Database Report

Introduction

This report aims to:

- Design, create, populate and query a data store from NB Gardens
- Understand the advantages and disadvantages of different database types and applications

The deliverables are as follows:

- 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

Oracle 12c is a Relational Database Management System which is able to interpret both logical (what) and physical (how) operations.

Database Design & Creation

Before creating a database, a model of the NB Gardens enterprise was developed. A catalogue of user stories covered the desired functionality to be implemented. These were written from the perspective of system users, including employees from various departments, and described a task they need to carry out to fulfil their role in the company along with why it is important. Two example user stories follow:

> 1. *As a member of the Accounts team, I want to log the time a customer order is placed so that the customer can be contacted at particular times*
> 2. *As a member of the Accounts team, I want to store a customer's delivery address in the database during the registration process so that orders can be fulfilled more efficiently*

The combined list of user stories can be found in Appendix A. These provided a basis for an Entity Relationship Diagram (Figure 5), representing the design of the database schema, focussing on the relationships between data. This was developed using the following steps:

- 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

Taking the previous user story as an example; the entity type **Customer** has the attribute **email**. There is no clear primary key resulting from this user story and only a single entity type so identifying relationship and cardinality is not needed. Combined with the second user story, however, we identify the entity type **Delivery Address** with attributes for each address line. Delivery address would have a one-to-many relationship - relationship types are shown in Figure 1 - with customer as a delivery address may belong to multiple

customers living in the same household. The resulting draft is displayed in Figure A1 and includes any relevant foreign key.

	Customer	Billing Address	Delivery Address	Card Info	Customer Order	Purchase Order	Product Location	Geo Distribution	Return	Supplier	Supplier Info	Call Centre	Warehouse	Product
Customer		has	has	has	places				sends back					
Billing Address	belongs to		is associated with											
Delivery Address	belongs to				receives									
Card Info	belongs to	is associated with												
Customer Order	is placed by		is sent to					is placed in						
Purchase Order												is processed by	is fulfilled by	contains
Product Location														belongs to
Geo Distribution					holds									
Return	is sent back by												is fulfilled by	contains
Supplier														supplies
Supplier Info														is related to
Call Centre					processes									
Warehouse					fulfills	fulfills			fulfills					
Product					is contained within	is contained within	has		is contained within	is supplied by	is related to			

Figure 1: Relationship matrix of database entities for NB Gardens

In the following stage, non-atomic values which could negatively affect performance or require complex code, were eliminated through normalisation. Many to many relationships, such as that of customer order and product, were split into separate entities, like order lines, to avoid data being repeated/replicated.

Once the initial Entity Relationship Diagram had been completed, I was able to begin creating the database in Oracle SQL. Initially, I modelled the entire database using PL/SQL in Oracle 12c. This was helpful to test and evaluate the suitability of a relational approach to different aspects of the inventory management system. The code used is provided in the accompanying .zip file.

Following this, the model was denormalised then recreated using NoSQL in MongoDB. Denormalisation is helpful in improving the read performance of a database. Again, the entire database was created using MongoDB for evaluation purposes. The code used is provided in the accompanying .zip file.

Evaluation & Redesign

The most significance between SQL and NoSQL is the way in which they store data; SQL databases are table-based relational databases with predefined schema whereas NoSQL databases are distributed databases where data is stored in key-value pairs, documents and wide-column stores with dynamic schema. NoSQL's flexibility makes it best suited to unstructured data.

The two database types also differ in terms of the strength of their respective querying languages; the SQL querying language is more powerful than its counterpart which lacks standard interfaces to perform complex queries.

NoSQL allows a customer to have multiple addresses stored on the system easily. Whereas the relationship between delivery address and customer was defined as one-to-many in the Entity Relationship diagram, NoSQL is not concerned with relationships and cardinality, meaning multiple addresses can be stored as an array or document within the customer entity type.

NoSQL is especially useful for documenting employee details and roles. For instance, it would be helpful to have a dataset including/encompassing all NB Gardens employees; however, this would need to include attributes such as **forkliftLicence** which is not relevant to Call Centre or Accounts staff. Thus, to avoid storing unnecessary data, employees in different departments would need to be represented as separate entities, increasing the complexity of the database.

MongoDB gives significantly more flexibility in entering information into the database since there are no restrictions when storing data; each entity in a collection could have an entirely different attributes if desired. This feature is useful for attributes which 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.

In addition to the above, another useful functionality of MongoDB is the ability to store documents and arrays within documents. This made it possible not only to store all customer order details within a single collection but also to embed a list of order IDs within a customer's records.

Figures 2 & 3 give the final model for the NB Gardens database system. The **Customer** entity was moved to the NoSQL database due to MongoDB's document capabilities, allowing address, payment and order information to be stored in a single **Customer** entity. The **Employee** entity was also moved to the NoSQL database following Denormalisation encompassing different roles within the company in order to improve read performance. Finally, **Customer Order** and **Customer Return** were migrated for similar reasons in addition to the benefits of the dynamic schemas offered by MongoDB. The Geographical Distribution was also moved as no direct relationship existed between the remaining entities in the relational database.

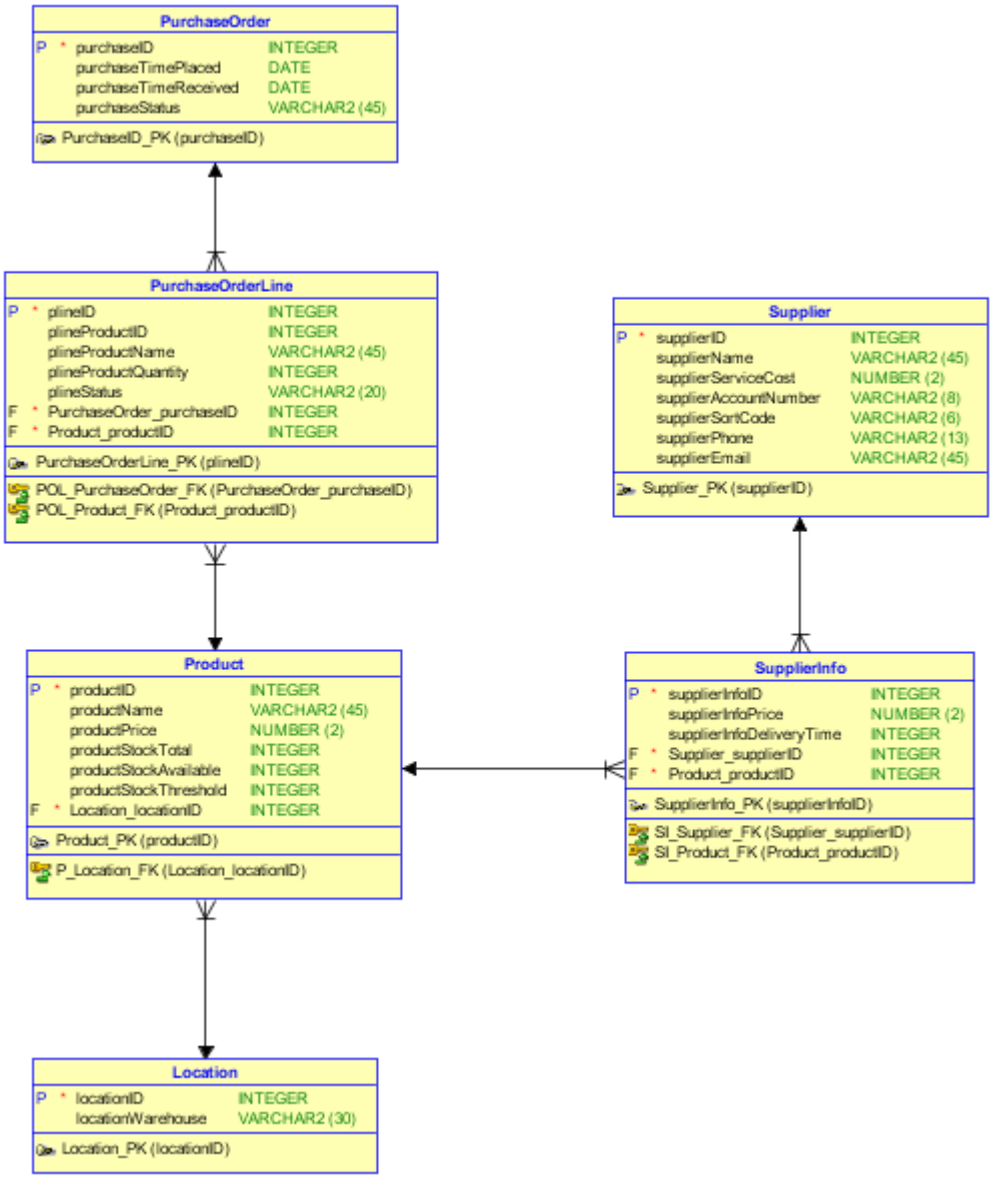

Figure 2: Revised Entity Relationship Diagram for NB Gardens

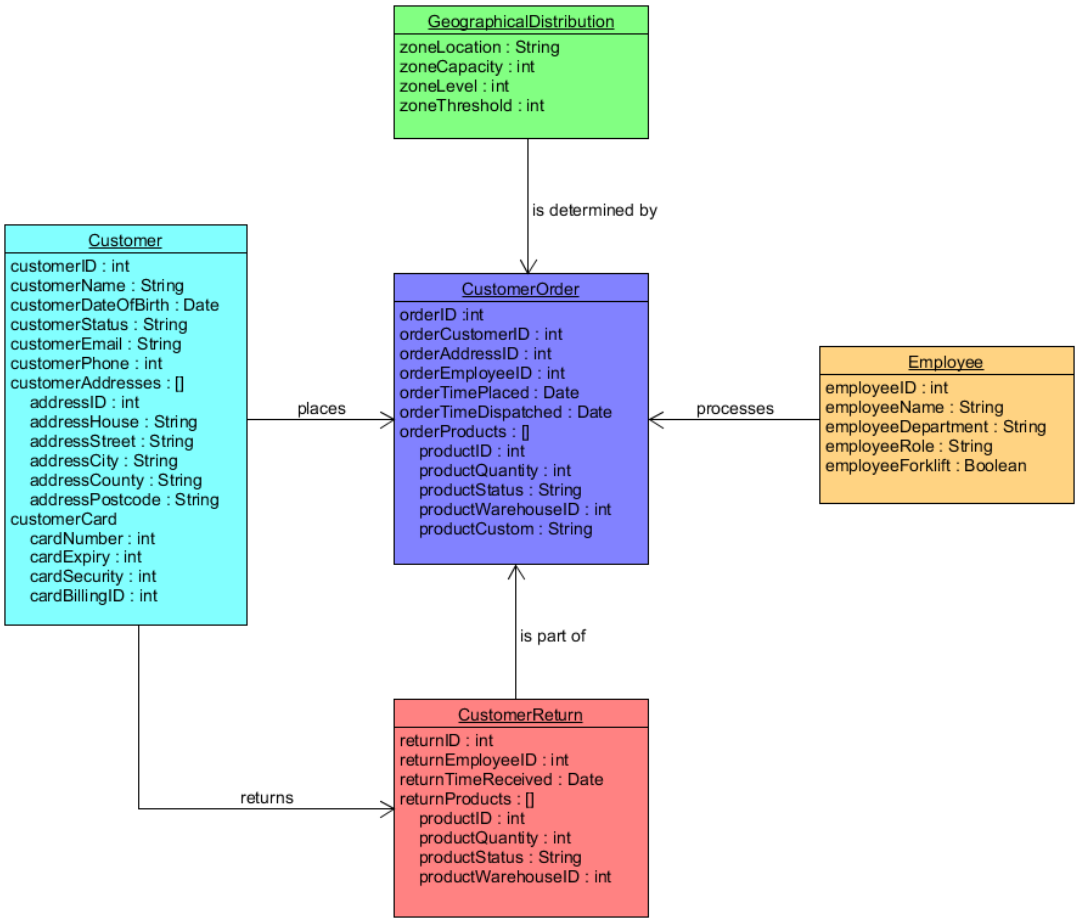

Figure 3: Revised class diagram for non-relational NB Gardens' database

A further consideration when designing and creating a database is security (i.e. Who will have access to which sections). Since the architecture of NB Gardens is relatively complex, user permissions will be granted according to a Permissions table. The table's values denote the level of access a particular user will have for areas of the database.

	Call Centre Employee	Warehouse Stocker	Inventory Manager	Accounts	Admin
Customer	R			CRU	CRUD
Address	RU	R		CRUD	CRUD
Card Info				CRUD	CRUD
Customer Order				RU	CRUD
Purchase Order			CRU	RU	CRUD
Product Location		RU	CRU		CRUD
Geo Distribution		RU	CRU		CRUD
Return		CRU	CRU		CRUD
Supplier			CRU		CRUD
Supplier Info			CRU		CRUD
Call Centre					CRUD
Warehouse					CRUD
Product		RU	CRU		CRUD

Figure 4: CRUD matrix for NB Gardens

Figure 5: Initial Entity Relationship Diagram for NB Gardens

Database Queries

A number of queries were tried and tested. An interesting query involved extracting information from embedded documents and arrays within MongoDB. The following code retrieves the product IDs of products within a customer order with a quantity of 1.

```
Db.CustomerOrder.find ( { products : { $elemMatch : { productQuantity : 1 } } } ,

    { "products.productID.$" : 1 , _id : 0 } ).pretty ()
```

Another interesting PL/SQL query constructed a join. The following code outputs a table with columns CustomerID, CustomerName and OrderID. This type of join is very useful for combining data from separate entities and demonstrates the power of PL/SQL querying.

```
SELECT customerID,  customerName, orderID
FROM Customer
INNER JOIN CustomerOrder
ON (Customer.customerID = CustomerOrder.customer_CustomerID) ;
```

Appendix A

New Customer

1. As a member of the Accounts team, I want to register a new customer so that their details can be stored and the ordering processes will be more efficient

2. As a member of the Accounts team, I want to store a customer's payment details in the database during the registration process so that orders can be fulfilled more efficiently

3. As a member of the Accounts team, I want to set a customer's status to "active" during registration so that they can place orders

4. As a member of the Accounts team, I want to store a customer's date of birth in the database during registration so that I can confirm they are over 18 and therefore eligible to place orders

5. As a member of the Accounts team, I want to store a customer's delivery address in the database during the registration process so that orders can be fulfilled more efficiently

6. As a member of the Accounts team, I want to store a customer's billing address in the database during the registration process so that orders can be fulfilled more efficiently

7. As a member of the Accounts team, I want to add new customers to the database so that their details can be stored and the ordering processes will be more efficient

8. As a member of the Accounts team, I want to log the time a customer order is placed so that the customer can be contacted at particular times

Process Payment

9. As a member of the Accounts team, I want to send a customer an order confirmation email so that they know the order has been successfully placed

10. As a member of the Accounts team, I want to process a customer's cash payment so that NB Gardens receives money for its services

11. As a member of the Accounts team, I want to process a customer's cheque payment so that NB Gardens receives money for its services

12. As a member of the Accounts team, I want to process a customer's card payment so that NB Gardens receives money for its services

13. As a member of the Accounts team, I want to send a customer a payment confirmation email so that they know the payment has been successful

14. As a member of the Accounts team, I want to make a payment to a supplier so that they will continue to supply their wares to the company

15. As a member of the Accounts team, I want to make a payment to a courier so that they will continue to deliver customer orders

Customer Satisfaction

16. As a member of the Accounts team, I want to input a customer satisfaction survey to the database so that the responses can be analysed

17. As an Accounts manager, I want to contact a customer by phone to follow up on negative reviews and understand how processes within the company and the company itself can improve

Purchase Order

18. As a member of the Accounts team, I want to file a purchase invoice so that NB Gardens has a record of all purchase orders made

19. As an Accounts manager, I want to approve or reject a purchase order so that only reasonable and realistic orders are made and there is more than one person responsible

20. As an Accounts manager, I want to send a purchase order to the supplier so that they can fulfil the request and the Warehouse can be restocked

Payment Follow Up

21. As a member of the Accounts team, I want to contact a customer by phone, asking how they are finding their new product(s) and gently reminding them to pay so that they feel good about the company and are reminded to pay for their order

22. As a member of the Accounts team, I want to send a customer a reminder email once a week after 14 days have elapsed since the order leaving the Warehouse so that they are more likely to pay for their order

23. As a member of the Accounts team, I want to send a reminder letter once a week after 14 days have elapsed since the order leaving the Warehouse so that they are more likely to pay for their order

Customer Status

24. As a member of the Accounts team, I want to update a customer's status (active, hold, inactive) so that others know whether or not they can place orders

25. As a member of the Accounts team, I want to blacklist a customer who has not paid for their products after 56 days so that they cannot place another order with NB Gardens

26. As a member of the Accounts team, I want to seek legal advice about a blacklisted customer so that I know whether it would be viable to pursue the customer

Search

27. As a member of the Sales team, I want to search for customers by their customer ID, name or postcode so that I can find a specific customer for whom the order will be placed

28. As a member of the Sales team, I want to search for products by their product ID or name so that I can check that it exists and confirm the price with the customer

Place Order

29. As a member of the Sales team, I want to check a customer's status so that I know whether they are able to place an order

30. As a member of the Sales team, I want to check if a product is in stock so that I can let the customer know if there will be a delay

31. As a member of the Sales team, I want to view a customer's order history so I can make small talk with the customer in order to build up a rapport

32. As a member of the Sales team, I want to add products and quantities to a customer's order so that the Warehouse operatives know which items to pick

33. As a member of the Sales team, I want to add details for a customised item so that the painter knows how the customer would like the item painted

34. As a member of the Sales team, I want to update a customer's delivery address so that the Warehouse operatives know in which GDZ an order should be placed and the courier will know where to deliver the products

35. As a member of the Sales team, I want to submit a customer's order so that it is on the system to help others fulfil the order

Process Payment

36. As a member of the Sales team, I want to forward a customer to the Accounts team to process their payment if they wish to pay immediately after ordering so that time consuming following up on payment is avoided

Operations

37. As a member of the Warehouse team, I want to update the status of individual stock items

38. As a member of the Warehouse team, I want to check a product's Warehouse location so that I can find it efficiently

39. As a member of the Warehouse team, I want to pick an item and place it in the packing area so that it can be packed

40. As a member of the Warehouse team, I want to pack an item so that it can eventually be delivered to the customer

41. As a member of the Warehouse team, I want to the customisation details for a particular item

42. As a member of the Warehouse team, I want to paint an item so that the customer's request for customisation can be fulfilled

43. As a member of the Warehouse team, I want to check a product's GDZ location so that I can put it in the appropriate location

44. As a member of the Warehouse team, I want to call the courier either at the end of the working day or when a GDZ has reached a certain capacity threshold so that they can pick up the parcels to deliver to customers

Returns

45. As a member of the Warehouse team, I want to unpack parcels with a returns label so that the return can be processed

46. As a member of the Warehouse team, I want to check the condition of a returned item so that the appropriate action can be taken

47. As a member of the Warehouse team, I want to mark a damaged returned item as such and place it in the damaged area so that it can be disposed of

48. As a member of the Warehouse team, I want to put a returned item in good condition in its appropriate Warehouse location and mark the return process complete so that it can be added back into the company's stock

Purchase Order

49. As a member of the Warehouse team, I want to receive a delivery so that the Warehouse can be restocked

50. As a member of the Warehouse team, I want to unpack a delivery so that the products can be easily accessed

51. As a member of the Warehouse team, I want to check whether an item is damaged or broken so that a customer does not receive an item in poor condition

52. As a member of the Warehouse team, I want to store items from a delivery in the appropriate Warehouse location so that they can be easily located

53. As the Inventory Manager, I want to view analytics relating to the sale of products so that I can decide whether any changes to stock thresholds are required or whether there is justification for ordering more or less of a particular product in the upcoming purchase order

Stock Levels

54. As the Inventory Manager, I want to check the automatically generated stock order form on a weekly basis so that I can judge whether any changes should be made

55. As the Inventory Manager, I want to update the threshold for individual products so that NB Gardens can respond to market demand

56. As the Inventory Manager, I want to update the price of individual products so that customers will either be more encouraged to buy or the company will obtain a higher profit margin

57. As the Inventory Manager, I want to add new products to the database so that stock levels can be tracked and thresholds set

Purchase Order

58. As the Inventory Manager, I want to submit a purchase order to Accounts so that they can review the request before sending it off to the supplier

59. As the Inventory Manager, I want to check the reason for a purchase order being rejected so that I can make adjustments accordingly

60. As the Inventory Manager, I want to negotiate with a supplier over missing or damaged items so that the company is compensated and receives new stock

Analysis

61. As an analyst, I want to access products and their attributes so that it can be used for analyses

62. As an analyst, I want to access customer sales data so that it can be used for analyses

63. As an analyst, I want to forecast product sales so that stock thresholds can be adapted to take account of any observed trends

64. As an analyst, I want to produce graphics describing the data so that the Inventory manager can easily interpret the analysis

65. As an analyst, I want to model product sales so that I can identify any related variables

66. As an analyst, I want to carry out an analysis on customer satisfaction surveys so that proposed changes will be justified

Administration

67. As an administrator, I want to add a new employee to the database so that accurate records can be maintained, the employee's access to the database can be modified and productivity can be monitored

68. As an administrator, I want to update an employee's level, role and licenses so that this information can be tracked and they can be given the relevant system privileges