**Terminal Assessment Introduction to Databases**

Project Name: Hibernian Companies

Student Name: Ronan Behan

Student Number: 19141815

Course: Higher Diploma in Science in Computing, Software Development

Brief Report on Terminal Introduction to Databases

The last deliverable is a brief report outlining each of the technologies used for analysis, high level modelling and physical implementation, and where they were used.

This was a fascinating project to be part of. My appreciation for Databases has grown as well as my knowledge. I feel like my understanding of High Level Modelling has grown quite a lot as well as my ability to create a database and all the queries in SQL.

I decided to create a Database around the idea of army companies, 3 fake tongue in cheek military companies called the Hibernian Companies.

I started with the High Level Model, an ERD. To create this I used LucidChart as I already have it on my computer. I quite like it. It’s easy to use. I started by writing my plans on paper to get an idea of what I am doing but quickly decided that LucidChart would be much easier.

In my ERD I have 8 entities – Company, Company\_Staff, Weapons, Vehicle, Job, Platoon, Officers and Soldiers. I gave them all their attributes. Soldiers and Officers having the majority. Every class has a primary key and many have foreign keys too.

I have indicated the Superclass, subclass. I have indicated attributes, relationships, the participation and cardinality are clear to see, and keys.

I think I did a decent job here and have fulfilled all the criteria asked. It is logical and straightforward.

Next I moved onto the Physical Implementation. I use MySQL workbench for this.

Here are a summary of the queries I used to create such an extensive table.

I created the Database Hibernian\_Companies. I created all the tables (The entities in the ERD).

I have inserted all the values required.

On line 22 I have used a COUNT query to the number of companies per location.

I renamed the Staff Table to Company\_Staff.

I created the table Job, I used the ALTER to add a new column called Rank\_Assigned. Then I used RENAME to rename the column Rank\_Info to ID to align it to other table’s Primary Keys. I did this to create a more fluid table.

Next I created the Officer table, similar to last time I renamed Officer\_ID to ID for Fluidity, same as the last rename.

Next drop a column from Officers; Rank\_info. I got rid of this as I decided to change the same column to ID in the Job Table.

I then went on to create the Soldiers table. I dropped the age column. I renamed the staff ID to ID, again for fluidity.

Again for the same reason as last time I removed the column Rank\_info from soldiers.

Next I wanted to delete a person from the soldiers. I delete the person with ID 22 for reasons stated in the code. WHERE being used to indicate the criteria from deletion.

Next I used a query to insert a new soldier, I gave him his attributes.

Then I proceeded to use the Update query to give the new soldier a new name.

Next I created a table Dogs.

I used the truncate query to wipe the attributes from the table.

Then I dropped the table as it was no longer of use.

After this I added a lot of queries to create the foreign key to link all the tables.

Next I did some aggregate queries to get an idea about salary. These are SUM and MAX.

Next I have a few Select Queries. The first is to bring up the ID, Name and Weapon values from the Officer table from the platoon No.1. WHERE the platoon ID is no.1.

The second is bring up the IDs for everyone with 1 year of service. WHERE there is 1 year of service.

Next is 2 querys for Group By.

Next My SELECT Query is to bring up the columns Salary, ID and Years of Service from the Job table where the salary is less than 50,000 and GROUP BY puts similar values together, order by puts the lowest at the top, ID is next and Years of Service as stated in the Select query.

Next I am selecting the years of Service from the Job table, where the years of service is less then 10 years and I’m grouping by the ID.

Next 2 queries are the Joins.

First I use a simple join. I joined the soldiers and officers tables together.

Second I am selecting all from the soldiers table, then joining it to the Job table, it's matched by the ID, and an inner join will give the values which are common.

This is my brief report for Section one. It took me a long time and I am quite proud. It contains most of the asked criteria.