School of Engineering

Diploma in

Electronic and Computer Engineering

EGE311 Database Design & Applications

Project Report

Nunez Myles Blasco (203149T) (E1)

Edit the following accordingly

Report Template

Scope

Key Specifications & Features of Your Solution

- Use of MySQL Database
  - To store, manipulate and retrieve Product, Staff, Customer, Order data

- Python Application
  - Login screen to verify authorised users only
  - Menu screens for each of the 4 user roles (Manager, Merchandiser, Packer, Deliverer)
  - Easy to use since interaction is mostly based on number input
  - Connected to database and can send SQL queries

Implementation

Database

Design Rationale

1. Identified the **4 entities**
   - Product, Staff, Customer (Outlet) table for Manager to CRUD

Manager
1. Add, Retrieve, Update, or Remove individual product (See Appendix A).
2. Add, Retrieve, Update, or Remove individual staff (See Appendix B).
3. Add, Retrieve, Update, or Remove individual outlet (See Appendix B).

   - Order table for Merchandiser to CRUD

Merchandiser
1. Add, Retrieve, Update, or Remove own order for individual outlets.

2. Created **attribute names** for each table, with reference from "*Project Questions (Set 01).docx*" appendix A, B

   - For Customer table, identified Customer name, Customer address, Customer phone as attributes from the image below:

Example of Outlet Details:

Customer Name: Cheers Hougang Central

Outlet Address: 810 Hougang Central, #03-10, Singapore 530810

Outlet Phone: 67589651

   - For Product table, identified Product category, Product item, Product code, Unit price as attributes from the image below:

Appendix A: Products Information

Category	Item	Product Code	Price
Diary	Milk	DKRS35501	$2.30
	Butter	DKRS35502	$4.50
	Cheese Slices	DKRS35503	$3.50
	Yogurt	DKRS35504	$0.95
Bakery	Bread	BVR30221	$2.70
	Cereal	BVR30222	$7.00
	Crackers	BVR30223	$3.10

   - For Staff table, identified Staff name, Staff role, Staff NRIC as attributes from the image below:

Example of Staff Details:

Berlin Liew, Manager, S7065843E

Peter Goh, Packer, S6556678E

   - For Order table, identified Merchandiser name, Order date, CustomerID, ProductID, Product quantity, Product revenue, Packer name, Deliverer name, Delivery date
     as important attributes from the image below:

Example of Sales Order Details:

Order By: Margaret Lim

Order Date: 22 Jun 2022

Delivery Customer: Cheers Hougang Central

Delivery Address: 810 Hougang Central, #03-10, Singapore 530810

Delivery Phone: 67589651

Product	Product Code	Unit Price	Quantity		Total
Tomato	CVOMS3681	$1.45	100		$145.00
Bread	BVR30221	$2.70	10		$27.00
				Total	$172.00

Packed By: Peter Goh

Delivered By: Samson Lim

Delivery Date: 24 Jun 2022

3. Identified **other attribute names** that are required, with reference from "*Project Questions (Set 01).docx*" functions of each role

   - For Customer table, identified Customer region as an attribute from the image below:

     9. See the sales by region (North, Central, East, West).

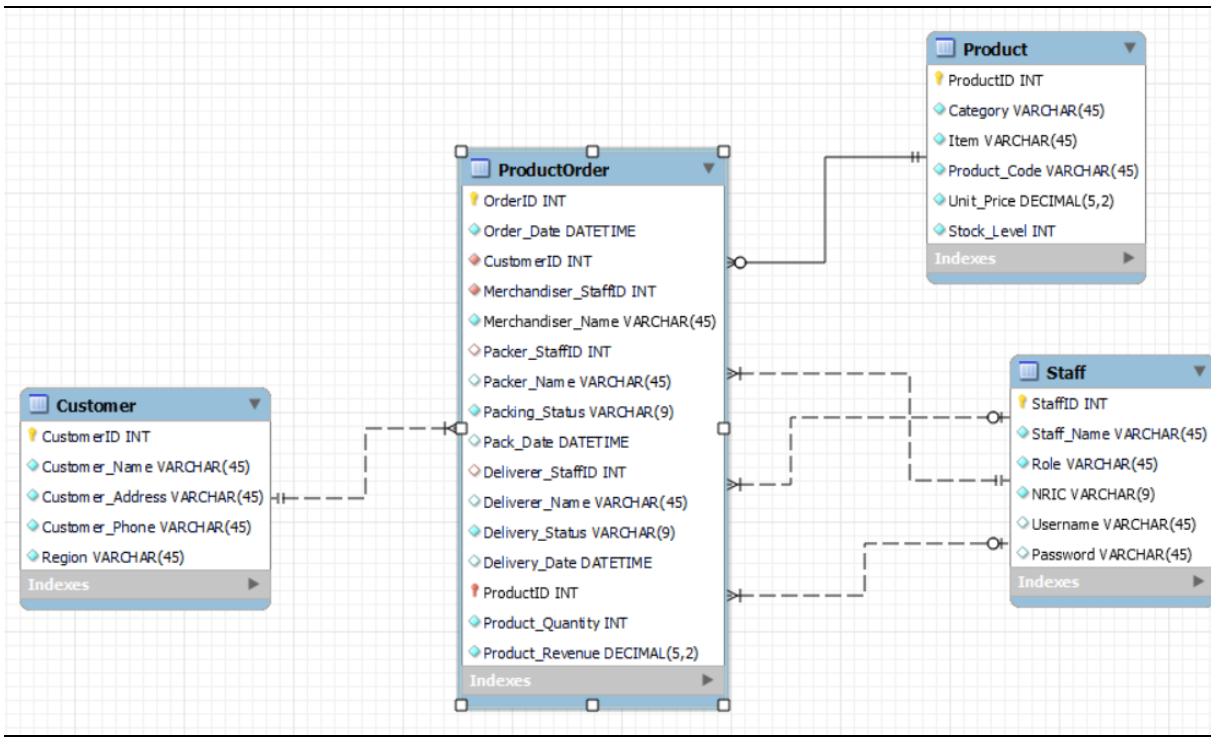
   - For Product table, identified Stock level as an attribute from the image below:

     2. Check stock level for individual product.

   - For Order table, identified Packing status, Delivery status as attributes from the image below:

     11.See individual sales order fulfilment details by packer and deliverer.

4. Created the **ER Diagram**, using the entities and attributes found above



Each ProductOrder can be made by **1 and only 1** customer while each customer can make **1 or many** ProductOrders

Each ProductOrder contains **1 and only 1** product while each product can be placed in **0 or more** ProductOrders

*It is possible to have a product not to be placed in any order, it could be the least favourite product of all customers and no one chooses it, hence 0 or more

Each ProductOrder is placed by **1 and only 1** Merchandiser Staff while each Merchandiser can place **1 or many** ProductOrders

Each ProductOrder can be handled by **0 or 1** Packer Staff while each Packer can pack **1 or many** ProductOrders

* Pack status with 0 packer = ‘pending’, pack status with 1 packer = ‘fulfilled’ and cannot have >1 packers handling an order

Each ProductOrder can be handled by **0 or 1** Deliverer Staff while each Deliverer can deliver **1 or many** ProductOrders

* Delivery status with 0 deliverer = ‘pending’, delivery status with 1 delivery = ‘fulfilled’ and cannot have >1 deliverers handling an order

5. Identified the **functional dependencies, partial dependencies and transitive dependencies**

   - Since each order can have 1 or many products and each product can be placed in 1 or many orders, we need OrderID and ProductID to form a composite key:

     {OrderID, ProductID} → ProductQuantity, ProductRevenue

   - OrderID → OrderDate, CustomerID, MerchandiserStaffID, PackerStaffID, PackingStatus, PackDate, DelivererStaffID, DeliveryStatus, DeliveryDate **(Partial dependencies)**

   - ProductID → Category, Item, ProductCode, UnitPrice, StockLevel

   - MerchandiserStaffID → MerchandiserName **(Transitive dependency)**

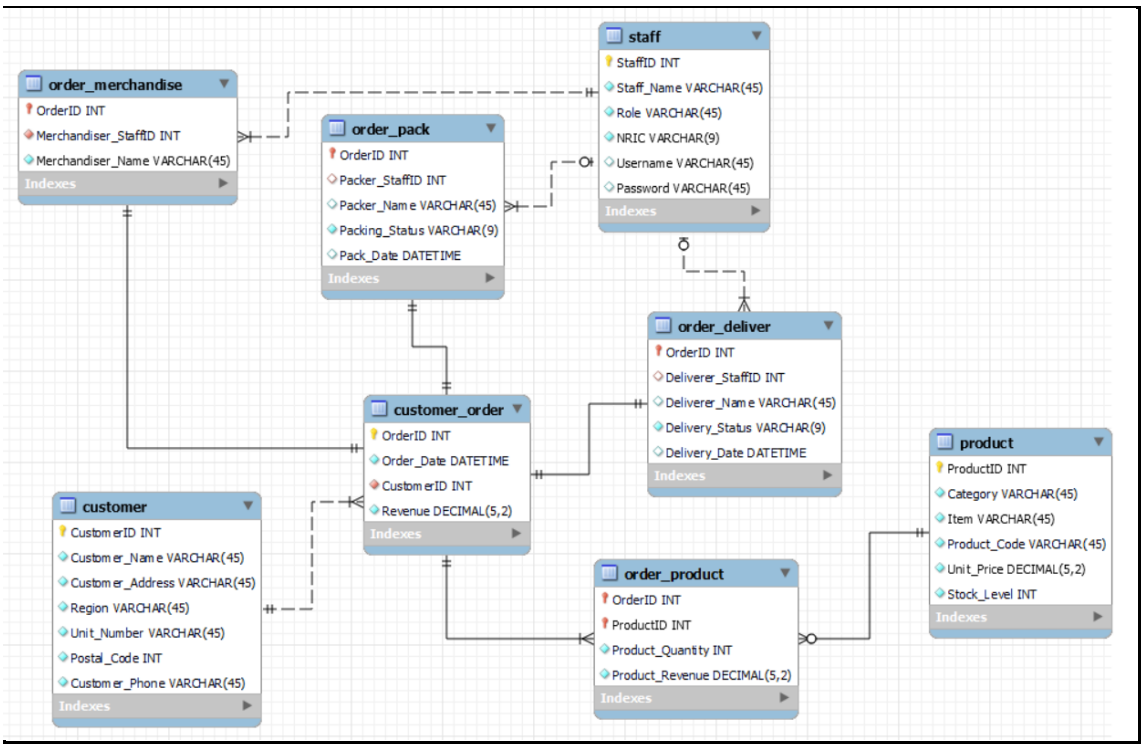
   - PackerStaffID → PackerName **(Transitive dependency)**

   - DelivererStaffID → DelivererName **(Transitive dependency)**

   - CustomerID → CustomerName, CustomerAddress, CustomerPhone, Region

   - StaffID → StaffName, Role, NRIC, Username, Password

6. **Normalised ERD** after removing the partial and transitive dependencies



Customer - customer_order

Each Customer can make **1 or many** Orders while each Order belongs to **1 and only 1** Customer.

Product - order_product

Each Product can be placed in **0 or more** Orders while each Order can contain **1 and only 1** Product.

*It is possible to have a product not to be placed in any order, it could be the least favourite product of all customers and no one chooses it, hence 0 or more

Staff - order_merchandise

Each merchandiser Staff can place **1 or many** Orders while each Order can be placed by **1 and only 1** merchandiser Staff

Staff – order_pack

Each packer Staff can pack **1 or many** Orders while each Order can contain **0 or 1** packer Staff

* Pack status with 0 packer = ‘pending’, pack status with 1 packer = ‘fulfilled’ and cannot have >1 packers handling an order

Staff – order_deliver

Each deliverer Staff can deliver **1 or many** Orders while each Order can contain **0 or 1** deliverer Staff

* Delivery status with 0 deliverer = ‘pending’, delivery status with 1 delivery = ‘fulfilled’ and cannot have >1 deliverers handling an order

<u>Customer_order – order_product</u>

Each order can contain **1 or many** products while each ordered product belongs to **1 and only 1** order

<u>Customer_order – order_merchandise</u>

Each order contains **1 and only 1** order merchandiser info while each order merchandiser info belongs to **1 and only 1** order

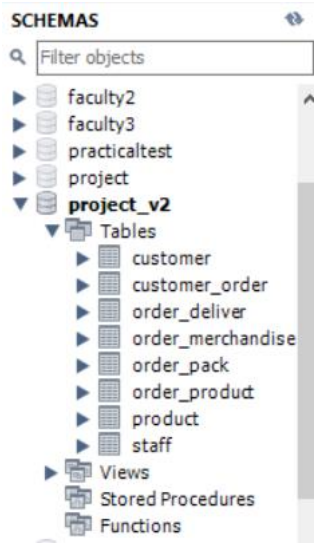
<u>Customer_order – order_pack</u>

Each order contains **1 and only 1** order packer info while each order packer info belongs to **1 and only 1** order

<u>Customer_order – order_deliver</u>

Each order contains **1 and only 1** order deliverer info while each order deliverer info belongs to **1 and only 1** order

Derived Schema



Customer table

CustomerID	Customer_Name	Customer_Address	Region	Unit_Number	Postal_Code	Customer_Phone
1	Cheers Hougang Central	810 Hougang Central	North East	#03-10	530810	67589651
3	bro	somewhere 711	North East	#06-14	770234	90352532
NULL	NULL	NULL	NULL	NULL	NULL	NULL

PK: CustomerID

CustomerOrder table

OrderID	Order_Date	CustomerID	Revenue
10	2022-07-23 15:39:26	1	166.00
12	2022-07-30 17:38:24	1	37.00
NULL	NULL	NULL	NULL

PK: OrderID, FK: CustomerID

OrderDeliver table

OrderID	Deliverer_StaffID	Deliverer_Name	Delivery_Status	Delivery_Date
10	1	Nunez Myles Blasco	Fulfilled	2022-07-30 17:37:37
12	NULL	NULL	Pending	NULL
NULL	NULL	NULL	NULL	NULL

PK: OrderID, FK: Deliverer_StaffID (StaffID)

OrderMerchandiser table

OrderID	Merchandiser_StaffID	Merchandiser_Name
10	1	Nunez Myles Blasco
12	1	Nunez Myles Blasco
NULL	NULL	NULL

PK: OrderID, FK: Merchandiser_StaffID (StaffID)

OrderPack table

OrderID	Packer_StaffID	Packer_Name	Packing_Status	Pack_Date
10	1	Nunez Myles Blasco	Fulfilled	2022-07-25 18:48:54
12	1	Nunez Myles Blasco	Fulfilled	2022-07-30 17:41:10
NULL	NULL	NULL	NULL	NULL

PK: OrderID, FK: Packer_StaffID (StaffID)

OrderProduct table

OrderID	ProductID	Product_Quantity	Product_Revenue
10	4	20	16.00
10	5	30	150.00
12	3	20	29.00
12	4	10	8.00
NULL	NULL	NULL	NULL

PK: {OrderID, ProductID}, FK: OrderID, ProductID

Product table

ProductID	Category	Item	Product_Code	Unit_Price	Stock_Level
1	Diary	Milk	DKRS35501	2.30	500
2	Bakery	Bread	BVR30221	2.70	500
3	Canned Goods	Tomato	CVOMS3681	1.45	480
4	Condiments	Fine Salt	CON2ASR801	0.80	470
5	Beverages	Green Tea	BRTE39861	5.00	470
NULL	NULL	NULL	NULL	NULL	NULL

PK: ProductID

Staff table

StaffID	Staff_Name	Role	NRIC	Username	Password
1	Nunez Myles Blasco	Admin	T0332991C	myles	myles991C
5	joe	Packer	T0112334C	joemama	joemama
NULL	NULL	NULL	NULL	NULL	NULL

PK: StaffID

Application Interface (Python)

Design Rationale

- Login screen
  - Input username and password
  - After successful login, display the menu that corresponds to user's role

- **Manager menu**

  - Task:

Manager

1. Add, Retrieve, Update, or Remove individual product (See Appendix A).
2. Add, Retrieve, Update, or Remove individual staff (See Appendix B).
3. Add, Retrieve, Update, or Remove individual outlet (See Appendix B).
4. Update quantity and unit selling price of existing product.
5. See the sales details made by individual merchandiser.
6. See the sales for individual product.
7. See the sales for all/individual product category.
8. See the sales for all/individual outlet.
9. See the sales by region (North, Central, East, West).
10. See the sales order details for individual outlet.
11. See individual sales order fulfilment details by packer and deliverer.

- Program flow
  - Manage products (task 1, 4)
  - Manage staff (task 2)
  - Manage outlets (task 3)
  - View sales
    - By product (task 6)
    - By product category (task 7)
    - By customer (task 8)
    - By region (task 9)
  - View order details by merchandiser (task 5)
  - View order details by outlet (task 10)
  - View order fulfilment details by packer & deliverer (task 11)

- Python interface

```
Manager Menu
====================
1. Manage products
2. Manage staff
3. Manage outlets
4. View sales
5. View order details by merchandiser
6. View order details by outlet
7. View order fulfilment details by packer & deliverer
8. Log out
====================
Enter Selection:
```

```
Manager Menu - Manage products
====================
1. Add product
2. Update product details
3. Remove product
4. Retrieve product details
5. Back
====================
Enter Selection:
```

```
Manager Menu - Manage staff
====================
1. Add staff
2. Update staff details
3. Remove staff
4. Retrieve staff details
5. Back
====================
Enter Selection:
```

```
Manager Menu - Manage Customer (Outlets)
====================
1. Add customer
2. Update customer details
3. Remove customer
4. Retrieve customer details
5. Back
====================
Enter Selection:
```

```
Manager Menu - View Sales
====================
1. By product
2. By product category
3. By customer
4. By region
5. Back
====================
Enter Selection:
```

- **Merchandiser menu**
  - Task:

Merchandiser
1. Add, Retrieve, Update, or Remove own order for individual outlets.
2. Check stock level for individual product.
3. Check the fulfilment status for individual outlet.

  - Program flow
    - Manage customer order (task 1)

- View product quantity (task 2)
- View customer order fulfilment status (task 3)

- Python interface

```
Merchandiser Menu
====================
1. Manage customer order
2. View product quantity
3. View customer order fulfilment status
4. Log out
====================
Enter Selection:
```

```
Merchandiser Menu - Manage customer order
====================
1. Add order
2. Update order details
3. Remove order
4. Retrieve order details
5. Back
====================
Enter Selection:
```

Packer menu

- Task:

Packer

1. Retrieve unfulfilled sales order.
2. Retrieve own fulfilled sales order.
3. Endorse sales order with date of fulfilment.

- Program flow

  - Retrieve unpacked orders (task 1)
  - Retrieve own packed orders (task 2)
  - Endorse customer order (task 3)

- Python interface

```
Packer Menu
====================
1. Retrieve unpacked orders
2. Retrieve own packed orders
3. Endorse customer order
4. Log out
====================
Enter Selection:
```

- **Deliverer menu**
  - Task:

Deliverer

1. Retrieve undelivered order.
2. Retrieve own delivered order.
3. Endorse sales order with date of delivery.

  - Program flow
    - Retrieve undelivered orders (task1)
    - Retrieve own delivered orders (task 2)
    - Endorse customer order (task 3)
  - Python interface

```
Deliverer Menu
====================
1. Retrieve undelivered orders
2. Retrieve own delivered orders
3. Endorse customer order
4. Log out
====================
Enter Selection:
```

Special Features

1. Added admin role to access all 4 roles menus for faster python interface development
2. Used parameterized query in python to protect against SQL injection attacks

Conclusion

- Accomplishment / Outcome
  - Fulfilled all tasks, where functions of all 4 roles are working

- Database has some security with the use of views and parameterized query
- Removed Insert, Update and Delete anomalies through normalisation which causes data to be more consistent and accurate

- Future Enhancement
  - For this project, it is assumed that if a product's unit price has been updated and when we retrieve past orders, the product revenue from past orders changes as well. However, in real life, that should not happen as details from past orders should stay constant. Hence, a future enhancement is to fix this problem by adding a 'Unit Price' column into 'ProductOrder' table to keep the old Unit Price and prevent any misinformation of revenue from past orders.
  - Hash password input in login screen for better security when logging in
  - Add a stock level checker where if the merchandiser places a product order with product's stock level at 0 OR exceeds the available stock level, the order should be invalid