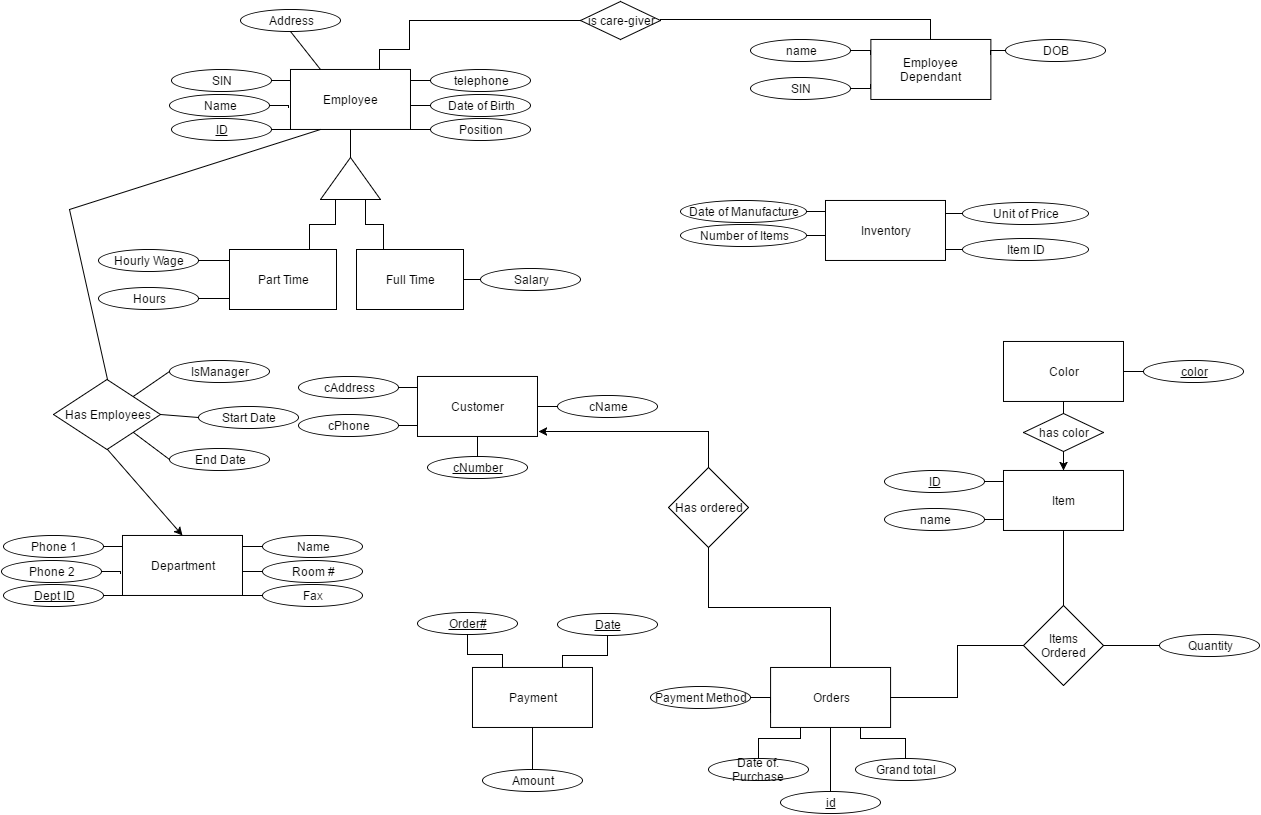
Concordia University  
Computer Science and Software Engineering  
COMP 353  
Database Project -Report

Patrick Young (27016085)  
Jonathan Bingham (27440545)  
Frederic Secco (26986080)  
Gabriel Hardy (6975739)

Submitted on: April 11, 2016

1. E/R diagram

2. Relational Schemas  
  
**Employee**(Id, SSN, Name, Tele, Email, DOB, Address, Position)  
  
**EmployeeDependant**(SSN, Name, DOB)  
  
**Customer**(CusNumber, Name, Tele, Address)  
  
**Orders**(Id, PaymentMethod, DOPurchase, Balance)  
  
**Item**(Id, Name)  
  
**Colors**(Color)  
  
**Department**(Id, Name, Tele1, Tele2, RoomNumber, FaxNumber)  
  
**PartTimeEmployee**(Id, HourlyWage, Hours)  
 Id is a foreign key referencing Employee.Id  
  
**FullTimeEmployee**(Id, Wage)   
Id is a foreign key referencing Employee.Id  
  
**IsCareGiver**(Id, SSN)   
Id references employee id, SSN is unique and references Employee.SSN  
  
**Inventory**(ItemId, Price, DOManufacture, Quantity)   
ItemId references item.id  
  
**Payment**(OrderId, PaymentDate, Amount)   
OrderId references Orders.Id  
  
**HasOrdered**(Cnumber, OrderId)  
Cnumber references Customer.CusNumber, OrderId references Orders.Id  
  
**HasColor**(ItemId, Color)  
ItemId references Item.Id, Color references Colors.Color  
  
**ItemsOrdered**(OrderId, ItemId, Quantity)  
OrderId references Orders.Id, ItemId references Item.Id  
  
**HasEmployees**(DeptId, EmpId, isManager, StartDate, EndDate)  
DeptId references Department.Id, EmpId references Employee.Id  
  
Based off of creation script:   
DROP TABLE IF EXISTS PartTimeEmployee;  
DROP TABLE IF EXISTS FullTimeEmployee;  
DROP TABLE IF EXISTS IsCareGiver;  
DROP TABLE IF EXISTS Inventory;  
DROP TABLE IF EXISTS Payment;  
DROP TABLE IF EXISTS HasOrdered;  
DROP TABLE IF EXISTS HasColor;  
DROP TABLE IF EXISTS ItemsOrdered;  
DROP TABLE IF EXISTS HasEmployees;  
DROP TABLE IF EXISTS Employee;  
DROP TABLE IF EXISTS EmployeeDependant;  
DROP TABLE IF EXISTS Customer;  
DROP TABLE IF EXISTS Orders;  
DROP TABLE IF EXISTS Item;  
DROP TABLE IF EXISTS Colors;  
DROP TABLE IF EXISTS Department;CREATE TABLE Employee (  
Id INT(6) PRIMARY KEY,  
SSN INT(9) NOT NULL,  
Name VARCHAR(30) NOT NULL,  
Telephone VARCHAR(12) NOT NULL,  
email VARCHAR(50) NOT NULL,  
DateOfBirth DATE NOT NULL,  
Address VARCHAR(30) NOT NULL,  
Position VARCHAR(30) NOT NULL  
);CREATE TABLE EmployeeDependant(  
SSN INT(9) PRIMARY KEY,  
Name VARCHAR(30) NOT NULL,  
DateOfBirth DATE NOT NULL  
);CREATE TABLE Customer(  
CustomerNumber INT(10) PRIMARY KEY,  
Name VARCHAR(30) NOT NULL,  
Telephone VARCHAR(12) NOT NULL,  
Address VARCHAR(30) NOT NULL  
);CREATE TABLE Orders(  
Id INT(10) PRIMARY KEY,  
PaymentMethod ENUM('FULL','INSTALLMENT') NOT NULL,  
DateOfPurchase DATE NOT NULL,  
Balance FLOAT(10,2) NOT NULL  
);CREATE TABLE Item(  
Id INT(10) PRIMARY KEY,  
Name VARCHAR(30) NOT NULL  
);CREATE TABLE Colors(  
Color VARCHAR(30) PRIMARY KEY  
);CREATE TABLE Department(  
Id INT(4) PRIMARY KEY,  
Name VARCHAR(30) NOT NULL,  
PhoneNumber1 VARCHAR(12) NOT NULL,  
PhoneNumber2 VARCHAR(12),  
RoomNumber INT(3) NOT NULL,  
FaxNumber VARCHAR(12)  
);CREATE TABLE PartTimeEmployee(  
Id INT(6) PRIMARY KEY,  
HourlyWage FLOAT(3,2) NOT NULL,  
Hours INT(2),  
FOREIGN KEY(Id) REFERENCES Employee(Id)  
);CREATE TABLE FullTimeEmployee(  
Id INT(6) PRIMARY KEY,  
Wage FLOAT(7,2) NOT NULL,  
FOREIGN KEY(Id) REFERENCES Employee(Id)  
);CREATE TABLE IsCareGiver(  
Id INT(6),  
SSN INT(9) UNIQUE,  
PRIMARY KEY (Id,SSN),  
FOREIGN KEY (Id) REFERENCES Employee(Id),  
FOREIGN KEY (SSN) REFERENCES EmployeeDependant(SSN)  
);CREATE TABLE Inventory(  
ItemId INT(10) PRIMARY KEY,  
Price FLOAT(10,2) NOT NULL,  
DateOfManufacture DATE NOT NULL,  
Quantity INT(15) NOT NULL,  
FOREIGN KEY (ItemId) REFERENCES Item(Id)  
);CREATE TABLE Payment(  
OrderId INT(10),  
PaymentDate DATE,  
Amount FLOAT(10,2) NOT NULL,  
PRIMARY KEY (OrderId, PaymentDate),  
FOREIGN KEY (OrderID) REFERENCES Orders(Id)  
);CREATE TABLE HasOrdered(  
CNumber INT(10),  
OrderId INT(10),  
PRIMARY KEY (CNumber, OrderId),  
FOREIGN KEY (CNumber) REFERENCES Customer(CustomerNumber),  
FOREIGN KEY (OrderId) REFERENCES Orders(Id)  
);CREATE TABLE HasColor(  
ItemId INT(10),  
Color VARCHAR(30),  
PRIMARY KEY (ItemId,Color),  
FOREIGN KEY (ItemId) REFERENCES Item(Id),  
FOREIGN KEY (Color) REFERENCES Colors(Color)  
);CREATE TABLE ItemsOrdered(  
OrderId INT(10),  
ItemId INT(10),  
Quantity INT(4),  
PRIMARY KEY (OrderId, ItemId),  
FOREIGN KEY (OrderId) REFERENCES Orders(Id),  
FOREIGN KEY (ItemId) REFERENCES Item(Id)  
);CREATE TABLE HasEmployees(  
DeptId INT(4),  
EmpId INT(6),  
IsManager BOOLEAN NOT NULL,  
StartDate DATE NOT NULL,  
EndDate DATE,  
PRIMARY KEY (DeptId, EmpId),  
FOREIGN KEY (DeptId) REFERENCES Department(Id),  
FOREIGN KEY (EmpId) REFERENCES Employee(Id)

3.   
**Technical details (implementation/design)**

   The project consists of a database system for corporate management. We have designed a web browser software product that is responsive in generating company based information. The project is designed with a front end using Bootstrap libraries and the back end is hosted on the Concordia server which uses PHP. We are also using a framework called Slim that is modeled after MVC (model, view and controller) architecture. The framework consists of PHP logic files associated with the calculation of data based on inputs generated in the front end which is a collection of HTML (actually phtml) files. The HTML files act as an interface and the PHP files are used to connect and fetch MySQL database information depending on the query. The implementation of CRUD (create, replace, update, delete) or Add/Modify/Delete is done by the Slim PHP framework; it handles routing, dependencies and auto loading.

**Special features**

-The implementation of Slim framework.  
-Bootstrap libraries  
-Mobile friendly.

**Functionalities**

The browser application allows the user, through an interface, to navigate and view the result of 6 queries. The queries are based upon user input (text area or button) which is sent to the PHP parser. Processes the information in-putted in accordance to the MySQL database query. Executes the query. Then returns the result of the query to the HTML View.

The user can also, through an interface, Add/Modify/Delete information regarding employees, customers, departments, orders, items, inventory and shipment details.

**Members contributions**  
**Gabriel Hardy**: E/R Diagrams and Schemas. Query implementation at PHP level. Git Hub master. Back end PHP design.   
**Fred Secco**: E/R Diagrams and Schemas. Query implementation at database level. MySql table creation and population.   
  
**Patrick Young**: E/R Diagrams and Schemas. Query implementation at PHP level. Git Hub master. SLIM framework Controller (Back end to Front end). Back end PHP design.   
**Jonathan Bingham**: E/R Diagrams and Schemas. Query implementation at HTML level. Front end HTML design. Author of report.  
  
Some contributions are shared amongst all team members, such as the use of Slack to communicate and share project related materiel, Git Hub was used by every team mate. All tea members assisted in planning and meeting up physically several times. All team members experienced a learning process with the combinations of technologies used to achieve our product. All technologies used were up-to-date project management tools such as Slack, Git Hub, Bootstrap and Slim are similar to a professional workplace environment. All members had to keep in mind abstract notions of what everyone else was doing and how their work combined to form an end product.

