**CSE 3241 Project Checkpoint 04**

**SQL and More SQL**

Names Date

**Submitted to the Carmen Dropbox**

1. Provide a current version of your ER Diagram and Relational Model as per Project Checkpoint 04. **If you were instructed to change the model for Project Checkpoint 03, make sure you use the revised versions of your models.** You must highlight and indicate the corrections/modifications.
2. Use the database populated with the data provided for Checkpoint 3 (with at least 20 sample records for each table and including data provided in the original project documents).
3. For Project Checkpoint 03, you were asked to come up with a set of queries and three additional interesting queries that your database can provide, outlined in Worksheet 02. If you were instructed to fix the queries in Checkpoint 03, make sure you include the fixed queries here. These queries should be provided in a plain text file named “ExtraQueries.txt”. Include also an updated version of the “Views.txt” file if you were instructed to fix them.
4. Given your relational schema, provide the SQL for the following more advanced queries. These queries may require you to use techniques such as nesting, aggregation using having clauses, and other techniques. If your database schema does not contain the information to answer to these queries, revise your ER Model and your relational schema to contain the appropriate information for these queries. **Note that if your database does contain the information but in non-aggregated form, you should NOT revise your model but instead figure out how to aggregate it for the query!** These queries should be provided in a plain text file named “AdvancedQueries.txt”.
   1. Provide a list of member names, along with the total combined amount of all items they have rented out.
   2. Provide a list of member names and email addresses for members who have rented more equipment than the average member.
   3. Provide a list of the equipment in the database and associated total copies rented to members, sorted from the equipment that has been rented the most to the movies that has been rented the least.
   4. Provide a list of the drones in the database, sorted from the ones that have been delivered the highest amount of items to the ones delivered the lowest.

* 1. Find the most popular manufacturer in the database (i.e. the one who has had the most rented items)
  2. Find the most used items in the database (use the running rented time of the item and number of times the item has been rented out to calculate)
  3. Provide a list of members information for members who have rented out anything by the most demanded equipment in the database.
  4. Provide a list of manufacturers who provided the items rented out by members who have rented more items than the average customer.

1. Integrate your Java application from Checkpoint #2 with the equipment renting database your are implementing, and provide the functionality (proposed and whatever you already implemented) but storing and accessing the data from the database. Use as starting point the application developed for Checkpoint #3 and follow the instructions for importing the SQL library API in to your java project (the Embedded\_SQL lab will help to practice the connection to a SQLite database). Once the database is imported and you are able to connect/access it, start implementing each one of the use cases provided in your menu.

Notes about the options:

**Add new records:**

* The user provides all the info needed to enter a new record (member, equipment, drone, ..) into the database.

**Edit/Delete records**

* The user selects an item option (member, equipment, drone, ..), provide the name and are presented with the option to edit/delete any field and then save it, updating the database.

**Search:**

* The user selects an item option (member, equipment, drone, ..) and provides the info needed for the search. The program then searches according to the criteria and retrieves the records
* You are free to design/provide the search criteria (name, year, title, ..)

**Useful reports:**

* Renting checkouts: Find the total number of equipment items rented by a single member patron (user designates the member) (see query from Checkpoint #3)
* Popular item: Find the most popular item in the database (use the renting time of the item and number of times the item has been rented out to calculate) (see query from Checkpoint #4)
* Popular Manufacturer: Find the most frequent equipment manufacturer in the database (i.e. the one who has had the most rented units) (see query from Checkpoint #4)
* Popular Drone: Find the most used drone in the database (use the flying distance of the drone -member distance- and number of deliveries the drone has been traveled to calculate) (see similar query from Checkpoint #4)
* Items checked out: Find the member who has rented out the most items and the total number of items they have rented out. (see query from Checkpoint #3).
* Equipment by Type of equipment: Find the description (name) of equipment by type released before YEAR. (see query from Checkpoint #3).

**SQLite JDBC library for Java and development tool - IDE**

* You are going to use eclipse (IDE).
* You will need the SQLite JDBC library. Download SQLite JDBC Driver and connect to the SQLite database via JDBC. Go to Carmen and download the sqlite-jdbc-3.32.3.2.jar

Add the jar file to your Java project. To add the .jar file to your eclipse project, right-click on the project name -> select build path -> select external archives -> navigate to the .jar file folder and select it.

*Note*: *For more general information about Installing the Java Database Connectivity (JDBC) API on different DBMS read instructions at:*

* [*https://www.codejava.net/java-se/jdbc/jdbc-driver-library-download*](https://www.codejava.net/java-se/jdbc/jdbc-driver-library-download) (links for different DBMS drivers)
* <https://docs.oracle.com/javase/tutorial/jdbc/basics/gettingstarted.html>.

.

Be sure to include in your program the best practices to prevent SQL injection, use prepared statements, sanitize input. Make improvements to the initial program according to best practices. (Name the text file with all the queries: “QueriesJavaApp.txt”. Include the file in the submission).

You need to submit:

* + The Java file with the Java project, database and any other file required in your implementation.
  + Run the program, test it and Include screen outputs for each one of the menu options

Once you have completed all of the questions for parts 1 - 5, create a ZIP archive containing the binary SQLite file and the text files. For part 5, export your Java project to a zip file and submit both, appropriately naming the zip files, to the Carmen Dropbox. Include screenshots of the output for each query.

**Make sure your queries work against your database and provide your expected output before you submit them! The Java application must compile, run, and behave as expected.**

No syntax or running errors should exist in the submission. If the grader cannot construct and populate the database with your code, nor compile/execute your application, then he will be unable to give you feedback.

1. Each team member, individually, needs to fill out the Peer-evaluation form provided and submit it to Carmen.

Please DO NOT zip the report file when you submit so that the grader can give you detailed feedback in Carmen.