

Lincoln Garden Center Database Assignment 1

Marcus Alderman: 20%

Bethany Davies: 20%

Aminah Hussain: 20%

Cameron Khan: 20%

Jordan Maslen: 20%

Appraise and Analyse

LGC is a garden center which sells plants, accessories, and offers certain services. From our briefing, we have made the assumption that the garden center itself is a small company. We inferred from this that all customers would be making purchases on a pay as you go basis.

The supplier for our company will only provide accessories to LGC, as the company already grow their own plants, and staff members will be qualified to provide services. We have made the assumption that each staff member is specialised in one type of service, meaning that customers can directly engage with the staff member responsible.

We plan to allow the customer to purchase plants, accessories, and to enquire with relevant staff to obtain quotes for services and further details. There will also be a feature on the database to allow customers to search for all products and services offered to be able to get a better idea before they make their purchase.

The database system will store information on staff, customers and suppliers. This will make it simpler to locate certain information and speed up necessary communications.

Once a customer has made a purchase, an invoice will be generated and sent to the customer along with their order.

Requirements specification - company

1. The system will need to store information on plants
2. The system will need to store information on accessories
3. The system will need to store information on services
4. Staff details will need to be stored on the system
5. Supplier details will need to be stored on the system, to allow for easy communication, and to purchase stock
6. There will be a calculation performed in the system to automatically re-order new stock from the supplier when the stock level gets below a certain level
7. The system will raise a 'Purchase Order' when new stock is required to send to the supplier
8. The system will be established with the assumption that each staff member provides one service
9. There will be a feature to add new staff members
10. There will be a feature to add new plants, accessories or services
11. There will be a feature to delete staff members from the system if they leave the company
12. There will be a feature to delete plants, accessories, or services from the system if the company no longer provides them
13. Staff will be able to add new customers to the system upon request

Requirements specification - customers

1. Customers will be able to purchase plants
2. Customers will be able to purchase accessories
3. Customers will be able to enquire about services the company provides
4. There will be a search function to find any plants, accessories, or services that the customer is interested in
5. Customer orders will be generated once an order is placed
6. Customers will be able to provide feedback to LGC once an order has been placed
7. Customers will have to enter complete forms when making an order as there will be presence checks (validation) to ensure only complete data is sent
8. Customers will have their personal data stored in the system to allow the company to contact them if necessary

Identifying entities

In order to identify our candidate entities, we went through the entirety of the text, picked out the nouns, and then did the following:

1. We removed all the duplicates that we found (e.g. item)
2. We then removed nouns which we knew meant the same thing (e.g. “customer” and “customers”)
3. We removed all the nouns which could be used later as attributes, or that we believed weren’t ideal for entities

The following is what we chose to be our candidate keys: plants; accessories; services; customers; customer order; staff; supplier; feedback; and purchase order

Identification of entities

■ Noun _ Candidate entity

Lincoln Garden Centre is a company based in Lincolnshire, which specialises in planting, growing and selling plants, accessories, and relevant services to customers. A Customer can place an order with the company for a specific item, such as a flower bouquet or garden pots. The Customer Order consists of the customer's name, the delivery address, the billing address, the item(s) ordered, the quantity of each item ordered, the order date, the order status (e.g. pending, confirmed, dispatched date, paid), the total price, the date the order was despatched, date the invoice was sent to the customer and the date the payment was received. The Customer Order also contains further detail that is required: price by item, how many are required, discount and total price by item. Based on the Customer Order, an invoice will be issued and sent to the customer after the customer has completely paid the invoice. The Customer Order is then sent to the garden department, which packs the ordered items and sent to the customer, along with a copy of the paid invoice and plant care instructions. This is shortly followed by a timely request for customer feedback. When standard Stock Items (e.g. apple-tree) are out of stock, a Purchase Order is raised from a third party supplier, which consists of the supplier's name, the supplier's address, the description of the stock items required, the quantity of each stock item required, the order date, the order status (e.g. pending, sent, received, paid), the total cost, the date the order was received, and the date the payment was sent. The Purchase Order is then sent to a Supplier who then sends the stock items to LGC, which are received and stored in a garden centre until needed. The Stock Items (flower, plant) information should include (at least): Latin name, Popular Name, Plant attributes (such as colour of foliage, colour of flowers, flowering period, height and spread for example, etc.), plant category (Shrub, tree, climber), Season of Interest (Any, Spring, Summer, Autumn, Winter), Full description, How to grow (Soil type, Moisture, Acid or Alkaline, sunlight {Full sun, half shade,} etc.) and Aftercare. To be able to process Customer Orders and Purchase Orders, the LGC ordering system also needs to store information (e.g. email, address, names, etc.) about Customers, Suppliers and LGC Staff

Justification of entities

The entities we chose to be apart of our system are:

1. Customer: we decided on this to be an entity on our system due to multiple factors. The company itself stated that they needed to be able to store information about the customer on the system. We also found that the customer would need to be able to place orders, therefore their information would need to be stored on the system as a order reference.
2. Plants: plants was decided as our next entity, due to the knowledge that the system itself is to be used as an ordering system where plants themselves would be purchased. We also knew that the company wanted to store information on the plants on the system, such as their latin name, given in the brief.
3. Accessories: accessories were chosen for similar reasons to plants. The company sells accessories to their customers, so it would be needed on the ordering system itself.
4. Services: services was chosen as an entity due to the fact that the LGC ordering company provides this to its customers, so it would be convenient to both the company and the customers to include it in the same system.
5. Staff: a staff entity was created for various reasons. We inferred that the staff themselves would be providing services, therefore there would need to be this entity to connect the two to make it more convenient for our customers. Not only this, but the LGC ordering system needs to have staff details stored on there for LGC, as we learnt from the brief.

Justification of entities

6. Supplier: we decided to create an entity for supplier in our ordering system as they would be an essential part to re-ordering stock items. The briefing from the LGC also specified that there needed to be information stored about the suppliers, which would allow staff to contact them easily as their contact details would be present
7. Purchase order: this was used as an entity, to allow an automated order to be raised by the system to take in information from the accessories entity and the supplier entity. This entity would enable the system to send a purchase order to the supplier automatically when the quantity of an accessory is below a specified level
8. Customer order: this entity was used to enable a customer to order products from the LGC. This is essential to the system, as ordering is the key purpose of the system itself
9. Feedback: we chose feedback to be our final entity on the system. From our briefing, we found out that the company would want the customer to provide feedback on their purchase order. This would allow customers to send feedback through the system to be stored in the system for staff to easily access.

Strong or weak entity?

After identifying the entities, we then proceeded to decide whether they were strong or weak.

Entity name	Strong/Weak	Justification
Plants	Strong	The entity 'Plants' is a key aspect of the database, and exist without depending on another entity
Accessories	Strong	The entity 'Accessories' is a key aspect of the database, and exist without depending on another entity
Services	Strong	The entity 'Services' is a key aspect of the database, and exist without depending on another entity
Customer	Strong	The entity 'Customer' is a key aspect of the database, and exist without depending on another entity
Customer Order	Weak	The 'Customer Order' entity is dependent on the 'Customer' entity, making it a weak entity
Supplier	Strong	The entity 'Supplier' is a key aspect of the database, and exist without depending on another entity
Feedback	Weak	The 'Feedback' entity is dependent on the 'Customer' entity, making it a weak entity
Purchase Order	Weak	The 'Purchase Order' entity is dependent on the 'Supplier' entity, making it a weak entity
Staff	Strong	The entity 'Staff' is a key aspect of the database, and exist without depending on another entity

Identification of relationships

 Noun - Entity  Verb  Relationship between entities

Lincoln Garden Centre is a company based in Lincolnshire, which specialises in planting, growing and selling plants, accessories, and relevant services to customers. A Customer can place an order with the company for a specific item, such as a flower bouquet or garden pots. The Customer Order consists of the customer's name, the delivery address, the billing address, the *item(s) ordered*, the *quantity* of each item ordered, the *order date*, the *order status* (e.g. pending, confirmed, *dispatched date*, paid), the *total price*, the date the order was despatched, date the *invoice* was sent to the customer and the date the payment was received. The Customer Order also contains further detail that is required: *price by item*, how many are required, discount and total price by item. Based on the Customer Order, an invoice will be issued and sent to the customer after the customer has completely paid the invoice. The Customer Order is then sent to the garden department, which packs the ordered items and sent to the customer, along with a copy of the paid invoice and plant care instructions. This is shortly followed by a timely request for customer feedback. When standard Stock Items (e.g. apple-tree) are out of stock, a Purchase Order is raised from a third party supplier, which consists of the *supplier's name*, the *supplier's address*, the description of the stock items required, the quantity of each stock item required, the order date, the order status (e.g. pending, sent, received, paid), the total cost, the date the order was received, and the *date the payment was sent*. The Purchase Order is then sent to a Supplier who then sends the stock items to LGC, which are received and stored in a garden centre until needed. The Stock Items (flower, plant) information should include (at least): *Latin name*, *Popular Name*, Plant attributes (such as *colour of foliage*, *colour of flowers*, *flowering period*, *height and spread* for example, etc.), *plant category* (Shrub, tree, climber), *Season of Interest* (Any, Spring, Summer, Autumn, Winter), Full description, How to grow (Soil type, Moisture, Acid or Alkaline, sunlight {Full sun, half shade,} etc.) and Aftercare. To be able to process Customer Orders and Purchase Orders, the LGC ordering system also needs to store information (e.g. *email*, *address*, *names*, etc.) about Customers, Suppliers and LGC Staff

Identifying relationships

After identifying most of our relationships, we inferred the connection between our other entities. We then decided whether they were one to one, one to many, or many to many.

From this we came up with:

- **Customer purchases Plants** (M:M)
- **Customer purchases Accessories** (M:M)
- **Customer purchases Services**(M:M)
- **Supplier supplies Accessories** (1:M)
- **Staff will provide Services** (1:1)
- **Customer can place Customer order** (1:N)
- **Customer requested to provide Feedback** (1:N)
- **Purchase order is sent to the Supplier** (N:1)
- **Purchase order** consists of **Accessories** (1:M)

Plants entity

Latin Name - primary key

Popular Name

Colour - composite attribute containing 'Colour of Foliage', and 'Colour of Flower'

Flowering Period

Dimensions - composite attribute containing height and spread

Plant Category - multi value attribute

Season of Interest - multi value attribute

Price

Plant Image

Our primary key for this entity is the Latin name. This was chosen due to the fact that each Latin name for all our plants in our future system will be different, therefore allowing each record to be uniquely identified. We've made the attributes 'Colour' and 'Dimensions' composite attributes to group relevant data together, so that when searched for, it will give all the data necessary. We've also created multi value attributes for 'Plant category' and 'Season of interest' so it's easier to input data, as well as providing a further form of data validation, so that staff input only correct data.

Accessory entity

Accessory Name - primary key

Accessory Description

Accessory Image

Price

We decided to use the attribute 'Accessory name' as our primary key in this entity, as each accessory will have a different name, meaning there would be no duplicates of the primary key.

Services

Type of Service - primary key

Service Description

Service Image

Price

For this entity, we chose 'Type of service' to uniquely identify a record. This was selected for the primary key due to the knowledge that each service will have a different name.

Customer

Email Address - primary key

Name - composite key consisting of 'First Name' and 'Surname'

Delivery Address - composite key consisting of 'Address Line 1', 'Address Line 2', 'Address Line 3' and 'Postcode'

Telephone Number

For the customer entity, the 'Email Address' was used as our primary key. Every customer would have a different email address, meaning that there will be no risk of duplicates. Both attributes 'Name' and 'Delivery Address' have been utilised as composite keys. This has been done to make sure that all data in our system will be atomic. Therefore, data will be streamlined, and the entity itself meets 3rd normal form standards.

Supplier

Supplier Email Address - primary key

Supplier Name

Supplier Telephone Number

Supplier Address - composite key which consists of 'Supplier Address Line 1', 'Supplier Address Line 2', 'Supplier Address Line 3' and 'Supplier Postcode'

We decided that the attribute 'Supplier email address' would be our primary key, as each supplier email address would be different, so there would be no clashes to compromise the integrity of the key itself. We made the 'supplier address' a composite value, to make sure that the data in this entity is atomic, and to conform to 3rd normal form.

Staff

Name - Composite attribute, containing first name and last name

NI Number - Primary Key

Staff Email Address

Staff Address - Composite attribute which consists of 'Address Line 1', 'Address Line 2', 'Address Line 3', 'Postcode'

We chose the NI Number as our Primary Key because it's a unique value, that will ensure each record is different. Both name and staff address are composite attributes to ensure data is atomic, and to conform to 3rd normal form

Customer order

Email Address - foreign key. This will be used to link to the customer table to access those details later. This will be the partial key.

Billing address - composite attribute consisting of 'Billing Address 1', 'Billing Address 2', 'Billing Address 3' and 'Postcode'

Item(s) Ordered

Quantity

Order Date - primary key

Order Status - multi value attribute

Total Price - derived attribute

Dispatch Date

Invoice Date

Payment Date

We combined email address (from the customer entity), and order date to create our primary key. This was done because there will never be multiple orders made at the same time and date by one customer, making each record unique. We also made the billing address as a composite attribute to make sure that data would be atomic. Order status is a multi value attribute, to allow only specific data to be entered, which would ensure that data validation was to a high standard.

Feedback

Email Address - foreign key from the 'Customer' entity

Order Date - foreign key from the 'Customer Order' entity

Feedback description

We chose to combine the partial keys 'Email Address' (from the customer entity) and the 'Order Date' (from the customer order entity) to make our primary key. We chose these two derived attributes knowing that email address would be unique already to the customer, and the order date would pinpoint the specific customer order the feedback is relating to.

Purchase order

Supplier Email Address - foreign key from the 'Supplier' entity

Purchase Order Date - partial key

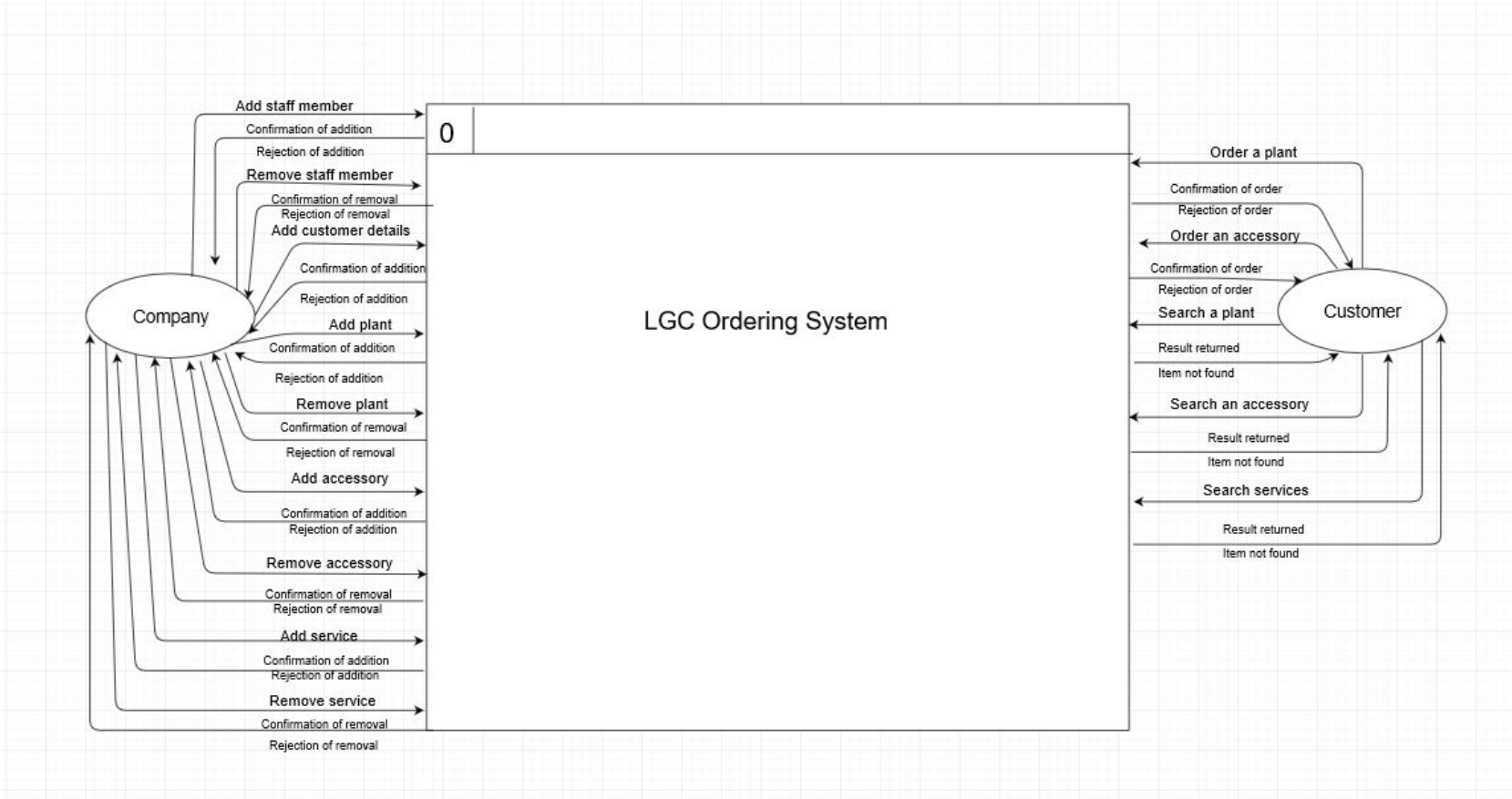
Accessory Name - foreign key from the 'Accessory' entity, primary key

Quantity

Price

We chose to have accessory name as the primary key plus the purchase order date as the partial key, so that each purchase order would be uniquely represented in the entity. We chose this over other combinations as it wouldn't be unique, for example, if we chose supplier email address as the primary key plus the purchase order date, if there was more than one purchase order made in one day for different products, we wouldn't be able to uniquely identify the order itself. The company itself grows their own plants and the staff themselves provide the services, so there is only the necessity for accessories to be purchased from the supplier.

DFD Level 0



Constraints

Not all customers will purchase plants

Not all customers will purchase Accessories

Not all customers will purchase services

Not all customers will provide feedback

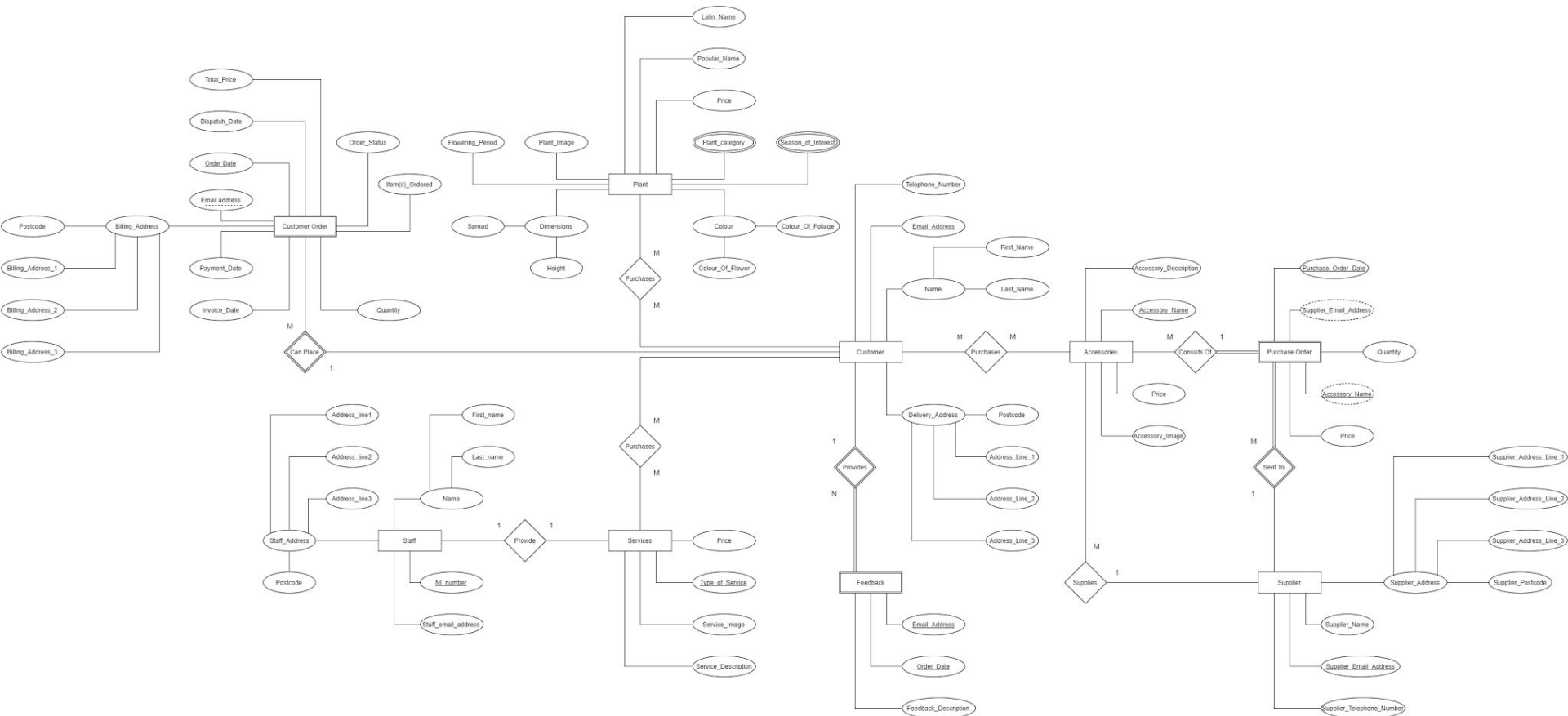
Not all staff are qualified to provide all services

A customer can place many customer orders

The supplier can receive many purchase orders

Purchase Order can consist of many accessories

ER-Model



Mapping

Strong:

Plant: Latin_Name, Popular_Name, Price, Plant_Image, Plant_Catergory, Flowering_Period, Dimensions, Season_Of_Interest, Colour_Of_Flower, Colour_Of_Foliage

Accessories: Accessory_Name, Accessory_Description, Accessory_Image, Price

Services: Type_Of_Service, Service_Description, Service_Image, Price

Customer: Customer_Email_Address, Customer_First_Name, Customer_Surename, Delivery_Address, Telephone_Number

Supplier: Sup_Email_Address, Sup_Tel_Number, Sup_Name, Sup_Add_Line_1, Sup_Add_Line_2, Sup_Add_Line_3, Sup_Postcode

Staff: NI_Number, Staff_Name, Staff_Email_Address, Staff_Add_Line_1, Staff_Add_Line_2, Staff_Add_Line_3, Staff_Postcode

Weak:

Customer Order: Order_Date, Email_Address, Billing_Address, Quantity, Order_Status, Invoice_Date, Dispatch_Date, Payment_Date, Total_Price

Feedback: Order_Date, Email_Address, Feedback_Description

Purchase Order: Purchase_Order_Date, Price, Quantity, Accesory_Name, Supplier_Email_Address

Mapping

1:1:

Staff *Provides* Services: NI_Number, Type_of_Service, hours available

1:M:

Supplier *supplies* Accessories: Accessorie_Names, Accesory_Description, Price, Accessory_Image, Supplier_Email_Address

Customer *can place* Customer order: Order_Date, Billing_Address, Order_Status, Quantity, Order_Status, Total_Price, Dispatch_Date, Invoice_Date, Payment_Date, Customer_Email_Address

Customer *provide* Feedback: Order_Date, Feedback_Description, Customer_Email_Address

Purchase order *is sent to the* Supplier: Accessory_Name, Purchase_Order_Date, Price, Quantity, Supplier_Email_Address

Purchase order *consists of* Accessories: Accessory_Name, Purchase_Order_Date, Accessory_Price, Accessory_Description, Accessory_Image

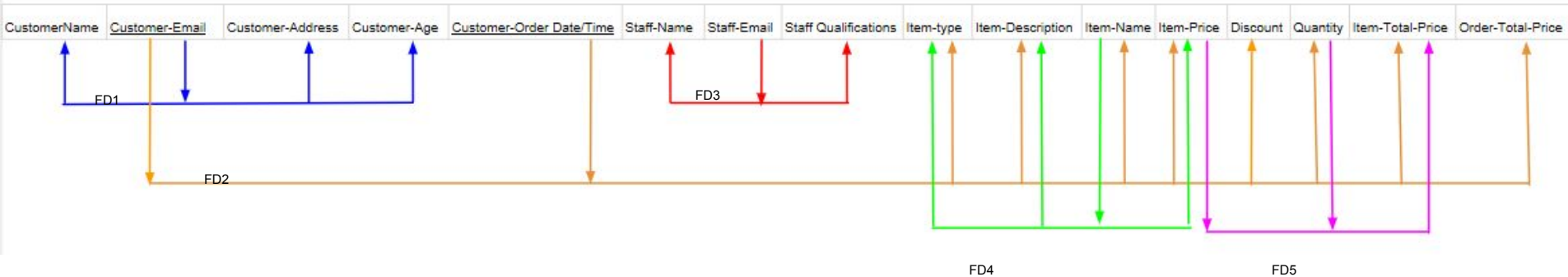
M:M:

Customer *purchases* Plants: Customer_Email_Address, Latin_Name

Customer *purchases* Accessories: Customer_Email_Address, Accesory_Name,

Customer *purchases* Services: Customer_Email_Address, Type_Of_Service,

Functional Dependencies



We identified two candidate keys from our relation:

- Customer-Email
- Customer-Order Date/Time

We chose these two attributes to be our candidate keys, as they would both contain unique records, ensuring that there would be no duplicate data which reduce redundancy. We also knew that these two attributes were relevant to the plant detail relation and would include all important data that we would need.

Normalisation -1NF

1st normal form – This relation is not in 1st normal form. In order for it to be in 1^{NF}, all attributes will need to be simple, single and stored values. For this reason, the following composite attributes have been changed:

- Customer-Name – this will now become Customer-Firstname, and Customer-Lastname
- Customer-Address – this will now become House-No/Name, Post-Code, City, County and Country
- Staff-Name - this will now become Staff-Firstname, and Staff-Lastname

This will ensure that data is atomic.

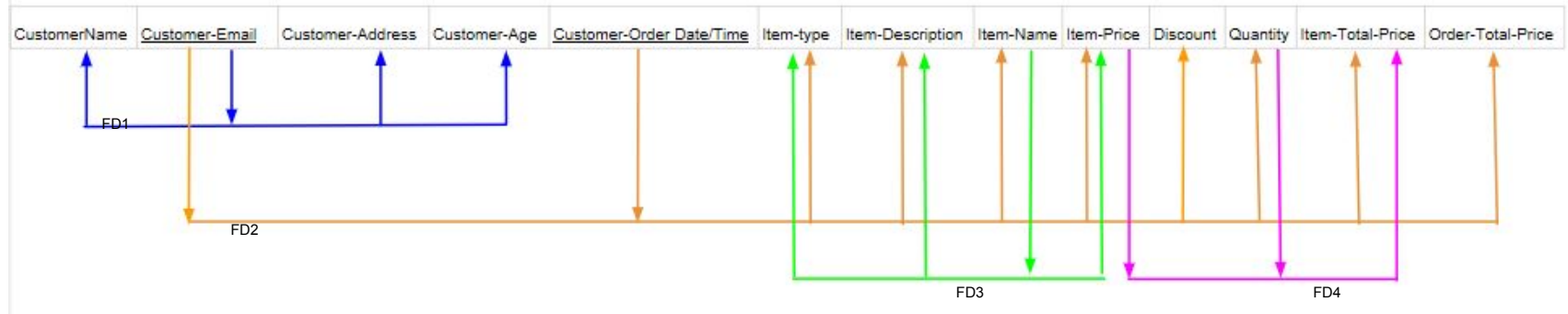
Not all attributes are dependent on the primary key, namely, Staff-Name, Staff-Email and StaffQualifications. To combat this we have created a separate relation to create two 1NF relations.

Staff-Name	<u>Staff-Email</u>	StaffQualifications
------------	--------------------	---------------------

This now meant that we have a separate 1NF relation with its primary key being Staff-Email.

All attributes are dependent on the primary key, all attributes are simple, single and stored values, and all keys are identified (see previous page), meaning that our relation is now in 1NF.

Normalisation - 2NF



FD1: Customer-Email \rightarrow {CustomerName, Customer-Address, Customer-Age}

FD2: {Customer-Email, Customer-Order Date/Time} \rightarrow {Item-Type, Item-Description, Item-Name, Item-Price, Discount, Quantity, Item-Total-Price, Order-Total-Price}

FD3: Item-Name \rightarrow {Item-Type, Item-Description, Item-Price}

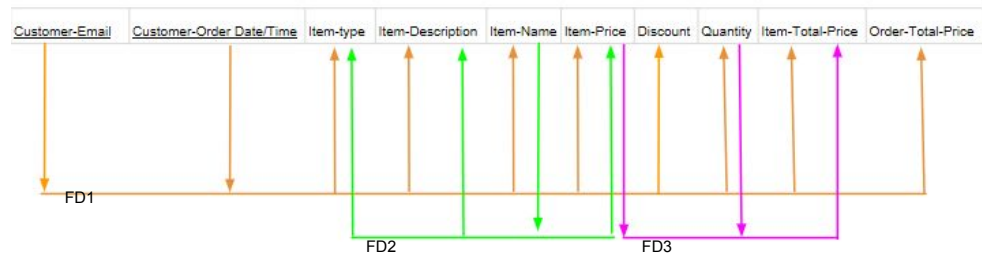
FD4: {Item-price, Quantity} \rightarrow Item-Total-Price

This relation is not in 2nd normal form. In order for it to be in 2nd normal form, all attributes will need to be fully functionally dependant on the primary key. For this reason, we separated FD1 due to its only link being to Customer-Email.

We created a separate relation for FD1 with the same attributes to have two tables in 2NF:

CustomerName	Customer-Email	Customer-Address	Customer-Age
--------------	----------------	------------------	--------------

Normalisation - 3NF



This table doesn't conform to 3NF. In order to remedy this, we have to ensure that every non-prime attribute is not transitively dependant on the primary key. Therefore, we removed FD2 and FD3 and placed them into their separate own relations. This gave us:

<u>Item-Name</u>	Item-Type	Item-Description	Item-Price
------------------	-----------	------------------	------------

<u>Item-Price</u>	<u>Quantity</u>	Item-Total-Price
-------------------	-----------------	------------------

However, FD1 did fit into 3NF whilst having the primary keys from FD2 and FD3 as foreign keys. This meant that our Plant-Relation consisted of the attributes {Customer-Email, Customer-Order Date/Time, Item-Name, Item-Price, Quantity, Order-Total-Price}.