**CSE 3241 Project Checkpoint 02**

**Functional Dependencies and Normalization**

Names: Seth Lamancusa, Connor Burke, Max Miller, Luke Alzapiedi

In a **NEATLY TYPED** document, provide the following:

1. Provide a current version of your ER Diagram and Relational Model as per Project Checkpoint 02. **If you were instructed to change the model for Project Checkpoint 02, make sure you use the revised versions of your models.** You must highlight and indicate the corrections/modifications.
2. Given your relational schema, provide the relational algebra to perform the following queries. If your schema cannot provide answers to these queries, revise your ER Model and your relational schema to contain the appropriate information for these queries:
   1. Find the type of all equipment by MANUFACTURERS released before YEAR

pTYPE(s­YEAR<2015(EQUIPMENT))

* 1. Give all the equipment and their date of their checkout from a single member (you choose how to designate the member)

(pDATE,MODEL\_NUM,SERIAL\_NUM(ORDERS\*(s­USER\_ID=12(MEMBER)))\*INV\_ORDERS\*EQUIPMENT

* 1. List all the equipment by TYPE and their unique identifiers with less than 5 copies held by the warehouse.

pTYPE,SERIAL\_NUM,MODEL\_NUM(s­­QUANTITY<5(EQUIPMENT))

* 1. Give all the members who checked out equipment by DRONE and the equipment they checked out.

sRETURN\_DATE!=NULL(ORDER)\*EQUIPMENT\*MEMBER

* 1. Find the total number of equipment items checked out by a single member (you choose how to designate the member)

FSUM NUM\_ITEMS(sUSER\_ID=12(MEMBER)\*ORDER)

* 1. Find the member who has checked out the most equipment and the total number of items they have checked out.

FMAX NUM\_ITEMS(MEMBER)

1. Come up with three additional interesting queries that your database can provide. Give what the queries are supposed to retrieve in plain English and then as relational algebra. Your queries should include joins and at least one should include an aggregate function. At least one of your queries should use “extra” entities you added to your model in Checkpoint 01.

List all orders placed by a given employee:

sSSN=12(EMPLOYEE)\*ORDER

List the average value of a customer order:

FAVG VALUE(sRETURN\_DATE!=NULL(ORDER))

List all drones and their max speed in a given warehouse:

pMODEL\_NUM,SERIAL\_NUM,MAX\_SPEED(sADDR=55 W Lane,CITY=COLUMBUS(WAREHOUSE)\*DRONE)

1. For each relation schema in your model, indicate the functional dependencies. Think carefully about what you are modeling here - make sure you consider all the possible dependencies in each relation and not just the ones from your primary keys. For example, a customer’s credit card number is unique, and so will uniquely identify a customer even if you have another key in the same table (in fact, if the customer can have multiple credit card numbers, the dependencies can get even more involved).
2. For each relation schema in your model, determine the highest normal form of the relation. If the relation is not in 3NF, rewrite your relation schema so that it is in at least 3NF (Normalize the relational schema).
3. For each relation schema in your model that is in 3NF but not in BCNF, either rewrite the relation schema to BCNF or provide a short justification for why this relation should be an exception to the rule of putting relations into BCNF.
4. The purpose of this part of the checkpoint is to refresh Java programming. You will re-take your conceptual design from Checkpoint 1 and review with your team any changes you mat want to make to the user interface (text-based interface) and then write the code to interact with the user.

The purpose is to write a Java program (no database are needed at this point). The program will present the user a menu with a list of options according to your user interface design, and then each option will ask the user for the input needed. You will need to store that data into any Java data structure that you consider more appropriate (classes, ArrayList, Sets, ..) and then be able to retrieve it as an output when an option requires it. Note: Implement a text-base interface (GUI is optional).

Note: You will continue developing this program, in future checkpoint you will interface with a database

The Java app needs to provide, at least, the following basic functionality:

* 1. **Add new records** (implement at least 3 of the main records, e.g. members, equipment, warehouse, ..)
  2. **Edit/delete records** (e.g. members, equipment, warehouse, ..)
  3. **Search** (e.g. members, equipment, warehouse, ..)
  4. **Useful reports** (not to be implemented yet)

Notes:

* All the options indicated must be implemented and tested (look for some real data)
* The program must compile and execute.
* The test output for each option must be included
* The program must run on eclipse so we can test it
* Use a text based interface. You don’t need a user graphical interface. Input and output would be from the console.
* DO NOT create databases at this time, this will be done on another checkpoint. Use classes, ArrayLists, sets, any other data structure as needed.

Zip the Java project and include it in the submission.

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

1. Each team member, individually, needs to fill out the Peer-evaluation form provided and submit it to Carmen (identify that is for Checkpoint #2). Use the same dropbox on Carmen.

**Note**: Your submission should include your ER-diagram, the relation schema before this checkpoint (before normalizing), the relational schema after this checkpoint (after normalizing) as well as the Java project.