**Database Report**

# Introduction

Aim:

* To design, create, populate and query a data store from NB Gardens

Deliverables:

* Produce a report justifying your design choice and documenting your thought processes and any changes you make
* Write about the queries you ran, the results and how you feel these may have been affected by the design of your database
* Create user stories for your databases to understand the functionality required
* Set up users for your database, including a database administrator, analysts and other departments

# Database Design & Creation

Before attempting to create a database for NB Gardens, a catalogue of user stories was created covering \_\_\_\_\_\_ functionality… The user stories were used as a basis for the Entity Relationship Diagram. For each story, I followed the steps below:

* Identify the entity
* Identify the attributes
* Identify the primary key
* Identify the relationships
* Identify the cardinality
* Draw a draft
* Map the attributes
* Refine the Entity Relationship Diagram

In the next stage, I aimed to eliminate non-atomic values that could negatively affect performance or require complex code, through normalisation. Many to many relationships, such as that of customer order and product, could be separated entities like order lines to avoid data being repeated.

Once the Entity Relationship Diagram had been completed, I was able to begin creating the database in Oracle SQL. Initially, I created the entire database using SQL. This was helpful in evaluating the suitability of a relational approach to different aspects of the inventory management system

NoSQL allows a customer to have multiple addresses stored on the system easily.

NoSQL is especially useful for employees & roles.

MongoDB gives significantly more flexibility in entering information into the database since there are no restrictions when storing data; each document in a collection could have entirely different attributes if desired. This feature is useful for attributes whose values are added at a later point or entities whose attributes are dependent on certain criteria. For instance, customisation details are only required for items to be customised. A solution would be to have separate entities for customised and non-customised products, but since customisation is a change of state, this would not work.

Another \_\_\_ I found was the ability to have documents within documents and arrays as attributes. This made it possible to store all customer order details within a single collection. However, I had difficulties querying nested documents due to how / the way in which MongoDB searches for things; in order to return results with a standard query, each key : value pair must be matched, which reduces the \_\_\_value of querying. This problem was solved using regular expression; however, if a match was found, the entire array was returned instead of the individual element. It is for this reason that I have decided to store orders in an SQL database.

## Database Population

Oracle is \_\_\_\_\_ in specifying data types, ensuring greater consistency than MongoDB.