



slington college
(इस्लिंग्टन कलेज)

Module Code & Module Title
CC4057NI Introduction to Information Systems

Assessment Weightage & Type
30% Individual Coursework

Year and Semester
2021 Spring

Student Name: Sujen Shrestha

Group: N4

London Met ID:

College ID: NP01NT4S210105

Assignment Due Date: April 30, 2021

Assignment Submission Date: April 29, 2021

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Table of Contents

1. Introduction.....	1
1.1 Databases.....	1
1.1.1 Data	1
1.1.2 Information.....	1
1.1.3 Database	1
1.1.4 Database Management System.....	1
1.2 Description of organization	2
1.3 Description of project	2
1.4 Features present in database.....	2
2. Database Model	3
2.1 Business Rule	3
2.2 Entity Relationship Model.....	4
2.3 Relational Diagram.....	5
2.4 Tables.....	6
2.4.1 Table 1: Customers.....	6
2.4.2 Table 2: Orders	7
2.4.3 Table 3: Mobile Phones.....	9
2.4.4 Table 4: Mobile Product Details.....	10
2.4.5 Table 5: Wholesaler.....	11
3. Data Dictionary	13
4. Query Table.....	18
5. Conclusion.....	22
References.....	23

List of Figures

Figure 1: Entity-Relation Diagram	4
Figure 2: Relational Diagram.....	5
Figure 3: Table Creation - Customers.....	6
Figure 4: Table Description - Customers.....	6
Figure 5: Data Insertion - Customers.....	7
Figure 6: Data Selection - Customers.....	7
Figure 7: Table Creation - Orders.....	8
Figure 8: Table Description - Orders	8
Figure 9: Data Insertion - Orders	8
Figure 10: Data Selection - Orders.....	8
Figure 11: Table Creation - Mobile Products.....	9
Figure 12: Table Description - Mobile Products.....	9
Figure 13: Data Insertion – Mobile Products	9
Figure 14: Data Selection – Mobile Products	10
Figure 15: Table Creation - Mobile Product Details.....	10
Figure 16: Table Description - Mobile Product Details.....	10
Figure 17: Data Insertion - Mobile Product Details.....	11
Figure 18: Data Selection - Mobile Product Details.....	11
Figure 19: Table Creation - Wholesalers.....	12
Figure 20: Table Description - Wholesalers.....	12
Figure 21: Data Insertion - Wholesalers.....	12
Figure 22: Data Selection - Wholesalers.....	12

List of Tables

Table 1: Customers Data Dictionary	13
Table 2: Orders Data Dictionary	14
Table 3: Mobile Products Data Dictionary.....	15
Table 4: Mobile Product Details Data Dictionary.....	16
Table 5: Wholesaler Data Dictionary.....	17
Table 6: Query – BETWEEN	18
Table 7: Query – LIKE	18
Table 8: Query - ORDER BY.....	18
Table 9: Query – LIMIT	19
Table 10: Query - DISTINCT	19
Table 11: Query - GROUP BY	19
Table 12: Query - COUNT	20
Table 13: Query - IN	20
Table 14: Query - HAVING.....	20
Table 15: Query - JOIN	21

1. Introduction

1.1 Databases

1.1.1 Data

Data can be defined as the facts and statistics collected together for reference or analysis. It is a unit of information which makes the basis for reasoning or calculation. For example, a list of values stored in a table of a report like names, numbers, etc.

1.1.2 Information

Information can be defined as the processed collection of data which gives detailed knowledge about any particular subject. It can be represented in the form of charts, tables, reports, etc. For example, the facts learned after a survey or research.

1.1.3 Database

A database is an organized collection of data and information held in a computer which can be stored, retrieved, modified or removed by the use of various data-processing operations. (Encyclopaedia Britannica, Inc., 2020)

1.1.4 Database Management System

A Database Management System is a software technology that manages and retrieves information through various queries from within a database. (Encyclopaedia Britannica, Inc., 2020)

1.2 Description of organization

The database in this project is created for a mobile phone store. The store sells mobile phones from various brands. The store imports its products from a wholesaler in bulk so they get some discount for the phones per handset. The units are then sold to the customers as according to the quantity ordered by the customers.

1.3 Description of project

The project is about creating a database management system for a mobile phone store. This project facilitates the shop owner to keep track of the mobile phones. This makes it easier for them to understand various information such as their best-selling phone, least selling phone, repeating customers, etc. By this knowledge the owner can decide to buy more of the most selling products in bulk and decide how much to buy the low selling products or avoid them completely. This way the store owner can maximize their revenue and avoid unnecessary expenditure by using this database management software.

1.4 Features present in database

The features present in the database are that it enforces integrity constraints in various fields such as phone number, name, etc. This means that the data type set in the system determines the data that can be added in a particular field. For example, only numbers are accepted in the phone number field. Likewise, for different data types only the relevant input can be inserted. Also, it makes sure that there is no redundant data as primary key ensures that no duplicates are entered. (Watt & Eng, 2014)

2. Database Model

2.1 Business Rule

In this model, a customer who wishes to buy a mobile phone can place an order for one model of a phone at a time. The customer can order multiple phones of same model but cannot order phones of different models at the same time. A product can be ordered multiple times but an order can only contain a single product. The order placed by the customer is then received for the products requested by the customer. The seller then checks the inventory for the product requested and if the requested product is available then they a transaction will be made. In case if the requested product is not available then the retail seller contacts the wholesaler for the number of products that he/she needs and continues the transaction.

2.2 Entity Relationship Model

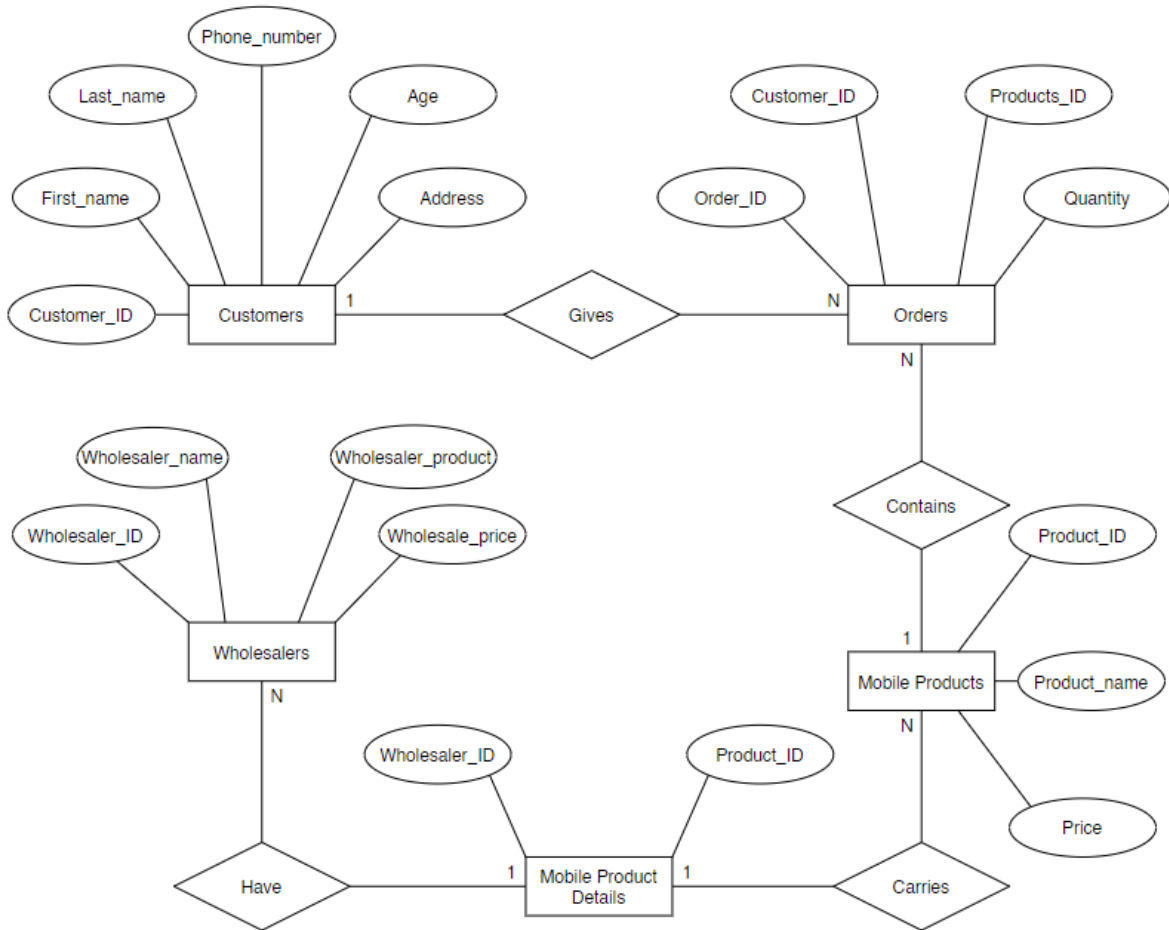


Figure 1: Entity-Relation Diagram

2.3 Relational Diagram

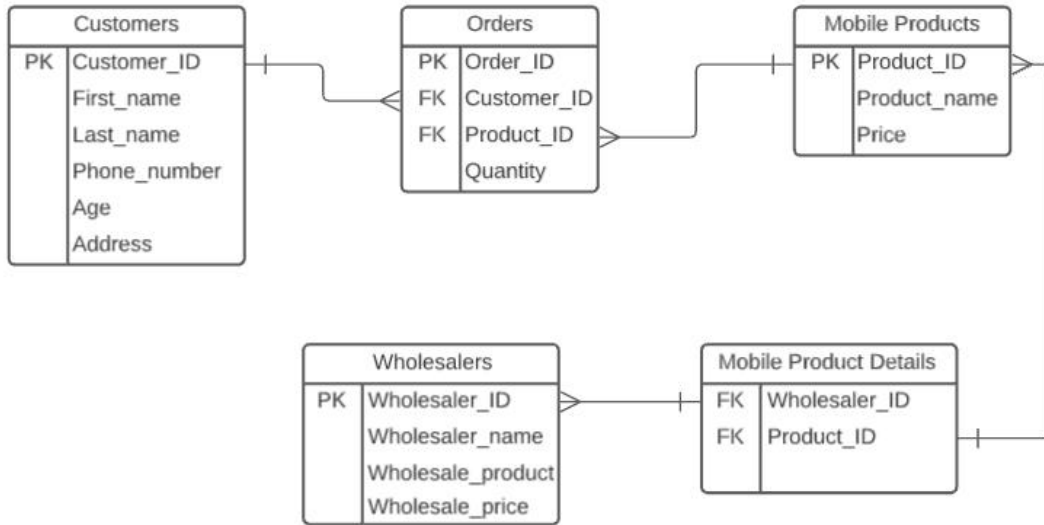


Figure 2: Relational Diagram

2.4 Tables

2.4.1 Table 1: Customers

Customers table is the entity in which all the information of the customer is stored. This table is important because it helps to identify each individual customer that purchased the products from the store. The following are the attributes present in this table and the type of data they store:

- Customer_ID: The unique identifier of each customer.
- First_name: The first name of the customer.
- Last_name: The family name of the customer.
- Phone_number: The contact number of the customer.
- Age: The age of the customer.
- Address: The residential location of the customer.

```
MariaDB [MobileStore]> CREATE TABLE Customers (
  -> Customer_ID INT PRIMARY KEY AUTO_INCREMENT,
  -> First_name VARCHAR(50) NOT NULL,
  -> Last_name VARCHAR(50) NOT NULL,
  -> Phone_number VARCHAR(10) NOT NULL UNIQUE,
  -> Age INT NOT NULL,
  -> Address VARCHAR(255) NOT NULL
  -> );
Query OK, 0 rows affected (0.517 sec)
```

Figure 3: Table Creation - Customers

Field	Type	Null	Key	Default	Extra
Customer_ID	int(11)	NO	PRI	NULL	auto_increment
First_name	varchar(50)	NO		NULL	
Last_name	varchar(50)	NO		NULL	
Phone_number	varchar(10)	NO	UNI	NULL	
Age	int(11)	NO		NULL	
Address	varchar(255)	NO		NULL	

6 rows in set (0.046 sec)

Figure 4: Table Description - Customers

```

MariaDB [MobileStore]> INSERT INTO Customers VALUES
-> (" ", "Samir", "Kansakar", "9800000000", 20, "Putalisadak"),
-> (" ", "Ajay", "Chhetri", "9811111111", 25, "New Baneshwor"),
-> (" ", "Sunny", "Jha", "9822222222", 33, "Gwarko"),
-> (" ", "Jenish", "Tandukar", "9833333333", 47, "Putalisadak"),
-> (" ", "Sampanna", "Basnet", "9844444444", 36, "New Plaza");
Query OK, 5 rows affected, 5 warnings (0.079 sec)
Records: 5 Duplicates: 0 Warnings: 5

```

Figure 5: Data Insertion - Customers

```

MariaDB [MobileStore]> SELECT * FROM Customers;
+-----+-----+-----+-----+-----+-----+
| Customer_ID | First_name | Last_name | Phone_number | Age | Address |
+-----+-----+-----+-----+-----+-----+
| 1 | Samir | Kansakar | 9800000000 | 20 | Putalisadak |
| 2 | Ajay | Chhetri | 9811111111 | 25 | New Baneshwor |
| 3 | Sunny | Jha | 9822222222 | 33 | Gwarko |
| 4 | Jenish | Tandukar | 9833333333 | 47 | Putalisadak |
| 5 | Sampanna | Basnet | 9844444444 | 36 | New Plaza |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.000 sec)

```

Figure 6: Data Selection - Customers

2.4.2 Table 2: Orders

Orders table is the entity in which the number of items ordered by the customer is stored. This table is helps to verify the products requested by the customer. The following are the attributes present in this table and the type of data they store:

- Order_ID: The unique identifier of each order.
- Customer: The foreign key which links to Customer_ID of Customers table.
- Product: The foreign key which links to Product_ID of Mobile Product Details table.
- Order_date: The date of when the order is placed.
- Quantity: The number of products ordered by the costumer.

```

MariaDB [MobileStore]> CREATE TABLE Orders (
  -> Order_ID INT PRIMARY KEY AUTO_INCREMENT,
  -> Customer INT NOT NULL,
  -> Product INT NOT NULL,
  -> Order_date DATE NOT NULL,
  -> Quantity INT NOT NULL DEFAULT "1",
  -> FOREIGN KEY(Customer) REFERENCES Customers(Customer_ID),
  -> FOREIGN KEY(Product) REFERENCES MobileProducts(Product_ID)
  -> );
Query OK, 0 rows affected (0.444 sec)

```

Figure 7: Table Creation - Orders

```

MariaDB [MobileStore]> DESCRIBE Orders;
+-----+-----+-----+-----+-----+-----+
| Field      | Type   | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| Order_ID   | int(11)| NO   | PRI | NULL    | auto_increment|
| Customer   | int(11)| NO   | MUL | NULL    |                |
| Product    | int(11)| NO   | MUL | NULL    |                |
| Order_date | date   | NO   |     | NULL    |                |
| Quantity   | int(11)| NO   |     | 1       |                |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.125 sec)

```

Figure 8: Table Description - Orders

```

MariaDB [MobileStore]> INSERT INTO Orders Values
  -> (" ", 2, 1, "2021-02-20", 4),
  -> (" ", 1, 3, "2021-02-27", 10),
  -> (" ", 5, 2, "2021-03-15", 5),
  -> (" ", 3, 5, "2021-03-30", "6"),
  -> (" ", 4, 4, "2021-04-20", 2);
Query OK, 5 rows affected, 5 warnings (0.141 sec)
Records: 5 Duplicates: 0 Warnings: 5

```

Figure 9: Data Insertion - Orders

```

MariaDB [MobileStore]> SELECT * FROM Orders;
+-----+-----+-----+-----+-----+
| Order_ID | Customer | Product | Order_date | Quantity |
+-----+-----+-----+-----+-----+
| 1        | 2        | 1        | 2021-02-20 | 4        |
| 2        | 1        | 3        | 2021-02-27 | 10       |
| 3        | 5        | 2        | 2021-03-15 | 5        |
| 4        | 3        | 5        | 2021-03-30 | 6        |
| 5        | 4        | 4        | 2021-04-20 | 2        |
+-----+-----+-----+-----+-----+
5 rows in set (0.000 sec)

```

Figure 10: Data Selection - Orders

2.4.3 Table 3: Mobile Phones

Mobile Phones is the entity in which contains all the information of the products stored. This table is developed because it is necessary to store all the information of the phones to provide to the customers. The following are the attributes present in this table and the type of data they store:

- Product_ID: The unique identifier of each mobile phone.
- Product_name: The name of products.
- Price: The cost of each products.

```
MariaDB [MobileStore]> CREATE TABLE MobileProducts (
  -> Product_ID INT PRIMARY KEY AUTO_INCREMENT,
  -> Product_name VARCHAR(255) NOT NULL,
  -> Price DECIMAL(10,2) NOT NULL
  -> );
Query OK, 0 rows affected (0.232 sec)
```

Figure 11: Table Creation - Mobile Products

```
MariaDB [MobileStore]> DESCRIBE MobileProducts;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| Product_ID    | int(11)       | NO   | PRI | NULL    | auto_increment |
| Product_name  | varchar(255) | NO   |     | NULL    |                |
| Price         | decimal(10,2) | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.045 sec)
```

Figure 12: Table Description - Mobile Products

```
MariaDB [MobileStore]> INSERT INTO MobileProducts Values
  -> (" ", "Samsung", 80000.00),
  -> (" ", "Apple", 120000.00),
  -> (" ", "Xiaomi", 40000.00),
  -> (" ", "Sony", 100000.00),
  -> (" ", "Oppo", 60000.00);
Query OK, 5 rows affected, 5 warnings (0.053 sec)
Records: 5 Duplicates: 0 Warnings: 5
```

Figure 13: Data Insertion – Mobile Products

```

MariaDB [MobileStore]> SELECT * FROM MobileProducts;
+-----+-----+-----+
| Product_ID | Product_name | Price |
+-----+-----+-----+
|          1 | Samsung      | 80000.00 |
|          2 | Apple        | 120000.00 |
|          3 | Xiaomi       | 40000.00 |
|          4 | Sony         | 100000.00 |
|          5 | Oppo         | 60000.00 |
+-----+-----+-----+
5 rows in set (0.000 sec)

```

Figure 14: Data Selection – Mobile Products

2.4.4 Table 4: Mobile Product Details

Mobile Product Details is the bridge-entity which contains all the details of the mobile products. This table is important because it links many other tables together that rely on its information. The following are the attributes present in this table and the type of data they store:

- Product: The foreign key which links to Product_ID of Mobile Product Details table.
- Wholesaler: The foreign key which links to Wholesaler_ID of Wholesalers table.

```

MariaDB [MobileStore]> CREATE TABLE MobileProductDetails (
  -> Wholesaler INT NOT NULL,
  -> Product INT NOT NULL,
  -> FOREIGN KEY(Wholesaler) REFERENCES Wholesalers(Wholesaler_ID),
  -> FOREIGN KEY(Product) REFERENCES MobileProducts(Product_ID)
  -> );
Query OK, 0 rows affected (0.271 sec)

```

Figure 15: Table Creation - Mobile ProductDetails

```

MariaDB [MobileStore]> DESCRIBE MobileProductDetails;
+-----+-----+-----+-----+-----+-----+
| Field      | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Wholesaler | int(11) | NO   | MUL | NULL    |       |
| Product    | int(11) | NO   | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.049 sec)

```

Figure 16: Table Description - Mobile Product Details

```
MariaDB [MobileStore]> INSERT INTO MobileProductDetails Values
-> (1,2),
-> (2,3),
-> (3,4),
-> (4,1),
-> (5,5);
Query OK, 5 rows affected (0.054 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

Figure 17: Data Insertion - Mobile Product Details

```
MariaDB [MobileStore]> SELECT * FROM MobileProductDetails;
+-----+-----+
| Wholesaler | Product |
+-----+-----+
|          1 |         2 |
|          2 |         3 |
|          3 |         4 |
|          4 |         1 |
|          5 |         5 |
+-----+-----+
5 rows in set (0.000 sec)
```

Figure 18: Data Selection - Mobile Product Details

2.4.5 Table 5: Wholesaler

Wholesaler is the entity in which the information of different wholesalers is stored. This is useful to know the different wholesalers which the store relies on to import the products. The following are the attributes present in this table and the type of data they store:

- Wholesaler_ID: The unique identifier of each wholesaler.
- Wholesaler_name: The name of wholesalers.
- Wholesale_products: The products sold by wholesaler.
- Wholesale_price: The discounted price per 50 pieces of products purchased from wholesale shop.

```

MariaDB [MobileStore]> CREATE TABLE Wholesalers (
  -> Wholesaler_ID INT PRIMARY KEY AUTO_INCREMENT,
  -> Wholesaler_name VARCHAR(255) NOT NULL,
  -> Wholesaler_product VARCHAR(255),
  -> Wholesale_price DECIMAL(10,2)
  -> );
Query OK, 0 rows affected (0.237 sec)

```

Figure 19: Table Creation - Wholesalers

```

MariaDB [MobileStore]> DESCRIBE Wholesalers;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| Wholesaler_ID | int(11)       | NO   | PRI | NULL    | auto_increment |
| Wholesaler_name | varchar(255)  | NO   |     | NULL    |                |
| Wholesaler_product | varchar(255) | YES  |     | NULL    |                |
| Wholesale_price | decimal(10,2) | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.045 sec)

```

Figure 20: Table Description - Wholesalers

```

MariaDB [MobileStore]> INSERT INTO Wholesalers Values
  -> (" ", "Shiva Electronic Suppliers", "Apple", 5600000.00),
  -> (" ", "Shankhar Mobile Suppliers", "Xiaomi", 1800000.00),
  -> (" ", "Bhairab Gadget Suppliers", "Sony", 4700000.00),
  -> (" ", "Mahakal IT Suppliers", "Samsung", 3600000.00),
  -> (" ", "Rudra Phone Suppliers", "Oppo", 2800000.00);
Query OK, 5 rows affected, 5 warnings (0.047 sec)
Records: 5 Duplicates: 0 Warnings: 5

```

Figure 21: Data Insertion - Wholesalers

```

MariaDB [MobileStore]> SELECT * FROM Wholesalers;
+-----+-----+-----+-----+
| Wholesaler_ID | Wholesaler_name          | Wholesaler_product | Wholesale_price |
+-----+-----+-----+-----+
| 1 | Shiva Electronic Suppliers | Apple               | 5600000.00     |
| 2 | Shankhar Mobile Suppliers  | Xiaomi              | 1800000.00     |
| 3 | Bhairab Gadget Suppliers  | Sony                | 4700000.00     |
| 4 | Mahakal IT Suppliers      | Samsung             | 3600000.00     |
| 5 | Rudra Phone Suppliers     | Oppo                | 2800000.00     |
+-----+-----+-----+-----+
5 rows in set (0.000 sec)

```

Figure 22: Data Selection - Wholesalers

3. Data Dictionary

Entity name	Entity description	Column name	Column description	Data Type	Length	Primary key	Foreign key	Nullable	Unique	Notes
Customers	A customer is someone who buys goods and services from a store.	Customer_ID	ID of the customer, for the unique identification of each customer	INT		True	False	False	True	Auto Incremented
		First_name	First name of the customer.	VARCHAR	50	False	False	False	False	
		Last_name	Surname of the customer.	VARCHAR	50	False	False	False	False	
		Phone_number	The contact number of a customer.	VARCHAR	10	False	False	False	True	
		Age	The age of the customer.	INT		False	False	False	False	
		Addresses	The residential location of the customer.	VARCHAR	255	False	False	False	False	

Table 1: Customers Data Dictionary

Entity name	Entity description	Column name	Column description	Data Type	Length	Primary key	Foreign key	Nullable	Unique	Notes
Orders	An order is the request placed by the customer to buy the products.	Order_ID	ID of the order, for the unique identification of each order	INT		True	False	False	True	Auto Incremented
		Customer	The identity of customers who place an order.	INT		False	True	False	False	References to Customer_ID column of Customers table.
		Product	The identity of list of phones in the store.	INT		False	True	False	False	References to the Product_ID of the Mobile Product Details table.
		Order_date	The date when the order is placed.	DATE		False	False	False	False	
		Quantity	The number of phones ordered by the customer.	INT		False	False	False	False	

Table 2: Orders Data Dictionary

Entity name	Entity description	Column name	Column description	Data Type	Length	Primary key	Foreign key	Nullable	Unique	Notes
Mobile Phones	Mobile phones are the list of products stored in the shop.	Product_ID	ID of the product, for the unique identification of each product.	INT		True	False	False	True	Auto Incremented
		Product_name	The name of product available.	INT		False	False	False	False	
		Price	The cost of the mobile phones.	DECIMAL	10,2	False	False	False	False	

Table 3: Mobile Products Data Dictionary

Entity name	Entity description	Column name	Column description	Data Type	Length	Primary key	Foreign key	Nullable	Unique	Notes
Mobile Product Details	Mobile Product Details is the bridge entity which links different tables together .	Product	ID of the product, for the unique identification of each product.	INT		False	True	False	False	Auto Incremented
		Wholesaler	ID of the product, for the unique identification of each product.	INT		False	True	False	False	

Table 4: Mobile Product Details Data Dictionary

Entity name	Entity description	Column name	Column description	Data Type	Length	Primary key	Foreign key	Nullable	Unique	Notes
Wholesaler	Wholesaler is the person who sells their products at a discounted price.	Wholesaler_ID	ID of the wholesaler, for the unique identification of each wholesaler.	INT		True	False	False	True	Auto Incremented
		Wholesaler_name	The name of the wholesaler.	VARCHAR	255	False	False	False	False	
		Wholesaler_product	The list of products taken from the wholesaler.	INT		False	False	False	False	
		Wholesaler_price	The discounted price of products per 50 pieces.	DECIMAL	10, 2	False	False	False	False	

Table 5: Wholesaler Data Dictionary

4. Query Table

Query Number	Query 1
Query	SELECT * FROM MobileProducts WHERE Price BETWEEN 40000 AND 60000;
Keyword Used	SELECT, FROM, WHERE, BETWEEN, AND
Purpose/Result	The records are listed whose price lies between 40000-60000.

Table 6: Query – BETWEEN

Query Number	Query 2
Query	SELECT * FROM Wholesalers WHERE Wholesaler_name LIKE "Sh%";
Keyword Used	SELECT, FROM, WHERE, LIKE
Purpose/Result	The records are shown where name starts with "Sh".

Table 7: Query – LIKE

Query Number	Query 3
Query	SELECT First_name, Last_name FROM Customers ORDER BY First_name;
Keyword Used	SELECT, FROM, ORDER BY
Purpose/Result	The records are displayed in ascending order according to the first name of customers.

Table 8: Query - ORDER BY

Query Number	Query 4
Query	SELECT * FROM MobileProducts ORDER BY Price DESC LIMIT 1;
Keyword Used	SELECT, FROM, ORDER BY, DESC, LIMIT
Purpose/Result	Only one record is displayed from descending order according to price.

Table 9: Query – LIMIT

Query Number	Query 5
Query	SELECT DISTINCT(Address) FROM Customers;
Keyword Used	SELECT, DISTINCT, FROM
Purpose/Result	The records with unique address are shown.

Table 10: Query - DISTINCT

Query Number	Query 6
Query	SELECT Product, Quantity FROM Orders GROUP BY Quantity DESC;
Keyword Used	SELECT FROM, GROUP BY, DESC;
Purpose/Result	The records are displayed in descending order according to quantity of products that have been ordered.

Table 11: Query - GROUP BY

Query Number	Query 7
Query	SELECT Order_ID, COUNT(Product) FROM Orders GROUP BY Order_ID;
Keyword Used	SELECT, COUNT, FROM, GROUP BY
Purpose/Result	The records are shown according the number of times a product has been ordered.

Table 12: Query - COUNT

Query Number	Query 8
Query	SELECT * FROM Customers WHERE Customer_ID IN (3,4,5);
Keyword Used	SELECT, FROM, WHERE. IN
Purpose/Result	The records are returned from Customers which contain the Customer_ID 3,4 and 5.

Table 13: Query - IN

Query Number	Query 9
Query	SELECT Product_name, Price FROM MobileProducts GROUP BY Product_name HAVING Price <=80000;
Keyword Used	SELECT, FROM, GROUP BY, HAVING
Purpose/Result	The records that contain Product_name and Price are displayed in ascending order that have the price equal to or less than 80000.

Table 14: Query - HAVING

Query Number	Query 10
--------------	----------

Query	SELECT * FROM Customers JOIN Orders ON Customers.Customer_ID = Orders.Customer;
Keyword Used	SELECT, FROM, JOIN, ON
Purpose/Result	The records from Customers and Orders are merged and displayed.

Table 15: Query - JOIN

5. Conclusion

In this model, a database was designed for Mobile Phone Store to give a logical structure by including the relationships and constraints to establish data to be stored and retrieved conveniently (Lucid Software Inc., 2021). In the process of creating the entity-relation diagram, I found that many-to-many relationships cannot be created directly in a database as we cannot determine which table is the child and which is the parent. So, to avoid this confusion, we can break the many-to-many relationship into two one-to-one relationships by using another entity, known as bridge-entity (FileMaker, Inc., n.d.). In this way, I learned about the various rules and structures to create the entity-relation diagram and relational diagram. Also, I learned about the syntax and semantics involved in database creation in MySQL and various queries which help to retrieve specific types of data as required by the user. I learned about various data types and found out where a certain data type is most appropriate to use. For example, DECIMAL should be used to store values such as price so that the exact amount can be recorded. I used the internet search about various topics related to this project and asked my module lecturers when I was confused about various concepts. This helped me to clearly understand the steps required to create a database. In this way, the database model was created through extensive research and various observations.

References

Encyclopaedia Britannica, Inc., 2020. *Encyclopaedia Britannica*. [Online]

Available at: <https://www.britannica.com/technology/database>

[Accessed 20 April 2021].

FileMaker, Inc., n.d. *FileMaker*. [Online]

Available at:

https://fmhelp.filemaker.com/help/18/fmp/en/index.html#page/FMP_Help/many-to-many-relationships.html

[Accessed 28 April 2021].

Lucid Software Inc., 2021. *Lucidchart*. [Online]

Available at: <https://www.lucidchart.com/pages/database-diagram/database-models#:~:text=A%20database%20model%20shows%20the,data%20model%20the%20designers%20adopt.>

[Accessed 28 April 2021].

Watt, A. & Eng, N., 2014. *Database Design*. 2nd ed. Victoria: BCcampus.