
Milestone 3

English versions of Queries to SQL:

1) List the top 5 characters with the highest number of successful combats attacks.

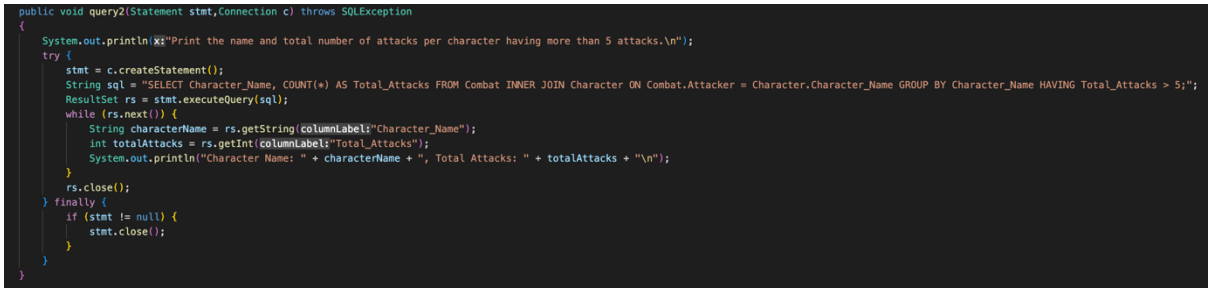


```
public void query1(Statement stmt,Connection c) throws SQLException
{
    System.out.println(x:"List the top 5 characters with the highest number of successful combats attacks.\n");
    try {
        stmt = c.createStatement();
        String sql = "SELECT Attacker, COUNT(*) AS Successful_Attacks FROM Combat WHERE Result='Hit' OR Result = 'Victory' GROUP BY Attacker ORDER BY Successful_Attacks DESC LIMIT 5;";
        ResultSet rs = stmt.executeQuery(sql);
        while (rs.next()) {
            String attacker = rs.getString(columnLabel:"Attacker");
            int successfulAttacks = rs.getInt(columnLabel:"Successful_Attacks");
            System.out.println("Attacker: " + attacker + ", Successful Attacks:" + successfulAttacks + "\n");
        }
        rs.close();
    } finally {
        if (stmt != null) {
            stmt.close();
        }
    }

}
```

As seen above the SQL statement is set so that it selects the Attacker column of Combat table and counts the successful attacks on the same row where the result column states a ‘Hit’ or ‘Victory’ which corresponds to a successful attack and based on that information of how many times there has been a successful attack for that player it is grouped together and ordered from highest to lowest.

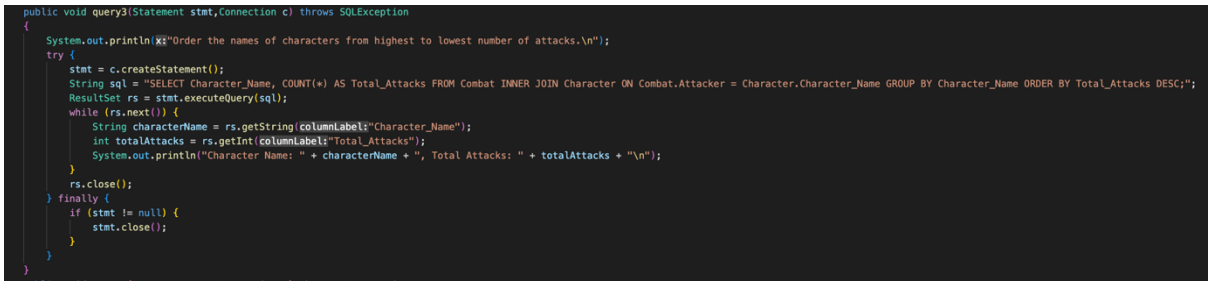
2) Print the name and total number of attacks per character having more than 5 attacks.



```
public void query2(Statement stmt,Connection c) throws SQLException
{
    System.out.println(x:"Print the name and total number of attacks per character having more than 5 attacks.\n");
    try {
        stmt = c.createStatement();
        String sql = "SELECT Character_Name, COUNT(*) AS Total_Attacks FROM Combat INNER JOIN Character ON Combat.Attacker = Character.Character_Name GROUP BY Character_Name HAVING Total_Attacks > 5;";
        ResultSet rs = stmt.executeQuery(sql);
        while (rs.next()) {
            String characterName = rs.getString(columnLabel:"Character_Name");
            int totalAttacks = rs.getInt(columnLabel:"Total_Attacks");
            System.out.println("Character Name: " + characterName + ", Total Attacks: " + totalAttacks + "\n");
        }
        rs.close();
    } finally {
        if (stmt != null) {
            stmt.close();
        }
    }
}
```

For this query it selects the Character_Name column from Character table and uses the count function to count the number of rows that match the character name to the attacker in the table and then an inner join takes place between Combat and Character table where the Attacker matches Character Name and then groups them by having total attacks more than 5 to fulfil the conditions of the question.

3) Order the names of characters from highest to lowest number of attacks.



```
public void query3(Statement stmt,Connection c) throws SQLException
{
    System.out.println(x:"Order the names of characters from highest to lowest number of attacks.\n");
    try {
        stmt = c.createStatement();
        String sql = "SELECT Character_Name, COUNT(*) AS Total_Attacks FROM Combat INNER JOIN Character ON Combat.Attacker = Character.Character_Name GROUP BY Character_Name ORDER BY Total_Attacks DESC;";
        ResultSet rs = stmt.executeQuery(sql);
        while (rs.next()) {
            String characterName = rs.getString(columnLabel:"Character_Name");
            int totalAttacks = rs.getInt(columnLabel:"Total_Attacks");
            System.out.println("Character Name: " + characterName + ", Total Attacks: " + totalAttacks + "\n");
        }
        rs.close();
    } finally {
        if (stmt != null) {
            stmt.close();
        }
    }
}
```

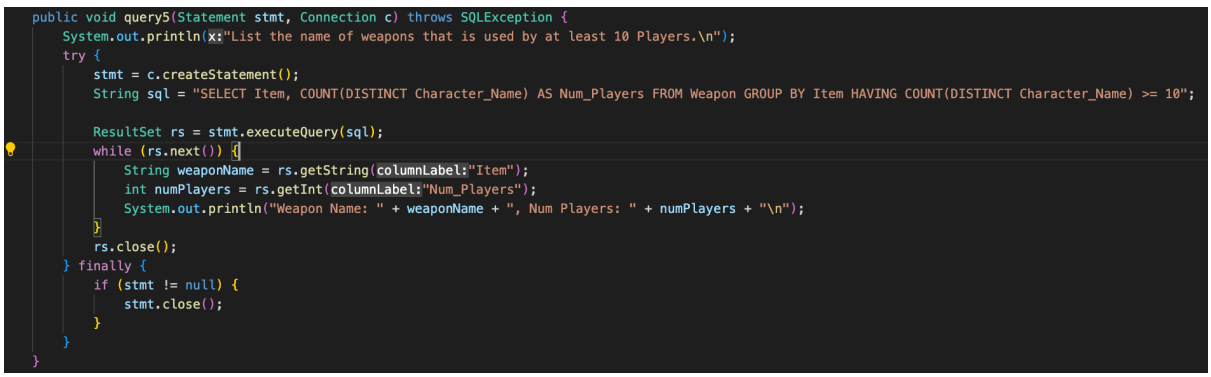
This is similar to question 2 in terms of selecting Character_Name column from Character table and uses the count function to count the number of rows that match the character name to the attacker in the table and then an inner join takes place between Combat and Character table where the Attacker matches Character Name and then groups them but instead of having total attacks above 5 in this case we are just ordering the Character_Name and total attacks in descending order.

4) List the name of Players with at least 5 characters.

```
public void query4(Statement stmt,Connection c) throws SQLException
{
    System.out.println(x:"List the name of Players with at least 5 characters\n");
    try {
        stmt = c.createStatement();
        String sql = "SELECT Account_Number, forename, surname FROM Player WHERE Account_Number IN (SELECT Account_Number FROM Character GROUP BY Account_Number HAVING COUNT(*) >= 5);";
        ResultSet rs = stmt.executeQuery(sql);
        while (rs.next()) {
            int accountNumber = rs.getInt(columnLabel:"Account_Number");
            String forename = rs.getString(columnLabel:"forename");
            String surname = rs.getString(columnLabel:"surname");
            System.out.println("Account Number: " + accountNumber + ", Name: " + forename + " " + surname + "\n");
        }
        rs.close();
    } finally {
        if (stmt != null) {
            stmt.close();
        }
    }
}
```

The first part of this SQL statement is selecting the Account_Number, forename, and surname columns of the player table where the account number is in the subquery of selecting Account_Number column from Character table and group the Account numbers after the query and count those that have 5 characters or more.

5) List the name of weapons that is used by at least 10 Players.



```
public void query5(Statement stmt, Connection c) throws SQLException {
    System.out.println(x:"List the name of weapons that is used by at least 10 Players.\n");
    try {
        stmt = c.createStatement();
        String sql = "SELECT Item, COUNT(DISTINCT Character_Name) AS Num_Players FROM Weapon GROUP BY Item HAVING COUNT(DISTINCT Character_Name) >= 10";

        ResultSet rs = stmt.executeQuery(sql);
        while (rs.next()) {
            String weaponName = rs.getString(columnLabel:"Item");
            int numPlayers = rs.getInt(columnLabel:"Num_Players");
            System.out.println("Weapon Name: " + weaponName + ", Num Players: " + numPlayers + "\n");
        }
        rs.close();
    } finally {
        if (stmt != null) {
            stmt.close();
        }
    }
}
```

In this query you are selecting the Item column and uses a count function to count the numner of distinct CHaracter_Name value in Weapon table and keep it in a resulting column

of Num_Players and group them by item where the Character_Name is more than or equal to 10.