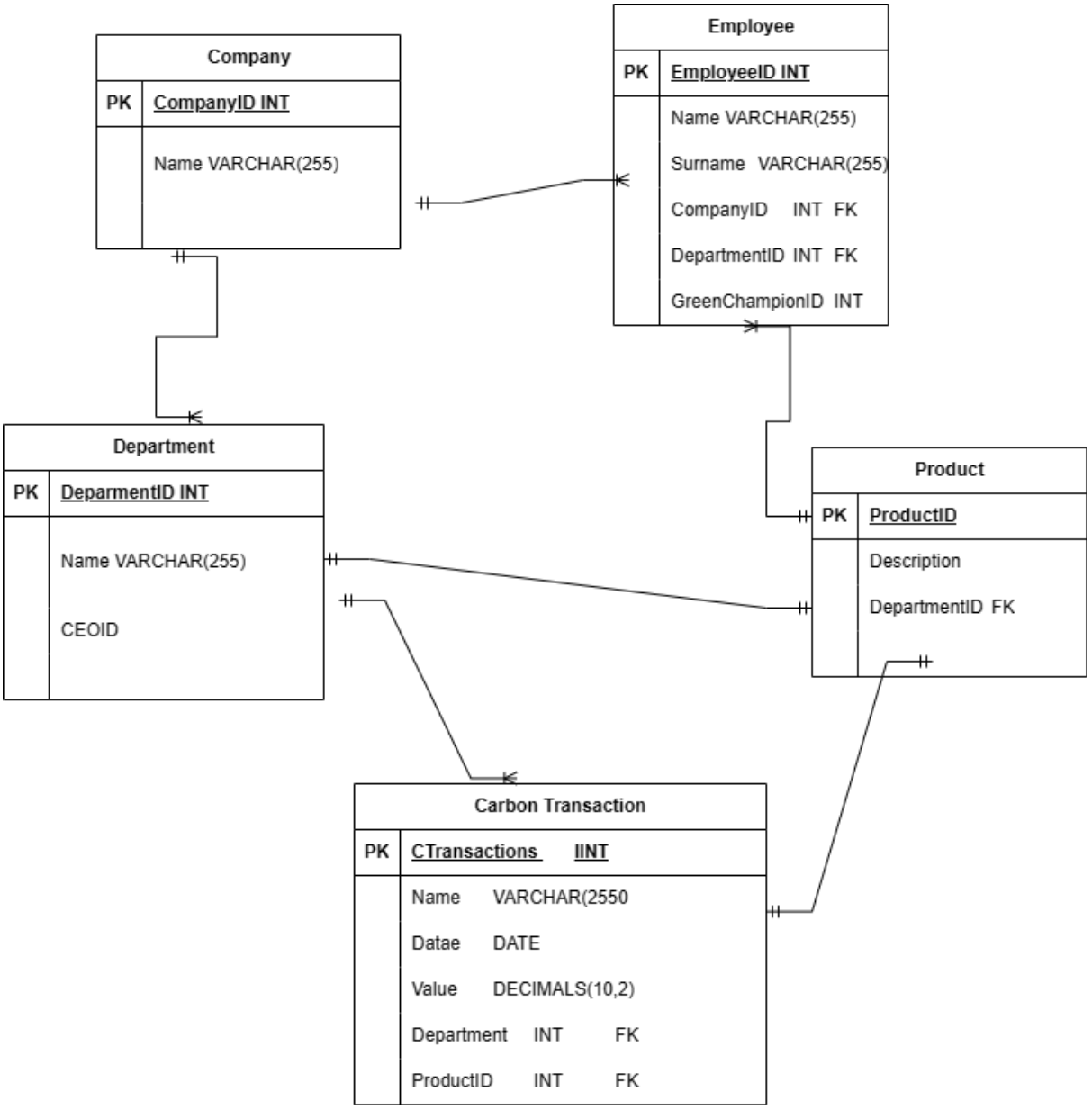
CO2102 Databases and Domain Modelling - Project1 - Ey51

Question 1 DB Green Project - Business Rules

1.1. Draw an ER model according to the above business rules

.



Question 2

Normalise the data below, showing all steps

Player Name	Player Date Joined	Character Name	Character Level	Character Max health
Bob	2014	Geo	50	800
Bob	2014	Fred	50	1200
Alice	2015	ZeAlice	21	350
CHarles	2020-02-29	His Royal Higness	30	470
Delta	2020	WarriorDelta	50	1205
Delta	2020	DeltaMage	2	60

Step 1 - Create a new table with a new primary key(unique identifier)

PlayerID	Player Name	Date Joined
1	Bob	2014
2	Alice	2015
3	Charles	2016
4	Delta	2020

Step 2 - Create another new table with a primary key relating to then characters

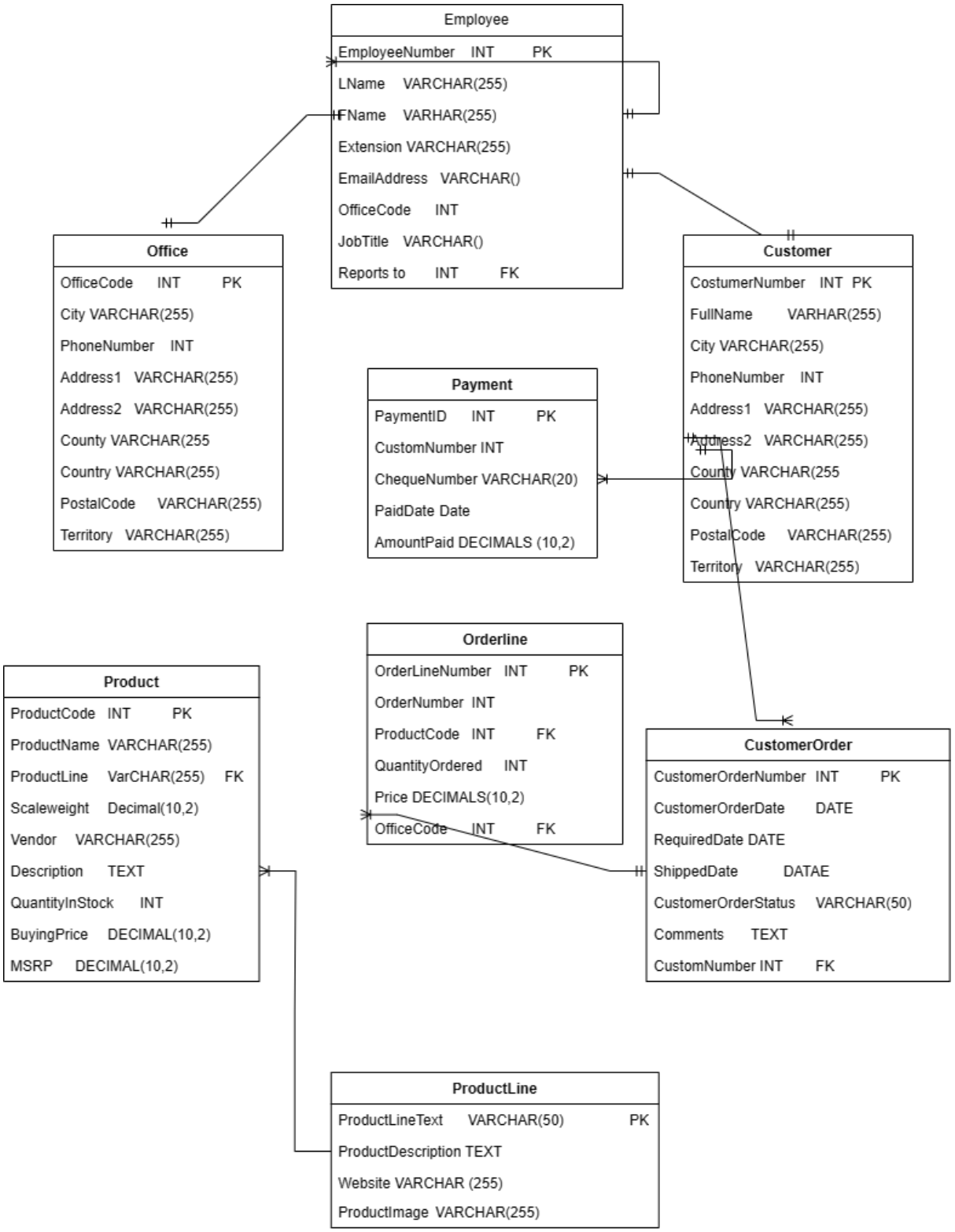
Character ID	Character Name
1	George
2	Fred
3	ZeAlice
4	HisRoyalHighness
5	WarriorDelta
6	Delta

Step 3 - Make a table that relates the two previously created tables

PlayerID	CharacterID	Character Level	Character Max Level
1	1	50	800
1	2	50	1200
2	3	21	350
3	4	30	470
4	5	50	1205
4	6	2	60

The original form is in an unnormalised state where essentially it is just a table that represents data of the player and characters. The transition to a normalised form involves breaking down the original data into smaller, related tables, with each table having a specific purpose and connected through relationships. This normalised form is more efficient, organised, and structured for data management and querying.
Furthermore allows for less data redundancy, ensures data integrity, and enhance data consistency.

Question 3
3.1



3.2-3.4 is below

```
-- Q3.2-- Create Employee table
CREATE TABLE Employee (
    EmployeeNumber INT PRIMARY KEY,
    LastName VARCHAR(255),
    FirstName VARCHAR(255),
    Extension VARCHAR(10),
    EmailAddress VARCHAR(255),
    OfficeCode INT,  -- Foreign key referencing Office
    JobTitle VARCHAR(255),
    ReportsTo INT,  -- Foreign key referencing EmployeeNumber
    FOREIGN KEY (ReportsTo) REFERENCES Employee(EmployeeNumber)
);

-- Create Office table
CREATE TABLE Office (
    OfficeCode INT PRIMARY KEY,
    City VARCHAR(255),
    PhoneNumber VARCHAR(15),
    AddressLine1 VARCHAR(255),
    AddressLine2 VARCHAR(255),
    County VARCHAR(255),
    Country VARCHAR(255),
    PostalCode VARCHAR(10),
    Territory VARCHAR(255)
);

-- Create Customer table
CREATE TABLE Customer (
    CustomerNumber INT PRIMARY KEY,
    FullName VARCHAR(255),
    PhoneNumber VARCHAR(15),
    AddressLine1 VARCHAR(255),
    AddressLine2 VARCHAR(255),
    City VARCHAR(255),
    County VARCHAR(255),
    PostalCode VARCHAR(10),
    Country VARCHAR(255),
    SalesAmount DECIMAL(10, 2),
    RepresentativeEmployeeNumber INT,  -- Foreign key referencing EmployeeNumber
    CreditLimitNumber INT
);

-- Create Payment table
CREATE TABLE Payment (
    PaymentID INT PRIMARY KEY,
    CustomerNumber INT,  -- Foreign key referencing CustomerNumber
```

```
    ChequeNumber VARCHAR(20),
    PaymentDate DATE,
    AmountPaid DECIMAL(10, 2)
);

-- Create Product table
CREATE TABLE Product (
    ProductCode INT PRIMARY KEY,
    ProductName VARCHAR(255),
    ProductLine VARCHAR(255),  -- Foreign key referencing ProductLineText
    ScaleWeight DECIMAL(10, 2),
    Vendor VARCHAR(255),
    ProductDescription TEXT,
    QuantityInStock INT,
    BuyingPrice DECIMAL(10, 2),
    MSRP DECIMAL(10, 2)
);

-- Create Order table
CREATE TABLE CustomerOrder (
    CustomerOrderNumber INT PRIMARY KEY,
    CustomerOrderDate DATE,
    RequiredDate DATE,
    ShippedDate DATE,
    CustomerOrderStatus VARCHAR(50),
    Comments TEXT,
    CustomerNumber INT,  -- Foreign key referencing CustomerNumber
    FOREIGN KEY (CustomerNumber) REFERENCES Customer(CustomerNumber)
);

-- Create OrderLine table
CREATE TABLE OrderLine (
    OrderLineNumber INT PRIMARY KEY,
    OrderNumber INT,  -- Foreign key referencing OrderNumber
    ProductCode INT,  -- Foreign key referencing ProductCode
    QuantityOrdered INT,
    Price DECIMAL(10, 2)
);

-- Create ProductLine table
CREATE TABLE ProductLine (
    ProductLineText VARCHAR(50) PRIMARY KEY,
    ProductDescription TEXT,
    Website VARCHAR(255),
    ProductImage VARCHAR(255)
);

-- Add foreign key constraint for Product-ProductLine relationship
```

```
ALTER TABLE Product
ADD FOREIGN KEY (ProductLine) REFERENCES ProductLine(ProductLineText);


-- Q3.3
-- Insert data into Employee table
INSERT INTO Employee (EmployeeNumber, LastName, FirstName, Extension,
EmailAddress, JobTitle, ReportsTo)
VALUES
    (1, 'Yusuf', 'Emmanuel', '2', 'eyusuf@gmail.com', 'Manager', NULL),
    (2, 'Taiwo', 'Moyo', '1', 'moyotaiwo03@icloud.com', 'Drug Supplier', 1),
    (3, 'Odelusi', 'Olu', '112', 'oluodelusi@.com', 'Customer Service Representative', 2),
    (4, 'Christie', 'Reona', '223', 'Rchristie@yahoo.com', 'Customer Service Representative',
2);

-- Insert data into Office table
INSERT INTO Office (OfficeCode, City, PhoneNumber, AddressLine1, AddressLine2,
County, Country, PostalCode, Territory)
VALUES
    (01, 'New York', 'nul'', 'nul', 'null', 'nul', 'nul', 'nul', 'null'),
    (02, 'Canberra', 'nul, 'nul', 'null', 'nul', 'nul', 'nul', 'null'),
    (03, 'London', 'null', 'nul', 'null', 'nul', 'nul', 'nul', 'null'),
    (04, 'Edinburgh', 'null', 'nul', 'null', 'nul', 'nul', 'nul', 'null');

-- Insert data into Customer table
INSERT INTO Customer (CustomerNumber, FullName, PhoneNumber, AddressLine1,
AddressLine2, City, County, PostalCode, Country, SalesAmount,
RepresentativeEmployeeNumber, CreditLimitNumber)
VALUES
    (1, 'Hannah Yusuf', '07897364228', '22 Raymere Street', 'Plumstead', 'London', 'Greater
London', 'SE186BP', 'England', 50.00, 2, 10000),
    (2, 'Michael Paul', '07625928135', '92 Rankin Drive', 'Blackford', 'Edinburgh', 'Midlothian',
'EH96NP', 'Scotland', 200.00, 1, 1000),
    (3, 'Angela Chu', '23745902', '65 Baltic Avenue', 'Atlanta City', 'Atlanta', 'Georgia',
'NJ08401', 'USA', 500.00, 4, 10000),
    (4, 'Eren Jeagar', '2357', '26 Old Ealling Road', 'Leckham', 'Majura', 'Canberra', 'CA95629',
'Australia', 75.00, 3, 5000);

-- Insert data into Payment table
INSERT INTO Payment (PaymentID, CustomerNumber, ChequeNumber, PaymentDate,
AmountPaid)
VALUES
    (1, 1, 'CHQ8262', '2023-10-28', 50.00),
    (2, 2, 'CHQ0192', '2023-11-01', 200.00),
    (3, 3, 'CHQ3243', '2023-11-05', 500.00),
    (4, 4, 'CHQ6483', '2023-11-07', 75.00);

-- Insert data into Product table
```

```
INSERT INTO Product (ProductCode, ProductName, ProductLine, ScaleWeight, Vendor, ProductDescription, QuantityInStock, BuyingPrice, MSRP)
VALUES
    (101, 'Codeine', 'Medicine', 0.5, 'EmmansPharmacy', 'Used to for short term pain relief', 300, 3.00, 8.99),
    (102, 'Painkillers', 'Medicine', 0.4, 'EmmansPharmacy', 'Used to relieve pain and aches', 300, 4.00, 7.99),
    (103, 'Ibuprofen', 'Medicine', 0.6, 'EmmansPharmacy', 'Used to manage and treat pain and fever', 300, 6.00, 14.99),
    (104, 'Vitamin D3 Pills', 'Supplements', 0.8, 'VitaBiotics', 'Used to increase the amount of Vitamin D in the body', 300, 8.00, 11.99);

-- Insert data into Order table
INSERT INTO CustomerOrder (CustomerOrderNumber, CustomerOrderDate, RequiredDate, ShippedDate, CustomerOrderStatus, Comments, CustomerNumber)
VALUES
    (1111, '2023-10-28', '2023-10-28', '2023-11-06', 'Delivered', 'Please enjoy your order', 1),
    (1112, '2023-11-01', '2023-11-11', '2023-11-09', 'Shipped', 'Congrats your order is on the way!', 2),
    (1113, '2023-11-05', '2023-11-15', '2023-11-12', 'Shipped', 'Your order is on transit', 4),
    (1114, '2023-11-07', '2023-11-17', NULL, 'Pending', 'Order processing', 3);


-- Insert data into OrderLine table
INSERT INTO OrderLine (OrderLineNumber, OrderNumber, ProductCode, QuantityOrdered, Price)
VALUES
    (1, 1111, 101, 20, 8.99),
    (2, 1112, 102, 10, 7.99),
    (3, 1113, 103, 10, 14.99),
    (4, 1114, 104, 5, 11.99);

-- Insert data into ProductLine table
INSERT INTO ProductLine (ProductLineText, ProductDescription, Website, ProductImage)
VALUES
    ('Medicine', 'Used to for short term pain relief', 'EmmanPharmacy.com/medicine/Codeniene', 'Codeniene.jpg'),
    ('Medicine2', 'Used to relieve pain and aches', 'EmmanPharmacy.com/medicine/Painkillers', 'Painkillers.jpg'),
    ('Medicine3', 'Used to manage and treat pain and fever', 'EmmanPharmacy.com/medicine/Ibuprofen', 'Ibuprofen.jpg'),
    ('Supplemenets', 'Used to increase the amount of Vitamin D in the body', 'Vitabiotitics.com/Supplements/VitaminD3', 'VitaminD3.jpg');

-- Q3.4
-- Create the Customer_Order_Restricted_Info view
CREATE VIEW Customer_Order_Restricted_Info AS
SELECT
```

```
    c.CustomerNumber AS Customer_ID,
    CONCAT(c.FirstName, ' ', c.LastName) AS Customer_FullName,
    CONCAT(c.AddressLine1, ' ', c.AddressLine2, ' ', c.PostalCode, ' ', c.Country) AS
Customer_FullAddress,
    o.CustomerOrderStatus AS OrderStatus,
    ol.QuantityOrdered
FROM
    Customer c
JOIN
    CustomerOrder o ON c.CustomerNumber = o.CustomerNumber
JOIN
    OrderLine ol ON o.CustomerOrderNumber = ol.OrderNumber
WHERE
    c.CreditLimitNumber > 1000
    AND YEAR(o.ShippedDate) < 2010;
```