Michael Garod, Odysseas Katsougrakis, and Zouhair Khallaf

Summary of “WAR GAME DATABASE”

In this team project we learned how to work in a team to design and create a fully functional database using Oracle SQL\*Plus. Our team designed and created a database representing a snapshot of an online war game.

Our virtual scenario is a game where every player has a first and last name, a starting amount of funds to make purchases, a unique player ID, and a unique username. Players can own different numbers of soldiers, tanks, and jets to be used in battles. Players can be members of alliances. Each alliance has a name, a unique identification number, and a territory that they are based in.

Players can place orders with different stores to supply any required war munitions. Each order, called a supply drop, is associated with a player, a store, and a particular munition. Supply drops are labeled with a timestamp of when they were ordered, as well as the quantity of the munition ordered. All munitions have a name, price, and unique munition number. Stores are labeled with a unique store ID, store name, and the location of the store. Players can order as many supply drops, of whichever munition, from whichever store that they would like.

Players can attack any other player. Records of each battle that takes places between players are kept with the unique ID of the attacking player, the unique ID of the defending player, unique ID of the victor, the location that the attack took place, and the timestamp of when the attack took place. Repeated attacks between the same attacker on the same defender are distinguished by the timestamp.

To fully map this abstract data model, seven entities were created: SUPPLYDROP, ALLIANCE, GAMELOCATION, MUNITION, PLAYER, STORE, and ATTACK. Their appropriate relationships and attributes are shown in our or Entity Relationship Diagram and Relational Diagram (please see the attached diagrams). A special thanks to Michael Garod for creating Python scripts which generate hundreds of INSERT statements full of randomized data for the purpose of testing. The database was tested by considering possible questions, and writing SQL queries to answer those questions. These SQL test queries and their answers are attached as well. The details of each table in the database are printed in the attached document named DATABASE TABLES. All files can be accessed on GitHub using this URL: <https://github.com/mgarod/CSCI435>.

In conclusion, our team felt that this project was very effective in teaching us how to build, design, and test our database, and above all how to work as a team.