I’m putting my SQL code here just in case you can’t get access to the sql file I have uploaded.

The second deliverable is the Physical Implementation and SQL Queries. Comments must be used in the code to provide an understanding to the reader. [Provide screenshots with your studentID]

The student ID is part of the File.

CREATE DATABASE Hibernian\_Companies;

USE Hibernian\_Companies;

CREATE TABLE Company(

Company\_ID INT PRIMARY KEY NOT NULL,

LOCATION VARCHAR(50) NOT NULL,

PHONE\_NO INT NOT NULL,

EMAIL VARCHAR(50),

SUPPLIED VARCHAR(50) NOT NULL);

ALTER TABLE Company ADD OFFICER\_ID INT; -- adding officer\_ID as it was left out

SELECT \* FROM Company;

SELECT \* FROM Company;

INSERT INTO Company VALUES (1, "Munster", 021, "Company1@gmail.com", "In Supply");

INSERT INTO Company VALUES (2, "Leinster", 022, "Company2@gmail.com", "In Supply");

INSERT INTO Company VALUES (3, "Connaught", 023, "Company3@gmail.com", "In Supply");

-- COUNT, this counts the number of companies per location

SELECT COUNT(COMPANY\_ID), LOCATION FROM Company Group BY LOCATION ORDER BY LOCATION DESC;

CREATE TABLE Staff(

ID INT PRIMARY KEY,

COMPANY\_ID INT,

Officer\_ID INT);

RENAME TABLE Staff TO Company\_Staff; -- here we are renaming the name of the table

ALTER TABLE Company\_Staff DROP COLUMN Officer\_ID; -- here we are dropping officer ID as we want to give all staff just an ID to make everything more fluid, it will also create greater harmony amongst the ranks

-- entering the soldiers IDs and company numbers

INSERT INTO Company\_Staff VALUES (1, 1); -- major of the 3 companies

INSERT INTO Company\_Staff VALUES (2, 1); -- captain of the first company

INSERT INTO Company\_Staff VALUES (3, 2); -- captain of the second company

INSERT INTO Company\_Staff VALUES (4, 3); -- captain of the third company

INSERT INTO Company\_Staff VALUES (5, 1); -- Lt

INSERT INTO Company\_Staff VALUES (6, 2); -- Lt

INSERT INTO Company\_Staff VALUES (7, 3); -- Lt

INSERT INTO Company\_Staff VALUES (8, 1); -- SGT

INSERT INTO Company\_Staff VALUES (9, 2); -- SGT

INSERT INTO Company\_Staff VALUES (10, 3); -- SGT

INSERT INTO Company\_Staff VALUES (11, 1); -- Cpl

INSERT INTO Company\_Staff VALUES (12, 2); -- Cpl

INSERT INTO Company\_Staff VALUES (13, 3); -- Cpl

INSERT INTO Company\_Staff VALUES (14, 1); -- Cpl

INSERT INTO Company\_Staff VALUES (15, 2); -- Cpl

INSERT INTO Company\_Staff VALUES (16, 3); -- Cpl

INSERT INTO Company\_Staff VALUES (17, 1); -- Pvt

INSERT INTO Company\_Staff VALUES (18, 2); -- Pvt

INSERT INTO Company\_Staff VALUES (19, 3); -- Pvt

INSERT INTO Company\_Staff VALUES (20, 1); -- Pvt

INSERT INTO Company\_Staff VALUES (21, 2); -- Pvt

INSERT INTO Company\_Staff VALUES (22, 3); -- Pvt

SELECT \* FROM Company\_Staff; -- here we can see all in the table

CREATE TABLE Weapons(

Weapon\_ID INT PRIMARY KEY NOT NULL,

Weapon\_Type VARCHAR(50));

INSERT INTO Weapons VALUES (1, "Comfy Chair"); -- major of the 3 companies

INSERT INTO Weapons VALUES (2, "Sword"); -- captain of the first company

INSERT INTO Weapons VALUES (3, "Blunderbuss"); -- captain of the second company

INSERT INTO Weapons VALUES (4, "Lance"); -- captain of the third company

INSERT INTO Weapons VALUES (5, "broken scissors"); -- Lt

INSERT INTO Weapons VALUES (6, "Knife"); -- Lt

INSERT INTO Weapons VALUES (7, "Rusty Spoon"); -- Lt

INSERT INTO Weapons VALUES (8, "Anti AirCraft Gun"); -- SGT

INSERT INTO Weapons VALUES (9, "MG"); -- SGT

INSERT INTO Weapons VALUES (10, "MG"); -- SGT

INSERT INTO Weapons VALUES (11, "Spoon"); -- Cpl

INSERT INTO Weapons VALUES (12, "Rifle"); -- Cpl

INSERT INTO Weapons VALUES (13, "Rifle"); -- Cpl

INSERT INTO Weapons VALUES (14, "Rifle"); -- Cpl

INSERT INTO Weapons VALUES (15, "Rifle"); -- Cpl

INSERT INTO Weapons VALUES (16, "Rifle"); -- Cpl

INSERT INTO Weapons VALUES (17, "Stone"); -- Pvt

INSERT INTO Weapons VALUES (18, "Rifle"); -- Pvt

INSERT INTO Weapons VALUES (19, "Rifle"); -- Pvt

INSERT INTO Weapons VALUES (20, "Rifle"); -- Pvt

INSERT INTO Weapons VALUES (21, "Rifle"); -- Pvt

INSERT INTO Weapons VALUES (22, "Rifle"); -- Pvt

SELECT \* FROM Weapons;

-- this selects all the data on the weapons table

CREATE TABLE Vehicle(

Vehicle\_ID INT PRIMARY KEY NOT NULL,

Vehicle\_Type VARCHAR(50));

INSERT INTO Vehicle VALUES (1, "Staff Car");

INSERT INTO Vehicle VALUES (2, "Golf");

INSERT INTO Vehicle VALUES (3, "Subaru Impreza WRX");

INSERT INTO Vehicle VALUES (4, "Lockheed Martin F-35 Lightning II");

INSERT INTO Vehicle VALUES (5, "Bob Semple tank");

INSERT INTO Vehicle VALUES (6, "Datsun");

INSERT INTO Vehicle VALUES (7, "Bob Semple tank");

CREATE TABLE Job(

Rank\_Info INT PRIMARY KEY NOT NULL,

Salary INT,

Years\_of\_Service INT);

ALTER TABLE Job ADD Rank\_Assigned VARCHAR(25); -- here we add another colume to give the name of their rank

ALTER TABLE Job RENAME COLUMN Rank\_Info to ID; -- as we are giving more info about the rank, we will just give IDs for the PK to help allign them with other tables

INSERT INTO Job VALUES (1, 100000, 30, "Major"); -- major of the 3 companies

INSERT INTO Job VALUES (2, 59000, 20, "Captain"); -- captain of the first company

INSERT INTO Job VALUES (3, 52000, 45, "Captain"); -- captain of the second company

INSERT INTO Job VALUES (4, 56000, 7, "Captain"); -- captain of the third company

INSERT INTO Job VALUES (5, 40000, 10, "Lieutenant"); -- Lt

INSERT INTO Job VALUES (6, 42000, 14, "Lieutenant"); -- Lt

INSERT INTO Job VALUES (7, 39000, 11, "Lieutenant"); -- Lt

INSERT INTO Job VALUES (8, 39000, 19, "Sergeant"); -- SGT

INSERT INTO Job VALUES (9, 35000, 3, "Sergeant"); -- SGT

INSERT INTO Job VALUES (10, 30000, 29, "Sergeant"); -- SGT

INSERT INTO Job VALUES (11, 25000, 5, "Corporal"); -- Cpl

INSERT INTO Job VALUES (12, 25000, 4, "Corporal"); -- Cpl

INSERT INTO Job VALUES (13, 23000, 2, "Corporal"); -- Cpl

INSERT INTO Job VALUES (14, 22000, 1, "Corporal"); -- Cpl

INSERT INTO Job VALUES (15, 21000, 1, "Corporal"); -- Cpl

INSERT INTO Job VALUES (16, 25000, 7, "Corporal"); -- Cpl

INSERT INTO Job VALUES (17, 20000, 3, "Private"); -- Pvt

INSERT INTO Job VALUES (18, 20000, 2, "Private"); -- Pvt

INSERT INTO Job VALUES (19, 20000, 0.5, "Private"); -- Pvt

INSERT INTO Job VALUES (20, 20000, 1, "Private"); -- Pvt

INSERT INTO Job VALUES (21, 20000, 1, "Private"); -- Pvt

INSERT INTO Job VALUES (22, 20000, 1, "Private"); -- Pvt

CREATE TABLE Platoon(

ID INT PRIMARY KEY NOT NULL,

trainingTarget VARCHAR(50),

Platoon\_ID INT);

INSERT INTO Platoon VALUES (1, "0KM Run", 1); -- major of the 3 companies

INSERT INTO Platoon VALUES (2, "100kg Deadlift", 2); -- captain of the first company

INSERT INTO Platoon VALUES (3, "5kg Run", 3); -- captain of the second company

INSERT INTO Platoon VALUES (4, "100km Run", 4); -- captain of the third company

INSERT INTO Platoon VALUES (5, "50kg Bench", 5); -- Lt

INSERT INTO Platoon VALUES (6, "40km Run", 6); -- Lt

INSERT INTO Platoon VALUES (7, "40km Run", 1); -- Lt

INSERT INTO Platoon VALUES (8, "40km Run", 1); -- SGT

INSERT INTO Platoon VALUES (9, "40km Run", 2); -- SGT

INSERT INTO Platoon VALUES (10, "40km Run", 3); -- SGT

INSERT INTO Platoon VALUES (11, "40km Run", 1); -- Cpl

INSERT INTO Platoon VALUES (12, "50kg Bench", 2); -- Cpl

INSERT INTO Platoon VALUES (13, "50kg Bench", 3); -- Cpl

INSERT INTO Platoon VALUES (14, "50kg Bench", 4); -- Cpl

INSERT INTO Platoon VALUES (15, "100kg Deadlift", 5); -- Cpl

INSERT INTO Platoon VALUES (16, "100kg Deadlift", 6); -- Cpl

INSERT INTO Platoon VALUES (17, "100kg Deadlift", 1); -- Pvt

INSERT INTO Platoon VALUES (18, "100kg Deadlift", 2); -- Pvt

INSERT INTO Platoon VALUES (19, "50kg Bench", 3); -- Pvt

INSERT INTO Platoon VALUES (20, "50kg Bench", 4); -- Pvt

INSERT INTO Platoon VALUES (21, "50kg Bench", 5); -- Pvt

INSERT INTO Platoon VALUES (22, "50kg Bench", 6); -- Pvt

CREATE TABLE Officers(

Officer\_ID INT PRIMARY KEY NOT NULL,

Name VARCHAR(50) NOT NULL,

Distinction VARCHAR(50),

Weapon\_ID INT NOT NULL,

Vehicle\_ID INT NOT NULL,

Platoon\_ID INT,

Rank\_Info INT NOT NULL);

ALTER TABLE Officers RENAME COLUMN Officer\_ID to ID; -- here we alter the name of a column, so that we this table's PK will be the same as that of Company\_Staff, the PK will aslo be a FK

ALTER TABLE Officers Drop COLUMN Rank\_info; -- we don't need this any more as we can just bring up the ID to find the rank info

INSERT INTO Officers VALUES (1, "John Burke", NULL, 1, 1, 1); -- major of the 3 companies

INSERT INTO Officers VALUES (2, "Jimmy Miggey","Cleaned the latrines during a storm", 2, 2, 2); -- captain of the first company

INSERT INTO Officers VALUES (3, "Francis Chute", NULL, 3,3,3); -- captain of the second company

INSERT INTO Officers VALUES (4, "Bridie McEllegot", "Fought the savages of Galway", 4,4,4); -- captain of the third company

INSERT INTO Officers VALUES (5, "John Doe", NULL, 5,5,5); -- Lt

INSERT INTO Officers VALUES (6, "Al Smith", "Not the Devil after all", 6,6,6 ); -- Lt

INSERT INTO Officers VALUES (7, "Mary Francis", "Decent Person", 7,7,1); -- Lt

CREATE TABLE Soldiers(

Staff\_ID INT PRIMARY KEY NOT NULL,

Name VARCHAR(50) NOT NULL,

Age INT NOT NULL,

Weapon\_ID INT NOT NULL,

Platoon\_ID INT NOT NULL,

Rank\_Info INT NOT NULL);

ALTER TABLE Soldiers DROP COLUMN age; -- here I am using the Alter column to get rid of the column age, as in this company we are not agist and we are as young as we feel

ALTER TABLE Soldiers RENAME COLUMN Staff\_ID to ID; -- here we alter the name of a column, so that we this table's PK will be the same as that of Company\_Staff, the PK will aslo be a FK

ALTER TABLE Soldiers Drop COLUMN Rank\_info; -- we don't need this any more as we can just bring up the ID to find the rank info

INSERT INTO Soldiers VALUES (8, "Karl Michaels", 8, 1); -- SGT

INSERT INTO Soldiers VALUES (9, "Amur Chitkakhana", 9, 2); -- SGT

INSERT INTO Soldiers VALUES (10, "Jake Ovens", 10, 3); -- SGT

INSERT INTO Soldiers VALUES (11, "Eustice O'Shea", 11, 1); -- Cpl

INSERT INTO Soldiers VALUES (12, "Owen Sheil", 12, 2); -- Cpl

INSERT INTO Soldiers VALUES (13, "Cristina Hara", 13, 3); -- Cpl

INSERT INTO Soldiers VALUES (14, "John Browne", 14, 4); -- Cpl

INSERT INTO Soldiers VALUES (15, "Happy Summers", 15, 5); -- Cpl

INSERT INTO Soldiers VALUES (16, "Patrick Lynch", 16, 6); -- Cpl

INSERT INTO Soldiers VALUES (17, "Brion O'Donnell", 17, 1); -- Pvt

INSERT INTO Soldiers VALUES (18, "David Burns", 18, 2); -- Pvt

INSERT INTO Soldiers VALUES (19, "Leo O'Connell", 19, 3); -- Pvt

INSERT INTO Soldiers VALUES (20, "Laura Goodwin", 20, 4); -- Pvt

INSERT INTO Soldiers VALUES (21, "Stephen Collins", 21, 5); -- Pvt

INSERT INTO Soldiers VALUES (22, "Erik YellowBelly", 22, 6); -- Pvt

DELETE FROM Soldiers WHERE ID= 22; -- here we delete the row from soldiers where the ID is 22, Erik YellowBelly always charges the enemy without orders to do so

SELECT ALL \* FROM Soldiers; -- you can see ID 22, Erik YellowBelly is gone

-- new conscript

INSERT INTO Soldiers VALUES (22, "Willy Dillon", 22, 6); -- Pvt

UPDATE Soldiers SET Name = 'William A. Dillane' WHERE ID= 22;

SELECT ALL \* FROM Soldiers; -- The misspelling of William's name has been corrected

CREATE TABLE Dogs(

Doggy\_ID INT PRIMARY KEY);

ALTER TABLE Dogs ADD Name VARCHAR(10);

INSERT INTO Dogs VALUES (1, "Nelly");

INSERT INTO Dogs VALUES (2, "Mindy");

INSERT INTO Dogs VALUES (3, "Bruce");

SELECT ALL \* FROM Dogs;

TRUNCATE TABLE Dogs; -- This is a truncate query, truncate gets rid of all the data in a table

SELECT ALL \* FROM Dogs; -- all the values we inserted is gone

-- here we Delete the table Dogs as the dogs have been give their own company, and I want to show how to get rid of a table from the database

DROP TABLE Dogs;

SELECT ALL \* FROM Dogs; -- if you use this you can see the table is gone

-- Later changes to the tables

-- NOW I AM LINKING ALL THE TABLES, THE PRIMARY OF SOME TABLES TO THE FOREIGN KEYS OF OTHERS

-- As I created a shell of tables then added data, I now want to assign Foreign Key's to link the tables

ALTER TABLE Company\_Staff ADD FOREIGN KEY (Company\_ID) REFERENCES Company (Company\_ID); -- linking the Company\_Staff to Company throught Company\_ID

ALTER TABLE Officers ADD FOREIGN KEY (ID) REFERENCES Company\_Staff (ID); -- maybe a fuck up here as we had ADD FOREIGN KEY (ID) for the ones down below

ALTER TABLE Officers ADD FOREIGN KEY (Weapon\_ID) REFERENCES Weapons (Weapon\_ID);

ALTER TABLE Officers ADD FOREIGN KEY (Vehicle\_ID) REFERENCES Vehicle (Vehicle\_ID);

ALTER TABLE Officers ADD FOREIGN KEY (ID) REFERENCES Platoon (ID);

ALTER TABLE Soldiers ADD FOREIGN KEY (ID) REFERENCES Company\_Staff (ID);

ALTER TABLE Soldiers ADD FOREIGN KEY (Weapon\_ID) REFERENCES Weapons (Weapon\_ID);

ALTER TABLE Soldiers ADD FOREIGN KEY (ID) REFERENCES Platoon (ID);

ALTER TABLE Job ADD FOREIGN KEY (ID) REFERENCES Company\_Staff (ID);

ALTER TABLE Platoon ADD FOREIGN KEY (ID) REFERENCES Job (ID);

-- here are some agregate functions

-- we want to find the sum of all salaries of all the people in our companies

SELECT SUM(salary) FROM Job;

-- we want to find the largest(The Major's) of all salaries of all the people in our companies

SELECT MAX(salary) FROM Job;

-- Here we are bringing up the ID, Name and Weapon values from the Officer table from the platoon No.1

SELECT ID, Name, Weapon\_ID FROM Officers WHERE Platoon\_ID = 1;

-- bring up the IDs for everyone with 1 year of service

SELECT ID FROM Job WHERE Years\_of\_Service = 1;

-- this is a query which selects all the salaries where they are below 50000 euros, 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

SELECT Salary, ID, Years\_of\_Service

FROM Job

WHERE Salary < 50000

GROUP BY Salary

ORDER BY Salary;

-- list the weapon IDs from every weapon ID 8 up

SELECT Weapon\_ID, Name, ID

FROM Soldiers

WHERE Weapon\_ID > 8

GROUP BY Weapon\_ID;

-- group by ID, so decending order of ID, the condition is where all the years above 10

SELECT Years\_of\_Service

FROM Job

WHERE Years\_of\_Service > 10

GROUP BY ID;

-- Joins

-- simple join, normal join which joins the soldier table with the Officer table

SELECT \*

FROM Soldiers

JOIN Officers;

-- Inner join, here we are selecting all from the soldiers table, then joining it to the Job table, it's match by the ID, inner join will give the values which are common

SELECT \* FROM Soldiers

Inner Join Job

ON Soldiers.ID = Job.ID;