B9IS100-Advanced Databases CA-1

Topic

Food Menu TakeAway Application Database Design

BY

Name: Michael Doyle
Reg No: 10636083

Name: Enoja Livinus Onyebuchi
Reg No: 10606259

Name: Onyeka Ebele
Reg No: 10621453

1

Table of Content

1 .0 Introduction

With the increased stability of the internet these days, people want to be able to order for items online and get the order delivered to their home address quickly. They can choose to pay online or pay on delivery. The crave for a highly reliable and efficient food delivery application is going high by the day. That is why my team decided to create a product that will help fill this gap.

"The food industry, like many others, has gained from the e-commerce boom. That is, by making food accessible on online channels, the growing number of food distribution apps and web-sites has transformed the landscape of the food industry. Basically, it's called online food ordering to order food through a web page or mobile application. During the last few years, online food ordering has grown steadily. Food delivery services have transitioned from ordering via telephone to digital ordering to satisfy customer needs, particularly with the continuous advances in technology. In other words, online platforms generate new possibilities for the food industry to attract more consumers" (R. Ramesha, 2021).

TakeAway project is a food delivery web application database. Customers come to the web application, search for food menu they like and select them for order. Once food menus are selected, all selected menu is added to cart. When menu selection is completed, the visitor clicks on checkout to view a summary of what was selected. The customer must be logged on as a valid customer with delivery address information captured. The customer proceeds to complete order by selecting payment option. Once payment option is selected or payment completed, the order is assigned to a delivery agent that ensures the order is delivered to the customer's home address. The recipe of the menu items and its ingredients are stored. The ingredients are supplied by a registered and vetted supplier in system. The managerial staff can view the history of order details and orders made by all registered customers.

1.1 Project Scope

The scope of this project is to develop a relational schema for a hybrid database system that will meet all the business cases required to serve users of this product effectively and reliably. We will create the entity relational diagram using draw.io. Normalise the entity diagram in 3 normal forms for more efficient reporting and write the SQL queries that will create the database tables in Microsoft SQL relational database server 2019.

We developed at least seven business rules and demonstrated the execution of each business rule using stored procedure.

1.2 Business Requirements

3

TakeAway is a system that automate day to day operations of food menu business from order to preparation and delivery. The system can be used by customers to place an order and track the status of their order until the order is delivered. The customer can search for food menu according to price range and food menu category. Before a customer can place an order, the customer must be registered on the system. The customers should be able to choose preferred mode of payment; pay with card or pay on delivery.

The system helps TakeAway staff to manage customer information, customer orders, create available food menu, place orders for customers, view order status, assign orders to food menu delivery agents and view order reports. The system keeps track of ingredients, ingredients suppliers and food menu order delivery agent information.

The TakeAway staff are classified into manager, waiter and cooks. The manager creates the food menu on the system and have access to entire backend management modules. The waiter can make order for customers, view orders, order status, update order status and can assign orders to food menu delivery agents. The cooks can view orders, update order status and update ingredients stock quantity.

4

2.0 Business Rules and Implementations

Business rules for a system describe the operations, definitions and constraints that apply to achieving the systems goals. These rules are used to help the system to better achieve goals, communicate among principals and agents, communicate between the system owner and interested third parties, demonstrate fulfillment of legal obligations, operate more efficiently, automate operations, perform analysis on current practices, etc (Wales, n.d.).

2.1 All Food Menu Orders in the System MUST be Assigned to a Customer and the Customer Cannot be Changed.

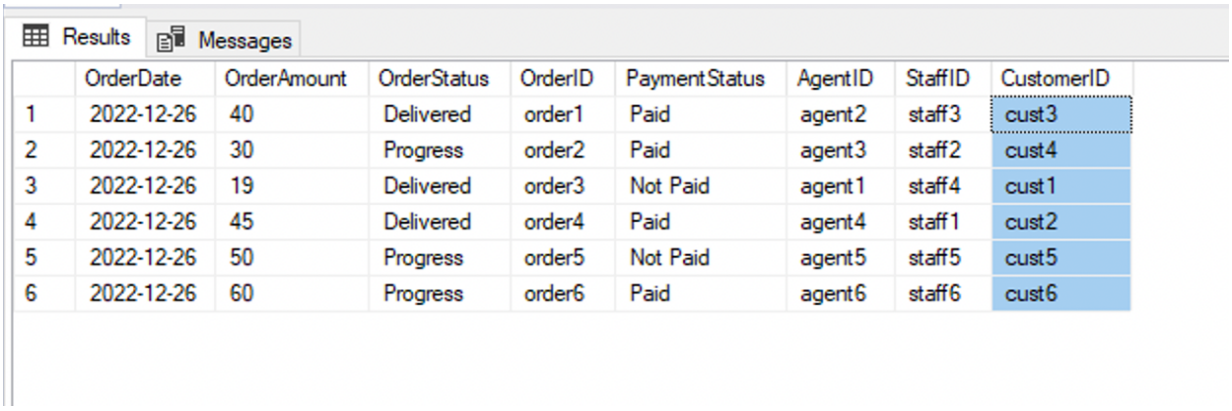
This rule helps to ensure that all orders in the system is created by a registered customer in the system. Imagine an order in the system that cannot be traced to a registered customer, this will mean that the order is not valid and cannot be serviced. The customer and order identifier cannot be changed once created to ensure the integrity of the system.

Implementations:

```
CREATE TABLE tbl_Order
(
OrderDate date not null,
OrderAmount float not null,
OrderStatus varchar(50) not null,
OrderID varchar(100) not null,
PaymentStatus varchar(10) not null,
AgentID varchar(30),
StaffID varchar(30),
CustomerID  varchar(30) not null, -- (this ensures that all orders MUST have a customer)
PRIMARY KEY (OrderID),
FOREIGN KEY (AgentID) REFERENCES tbl_DeliveryAgent (AgentID),
FOREIGN KEY (StaffID) REFERENCES tbl_Staff (StaffID),
FOREIGN KEY (CustomerID) REFERENCES tbl_Customer (CustomerID),
)
```

Sample:

5



Results | Messages

	OrderDate	OrderAmount	OrderStatus	OrderID	PaymentStatus	AgentID	StaffID	CustomerID
1	2022-12-26	40	Delivered	order1	Paid	agent2	staff3	cust3
2	2022-12-26	30	Progress	order2	Paid	agent3	staff2	cust4
3	2022-12-26	19	Delivered	order3	Not Paid	agent1	staff4	cust1
4	2022-12-26	45	Delivered	order4	Paid	agent4	staff1	cust2
5	2022-12-26	50	Progress	order5	Not Paid	agent5	staff5	cust5
6	2022-12-26	60	Progress	order6	Paid	agent6	staff6	cust6

2.2 Order Status Cannot be Set to Delivered Unless Payment Status is “Paid”

It is important to ensure that payment for an order is confirmed before the order status can be changed to “delivered”. Else, it will not make sense to update the order status to “delivered” if payment for the order is not confirmed. Order status is a confirmation whether value has been delivered to the customer or not. We believe that a customer will not complete payment if value is not delivered.

Implementations:

```
CREATE TABLE tbl_Order
(
OrderDate date not null,
OrderAmount float not null,
OrderStatus varchar(50) not null,   --(This cannot be “Delivered” unless Payment Status is “Paid”)
OrderID varchar(100) not null,
PaymentStatus varchar(10) not null, -- (Only when this is “Paid” can order status be “delivered”)
AgentID varchar(30),
StaffID varchar(30),
CustomerID  varchar(30) not null,
PRIMARY KEY (OrderID),
FOREIGN KEY (AgentID) REFERENCES tbl_DeliveryAgent (AgentID),
FOREIGN KEY (StaffID) REFERENCES tbl_Staff (StaffID),
FOREIGN KEY (CustomerID) REFERENCES tbl_Customer (CustomerID),
)
```

Sample:

6

Results | Messages

	OrderDate	OrderAmount	OrderStatus	OrderID	PaymentStatus	AgentID	StaffID	CustomerID
1	2022-12-26	40	Delivered	order1	Paid	agent2	staff3	cust3
2	2022-12-26	30	Progress	order2	Paid	agent3	staff2	cust4
3	2022-12-26	19	Delivered	order3	Not Paid	agent1	staff4	cust1
4	2022-12-26	45	Delivered	order4	Paid	agent4	staff1	cust2
5	2022-12-26	50	Progress	order5	Not Paid	agent5	staff5	cust5
6	2022-12-26	60	Progress	order6	Paid	agent6	staff6	cust6

2.3 Ingredients in the System Must Have a Registered Supplier

All ingredients recorded in the system should be traceable to a registered supplier. This is important for accounting purpose and reconciliation needs. Imagine if an issue is noticed on an ingredient and the business cannot trace the supplier. This constraint will help solidify the integrity of the suppliers knowing that any bad ingredients can be traced back to the supplier when required.

Implementations:

```
CREATE TABLE tbl_Ingredient
(
IngredientID varchar(30) not null,
IngredientName varchar(20) not null,
IngredientPrice float not null,
IngredientCreateDate  date not null,
SupplierID varchar(30) not null,    -- (This must be set for all ingredients)
PRIMARY KEY (IngredientID),
FOREIGN KEY (SupplierID) REFERENCES tbl_Supplier (SupplierID),
)
```

Sample:

Results | Messages

	IngredientID	IngredientName	IngredientPrice	IngredientCreateDate	SupplierID
1	ing1	Flower	15	2022-12-26	supplyer3
2	ing2	Vegetable Oil	25	2022-12-26	supplyer2
3	ing3	Frozen Beef	34	2022-12-26	supplyer1
4	ing4	Food Seasoning	20.5	2022-12-26	supplyer5
5	ing5	Salt	45.8	2022-12-26	supplyer4
6	ing6	Hot Pepper	35	2022-12-26	supplyer2

7

2.4 Menu Items Must Belong to a Category

For proper organization of the menu items on the system, it is important that all menu items are categorized according to their features, food type or target customers. The categorization of menu items will help for easier searching or filtering of menu items.

Implementations:

```
CREATE TABLE tbl_MenuItem
(
MenuItemID  varchar(30) not null,
MenuName    varchar(50) not null,
MenuPrice   float   not null,
CategoryID  varchar(20) not null, -- (This must exist for a menu item to be created)
RecipeID    varchar(20) not null,
MenuDescription varchar(200) not null,
MenuImage   image,
Alegies varchar(50),
PRIMARY KEY (MenuItemID),
FOREIGN KEY (CategoryID)    REFERENCES  tbl_Category(CategoryID),
FOREIGN KEY (RecipeID)  REFERENCES  tbl_Recipe(RecipeID),
)
```

Sample:

Results | Messages

	MenuItemID	MenuName	MenuPrice	CategoryID	RecipeID	MenuDescription	MenuImage	Alegies
1	menu1	Spring Rolls	5	cat4	recipe1	Spring roll is a traditional Chinese savory snack wher...	0xFFD8FFE000104A46494600010100000100010000FFDB00...	
2	menu2	Roast Pork with Cracle	7	cat5	recipe2	Transfer pork to a roasting dish and roast for 50 minu...	0xFFD8FFE000104A46494600010100000100010000FFDB00...	
3	menu3	Coca Cola	2	cat1	recipe3	Coca-Cola, or Coke, is a carbonated soft drink manu...	0xFFD8FFE000104A46494600010100000100010000FFDB00...	
4	menu4	Crispy Chilli Chicken Udon	10	cat2	recipe4	Chilli Chicken Udon Noodle Soup by Chef Kwanghi ...	0xFFD8FFE000104A46494600010100000100010000FFDB00...	
5	menu5	Teriyaki Sauce Japanese Style	9	cat3	recipe5	Teriyaki is a Japanese cooking technique used to c...	0xFFD8FFE000104A46494600010100000100010000FFDB00...	
6	menu6	Pepper Chicken Recipe	9	cat4	recipe6	Pepper Chiken is a Japanese cooking technique us...	0xFFD8FFE000104A46494600010100000100010000FFDB00...	

2.5 Registered Suppliers Must Have an Email, Address and Mobile Number

In other to easily communicate with registered suppliers in the system, it is pertinent that the systems capture the suppliers contact details like the email address, office address and mobile number or phone number. This business rule demands that all registered suppliers in the TakeAway system must have their email address, office address and mobile or phone number recoded.

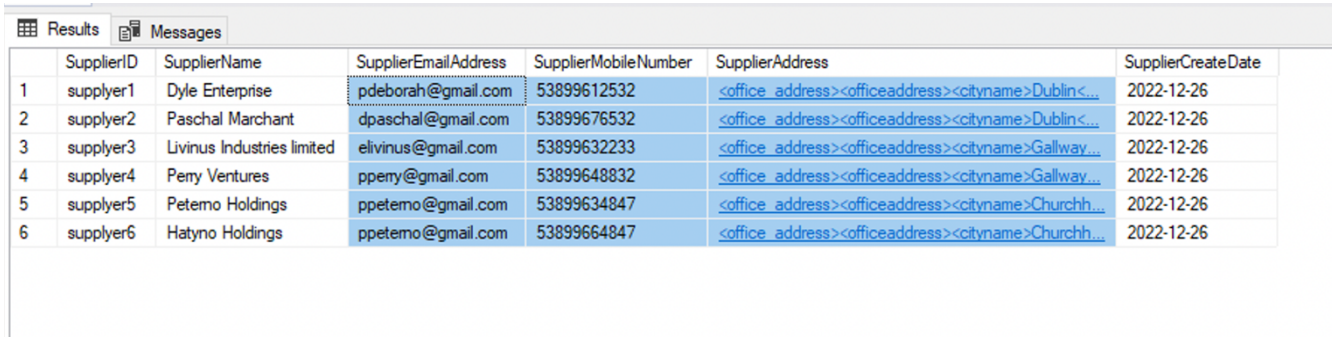
Implementations:

```
CREATE TABLE tbl_Supplier
```


8

```
(
SupplierID varchar(30) not null,
SupplierName varchar(50) not null,
SupplierEmailAddress varchar(50) not null, -- (this must exist)
SupplierMobileNumber varchar(14) not null, -- (this must exist)
SupplierAddress  xml (SupplierAddressCollection), -- (This must exist)
SupplierCreateDate  date not null,
PRIMARY KEY (SupplierID),
)
```

Sample:



Results | Messages

	SupplierID	SupplierName	SupplierEmailAddress	SupplierMobileNumber	SupplierAddress	SupplierCreateDate
1	supplyer1	Dyle Enterprise	pdeborah@gmail.com	53899612532	<office_address><officeaddress><cityname>Dublin<...	2022-12-26
2	supplyer2	Paschal Marchant	dpaschal@gmail.com	53899676532	<office_address><officeaddress><cityname>Dublin<...	2022-12-26
3	supplyer3	Livinus Industries limited	elivinus@gmail.com	53899632233	<office_address><officeaddress><cityname>Gallway...	2022-12-26
4	supplyer4	Perry Ventures	pperry@gmail.com	53899648832	<office_address><officeaddress><cityname>Gallway...	2022-12-26
5	supplyer5	Peterno Holdings	ppeterno@gmail.com	53899634847	<office_address><officeaddress><cityname>Churchh...	2022-12-26
6	supplyer6	Hatyno Holdings	ppeterno@gmail.com	53899664847	<office_address><officeaddress><cityname>Churchh...	2022-12-26

2.6 Every Customer Must Have Email, Phone and Home Address

In the system, all orders must be made by a registered customer. To deliver the order, the customers home address or phone will be required. The home address and phone number must be valid and can be updated by customers in case there is a change. It is the responsibility of the customers to ensure that their contact information is accurate all times.

Implementations:

```
CREATE TABLE tbl_Customer
(
CustomerID  varchar(30) not null,
CustomerFirstName varchar(20) not null,
CustomerLastName varchar(20) not null,
CustomerEmailAddress varchar(50) not null, -- (This must exist)
CustomerMobileNumber varchar(14) not null, -- (This must exist)
CustomerPostCode varchar(10) not null,
CustomerCity varchar(10) not null,
CustomerHomeAddress varchar(100) not null, -- (This must exist
CustomerCreateDate  date not null,
CustomerPassword varchar(100) not null
PRIMARY KEY (CustomerID),
)
```

Samples:

9

Results | Messages

	CustomerID	CustomerFirstName	CustomerLastName	CustomerEmailAddress	CustomerMobileNumber	CustomerPostCode	CustomerCity	CustomerHomeAddress	CustomerCreateDate	CustomerPassword
1	cust1	John	Gowel	gjohn@gmail.com	353899634532	G2T4	Dublin	No 23, Shagan Avenue	2022-12-26	P@ssJ0hn#
2	cust2	Michael	Doyle	dmichael@gmail.com	353899635532	N2G2	Dublin	No 33 Tayak Road	2022-12-26	P@ss@D0ylE#
3	cust3	Livinus	Enoja	elivinus@gmail.com	353899634533	Y2f5	Cork	No 67 Bijock Avenue	2022-12-26	en0j@P@ss#
4	cust4	Ebele	Onyeka	oebele@gmail.com	353899644532	B3Y4	Dublin	89 Asuka Avenue	2022-12-26	0ny3k@PAss#
5	cust5	Jackson	Peters	pjackson@gmail.com	353899634536	H7D3	Baywood	44 Badeye Cross Road	2022-12-26	P3t3rzP@ss#
6	cust6	Jack	Pet	pjack@gmail.com	353899674536	H7U3	Baywood	42 Badeye Cross Road	2022-12-26	P3t3rzP@ss#

2.7 Customers Cannot be Created Without a Password

Customers will need to authenticate their access to the system by providing a valid username and password. The system will also support multi-factor authentication by sending a valid token to the customers phone number or email address. For this authentication to be possible, all customers must provide a valid password that meets the password standards as proscribed by security standard like ISO 27001 (CISA, 2021).

Implementations:

```
CREATE TABLE tbl_Customer
(
CustomerID  varchar(30) not null,
CustomerFirstName varchar(20) not null,
CustomerLastName varchar(20) not null,
CustomerEmailAddress varchar(50) not null,
CustomerMobileNumber varchar(14) not null,
CustomerPostCode varchar(10) not null,
CustomerCity varchar(10) not null,
CustomerHomeAddress varchar(100) not null,
CustomerCreateDate  date not null,
CustomerPassword varchar(100) not null -- (This must exist
PRIMARY KEY (CustomerID),
)
```

Sample:

Results | Messages

	CustomerID	CustomerFirstName	CustomerLastName	CustomerEmailAddress	CustomerMobileNumber	CustomerPostCode	CustomerCity	CustomerHomeAddress	CustomerCreateDate	CustomerPassword
1	cust1	John	Gowel	gjohn@gmail.com	353899634532	G2T4	Dublin	No 23, Shagan Avenue	2022-12-26	P@ssJ0hn#
2	cust2	Michael	Doyle	dmichael@gmail.com	353899635532	N2G2	Dublin	No 33 Tayak Road	2022-12-26	P@ss@D0ylE#
3	cust3	Livinus	Enoja	elivinus@gmail.com	353899634533	Y2f5	Cork	No 67 Bijock Avenue	2022-12-26	en0j@P@ss#
4	cust4	Ebele	Onyeka	oebele@gmail.com	353899644532	B3Y4	Dublin	89 Asuka Avenue	2022-12-26	0ny3k@PAss#
5	cust5	Jackson	Peters	pjackson@gmail.com	353899634536	H7D3	Baywood	44 Badeye Cross Road	2022-12-26	P3t3rzP@ss#
6	cust6	Jack	Pet	pjack@gmail.com	353899674536	H7U3	Baywood	42 Badeye Cross Road	2022-12-26	P3t3rzP@ss#

2.8 Order Status Cannot be Set to "out for delivery" Unless an Agent is Assigned to the Order

All orders are to be delivered by a valid delivery agent in the system. Therefore, for an order to be marked as “out for delivery”, it must have been assigned to a delivery agent. This will help the business know when delivered all orders for reconciliation and accounting purpose.

10

Implementations:

```
CREATE TABLE tbl_Order
(
OrderDate date not null,
OrderAmount float not null,
OrderStatus varchar(50) not null,   --(This cannot be “Out for Delivery” unless AgentID is set)
OrderID varchar(100) not null,
PaymentStatus varchar(10) not null,
AgentID varchar(30), -- (Only when this is set can order status be “Out for Delivery”)
StaffID varchar(30),
CustomerID  varchar(30) not null,
PRIMARY KEY (OrderID),
FOREIGN KEY (AgentID) REFERENCES tbl_DeliveryAgent (AgentID),
FOREIGN KEY (StaffID) REFERENCES tbl_Staff (StaffID),
FOREIGN KEY (CustomerID) REFERENCES tbl_Customer (CustomerID),
)
```

Sample:

Results Messages

	OrderDate	OrderAmount	OrderStatus	OrderID	PaymentStatus	AgentID	StaffID	CustomerID
1	2022-12-26	40	Delivered	order1	Paid	agent2	staff3	cust3
2	2022-12-26	30	Progress	order2	Paid	agent3	staff2	cust4
3	2022-12-26	19	Delivered	order3	Not Paid	agent1	staff4	cust1
4	2022-12-26	45	Delivered	order4	Paid	agent4	staff1	cust2
5	2022-12-26	50	Out for Delivery	order5	Not Paid	agent5	staff5	cust5
6	2022-12-26	60	Progress	order6	Paid	agent6	staff6	cust6

2.9 Order Status Cannot be Set to "cooking" Unless a Cooking Staff is Assigned to the Order

All orders must have a registered cooking staff assigned. Only when a cooking staff is assigned can the order status be “cooking”. This will help ensure that all orders can be traced to a cooking staff.

Implementations:

```
CREATE TABLE tbl_Order
(
OrderDate date not null,
OrderAmount float not null,
```


11

```
OrderStatus varchar(50) not null,   --(This cannot be "Cooking" unless StaffID is set)
OrderID varchar(100) not null,
PaymentStatus varchar(10) not null,
AgentID varchar(30),
StaffID varchar(30), -- (Only when this is set can order status be "Cooking")
CustomerID  varchar(30) not null,
PRIMARY KEY (OrderID),
FOREIGN KEY (AgentID) REFERENCES tbl_DeliveryAgent (AgentID),
FOREIGN KEY (StaffID) REFERENCES tbl_Staff (StaffID),
FOREIGN KEY (CustomerID) REFERENCES tbl_Customer (CustomerID),
)
```

Sample:

Results | Messages

	OrderDate	OrderAmount	OrderStatus	OrderID	PaymentStatus	AgentID	StaffID	CustomerID
1	2022-12-26	40	Delivered	order1	Paid	agent2	staff3	cust3
2	2022-12-26	30	Cooking	order2	Paid	agent3	staff2	cust4
3	2022-12-26	19	Delivered	order3	Not Paid	agent1	staff4	cust1
4	2022-12-26	45	Delivered	order4	Paid	agent4	staff1	cust2
5	2022-12-26	50	Out for Delivery	order5	Not Paid	agent5	staff5	cust5
6	2022-12-26	60	Progress	order6	Paid	agent6	staff6	cust6

12

3.0 TakeAway Relational Schema

A relational schema is a set of relational tables and associated items that are related to one another (Ricardo, 2003). Simply put, relational schema defines the design and structure of the relation. It consists of the relations name, set of attributes/field names/column names (Saxena, 2021). The TakeAway schema consists of ten relational tables and attributes that are related to one another.

3.1 Relational Diagram

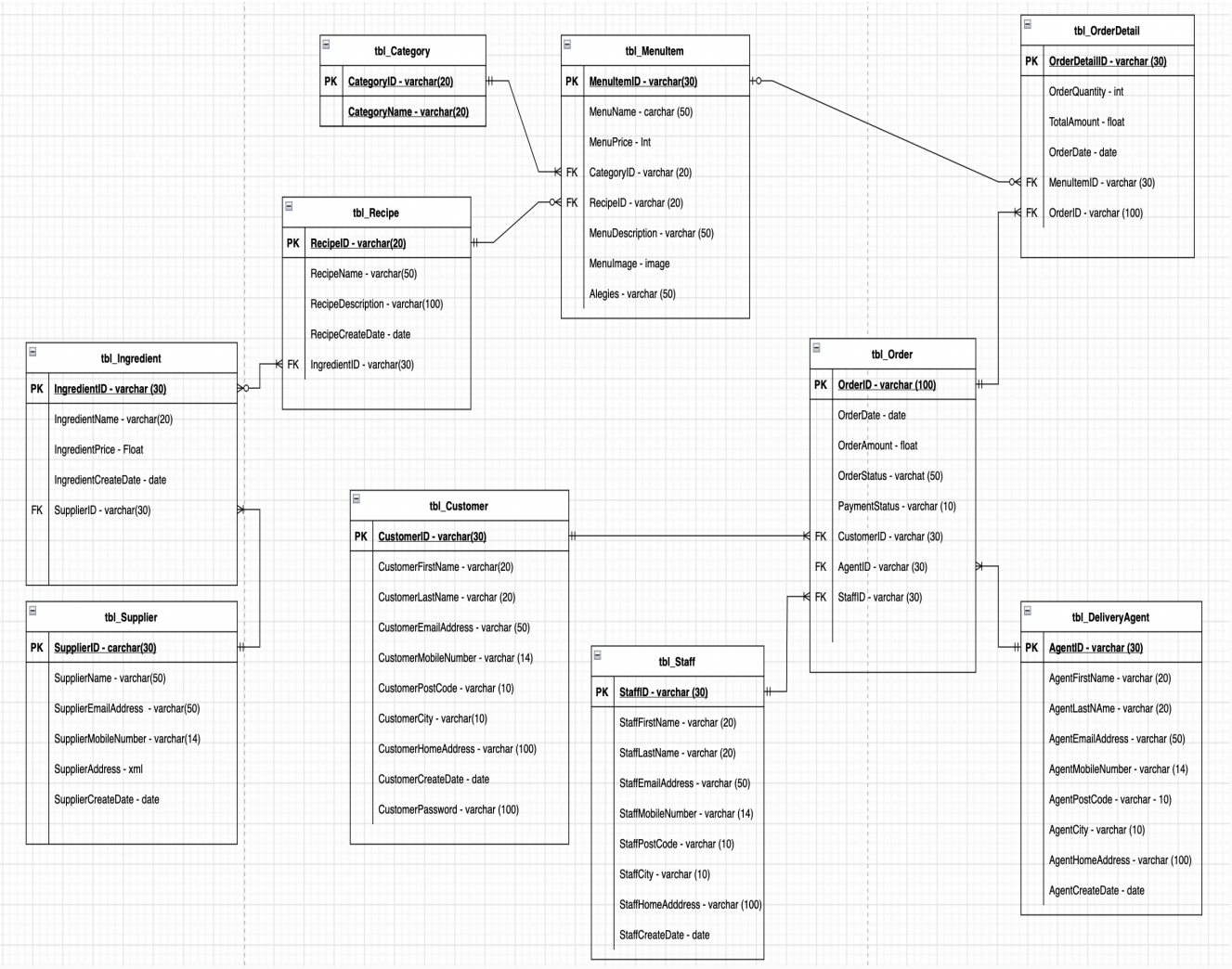


Fig 1 Relational Schema

3.2 Schema Normalisation

Normalization is the branch of relational theory that provides design insights. It is the process of determining how much redundancy exists in a table. Normal forms involve a set of dependency properties that a schema must satisfy. Each normal form gives guarantees about the presence and/or absence of update anomalies (Adrienne Watt and Watt, 2014).

13

All the tables in the TakeWay schema are in 3 normal forms because of the following reasons.

- Only single values are permitted in the intersection of each row and column therefore there is not repeating group in all tables in the schema.
- The primary key comprises a single attribute. In other words, all columns in the tables are dependent on the primary key.
- All transitive dependencies were removed.
- No non-key attribute is functionally dependent on another non-key attributes.

3.3 XML in Schema

We chose to represent the supplier office address in xml format to accommodate the office address of each supplier. A supplier can have one or more office in different cities or counties with different postal codes.

XML Schema Collection Create Query:

```
CREATE XML SCHEMA COLLECTION SupplierAddressCollection
AS
'<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <xsd:element name="office_address">
         <xsd:complexType>
            <xsd:sequence>
               <xsd:element ref="officeaddress" minOccurs="1" maxOccurs="unbounded" />
            </xsd:sequence>
         </xsd:complexType>
   </xsd:element>
   <xsd:element name="officeaddress">
         <xsd:complexType>
            <xsd:sequence>
               <xsd:element name="cityname" type="xsd:string" />
               <xsd:element name="postcodedetails" type="xsd:string" />
            </xsd:sequence>
         </xsd:complexType>
   </xsd:element>
</xsd:schema>'
```

Data Insert Query:

```
INSERT INTO tbl_Supplier VALUES
('supplyer1','Dyle Enterprise','pdeborah@gmail.com','53899612532','<?xml version="1.0"?>
<office_address>
   <officeaddress>
```


14

```
    <cityname>Dublin</cityname>
    <postcodedetails>D07 D3C5</postcodedetails>
  </officeaddress>
  <officeaddress>
    <cityname>Cork</cityname>
    <postcodedetails>C11 G7C5</postcodedetails>
  </officeaddress>
  <officeaddress>
    <cityname>Baywood</cityname>
    <postcodedetails>B03 H7R5</postcodedetails>
  </officeaddress>
</office_address>',GETDATE()),
```

XML Data Select Query:

```
CREATE PROCEDURE GetSupplier_withparameters
(@SupplierID varchar)
AS
BEGIN
SET NOCOUNT ON

SELECT m.SupplierName, m.SupplierEmailAddress, m.SupplierAddress
FROM tbl_Supplier m
WHERE m.SupplierID=@SupplierID
END
```

XML Data Update Query:

```
CREATE PROCEDURE UpdateSupplier_withparameters
(@SupplierID varchar)
AS
BEGIN
SET NOCOUNT ON

UPDATE tbl_Supplier
SET SupplierAddress.modify('insert
(<officeaddress><cityname>Berlin</cityname><postcodedetails>A09
F3X8</postcodedetails></officeaddress>) after (/office_address/officeaddress)[3]' )
WHERE SupplierID =@SupplierID
END
```

Search XML Data Query:

```
CREATE PROCEDURE GetSupplierAndSupplierAddress_withparameters
(@SupplierID varchar)
AS
BEGIN
SET NOCOUNT ON

SELECT m.SupplierName, m.SupplierEmailAddress, m.SupplierAddress,
m.SupplierAddress.value('(/office_address/officeaddress[2]/postcodedetails)[1]', 'varchar(50)')
as postcode
FROM tbl_Supplier m
WHERE m.SupplierID=@SupplierID
```


15

```
END
```

3.4 Triggers

Database triggers are special type of stored procedure that automatically runs when an event occurs in the database server. We created two triggers to help guarantee the integrity of the data captured. Once any of the business rule is violated, the trigger help to ensure that wrong manipulation does not get persisted.

- Trigger to ensure that CustomerID and OrderID Cannot be modified once an order is created

```
GO
CREATE TRIGGER customer_edit_alert  ON [dbo].[tbl_Order]
AFTER UPDATE
AS
IF ( UPDATE (CustomerID) OR UPDATE (OrderID) )
BEGIN
RAISERROR (50009, 16, 10)
END;
GO
```

- Trigger to Ensure All Order with Delivered Status are Fully Paid for.

```
CREATE TRIGGER order_status_update ON [dbo].[tbl_Order]
AFTER UPDATE
AS
BEGIN

declare @OrderDAte date;
declare @OrderAmount float;
declare @OrderStatus varchar(50);
declare @OrderID varchar(100);
declare @PaymentStatus varchar(10);
declare @AgentID varchar(30);
declare @StaffID varchar(30);
declare @CustomerID varchar(30);

select @OrderDate = orderlist.OrderDate from inserted orderlist
select @OrderAmount = orderlist.OrderAmount from inserted orderlist
select @OrderStatus = orderlist.OrderStatus from inserted orderlist
select @OrderID = orderlist.OrderID from inserted orderlist
select @AgentID = orderlist.AgentID from inserted orderlist
select @StaffID = orderlist.StaffID from inserted orderlist
select @CustomerID = orderlist.CustomerID from inserted orderlist
```


16

```
    IF @OrderStatus = 'Delivered' AND @PaymentStatus != 'Paid'
        UPDATE tbl_Order
        SET PaymentStatus = 'Paid'
        WHERE @OrderStatus = 'Delivered' AND  @PaymentStatus != 'Paid'

    END
        GO
```

3.5 Views

```
USE TakeAwayDBCA1
GO

--View All Orders

CREATE VIEW Allorders
AS
SELECT * FROM tbl_Order
GO

--View All Orders Paid

CREATE VIEW Paidorders
AS
SELECT tbl_Order.OrderDate, tbl_Order.OrderID, tbl_Order.OrderStatus, tbl_Order.PaymentStatus
FROM tbl_Order
WHERE tbl_Order.PaymentStatus = 'Paid'
GO

--View All Orders not Delivered

CREATE VIEW NotDeliveredOrders
AS
SELECT tbl_Order.OrderDate, tbl_Order.OrderID, tbl_Order.OrderStatus, tbl_Order.PaymentStatus
FROM tbl_Order
WHERE tbl_Order.OrderStatus != 'Delivered'
GO
```

3.6 JOIN, Group By, HAVING and Function Queries

```
USE TakeAwayDBCA1
GO
```


17

```
-- Function that get Order list by Date Range
CREATE FUNCTION function_Orderlist_by_DateRange (@start as Date,@end as Date)
returns TABLE
AS
return
   SELECT * FROM tbl_Order c
   WHERE c.OrderDate BETWEEN  @start AND @end
GO

-- SELECT Query with JOIN
SELECT tbl_MenuItem.MenuName, tbl_MenuItem.MenuPrice, tbl_MenuItem.CategoryID,
tbl_Category.CategoryName
FROM tbl_MenuItem
INNER JOIN tbl_Category
ON tbl_MenuItem.CategoryId = tbl_Category.CategoryID

-- SELECT Query with Group BY , HAVING and Order BY
SELECT PaymentStatus, AgentID, OrderAmount
FROM tbl_Order
GROUP BY AgentID,PaymentStatus,OrderAmount HAVING PaymentStatus = 'Paid'
ORDER BY OrderAmount
```

4.0 Implementation in SQL Server

18

This chapter discussed the implementations in our SQL Server starting from database server setup, database creation, database schema query writing, creation of data diagram, creating of insertion queries to populate the tables in the database with data and creation of view, triggers and stored procedures.

4.1 Database Setup

My team used Microsoft SQL Server 2019 installed in a Windows server virtual machine provisioned in Azure. We use Microsoft SQL Server Management Studio 18 to interact with the SQL server using valid SQL privileged user for authentication. Once the SQL server and Management studio is fully setup and new user created, we created a database called **TakeAwayDBCA1** and assigned the newly created privileged user ("dbuser") as the owner of the database.

- The create XML schema collection query "TakeAwaySchemaQuery.sql" was created as shown in **Appendix A**.
- The create table query "CreateTableQueriesTakeAway.sql" was written as shown in **Appendix B**.
- The Data Insert query; "InsertIntoTakeAwayTablesQuery.sql" was created as shown in **Appendix C**.
- The queries were executed in the order of the Appendixes to form a fully functional database with enough records to create all necessary business views and stored procedures.

4.2 TakeAway Data Diagram and Customer Journey

A data diagram is a visual representation of the information flow through a process or a system. It helps us better understand process or system operation to discover potential problems, improve efficiency, and develop better processes. Data diagrams range from simple overview to detailed, complex displays of process or system (Chi, 2021).

The user journey of TakeAway system is as follows.

19

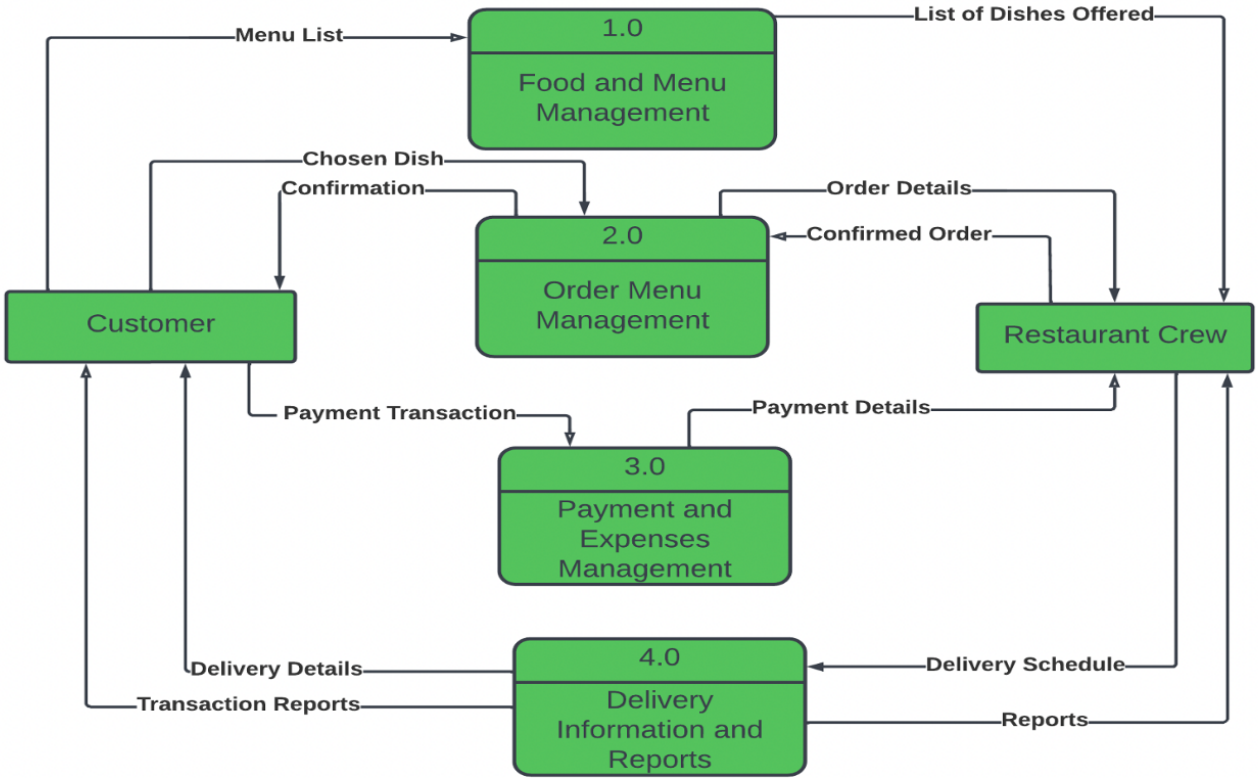


Fig 2: Customer Journey

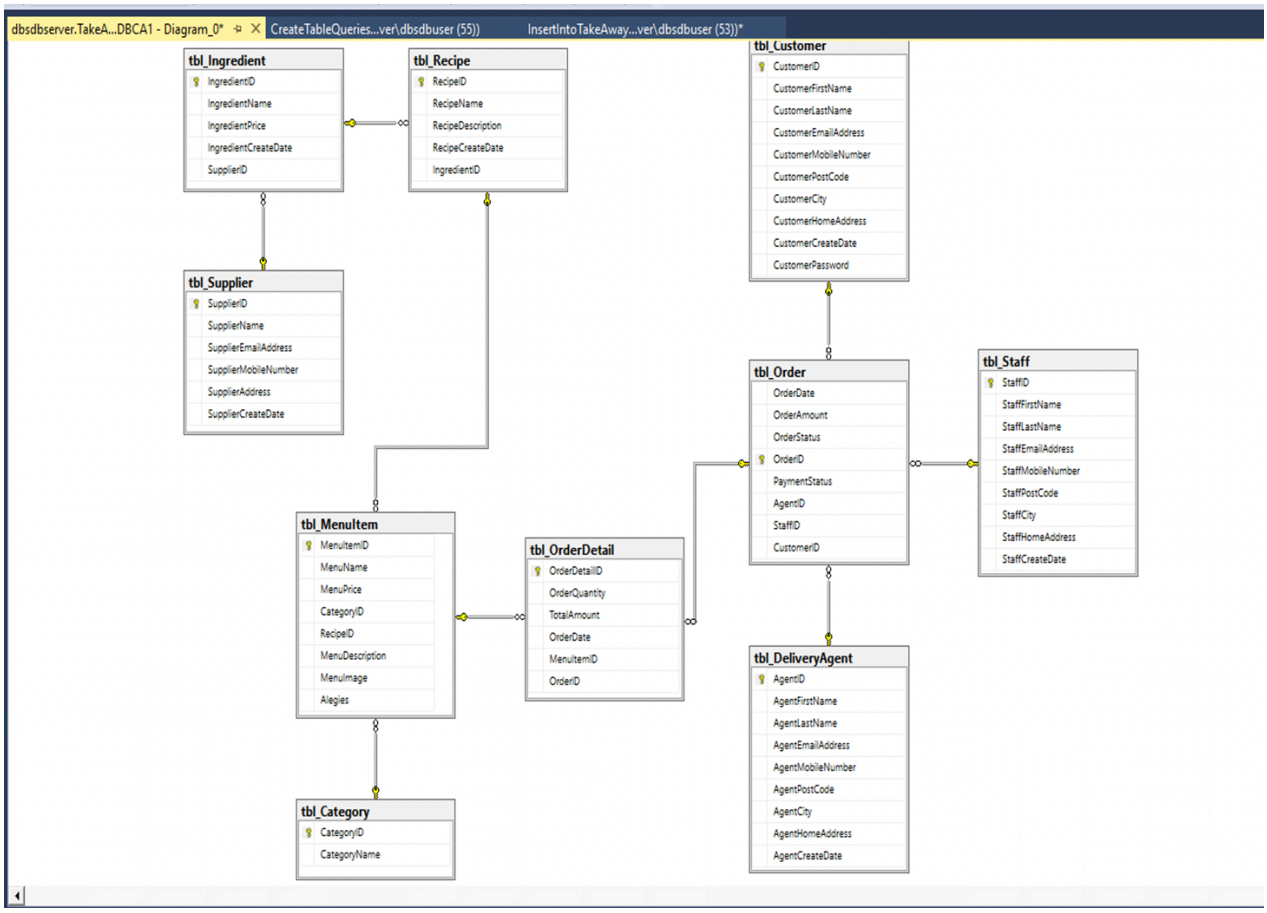


Fig 3: Database Diagram for TakeAway

20

4.3 Referential Integrity Constraints Used

In this project, we introduced many referential integrity constraints on the database tables to ensure that the reliability the data and ease of access of records for reporting. The referential integrity constraints on each table will be described as follows.

4.3.1 Menu Item Table

This table holds the available menu item in the system that customers can order. The menu item must belong to a category defined in the category table.

Constraints:

- All foreign keys cannot be null. This ensure that all menu item created has category and recipe.
- Menu Item image can be null to all items that does not have image to be created.
- Menu Item can have allergies or not.
- Specific character length is allowed for the various attributes in menu item table.
- All menu Item must have a unique ID.

4.3.2 Customer Table

Customer table contains all customers details captured during signup. This table is very important because the system need it for order delivery and reconciliation purposes when needed. The information captured must adhere to the GDPR stipulated processes for customer data acquisition.

Constraints:

- All customers my provide a valid password that will be confirmed at the point of collection.
- Customers must provide valid email and verifiable home address at the point of sign up.
- The creation date of all customers must be recorded.
- No two customer should have the same customer ID. Hence, we had to make the customer ID our primary key.

4.3.3 Category Table

The category table holds all the defined categories for the available menu items in the system. The categories names must be distinct and reflect the common features of a group of menu items.

Constraints:

- All categories must have a unique identifier.

21

- Category names cannot be null

4.3.4 Order Table

Order table contains all orders in the system with details of their delivery and payment status. It also has information about the summation of the cost of all ordered items.

Constraints:

- All orders must be assigned to a customer
- At the point of creation, order may not have been assigned to a staff or delivery agent.
- The date and time the order was made should be recorded.
- An order must have an order amount expected to be paid by the customer.
- Order must have payment status clearly captured.
- All order identifiers must be unique
- Order status cannot be “Delivered” if payment status is not “Paid”

4.3.5 Order Details Table

This table holds the details of a unique orders for menu item placed by customers. All the orders in this table can be traced to a valid registered customer in the system. This order details hold added information about orders in order table.

Constraints:

- All order details must have a unique identifier
- Every order must belong to a registered customer
- Order details must be linked to order in order table.
- The quantity attribute must be set.
- Total amount attribute must be defined.

4.3.6 Ingredients Table

This table hold the list of ingredients available in the system. All ingredients must be traceable to a supplier. These ingredients are the constituents of the recipe for the menu items.

Constraints:

- Ingredients must have unique identifiers.
- All ingredients should be traceable to a registered supplier.
- All ingredients must have a recorded price

22

- We must record the date the ingredients were added to the system.

4.3.7 Recipe Table

The recipe is a documents constituents and process for preparing a menu item. Every menu item must have a recipe. All recipes must have one or many ingredients.

Constraints:

- All recipes must have at least one ingredient
- Recipes must have a unique identifier
- The date a recipe was added to the system must be recorded.
- Recipe in the system must have descriptions.

4.3.8 Supplier Table

All ingredients are supplied by registered suppliers in the system. This table hold all information about suppliers and the information should be updatable by the suppliers in the case of information change.

Constraints:

- Suppliers mu have a verifiable office address. The office address should have the city name and postal code.
- All Supplier identifiers must be unique.
- The date the supplier is created must be recorded.
- Suppliers contact details like phone number, mobile number and email address must be captured.

4.3.9 Staff Table

This table contains the information of all staff in the business. It is the responsibility of this staff to manage the day-to-day operation of the business ranging from cooking, procurement, updating data in the system that has to do with menu item or third-party information.

Constraints:

- Staff must have unique identifier
- The contact details of each staff must be captured ranging from email, phone number, home address.
- The first and last name of each staff must be captured.

23

- Date the staff was created in the system must be recorded.

4.3.10 Delivery Agent Table

This holds the list of all registered delivery agents that delivers orders to customers upon assignment. The home address and contact information of the agent are captured during registration.

Constraints:

- Delivery Agent must have unique identifier
- The contact details of each delivery agent must be captured ranging from email, phone number, home address.
- The first and last name of each delivery agent must be captured.
- The Date each delivery agent was created in the system must be recorded.

4.4 Innovations

We integrated our database to a robust analytic tool that is used by many enterprises to make access to business data easy. Metabase has a visual query builder that help free our business data from the confines of SQL, letting everyone that have access to the metabase to query in a user interface (UI) humans love. The tool allows users to easily create interactive dashboard from terabyte-scale analytic workloads to day-to-day operational workflows with over 15 visualisation types. It also empowers our system users to be able to craft metadata-rich, semantic models which let people query the database on their own while keeping things consistent and avoid repetitions (Abraham, 2020).

Metabase Report Sample 1:

24

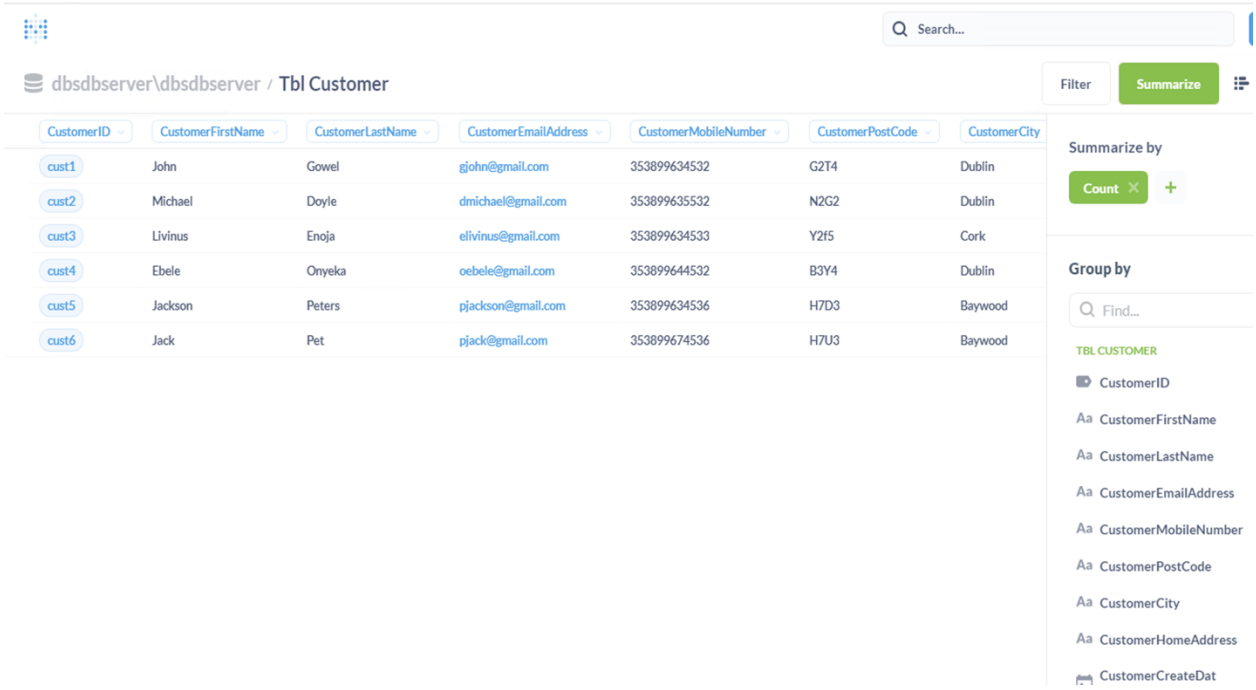


Fig 4: Customer Analytics Interface

Metabase Report Sample 2:

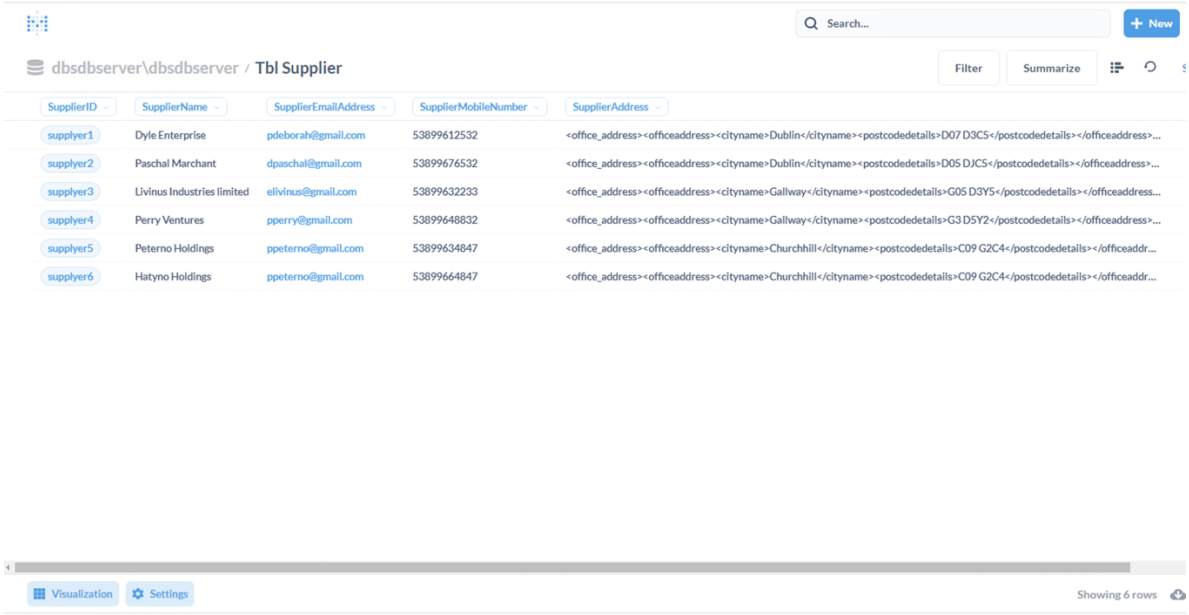


Fig 5: Supplier Analytics Interface

25

Metabase Report Sample 3:

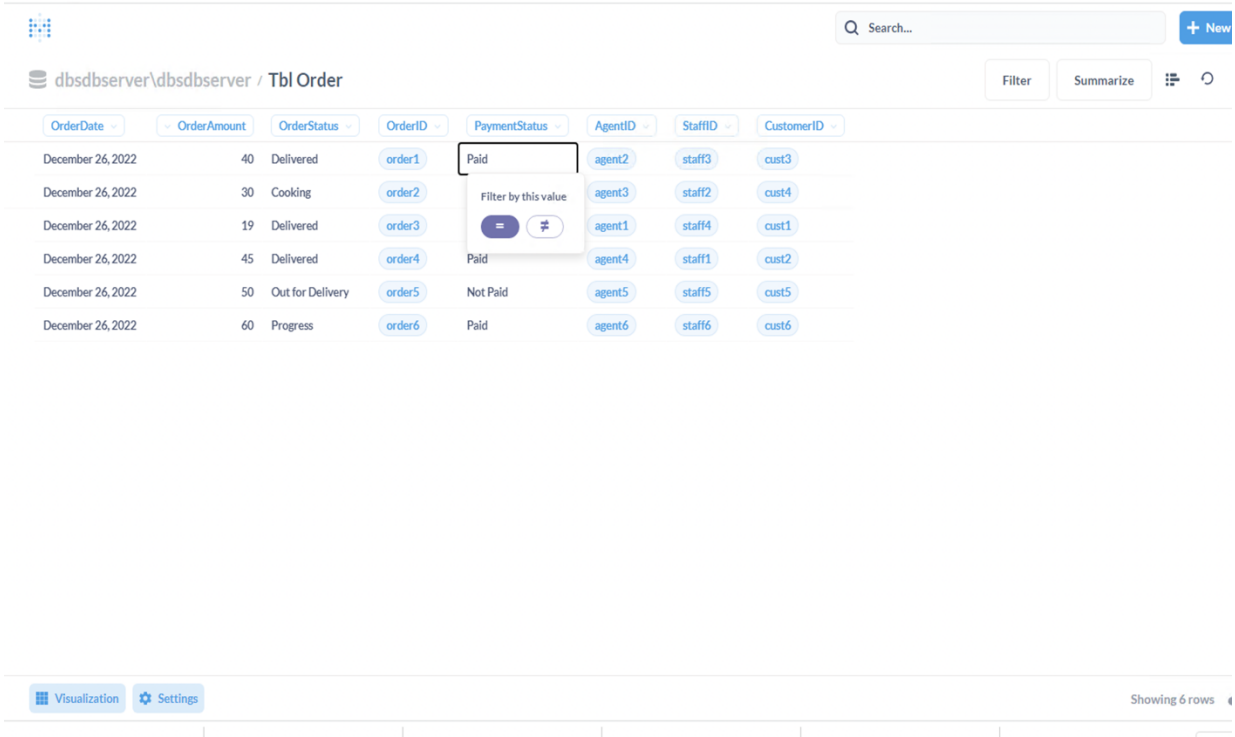


Fig 6: Order Analytics Interface

Metabase Report Sample 4:

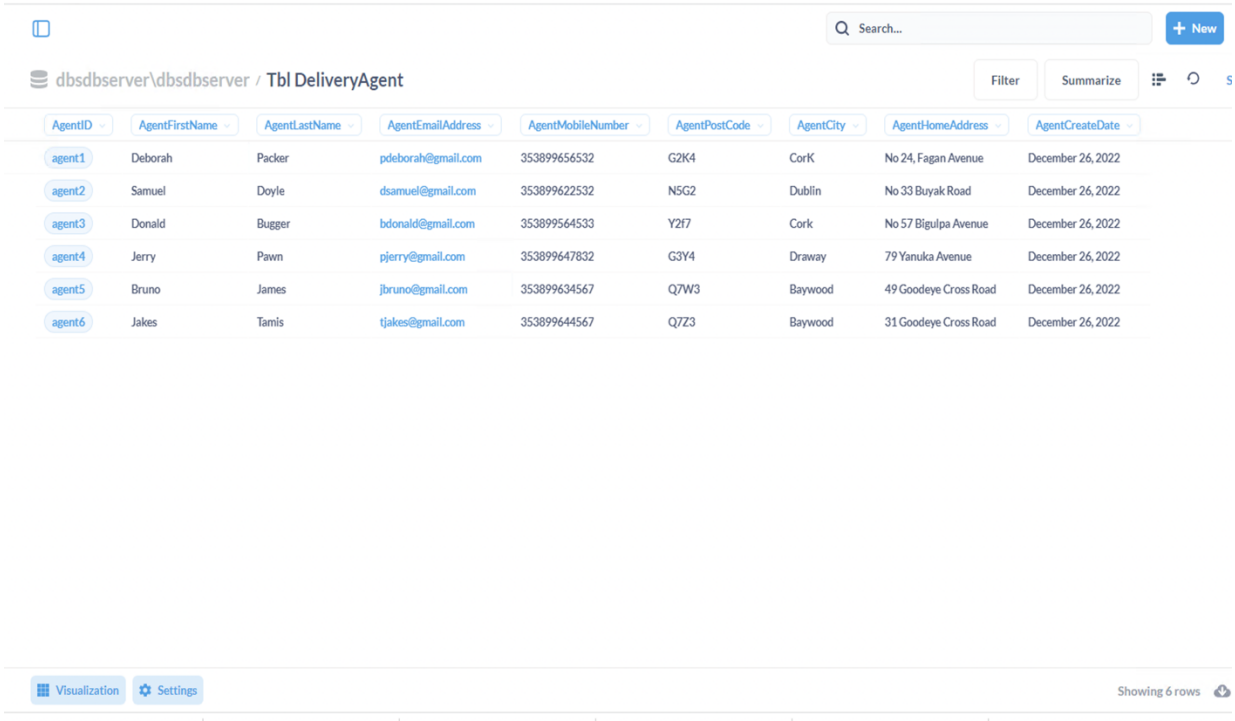


Fig 7: Delivery Agent Analytics Interface

26

5.0 Conclusions

In the implementation of this project, my team have learned a lot of enterprise skills in database design concepts, SQL query writing, hybrid database design, schema normalization, and data analytics. We learned that Database Design as a collection of processes that facilitate the designing, development, implementation and maintenance of enterprise data management systems is an important part of business system development that should be given utmost attention in the beginning of the project. Properly designed databases makes it easy to maintain, it improves data consistency and makes our product more cost effective in terms of disk storage space needed to store the data. It is responsibility of the database designer to decide how the data elements correlate and what data must be stored during product design (Peterson, 2022).

If given more time, we would like to fully implement the web portal and make it available for customers. We will consider continuing work on it to make the product full functional and hosted.

6.0 References

Wales, J. (n.d.). *https://en.wikipedia.org*. From Wikipedia, the free encyclopedia: https://en.wikipedia.org/wiki/Business_rule

Ricardo, C. M. (2003). https://www.sciencedirect.com. *Encyclopedia of Information Systems*, Pages 279-297. From Science Direct: https://www.sciencedirect.com/topics/computer-science/relational-schema

Saxena, S. (2021, 08 06). *https://www.geeksforgeeks.org*. From Geeks for Geeks: https://www.geeksforgeeks.org/relation-schema-in-dbms/

Adrienne Watt and Watt, A. (2014). *Database Design.* Pressbooks, BCcampus.

Chi, C. (2021, 5 13). *A Beginner's Guide to Data Flow Diagrams*. From Hub Spot: https://blog.hubspot.com/marketing/data-flow-diagram

R. Ramesha, S. V. (2021). Emperical Study of Online Food Delivery Services From Application Perspective. *Elsevier*, 1.

Abraham, T. (2020). *Metabase Up and Running.* Packt Publishing.

Peterson, R. (2022, 12 3). *database-design*. From https://www.guru99.com: https://www.guru99.com/database-design.html

CISA, R. (2021). *ISO 27001 Controls and Objectives.* Academia.edu.

Appendix A: XML Schema Collection Query

```
USE TakeAwayDBCA1
GO
CREATE XML SCHEMA COLLECTION SupplierAddressCollection
AS
'<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <xsd:element name="office_address">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element ref="officeaddress" minOccurs="1" maxOccurs="unbounded" />
                </xsd:sequence>
            </xsd:complexType>
    </xsd:element>
    <xsd:element name="officeaddress">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="cityname" type="xsd:string" />
                    <xsd:element name="postcodedetails" type="xsd:string" />
                </xsd:sequence>
            </xsd:complexType>
    </xsd:element>
</xsd:schema>'
```

Appendix B: Create Table Query

```
USE TakeAwayDBCA1
GO

CREATE TABLE tbl_Category
(
CategoryID   varchar(20) not null,
CategoryName varchar(20) not null,
PRIMARY KEY (CategoryID),
)

GO
CREATE TABLE tbl_Supplier
(
SupplierID varchar(30) not null,
SupplierName varchar(50) not null,
SupplierEmailAddress varchar(50) not null,
SupplierMobileNumber varchar(14) not null,
SupplierAddress  xml (SupplierAddressCollection),
SupplierCreateDate  date not null,
PRIMARY KEY (SupplierID),
)
GO

CREATE TABLE tbl_Ingredient
(
IngredientID varchar(30) not null,
IngredientName varchar(20) not null,
IngredientPrice float not null,
IngredientCreateDate  date not null,
SupplierID varchar(30) not null,
```


28

```
PRIMARY KEY (IngredientID),
FOREIGN KEY (SupplierID) REFERENCES tbl_Supplier (SupplierID),
)
GO
CREATE TABLE tbl_Recipe
(
RecipeID   varchar(20) not null,
RecipeName varchar(50) not null,
RecipeDescription varchar(100) not null,
RecipeCreateDate  date not null,
IngredientID varchar (30),
PRIMARY KEY (RecipeID),
FOREIGN KEY (IngredientID) REFERENCES tbl_Ingredient (IngredientID),
)

GO

CREATE TABLE tbl_MenuItem
(
MenuItemID    varchar(30) not null,
MenuName      varchar(50) not null,
MenuPrice     float  not null,
CategoryID    varchar(20) not null,
RecipeID      varchar(20) not null,
MenuDescription      varchar(200) not null,
MenuImage     image,
Alegies       varchar(50),
PRIMARY KEY   (MenuItemID),
FOREIGN KEY   (CategoryID)  REFERENCES    tbl_Category(CategoryID),
FOREIGN KEY   (RecipeID)    REFERENCES    tbl_Recipe(RecipeID),
)
Go

CREATE TABLE tbl_Customer
(
CustomerID    varchar(30) not null,
CustomerFirstName varchar(20) not null,
CustomerLastName varchar(20) not null,
CustomerEmailAddress varchar(50) not null,
CustomerMobileNumber varchar(14) not null,
CustomerPostCode varchar(10) not null,
CustomerCity varchar(10) not null,
CustomerHomeAddress varchar(100) not null,
CustomerCreateDate  date not null,
CustomerPassword varchar(100) not null
PRIMARY KEY (CustomerID),
)
GO

CREATE TABLE tbl_DeliveryAgent
(
AgentID       varchar(30) not null,
AgentFirstName varchar(20) not null,
AgentLastName varchar(20) not null,
AgentEmailAddress varchar(50) not null,
AgentMobileNumber varchar(14) not null,
AgentPostCode varchar(10) not null,
AgentCity varchar(10) not null,
AgentHomeAddress varchar(100) not null,
AgentCreateDate  date not null
PRIMARY KEY (AgentID),
)
GO
```


29

```
CREATE TABLE tbl_Staff
(
StaffID       varchar(30) not null,
StaffFirstName varchar(20) not null,
StaffLastName varchar(20) not null,
StaffEmailAddress varchar(50) not null,
StaffMobileNumber varchar(14) not null,
StaffPostCode varchar(10) not null,
StaffCity varchar(10) not null,
StaffHomeAddress varchar(100) not null,
StaffCreateDate  date not null
PRIMARY KEY (StaffID),
)
GO

CREATE TABLE tbl_Order
(
OrderDate date not null,
OrderAmount float not null,
OrderStatus varchar(50) not null,
OrderID varchar(100) not null,
PaymentStatus varchar(10) not null,
AgentID       varchar(30),
StaffID       varchar(30),
CustomerID    varchar(30) not null,
PRIMARY KEY (OrderID),
FOREIGN KEY (AgentID) REFERENCES tbl_DeliveryAgent (AgentID),
FOREIGN KEY (StaffID) REFERENCES tbl_Staff (StaffID),
FOREIGN KEY (CustomerID) REFERENCES tbl_Customer (CustomerID),
)

GO
CREATE TABLE tbl_OrderDetail
(
OrderDetailID varchar(30) not null,
OrderQuantity int not null,
TotalAmount float not null,
OrderDate date not null,
MenuItemID    varchar(30) not null,
OrderID       varchar(100) not null,
PRIMARY KEY (OrderDetailID),
FOREIGN KEY (MenuItemID) REFERENCES tbl_MenuItem (MenuItemID),
FOREIGN KEY (OrderID) REFERENCES tbl_Order (OrderID),
)
```

Appendix C: Insert Data to Tables Query

```
USE TakeAwayDBCA1
GO
INSERT INTO tbl_Category VALUES
('cat1','Drinks'),
('cat2', 'Vegetarian'),
('cat3', 'Noodles'),
('cat4', 'Starter'),
('cat5','Desserts'),
('cat6','African')

GO
INSERT INTO tbl_Supplier VALUES
('supplyer1','Dyle Enterprise','pdeborah@gmail.com','53899612532','<?xml version="1.0"?>
<office_address>
```


30

```
    <officeaddress>
        <cityname>Dublin</cityname>
        <postcodedetails>D07 D3C5</postcodedetails>
    </officeaddress>
    <officeaddress>
        <cityname>Cork</cityname>
        <postcodedetails>C11 G7C5</postcodedetails>
    </officeaddress>
    <officeaddress>
        <cityname>Baywood</cityname>
        <postcodedetails>B03 H7R5</postcodedetails>
    </officeaddress>
</office_address>',GETDATE()),
('supplyer2', 'Paschal Marchant','dpaschal@gmail.com','53899676532','<?xml version="1.0"?>
<office_address>
    <officeaddress>
        <cityname>Dublin</cityname>
        <postcodedetails>D05 DJC5</postcodedetails>
    </officeaddress>
    <officeaddress>
        <cityname>Gallway</cityname>
        <postcodedetails>G11 G7T5</postcodedetails>
    </officeaddress>
</office_address>',GETDATE()),
('supplyer3', 'Livinus Industries limited','elivinus@gmail.com','53899632233','<?xml
version="1.0"?>
<office_address>
    <officeaddress>
        <cityname>Gallway</cityname>
        <postcodedetails>G05 D3Y5</postcodedetails>
    </officeaddress>
    <officeaddress>
        <cityname>Cork</cityname>
        <postcodedetails>C05 G8C5</postcodedetails>
    </officeaddress>
    <officeaddress>
        <cityname>Baywood</cityname>
        <postcodedetails>B12 A7R5</postcodedetails>
    </officeaddress>
</office_address>',GETDATE()),
('supplyer4', 'Perry Ventures','pperry@gmail.com','53899648832','<?xml version="1.0"?>
<office_address>
    <officeaddress>
        <cityname>Gallway</cityname>
        <postcodedetails>G3 D5Y2</postcodedetails>
    </officeaddress>
    <officeaddress>
        <cityname>Cork</cityname>
        <postcodedetails>C15 G8C4</postcodedetails>
    </officeaddress>
    <officeaddress>
        <cityname>Baywood</cityname>
        <postcodedetails>B12 A7S5</postcodedetails>
    </officeaddress>
       <officeaddress>
        <cityname>Gitall</cityname>
        <postcodedetails>GI12 A7R5</postcodedetails>
    </officeaddress>
</office_address>',GETDATE()),
('supplyer5','Peterno Holdings','ppeterno@gmail.com','53899634847','<?xml version="1.0"?>
<office_address>
    <officeaddress>
        <cityname>Churchhill</cityname>
        <postcodedetails>C09 G2C4</postcodedetails>
```


31

```
    </officeaddress>
    <officeaddress>
        <cityname>Baywood</cityname>
        <postcodedetails>B03 A7X5</postcodedetails>
    </officeaddress>
       <officeaddress>
        <cityname>Gitall</cityname>
        <postcodedetails>GI01 Q7P5</postcodedetails>
    </officeaddress>
</office_address>',GETDATE()),
('supplyer6','Hatyno Holdings','ppeterno@gmail.com','53899664847','<?xml version="1.0"?>
<office_address>
    <officeaddress>
        <cityname>Churchhill</cityname>
        <postcodedetails>C09 G2C4</postcodedetails>
    </officeaddress>
    <officeaddress>
        <cityname>Baywood</cityname>
        <postcodedetails>B03 A7X5</postcodedetails>
    </officeaddress>
       <officeaddress>
        <cityname>Gitall</cityname>
        <postcodedetails>GI01 Q7P5</postcodedetails>
    </officeaddress>
</office_address>',GETDATE())

GO
INSERT INTO tbl_Ingredient VALUES
('ing1','Flower',15.0,GETDATE(),'supplyer3'),
('ing2','Vegetable Oil',25.0,GETDATE(),'supplyer2'),
('ing3','Frozen Beef',34.0,GETDATE(),'supplyer1'),
('ing4','Food Seasoning',20.5,GETDATE(),'supplyer5'),
('ing5','Salt',45.8,GETDATE(),'supplyer4'),
('ing6','Hot Pepper',35.0,GETDATE(),'supplyer2')

GO
INSERT INTO tbl_Recipe VALUES
('recipe1','Flower Recipe','This recipe is for Flower',GETDATE(),'ing1'),
('recipe2','Vegetable Oil Recipe','This recipe is for Vegitable Oil',GETDATE(),'ing2'),
('recipe3','Frozen Beef Recipe','This recipe is for Frozen Beef',GETDATE(),'ing3'),
('recipe4','Food Seasoning Recipe','This recipe is for Seasoning',GETDATE(),'ing4'),
('recipe5','Table Salt Recipe','This recipe is for table salt',GETDATE(),'ing5'),
('recipe6','Hot Pepper Recipe','This recipe is for Hot Pepper',GETDATE(),'ing6')

GO
insert into tbl_MenuItem
(MenuItemID,MenuName,MenuPrice,CategoryID,RecipeID,MenuDescription,MenuImage,Alegies)
SELECT 'menu1','Spring Rolls',5,'cat4','recipe1','Spring roll is a traditional Chinese savory
snack where a pastry sheet is filled with vegetables, rolled & fried.',BulkColumn,''
FROM Openrowset( Bulk 'C:\imgs\springrolls.png', Single_Blob) as img

insert into tbl_MenuItem
(MenuItemID,MenuName,MenuPrice,CategoryID,RecipeID,MenuDescription,MenuImage,Alegies)
SELECT 'menu2', 'Roast Pork with Cracle',7,'cat5','recipe2','Transfer pork to a roasting dish
and roast for 50 minutes, or until the rind crackles.',BulkColumn,''
FROM Openrowset( Bulk 'C:\imgs\roast-pork-with-cracle.png', Single_Blob) as img

insert into tbl_MenuItem
(MenuItemID,MenuName,MenuPrice,CategoryID,RecipeID,MenuDescription,MenuImage,Alegies)
SELECT 'menu3', 'Coca Cola',2,'cat1','recipe3','Coca-Cola, or Coke, is a carbonated soft drink
manufactured by the Coca-Cola Company.',BulkColumn,''
FROM Openrowset( Bulk 'C:\imgs\coca-cola.png', Single_Blob) as img
```


32

```
insert into tbl_MenuItem
(MenuItemID,MenuName,MenuPrice,CategoryID,RecipeID,MenuDescription,MenuImage,Alegies)
SELECT 'menu4', 'Crispy Chilli Chicken Udon',10,'cat2','recipe4','Chilli Chicken Udon Noodle
Soup by Chef Kwanghi Chan.',BulkColumn,''
FROM Openrowset( Bulk 'C:\imgs\crispy-chilli-chicken-udon.png', Single_Blob) as img

insert into tbl_MenuItem
(MenuItemID,MenuName,MenuPrice,CategoryID,RecipeID,MenuDescription,MenuImage,Alegies)
SELECT 'menu5','Teriyaki Sauce Japanese Style',9,'cat3','recipe5','Teriyaki is a Japanese
cooking technique used to cook and glaze proteins with a savory-sweet sauce.',BulkColumn,''
FROM Openrowset( Bulk 'C:\imgs\teriyaki-sauce-japanese-style.png', Single_Blob) as img

insert into tbl_MenuItem
(MenuItemID,MenuName,MenuPrice,CategoryID,RecipeID,MenuDescription,MenuImage,Alegies)
SELECT 'menu6','Pepper Chicken Recipe',9,'cat4','recipe6','Pepper Chiken is a Japanese cooking
technique used to cook and glaze proteins with a savory-sweet sauce.',BulkColumn,''
FROM Openrowset( Bulk 'C:\imgs\pepper-chiken-recipe.png', Single_Blob) as img

GO
INSERT INTO tbl_Customer VALUES
('cust1','John','Gowel','gjohn@gmail.com','353899634532','G2T4','Dublin','No 23, Shagan
Avenue',GETDATE(),'P@ssJ0hn#'),
('cust2', 'Michael','Doyle','dmichael@gmail.com','353899635532','N2G2','Dublin','No 33 Tayak
Road',GETDATE(),'P@ss@D0ylE#'),
('cust3', 'Livinus','Enoja','elivinus@gmail.com','353899634533','Y2f5','Cork','No 67 Bijock
Avenue',GETDATE(),'en0j@P@ss#'),
('cust4', 'Ebele','Onyeka','oebele@gmail.com','353899644532','B3Y4','Dublin','89 Asuka
Avenue',GETDATE(),'0ny3k@PAss#'),
('cust5','Jackson','Peters','pjackson@gmail.com','353899634536','H7D3','Baywood','44 Badeye
Cross Road',GETDATE(),'P3t3rzP@ss#'),
('cust6','Jack','Pet','pjack@gmail.com','353899674536','H7U3','Baywood','42 Badeye Cross
Road',GETDATE(),'P3t3rzP@ss#')
GO
INSERT INTO tbl_DeliveryAgent VALUES
('agent1','Deborah','Packer','pdeborah@gmail.com','353899656532','G2K4','CorK','No 24, Fagan
Avenue',GETDATE()),
('agent2', 'Samuel','Doyle','dsamuel@gmail.com','353899622532','N5G2','Dublin','No 33 Buyak
Road',GETDATE()),
('agent3', 'Donald','Bugger','bdonald@gmail.com','353899564533','Y2f7','Cork','No 57 Bigulpa
Avenue',GETDATE()),
('agent4', 'Jerry','Pawn','pjerry@gmail.com','353899647832','G3Y4','Draway','79 Yanuka
Avenue',GETDATE()),
('agent5','Bruno','James','jbruno@gmail.com','353899634567','Q7W3','Baywood','49 Goodeye Cross
Road',GETDATE()),
('agent6','Jakes','Tamis','tjakes@gmail.com','353899644567','Q7Z3','Baywood','31 Goodeye Cross
Road',GETDATE())

GO
INSERT INTO tbl_Staff VALUES
('staff1','Dyle','Jacob','pdeborah@gmail.com','353899612532','Y2K4','CorK','No 24, Hagan
Avenue',GETDATE()),
('staff2', 'Paschal','Doyle','dpaschal@gmail.com','353899676532','N5G2','Dublin','No 33 Beer
Road',GETDATE()),
('staff3', 'Livinus','Enoja','elivinus@gmail.com','353899632233','Y2f5','Cork','No 67 Jock
Avenue',GETDATE()),
('staff4', 'Perry','Poter','pperry@gmail.com','353899648832','G3Y4','Draway','79 Sanuka
Avenue',GETDATE()),
('staff5','Peterno','Pecky','ppeterno@gmail.com','353899634847','Q7W3','Baywood','49 Codeye
Cross Road',GETDATE()),
('staff6','Yerno','Azicky','ayerno@gmail.com','353899234847','Q7V3','Baywood','4 Codeye Cross
Road',GETDATE())

GO
INSERT INTO tbl_Order VALUES
```


33

```
(GETDATE(),40.0,'Delivered','order1','Paid','agent2','staff3','cust3'),
(GETDATE(),30.0,'Progress','order2','Paid','agent3','staff2','cust4'),
(GETDATE(),19.0,'Delivered','order3','Not Paid','agent1','staff4','cust1'),
(GETDATE(),45.0,'Delivered','order4','Paid','agent4','staff1','cust2'),
(GETDATE(),50.0,'Progress','order5','Not Paid','agent5','staff5','cust5'),
(GETDATE(),60.0,'Progress','order6','Paid','agent6','staff6','cust6')


GO
INSERT INTO tbl_OrderDetail VALUES
('orderd1',5,35.0,GETDATE(),'menu2','order1'),
('orderd2',5,25.0,GETDATE(),'menu1','order2'),
('orderd3',7,14.0,GETDATE(),'menu3','order3'),
('orderd4',4,40.0,GETDATE(),'menu5','order4'),
('orderd5',5,45.0,GETDATE(),'menu4','order5'),
('orderd6',3,35.0,GETDATE(),'menu6','order6')
```


34