Script – Jacob Dyer

Hello. My name is Jacob Dyer, and this is a portion of my portion of the final video for CS 560. About halfway Luke Duball will take over and then I will return to wrap up our discussion.

Our project focuses on comparing three database systems: MySQL, MongoDB, and Neo4j. What we would like to determine is under what circumstances do these systems perform well in addition to when they perform poorly. To do this we have created a list of queries that we will run on each system and we will use the time it takes to return an answer as a performance benchmark. All the systems will be running locally on our machines so there should not be any network delay involved in our analysis; keeping the results confined to whatever the database is doing to complete the query.

Before I hand off to Luke, I would like to take some time to discuss the data we used for the project.

To make sense of the data we need to first understand where it comes from. Our dataset was pulled from an online nightly data dump of Elite Dangerous. Elite Dangerous is an MMO game set in space where players are the commanders of space vessels. The goal of the game is to amass wealth and to become powerful within the factions. These factions control many star systems within the Milky Way and are usually at constant war. Typically, everything in the game is decided on how many credits you have, credits are the primary currency. There are many ways to accumulate credits. To list a few there is: Trading, Asteroid Mining, Piracy, Private Faction Jobs, and Bounty Hunting. The game is made so that if a player jumps to another planetary system, and that system has yet to be visited by a player, it will procedurally generate a new system and all the planets within that system. This makes the play space somewhat infinite. The nightly data dump is publicly available but a bit limited. The owner of the data dump used to dump all the celestial body information but has stopped doing that because of how many celestial bodies are currently in the game. I think I read it was last about 42 gigs. The files we used cover less about the natural elements, such as celestial bodies, and more about the civilization that has been made.

Most of the people in this universe live on space stations. These stations orbit some celestial body and are central to the entire game. Stations are where the player will trade/barter, sell/buy ships, as well as interact with non-player factions and player factions alike. All this information is located within our first set of data: Stations. Most of the data we get from the dump is formatted in json so there are nested elements beyond properties like a name and location. Stations also contain services like repairing, refueling, and a marketplace. Important foreign keys here are: System ID, which denotes which planetary system the station is in, and imports/exports, which is an array of commodity IDs that show what this station might produce and what it needs to produce that commodity.

The systems file we chose to use for our project only contains the systems that have people living there. The other system file was over 5 Gigs and we can justify this because the main concern with our queries is trade and services, so we did not see a need to bog everything down with systems that had neither. Important foreign keys to notice are