

**DATABASES**

October 2020 -- February 2021

**SCHOOL OF INFOCOMM TECHNOLOGY**

Diploma in Information Technology

Diploma in Financial Informatics

Diploma in Cyber Security & Forensic

**(50 % of DB Module)**

**18 Jan 2021 – 29 Jan 2021 (2 Weeks)**

|  |  |  |
| --- | --- | --- |
| Deliverables | Deadline | Submission |
| Softcopy | Friday 29 Jan 2021, 23:59 pm | MEL Discussion Board |
| Hardcopy | Monday 01 Feb 2021, 10:00 am | Level 8 (Admin Office),  Pigeon-holes (Databases) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DB Class Group | | | P08 | |
| Team Number | | | 3 | Team Grade: |
| Tutor | | | Mr Arthur Tan | |
| Members | Student No. | Student Name | | Individual Grade |
| S10204829 | Neo Rei Siang Edward | |  |
| S10202940 | Yang Molei | |  |
| S10205166 | Chiang Jun Han | |  |
| S10194040 | Evan Ong Shui Fu | |  |
| S10204529 | Sng Wei Liang Jeremy | |  |

**Penalty for late submission:**

10% of the marks will be deducted for everyday (inclusive of Saturdays, Sundays, and Public Holidays) after the deadline.

**NO** submission will be accepted after 05 Feb 2021, 23:59 pm.

Table of Contents

[**SCHOOL OF INFOCOMM TECHNOLOGY** 1](#_Toc62855705)

[**1.** **Changes Made** 3](#_Toc62855706)

[**2.** **Additional Assumptions Made** 3](#_Toc62855707)

[**3.** **Mapping** 3](#_Toc62855708)

[**4.** **Data Dictionary** 5](#_Toc62855709)

[**5.** **SQL Creation of Database & Tables** 18](#_Toc62855710)

[**6.** **Insertion of Values** 28](#_Toc62855711)

[**7.** **Display all tables** 47](#_Toc62855712)

[**8.** **SQL Queries** 59](#_Toc62855713)

[Edward’s Queries: 59](#_Toc62855714)

[Molei’s Queries: 60](#_Toc62855715)

[Jeremy’s Queries: 61](#_Toc62855716)

[Jun Han’s Queries: 63](#_Toc62855717)

[Evan’s Queries: 64](#_Toc62855718)

# **Changes Made**

There were some changes made to the ER-Diagram.

One relationship that is added is **Enjoys (<orderID>, <promoID>),** to signify the M-N relationship between the custOrder entity and Promotion entity. Another relationship that is added is **Runs (<promoID>, <outletID>, maxCount),** to signify the M-N relationship between the Promotion entity and Outlet entity. The next relationship added is **Belongs (<outletID>, <cuisineID>),** to signify the M-N relationship between the Outlet entity and Cuisine entity. Moreover, the next relationship added is **Has (<outletID, menuNo>, <itemID>),** to signify the M-N relationship between the weak entity, Menu, and item entity. Another relationship added is **Purchase (<riderID, purchaseDateTime>, <equipID>, purchaseQty),** to signify the M-N relationship between the Rider entity and Equipment entity. Additionally, another relationship added is **Achieve (<riderID, winDate>, <awardID>),** to signify the M-N relationship between the Rider entity and Award entity. Another relationship added is **Assign (<orderID>, <riderID>, status),** to signify the M-N relationship between Order entity and Rider entity. Lastly, the last relationship that is added is **Contact (<outletID>, ContactNo),** to signify the multi-value attribute of contactNo in the Outlet entity.

1. **Additional Assumptions Made**

Delivery Fee is standardised for all outlets and for each business.

# **Mapping**

Customer (custID, custName, custAddress, custContact, custEmail)

Voucher (voucherID, voucherStatus, voucherDesc, startDate, expiryDate, minOrder, dollarValue, <custID>)

CustOrder (orderID, orderStatus, orderDateTime, <voucherID>, <custID>, <outletID>)

Payment (pmtID, pmtMode, pmtType, pmtAmt, <orderID>)

Pickup (<orderID>, pickupRefNo, pickupDateTime)

Delivery(<orderID>, deliveryAddress, deliveryDateTime, <riderID>)

Rider (riderID, riderNRIC, riderName, riderContact, riderAddress, riderDOB, deliveryMode, <teamID>)

Equipment (equipID, equipName, equipPrice, <setID>)

Award (awardID, awardName, awardType)

Team (teamID, teamName, <awardID>, <leaderID>)

OrderItem (<orderID>, <itemID>, orderQty, unitPrice)

Item (itemID, itemName, itemDesc, itemPrice)

Menu (<OutletID, menuNo>, menuName)

Outlet (outletID, outletName, outletAddress, deliveryFee, openTime, closeTime, startDeliveryTime, endDeliveryTime, <bizID>, <zoneID>)

OutletContact (<outletID>, ContactNo)

Cuisine (cuisineID, cuisineName)

Promotion (promoID, promoName, promoDesc, percentDiscount, isFreeDelivery)

Zone (zoneID, zoneName)

Business (bizID, bizName)

Enjoys (<orderID>, <promoID>)

Runs (<promoID>, <outletID>, maxCount)

Belongs (<outletID>, <cuisineID>)

Has (<outletID, menuNo>, <itemID>)

Purchase (<riderID, purchaseDateTime>, <equipID>, purchaseQty)

Achieve (<riderID, winDate>, <awardID>)

Assign (<orderID>, <riderID>, status)

# **Data Dictionary**

**Customer**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Data Type | Constraint | Null value |
| custID | Unique number to identify customer | char(8) | Primary key | No |
| custName | Name of customer | varchar(50) |  | No |
| custAddress | Physical Address of customer | varchar(150) |  | No |
| custContact | Contact number of customer | char(8) | All digits  First digit starts with ‘8’ or ‘9’ | No |
| custEmail | Email Address of customer | varchar(50) |  | Yes |

**Voucher**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Data Type | Constraint | Null value |
| voucherID | Unique number to identify voucher | char(10) | Primary key | No |
| voucherStatus | Status of voucher | varchar(10) |  | Yes |
| voucherDesc | Details of the voucher code | varchar(50) |  | Yes |
| startDate | Starting date of the voucher's use | date |  | No |
| expiryDate | Expiry date of the voucher's | date | Must not be earlier than startDate | No |
| minOrder | Minimum order value of customer order for voucher's use | smallmoney |  | No |
| dollarValue | Value of the voucher | smallmoney |  | No |
| custID | Unique number to identify customer | char(8) | Foreign Key  → Customer  (custID) | Yes |

**CustOrder**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Data Type | Constraint | Null value |
| orderID | Unique number to identify customer's order | char(10) | Primary key | No |
| orderStatus | Status of customer's order | varchar(10) | ‘Completed’, ‘Preparing’, ‘Accepted’, ‘Dispatched’, ‘Cancelled’ | No |
| orderDateTime | Date of order | smalldatetime |  | No |
| voucherID | Unique number to identify voucher | char(10) | Foreign key  → Voucher  (voucherID) | Yes |
| custID | Unique number to identify customer | char(8) | Foreign key  → Customer  (custID) | Yes |
| outletID | Unique number to identify outlet | char(10) | Foreign key  → Outlet  (outletID) | Yes |

**Payment**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Data Type | Constraint | Null value |
| pmtID | Unique number to identify payment | char(10) | Primary key | No |
| pmtMode | Payment mode of customer's other | varchar(20) | ‘NETS’, ‘Credit Card’, ‘Internet Banking’, ‘VISA’, ‘Electronic Wallet’ | No |
| pmtType | The mode of payment. E.g. NETS, Credit Card, Electronic Wallet. | varchar(20) | ‘Order Payment’, ‘Refund’ | No |
| pmtAmt | Total price to be paid. | smallmoney |  | No |
| orderID | Unique number to identify customer's order | char(10) | Primary key  Foreign key  → CustOrder  (orderID) | Yes |

**Pickup**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Data Type | Constraint | Null value |
| orderID | Unique number to identify customer's order | char(10) | Primary key  Foreign key  → CustOrder  (orderID) | No |
| pickupRefNo | Reference number for customer's order | varchar(10) |  | Yes |
| pickupDateTime | Date/Time when customer receives order | smalldatetime |  | Yes |

**Delivery**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Data Type | Constraint | Null value |
| orderID | Unique number to identify customer's order | char(10) | Primary key  Foreign key  → CustOrder  (orderID) | No |
| deliveryAddress | Address to deliver order to | varchar(50) |  | No |
| deliveryDateTime | Date/Time that an order was delivered to a customer. | smalldatetime |  | No |
| riderID | Unique number to identify rider | char(10) | Foreign Key  → Rider  (riderID) | No |

**Rider**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Data Type | Constraint | Null value |
| riderID | Unique number to identify rider | char(10) | Primary key | No |
| riderNRIC | NRIC of rider | char(9) | First and last character must be alphabet  First character must be either ‘S’, ‘T’, ‘F’, ‘G’ | No |
| riderName | Name of Rider | varchar(50) |  | No |
| riderContact | Contact number of rider's phone | char(8) | All digits  First digit starts with ‘8’ or ‘9’ | No |
| riderAddress | Physical Address of rider | varchar(50) |  | No |
| riderDOB | Date of Birth of rider | date |  | Yes |
| deliveryMode | Mode of transport rider is using | varchar(20) |  | No |
| teamID | Unique number to identify rider's team | char(4) | Foreign key  →Team  (teamID) | Yes |

**Team**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Data Type | Constraint | Null value |
| teamID | Unique number to identify team | char(4) | Primary key | No |
| teamName | Name of team | varchar(50) |  | No |
| awardID | Unique number to identify award. | char(10) | Foreign key  → Award  (awardID) | Yes |
| leaderID | Unique number to identify rider who is team leader | char(10) | Foreign key  → Rider  (riderID) | Yes |

**Award**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Data Type | Constraint | Null value |
| awardID | Unique number to identify award. | char(10) | Primary key | No |
| awardName | Name of award | varchar(50) |  | No |
| awardType | Type of award | varchar(50) | ‘Individual’, ‘Team’ | No |

**Equipment**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Data Type | Constraint | Null value |
| equipID | Unique number to identity equipment | char(10) | Primary key | No |
| equipName | Name of equipment | varchar(50) |  | No |
| equipPrice | Price of equipment | smallmoney |  | Yes |
| setID | Unique number to identify set | char(10) | Foreign key  → Equipment  (equipID) | Yes |

**Outlet**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Data Type | Constraint | Null value |
| outletID | Unique number to identity outlet | char(10) | Primary key | No |
| outletName | Name of outlet | varchar(50) |  | No |
| outletAddress | Address of outlet | varchar(50) |  | No |
| deliveryFee | Delivery fee of order | smallmoney |  | No |
| openTime | Time when outlet opens | time(0) |  | No |
| closeTime | Time when outlet closes | time(0) | Must not be earlier than openTime | No |
| startDeliveryTime | Start time of delivery | time(0) | Must not be earlier than openTime | Yes |
| endDeliveryTIme | End time of delivery | time(0) | Must not be earlier than startDeliveryTime | Yes |
| bizID | Unique number to identify each Business | char(4) | Foreign key  →Business  (bizID) | Yes |
| zoneID | Unique number to identify outlet zone | char(4) | Foreign key  → Zone  (zoneID) | Yes |

**Business**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Data Type | Constraint | Null value |
| bizID | Unique number to identify business. | char(4) | Primary key | No |
| bizName | Name of Business | varchar(50) |  | No |

**Zone**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Data Type | Constraint | Null value |
| zoneID | Unique number to identify outlet's zone | char(4) | Primary key | No |
| zoneName | Name of zone | varchar(20) |  | No |

**Cuisine**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Data Type | Constraint | Null value |
| cuisineID | Unique number to identify cuisine | char(10) | Primary key | No |
| cuisineName | Name of cuisine | varchar(50) |  | No |

**Menu**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Data Type | Constraint | Null value |
| menuNo | Unique number to identify meny | char(10) | Primary key | No |
| outletID | Unique number to identify outlet | char(10) | Primary key  Foreign key  →Outlet  (outletID) | No |
| menuName | Name of menu | varchar(100) |  | No |

**Item**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Data Type | Constraint | Null value |
| itemID | Unique number to identify item | tinyint | Primary key | No |
| itemName | Name of food item | varchar(50) |  | No |
| itemDesc | Description of food item | varchar(200) |  | Yes |
| itemPrice | Current price of food item | smallmoney |  | Yes |

**Orderitem**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Data Type | Constraint | Null value |
| orderID | Unique number to identify customer's order | char(10) | Primary key  Foreign key  →custOrder  (orderID) | No |
| itemID | Unique number to identify item | tinyint | Primary key  Foreign key  → Item(itemID) | No |
| orderQty | Number of orders made | smallint |  | No |
| unitPrice | Actual price paid for food item | smallmoney |  | No |

**Promotion**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Data Type | Constraint | Null value |
| promoID | Unique number to identity promotion | char(8) | Primary key | No |
| promoName | Name of promotion | varchar(20) |  | No |
| promoDesc | Description of promotion | varchar(50) |  | No |
| percentDiscount | Discount percent of coupon | decimal(3,2) |  | Yes |
| isFreeDelivery | Check if promotion includes free delivery | varchar(5) | Must be either ‘Yes’ or ‘No’ | No |

**OutletContact**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Data Type | Constraint | Null value |
| outletID | Unique number to identify outlet | char(10) | Primary key  Foregin key  →Outlet  (custID) | No |
| contactNo | Outlet contact number | char(8) | Primary key  All digits  First digit starts with ‘6’ | No |

**Enjoy**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Data Type | Constraint | Null value |
| orderID | Unique number to identify customer's order | char(10) | Primary key  Foreign key  → CustOrder  (orderID) | No |
| promoID | Unique number to identify promotion | char(8) | Primary key  Foreign key  →Promotion  (promoId) | No |

**Runs**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Data Type | Constraint | Null value |
| promoID | Unique number to identity promotion | char(8) | Primary key  Foreign key  → Promotion  (promoID) | No |
| outletID | Unique number to identify outlet | char(10) | Primary key  Foreign key  → Outlet(outletID) | No |
| maxCount | Cap on a promotion run by an outlet | smallint |  | No |

**Belongs**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Data Type | Constraint | Null value |
| outletID | Unique number to identify outlet | char(10) | Primary key  Foreign key  →Outlet(outletID) | No |
| cuisineID | Unique number to identify cuisine | char(10) | Primary key  Foreign key  → Cuisine  (cuisineID) | No |

**Has**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Data Type | Constraint | Null value |
| outletID | Unique number to identify outlet | char(10) | Primary key  Foreign key  → Menu  (menuNo, outletID) | No |
| menuNo | Unique number to identify menu | char(10) | Primary key  Foreign key  → Menu  (menuNo,outletID) | No |
| itemID | Unique number to identify item | tinyint | Primary key  Foreign key  → Item(itemID) | No |

**Assign**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Data Type | Constraint | Null value |
| orderID | Unique number to identify customer's order | char(10) | Primary key  Foreign key  → CustOrder  (orderID) | No |
| riderID | Unique number to identify rider | char(10) | Primary key  → Foreign key  Rider(riderID) | No |
| status | Delivered or Delivering | varchar(10) | Must be either  ‘Delivered’ or ‘Delivering’ | No |

**Purchase**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Data Type | Constraint | Null value |
| riderID | Unique number to identify rider | char(10) | Primary key  Foreign key  → Rider(riderID) | No |
| purchaseDateTime | Date when equipment was purchased by rider | smalldatetime | Primary Key | No |
| equipID | Unique number to identify equipment | char(10) | Primary key  Foreign key  →Equipment  (equipID) | No |
| purchaseQty | Number of purchases made | smallint |  | No |

**Achieve**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Description | Data Type | Constraint | Null value |
| riderID | Unique number to identify rider | char(10) | Primary key  Foreign key  →Rider  (riderID,winDate) | No |
| awardID | Unique number to identify award | char(10) | Primary key  Foreign key  → Award  (awardID) | No |
| winDate | Date when award was won | date | Primary Key | No |

# **SQL Creation of Database & Tables**

Create Database MokeFoods

GO

USE MokeFoods

GO

**--Customer**

CREATE TABLE Customer(

custID char(8),

custName varchar(50) NOT NULL,

custAddress varchar(150) NOT NULL,

custContact char(8) NOT NULL,

custEmail varchar(50) NULL,

CONSTRAINT PK\_Customer PRIMARY KEY (custID),

CONSTRAINT CHK\_Customer\_custContact

CHECK (custContact LIKE '[8][0-9][0-9][0-9][0-9][0-9][0-9][0-9]' OR custContact LIKE '[9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'))

**--Voucher**

CREATE TABLE Voucher(

voucherID char(10),

voucherStatus varchar(10) NULL,

voucherDesc varchar(50) NULL,

startDate date NOT NULL,

expiryDate date NOT NULL,

minOrder smallmoney NOT NULL,

dollarValue smallmoney NOT NULL,

custID char(8) NULL,

CONSTRAINT PK\_Voucher PRIMARY KEY (voucherID),

CONSTRAINT FK\_Voucher\_custID FOREIGN KEY(custID) REFERENCES Customer(custID),

CONSTRAINT CHK\_Voucher\_ExpiryDate CHECK(expiryDate > startDate))

**--Business**

CREATE TABLE Business(

bizID char(4),

bizName varchar(50) NOT NULL,

CONSTRAINT PK\_Business PRIMARY KEY(bizID))

**--Zone**

CREATE TABLE Zone(

zoneID char(4),

zoneName varchar(20) NOT NULL,

CONSTRAINT PK\_Zone PRIMARY KEY (zoneID))

**--Outlet**

CREATE TABLE Outlet(

outletID char(10),

outletName varchar(50) NOT NULL,

outletAddress varchar(50) NOT NULL,

deliveryFee smallmoney NOT NULL,

openTime time(0) NOT NULL,

closeTime time(0) NOT NULL,

startDeliveryTime time(0) NULL,

endDeliveryTime time(0) NULL,

bizID char(4) NULL,

zoneID char(4) NULL,

CONSTRAINT PK\_Outlet PRIMARY KEY(outletID),

CONSTRAINT FK\_Outlet\_bizID FOREIGN KEY(bizID) REFERENCES Business(bizID),

CONSTRAINT FK\_Outlet\_zoneID FOREIGN KEY(zoneID) REFERENCES Zone(zoneID),

CONSTRAINT CHK\_Outlet\_closeTime CHECK( closeTime > openTime),

CONSTRAINT CHK\_Outlet\_endDeliveryTime CHECK( endDeliveryTime > startDeliveryTime),

CONSTRAINT CHK\_Outlet\_startDeliveryTime CHECK( startDeliveryTime >= openTime))

**--OutletContact**

CREATE TABLE OutletContact(

outletID char(10),

contactNo char(8),

CONSTRAINT PK\_OutletContact PRIMARY KEY(outletID, ContactNo),

CONSTRAINT FK\_OutletContact\_outletID FOREIGN KEY(outletID) REFERENCES Outlet(outletID),

CONSTRAINT CHK\_OutletContact\_contactNo

CHECK (contactNo LIKE '[6][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'))

**--CustOrder**

CREATE TABLE CustOrder(

orderID char(10),

orderStatus varchar(10) NOT NULL,

orderDateTime smalldatetime NOT NULL,

voucherID char(10) NULL,

custID char(8) NULL,

outletID char(10) NULL,

CONSTRAINT PK\_CustOrder PRIMARY KEY (orderID),

CONSTRAINT FK\_CustOrder\_voucherID FOREIGN KEY(voucherID) REFERENCES Voucher(voucherID),

CONSTRAINT FK\_CustOrder\_custID FOREIGN KEY(custID) REFERENCES Customer(custID),

CONSTRAINT FK\_CustOrder\_outletID FOREIGN KEY(outletID) REFERENCES Outlet(outletID),

CONSTRAINT CHK\_CustOrder\_orderStatus CHECK (orderStatus = 'Preparing' OR orderStatus = 'Completed' OR orderStatus = 'Dispatched' OR orderStatus = 'Accepted' OR orderStatus = 'Cancelled'))

**--Payment**

CREATE TABLE Payment(

pmtID char(10),

pmtMode varchar(20) NOT NULL,

pmtType varchar(20) NOT NULL,

pmtAmt smallmoney NOT NULL,

orderID char(10) NULL,

CONSTRAINT PK\_Payment PRIMARY KEY (pmtID),

CONSTRAINT FK\_Payment\_orderID FOREIGN KEY(orderID) REFERENCES CustOrder(orderID),

CONSTRAINT CHK\_Payment\_pmtMode CHECK (pmtMode = 'NETS' OR pmtMode = 'Credit Card' OR pmtMode = 'Internet Banking' OR pmtMode = 'VISA' OR pmtMode = 'Electronic Wallet'),

CONSTRAINT CHK\_Payment\_pmtType CHECK (pmtType = 'Order Payment' OR pmtType = 'Refund'))

**--Pickup**

CREATE TABLE Pickup(

orderID char(10),

pickupRefNo varchar(10) NULL,

pickupDateTime smalldatetime NULL,

CONSTRAINT PK\_Pickup PRIMARY KEY (orderID),

CONSTRAINT FK\_Pickup\_orderID FOREIGN KEY(orderID) REFERENCES CustOrder(orderID))

**--Award**

CREATE TABLE Award(

awardID char(10),

awardName varchar(50) NOT NULL,

awardType varchar(50) NOT NULL,

CONSTRAINT PK\_Award PRIMARY KEY (awardID),

CONSTRAINT CHK\_Award\_awardType CHECK (awardType = 'Team' OR awardType = 'Individual'))

**--Team**

CREATE TABLE Team(

teamID char(4),

teamName varchar(50) NOT NULL,

awardID char(10) NULL,

leaderID char(10) NULL,

CONSTRAINT PK\_Team PRIMARY KEY (teamID),

CONSTRAINT FK\_Team\_awardID FOREIGN KEY (awardID) REFERENCES Award(awardID))

**--Rider**

CREATE TABLE Rider(

riderID char(10),

riderNRIC char(9) NOT NULL,

riderName varchar(50) NOT NULL,

riderContact char(8) NOT NULL,

riderAddress varchar(50) NOT NULL,

riderDOB date NULL,

deliveryMode varchar(20) NOT NULL,

teamID char(4) NULL,

CONSTRAINT PK\_Rider PRIMARY KEY (riderID),

CONSTRAINT FK\_Rider\_teamID FOREIGN KEY (teamID) REFERENCES Team(teamID),

CONSTRAINT CHK\_Ridert\_riderNRIC

CHECK (riderNRIC LIKE '[S][0-9][0-9][0-9][0-9][0-9][0-9][0-9][A-Z]' OR

riderNRIC LIKE '[T][0-9][0-9][0-9][0-9][0-9][0-9][0-9][A-Z]' OR

riderNRIC LIKE '[F][0-9][0-9][0-9][0-9][0-9][0-9][0-9][A-Z]' OR

riderNRIC LIKE '[G][0-9][0-9][0-9][0-9][0-9][0-9][0-9][A-Z]'),

CONSTRAINT CHK\_Ridert\_riderContact CHECK (riderContact LIKE '[8][0-9][0-9][0-9][0-9][0-9][0-9][0-9]' OR riderContact LIKE '[9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'))

ALTER TABLE Team

ADD CONSTRAINT FK\_Team\_leaderID FOREIGN KEY (leaderID) REFERENCES

Rider(riderID)

**--Delivery**

CREATE TABLE Delivery(

orderID char(10),

deliveryAddress varchar(50) NOT NULL,

deliveryDateTime smalldatetime NOT NULL,

riderID char(10) NULL,

CONSTRAINT PK\_Delivery PRIMARY KEY (orderID),

CONSTRAINT FK\_Delivery\_riderID FOREIGN KEY(riderID) REFERENCES Rider(riderID))

**--Equipment**

CREATE TABLE Equipment(

equipID char(10),

equipName varchar(50) NOT NULL,

equipPrice smallmoney NULL,

setID char(10) NULL,

CONSTRAINT PK\_Equipment PRIMARY KEY NONCLUSTERED (equipID),

CONSTRAINT FK\_Equipment\_setID FOREIGN KEY (setID) REFERENCES Equipment(equipID))

**--Item**

CREATE TABLE Item(

itemID tinyint,

itemName varchar(50) NOT NULL,

itemDesc varchar(200) NULL,

itemPrice smallmoney NULL,

CONSTRAINT PK\_Item PRIMARY KEY (itemID))

**--OrderItem**

CREATE TABLE OrderItem(

orderID char(10),

itemID tinyint,

orderQty smallint NOT NULL,

unitPrice smallmoney NOT NULL,

CONSTRAINT PK\_OrderItem PRIMARY KEY (orderID,itemID),

CONSTRAINT FK\_OrderItem\_orderID FOREIGN KEY (orderID) REFERENCES CustOrder(orderID),

CONSTRAINT FK\_OrderItem\_itemID FOREIGN KEY (itemID) REFERENCES Item(itemID))

**--Menu**

CREATE TABLE Menu(

menuNo char(10) NOT NULL,

outletID char(10) NOT NULL,

menuName varchar(100) NOT NULL,

CONSTRAINT PK\_Menu PRIMARY KEY (menuNo,outletID),

CONSTRAINT FK\_Menu\_outletID FOREIGN KEY (outletID) REFERENCES Outlet(outletID))

**--Cuisine**

CREATE TABLE Cuisine(

cuisineID char(10),

cuisineName varchar(50) NOT NULL,

CONSTRAINT PK\_Cuisine PRIMARY KEY(cuisineID))

**--Promotion**

CREATE TABLE Promotion(

promoID char(8),

promoName varchar(20) NOT NULL,

promoDesc varchar(50) NOT NULL,

percentDiscount decimal(3,2) NULL,

isFreeDelivery varchar(5) NOT NULL,

CONSTRAINT PK\_Promotion PRIMARY KEY(promoID),

CONSTRAINT CHK\_Promotion\_isFreeDelivery CHECK (isFreeDelivery = 'Yes' OR isFreeDelivery = 'No'))

**--Enjoys**

CREATE TABLE Enjoys(

orderID char(10),

promoID char(8),

CONSTRAINT PK\_Enjoys PRIMARY KEY(orderID,promoID),

CONSTRAINT FK\_Enjoys\_orderID FOREIGN KEY (orderID) REFERENCES CustOrder(orderID),

CONSTRAINT FK\_Enjoys\_promoID FOREIGN KEY (promoID) REFERENCES Promotion(promoID))

**--Runs**

CREATE TABLE Runs(

promoID char(8),

outletID char(10),

maxCount smallint NOT NULL,

CONSTRAINT PK\_Runs PRIMARY KEY(promoID,outletID),

CONSTRAINT FK\_Runs\_promoID FOREIGN KEY (promoID) REFERENCES Promotion(promoID),

CONSTRAINT FK\_Runs\_outletID FOREIGN KEY (outletID) REFERENCES Outlet(outletID))

**--Belongs**

CREATE TABLE Belongs(

outletID char(10),

cuisineID char(10),

CONSTRAINT PK\_Belongs PRIMARY KEY(outletID,cuisineID),

CONSTRAINT FK\_Belongs\_outletID FOREIGN KEY(outletID) REFERENCES Outlet(outletID),

CONSTRAINT FK\_Belongs\_cuisineID FOREIGN KEY (cuisineID) REFERENCES Cuisine(cuisineID))

**--Has**

CREATE TABLE Has(

menuNo char(10) NOT NULL,

outletID char(10) NOT NULL,

itemID tinyint NOT NULL,

CONSTRAINT PK\_Has PRIMARY KEY(menuNo, outletID, itemID),

CONSTRAINT FK\_Has\_outletID\_menuNo FOREIGN KEY(menuNo,outletID) REFERENCES Menu(menuNo,outletID),

CONSTRAINT FK\_Has\_itemID FOREIGN KEY(itemID) REFERENCES Item(itemID))

**--Purchase**

CREATE TABLE Purchase(

riderID char(10),

purchaseDateTime smalldatetime,

equipID char(10),

purchaseQty smallint NOT NULL,

CONSTRAINT PK\_Purchase PRIMARY KEY(riderID,purchaseDateTime,equipID),

CONSTRAINT FK\_Purchase\_riderID FOREIGN KEY(riderID) REFERENCES Rider(riderID),

CONSTRAINT FK\_Purchase\_equipID FOREIGN KEY(equipID) REFERENCES Equipment(equipID))

**--Achieve**

CREATE TABLE Achieve (

riderID char(10),

winDate date,

awardID char(10),

CONSTRAINT PK\_Achieve PRIMARY KEY(riderID,winDate,awardID),

CONSTRAINT FK\_Achieve\_riderID FOREIGN KEY (riderID) REFERENCES Rider(riderID),

CONSTRAINT FK\_Achieve\_awardID FOREIGN KEY (awardID) REFERENCES Award(awardiD))

**--Assign**

CREATE TABLE Assign(

orderID char(10),

riderID char(10),

status varchar(10) NOT NULL,

CONSTRAINT PK\_Assign PRIMARY KEY(orderID,riderID),

CONSTRAINT FK\_Assign\_orderID FOREIGN KEY(orderID) REFERENCES CustOrder(orderID),

CONSTRAINT FK\_Assign\_riderID FOREIGN KEY(riderID) REFERENCES Rider(riderID),

CONSTRAINT CHK\_Assign\_status CHECK (status = 'Delivered' OR status = 'Delivering'))

# **Insertion of Values**

**--Customer**

INSERT INTO Customer

VALUES

('88020501', 'Johnny Tan Wei Li', '36, Long Kang Ave 6 #06-12, Singapore 134825', '81234567', 'johnnytan31@gmail.com'),

('88020502', 'Kok Leong Schlong', '69, Ang Mo Kio Ave 6 #12-69, Singapore 126929', '83894131', 'klschlong69@gmail.com'),

('88020503', 'Willie Tan Cheng Xian', '3, Toa Payoh Lor 6 #16-34, Singapore 126391', '98765123', 'williebillie19@gmail.com'),

('88020504', 'Annie Kuan Xin Tong', '67, Pasir Panjang Rd #02-21, Singapore 139241', '83891034', 'anniekxt17@gmail.com'),

('88020505', 'Krystal Chew Xiao Li', '31, Bukit Timah Rd #21-87, Singapore 187402', '92345674', 'krystalball30@gmail.com'),

('88020506', 'Xenon Affroe Zacheus', '17, Kim Keat View #14-17, Singapore 128301', '84572130', 'xenozachy@gmail.com'),

('88020507', 'Peter Phua Par Yar', '7, Bishan Street 12 #03-85, Singapore 112010', '93278909', 'phuaparyar99@gmail.com'),

('88020508', 'Eva Liew', '9, Choa Chu Kang Loop #08-33, Singapore 168215', '91745690', 'evaliew09@gmail.com'),

('88020509', 'Greg Meister', '49, Woodlands Ave 5 #18-22, Singapore 111456', '83478103', 'gregmeister56@gmail.com'),

('88020510', 'Noah Chow Wei Ming', '54, Yishun Street 21 #05-89, Singapore 123456', '90762834', 'noahchowneymar32@gmail.com'),

('88020511', 'Lesley Ong Shui Yan', '78, Sembawang Dr #06-73, Singapore 129921', '91348972', 'lesleyongsy93@gmail.com'),

('88020512', 'Monica Cheng Ke Yi', '23, Hougang Ave 1 #10-44, Singapore 171935', '84012939', 'kymonicacheng12@gmail.com'),

('88020513', 'Eddie Chong Chia Ming', '81, Serangoon Ave 1 #08-09, Singapore 149182', '88881234', 'eddieccharming81@gmail.com'),

('88020514', 'Melvin Wang Jing Kai', '72, Clementi Ave 3 #04-25, Singapore 156712', '90867212', 'melvinkai@gmail.com'),

('88020515', 'Mary Lim Si Tian', '41, Jurong West Street 64 #17-05, Singapore 189124', '88112233', 'limtiantian85@gmail.com'),

('88020516', 'Lester Wee Wei Feng', '15, Sengkang Central #25-23, Singapore 162910', '98981235', 'mlesterwee296@gmail.com');

**--Voucher**

INSERT INTO Voucher

VALUES

('1182050001', 'VALID', '$10 Discount e-Voucher', '2021-01-15', '2021-03-10', 40.00, 10.00, '88020510'),

('1182050002', 'VALID', '$10 Discount e-Voucher', '2021-01-05', '2021-04-05', 30.00, 10.00, '88020511'),

('1182050003', 'INVALID', '$25 Discount e-Voucher', '2020-12-18', '2021-01-01', 80.00, 25.00, '88020506'),

('1182050004', 'VALID', '$20 Discount e-Voucher', '2021-01-01', '2021-03-25', 75.00, 20.00, '88020508'),

('1182050005', 'INVALID', 'Buy 2 Get $15 Discount e-Voucher', '2020-12-31', '2021-01-07', 0.00, 15.00, '88020515'),

('1182050006', 'VALID', '$3 Delivery Fee Promo e-Voucher', '2021-01-05', '2021-03-01', 15.00, 3.00, '88020514'),

('1182050007', 'VALID', '$5 Discount e-Voucher', '2021-01-03', '2021-04-20', 20.00, 5.00, '88020507'),

('1182050008', 'VALID', '$15 Discount e-Voucher', '2021-01-10', '2021-05-01', 45.00, 15.00, '88020514'),

('1182050009', 'INVALID', '$10 Discount e-Voucher', '2020-12-25', '2021-01-01', 35.00, 10.00, '88020506'),

('1182050010', 'INVALID', '$3 Delivery Fee Promo e-Voucher', '2021-01-01', '2021-01-06', 15.00, 3.00, '88020511'),

('1182050011', 'VALID', 'Buy 3 Get $20 Discount e-Voucher', '2021-01-02', '2021-03-01', 0.00, 20.00, '88020506'),

('1182050012', 'INVALID' , '$20 Discount e-Voucher', '2020-12-28', '2021-01-10', 50.00, 20.00, '88020509'),

('1182050013', 'VALID', 'Buy 2 Get $10 Discount e-Voucher', '2021-01-15', '2021-02-27', 0.00, 10.00, '88020516'),

('1182050014', 'INVALID', '$15 Discount e-Voucher', '2021-01-10', '2021-01-20', 40.00, 15.00, '88020509'),

('1182050015', 'VALID', '$5 Discount e-Voucher', '2021-01-20', '2021-06-20', 15.00, 5.00,'88020501');

**--Business**

INSERT INTO Business

VALUES

('B001', 'RonaldDonald'),

('B002', 'Long Schlong Silver'),

('B003', 'Kai Xing Hainanese Chicken Rice'),

('B004', 'Ho Jiak Dim Sum'),

('B005', 'Vasantham Prata'),

('B006', 'HuoLong BubbleTea'),

('B007', 'Burger Queen'),

('B008', 'Earthly Goodness'),

('B009', 'Pui Pui Burger'),

('B010', 'Fish & Chics'),

('B011', 'Ah Bee Yong Tow Foo'),

('B012', 'Yee Haw Western Food'),

('B013', 'A-Yum Penyet'),

('B014', 'Makcik Nasi Lemak'),

('B015', 'Oishi Raja');

**--Zone**

INSERT INTO Zone

VALUES

('Z001', 'North'),

('Z002', 'South'),

('Z003', 'East'),

('Z004', 'West'),

('Z005', 'Central'),

('Z006', 'North-East'),

('Z007', 'North-West'),

('Z008', 'South-East'),

('Z009', 'South-West'),

('Z010', 'North-Central'),

('Z011', 'South-Central'),

('Z012', 'East-Central'),

('Z013', 'West-Central');

**--Outlet**

INSERT INTO Outlet

VALUES

('6692050001', 'Vasantham Prata @ Jurong West', '762, Jurong West Street 75, Singapore 640762', 4.00, '08:00', '21:00', '08:00', '20:45', 'B005', 'Z009'),

('6692050002', 'Ronald Donald @ Orchard', '2, Orchard Turn, Singapore 238801', 3.00, '00:00', '23:59','00:00','23:59', 'B001', 'Z005'),

('6692050003', 'Fish & Chics @ Tampines', '4, Tampines Central 5, Singapore 529510', 3.00, '09:00', '21:00', '09:00', '20:45', 'B010', 'Z003'),

('6692050004', 'Ah Bee Yong Tow Foo @ Tanjong Pagar', '7, Wallich St, Singapore 078884', 4.00, '08:00', '15:00', '08:00', '14:45', 'B011', 'Z002'),

('6692050005', 'Burger Queen @ Clark Quay', '6, Eu Tong Sen St, Singapore 059817', 3.00, '09:00', '22:00', '09:00', '21:45', 'B007', 'Z011'),

('6692050006', 'Ho Jiak Dim Sum @ Tampines', '5, Tampines Central 6, Singapore 749932' , 4.00, '09:00', '21:00', '09:00', '20:45', 'B004', 'Z003'),

('6692050007', 'Makcik Nasi Lemak @ Yio Chu Kang', '5209, Ang Mo Kio Ave 6, Singapore 569845', 4.00, '08:00', '15:00', '08:00', '14:45', 'B014', 'Z010'),

('6692050008', 'Pui Pui Burger @ Serangoon', '23, Serangoon Central, Singapore 556083', 4.00, '09:00', '21:00', '09:00', '20:45', 'B009', 'Z012'),

('6692050009', 'Ronald Donald @ Marina Bay', '2, Bayfront Ave Marina Bay Sands, 018972', 3.00, '00:00', '23:59', '00:00', '23:59', 'B001', 'Z002'),

('6692050010', 'A-Yum Penyet @ Serangoon', '202, Serangoon Central, Singapore 550202', 3.00, '09:00', '21:00', '09:00', '20:45', 'B013', 'Z012'),

('6692050011', 'Oishi Raja @ Yishun', '930, Yishun Ave 2, Singapore 769098', 4.00, '09:00', '21:00', '09:00', '20:45', 'B015', 'Z001'),

('6692050012', 'Earthly Goodness @ Kranji', '55, Sungei Kadut Street 1, Singapore 729358', 4.00, '09:00', '21:00', '09:00', '20:45', 'B008', 'Z004'),

('6692050013', 'Kai Xing Hainanese Chicken Rice @ Orchard', '313, Orchard Rd, Singapore 238895', 4.00, '09:00', '21:00', '09:00', '20:45', 'B003', 'Z005'),

('6692050014', 'Yee Haw Western Food @ Somerset', '181, Orchard Rd, Singapore 238896', 4.00, '09:00', '21:00', '09:00', '20:45', 'B012', 'Z005'),

('6692050015', 'HuoLong BubbleTea @ Bishan', '9, Bishan Pl, Singapore 579837', 4.00, '09:00', '21:00', '09:00', '20:45', 'B006', 'Z013'),

('6692050016', 'Long Schlong Silver @ Sengkang', '1, Sengkang Square, Singapore 545078', 3.00, '09:00', '22:00', '09:00', '21:45', 'B002', 'Z006'),

('6692050017', 'Ronald Donald @ Woodlands', '2, Woodlands Square, Singapore 738099', 3.00, '00:00', '23:59', '00:00', '23:59', 'B001', 'Z002'),

('6692050018', 'HuoLong BubbleTea @ Bukit Batok', '3, Bukit Batok Central Link, Singapore 658713', 4.00, '09:00', '21:00', '09:00', '20:45', 'B006', 'Z007'),

('6692050019', 'A-Yum Penyet @ Hillview', '4, Hillview Rise, Singapore 667979', 3.00, '09:00', '21:00', '09:00', '20:45', 'B013', 'Z004'),

('6692050020', 'Kai Xing Hainanese Chicken Rice @ Yew Tee', '666, Choa Chu Kang Cres, Singapore 680666', 4.00, '09:00', '21:00', '09:00', '20:45', 'B003', 'Z007'),

('6692050021', 'Ho Jiak Dim Sum @ Toa Payoh', '510, Lor 6 Toa Payoh, Singapore 319398', 4.00, '09:00', '21:00', '09:00', '20:45', 'B004', 'Z005'),

('6692050022', 'Oishi Raja @Tampines', '60, Tampines North Drive 2, Singapore 528764', 4.00, '09:00', '21:00', '09:00', '20:45', 'B015', 'Z003'),

('6692050023', 'Ah Bee Yong Tow Foo - Clementi', '315, Commonwealth Ave W, Singapore 129588', 4.00, '08:00', '15:00', '08:00', '14:45', 'B011', 'Z009'),

('6692050024', 'Makcik Nasi Lemak @ Marina South', '10, Bayfront Ave, Singapore 018956', 4.00, '08:00', '15:00', '08:00', '14:45', 'B014', 'Z008'),

('6692050025', 'Long Schlong Silver @ Novena', '101, Thomson Rd, Singapore 307591', 3.00, '09:00', '22:00', '09:00', '21:45', 'B002', 'Z012');

**--OutletContact**

INSERT INTO OutletContact

VALUES

('6692050001', '67891011'),

('6692050002', '68910237'),

('6692050003', '62832020'),

('6692050004', '61891108'),

('6692050005', '68910339'),

('6692050006', '63229019'),

('6692050007', '62920183'),

('6692050008', '62830348'),

('6692050009', '65820201'),

('6692050010', '68283738'),

('6692050011', '64220138'),

('6692050012', '68011283'),

('6692050013', '69219201'),

('6692050014', '66929338'),

('6692050015', '63020283'),

('6692050016', '62820101'),

('6692050017', '63829201'),

('6692050018', '67391012'),

('6692050019', '68292389'),

('6692050020', '68391028'),

('6692050021', '67929128'),

('6692050022', '61118390'),

('6692050023', '68301029'),

('6692050024', '67392988'),

('6692050025', '63722367');

**--CustOrder**

INSERT INTO CustOrder

VALUES

('5402050001', 'Completed', '2021-01-09 12:30:30', '1182050005', '88020508', '6692050022'),

('5402050002', 'Preparing', '2021-02-01 14:37:32', '1182050012', '88020506', '6692050018'),

('5402050003', 'Accepted', '2021-02-01 14:38:27', '1182050014', '88020504', '6692050009'),

('5402050004', 'Dispatched', '2021-02-01 14:26:39', '1182050004', '88020512', '6692050017'),

('5402050005', 'Cancelled', '2020-12-25 18:45:18', '1182050011', '88020504', '6692050006'),

('5402050006', 'Completed', '2021-01-03 15:34:25', '1182050013', '88020505', '6692050016'),

('5402050007', 'Preparing', '2021-02-01 14:39:00', '1182050006', '88020516', '6692050009'),

('5402050008', 'Accepted', '2021-02-01 14:40:00', '1182050012', '88020513', '6692050007'),

('5402050009', 'Completed', '2021-12-29 16:50:40', '1182050005', '88020511', '6692050005'),

('5402050010', 'Completed', '2021-01-07 18:12:33', '1182050003', '88020501', '6692050022'),

('5402050011', 'Dispatched', '2021-02-01 14:45:12', '1182050007', '88020510', '6692050018'),

('5402050012', 'Cancelled', '2020-11-27 20:01:24', '1182050001', '88020502', '6692050016'),

('5402050013', 'Preparing', '2021-02-01 14:33:40', '1182050014', '88020508', '6692050020'),

('5402050014', 'Completed', '2021-01-01 08:39:01', '1182050008', '88020503', '6692050009'),

('5402050015', 'Completed', '2020-11-24 12:17:15', '1182050010', '88020507', '6692050024'),

('5402050016', 'Accepted', '2021-02-01 14:42:53', '1182050002', '88020509', '6692050001'),

('5402050017', 'Completed', '2021-01-13 16:03:39', '1182050015', '88020501', '6692050013'),

('5402050018', 'Dispatched', '2021-02-01 14:45:17', '1182050006', '88020501', '6692050012'),

('5402050019', 'Completed', '2021-01-07 11:47:10', '1182050003', '88020502', '6692050015'),

('5402050020', 'Accepted', '2021-02-01 14:37:07', '1182050009', '88020515', '6692050004');

**--Payment**

INSERT INTO Payment

VALUES

('4403070001', 'Credit Card', 'Order Payment', 35.50, '5402050014'),

('4403070002', 'NETS', 'Order Payment', 17.30, '5402050011'),

('4403070003', 'Internet Banking', 'Refund', 9.40, '5402050012'),

('4403070004', 'Electronic Wallet', 'Order Payment', 56.10, '5402050002'),

('4403070005', 'VISA', 'Order Payment', 29.90, '5402050009'),

('4403070006', 'NETS', 'Order Payment', 24.80, '5402050004'),

('4403070007', 'Credit Card', 'Order Payment', 5.75, '5402050010'),

('4403070008', 'NETS', 'Order Payment', 13.90, '5402050008'),

('4403070009', 'Internet Banking', 'Order Payment', 19.95, '5402050007'),

('4403070010', 'Electronic Wallet', 'Order Payment', 26.60, '5402050001'),

('4403070011', 'Internet Banking', 'Order Payment', 14.80, '5402050019'),

('4403070012', 'VISA', 'Order Payment', 38.40, '5402050003'),

('4403070013', 'NETS', 'Refund', 20.00, '5402050005'),

('4403070014', 'NETS', 'Order Payment', 8.30, '5402050018'),

('4403070015', 'Electronic Wallet', 'Order Payment', 17.95, '5402050015'),

('4403070016', 'Credit Card', 'Order Payment', 58.50, '5402050006'),

('4403070017', 'VISA', 'Order Payment', 13.35, '5402050016'),

('4403070018', 'Internet Banking', 'Order Payment', 16.20, '5402050017'),

('4403070019', 'Credit Card', 'Order Payment', 78.20, '5402050020'),

('4403070020', 'Credit Card', 'Order Payment', 109.50, '5402050013');

**--Pickup**

INSERT INTO Pickup

VALUES

('5402050001', '737', '2021-01-09 12:58:16'),

('5402050005', NULL, NULL),

('5402050006', '1269', '2021-01-03 15:33:25'),

('5402050008', '138', '2021-02-01 14:59:04'),

('5402050010', '961', '2021-01-7 18:11:36'),

('5402050012', NULL, NULL),

('5402050014', '3651', '2021-01-01 08:37:49'),

('5402050016', '873', '2021-02-01 15:08:33'),

('5402050017', '946', '2021-01-13 16:02:45'),

('5402050020', '083', '2021-01-01 15:03:13');

**--Award**

INSERT INTO Award

VALUES

('5272050001', 'Highest Number of Order Fulfillment', 'Individual'),

('5272050002', 'Highest Number of Order Fulfillment', 'Team'),

('5272050003', 'Highest Acceptance Rate', 'Individual'),

('5272050004','Highest Acceptance Rate', 'Team'),

('5272050005','Highest Total Ratings', 'Individual'),

('5272050006', 'Highest Total Ratings', 'Team'),

('5272050007', 'Fastest Average Delivery Time', 'Individual'),

('5272050008', 'Fastest Average Delivery Time', 'Team'),

('5272050009', 'Million Mile Distance Coverage', 'Individual'),

('5272050010', 'Longest Service Award', 'Individual');

**--Team**

INSERT INTO Team

VALUES

('T001', 'KenOneLah', '5272050004', NULL),

('T002', 'EzClap', NULL, NULL),

('T003', 'Nyoooom', NULL, NULL),

('T004', 'VroomVroom', '5272050008', NULL),

('T005', 'KenaLanga', NULL, NULL),

('T006', 'MacikRules', '5272050002', NULL),

('T007', 'ZeroAccident', NULL, NULL),

('T008', 'SuperSonic', '5272050006', NULL),

('T009', 'NootNation', NULL, NULL),

('T010', 'GottaCatchThemAll', NULL, NULL);

**--Rider**

INSERT INTO Rider

VALUES

('5342050001', 'S7518723Z', 'Laila Lim Bee Huey', '83717100', '357, Clementi Ave 2, Singapore 120654', '1975-08-19', 'Motorcycle', 'T008'),

('5342050002', 'S6819320N', 'Aleeza Tan Hui Min', '92728281', '20, Lorong 16 Geylang, Singapore 398863', '1968-07-12', 'Bicycle', 'T004'),

('5342050003', 'T0030320B', 'Thomas Chan Guo Hao', '82729177', '110, Jalan Bukit merah, Singapore 160110', '2000-01-28', 'Bicycle', 'T007'),

('5342050004', 'S8049274L', 'Ardii Cheng Hao Wei', '82300126', '163, Bedok S Rd, Singapore 560163', '1980-11-29', 'Motorcycle', 'T002'),

('5342050005', 'S8229830P', 'Patrick Lim Ah Heok', '87271903', 'Rivervale Cres, Singapore 54216', '1982-03-17', 'Motorcycle', 'T004'),

('5342050006', 'S9124901A', 'Jefferson Wong Jun Wei', '97868932', '69, Jurong West Central 1, Singapore 640543', '1991-09-24', 'Bicycle', 'T006'),

('5342050007', 'S6939834K', 'Chloe Cheng Hui Xin', '96897796', '80, Mandai Lake Rd, Singapore 72986', '1969-12-27', 'Motorcycle', 'T010'),

('5342050008', 'S6639220L', 'Gan Heong Wei', '83409737', '207, Toa Payoh N, Singapore 310207', '1996-11-30', 'Motorcycle', 'T002'),

('5342050009', 'T0187202I', 'Lucas Teo Wei Ming', '96114533', '357, Admiralty Dr, Singapore 750357', '2001-04-06', 'Bicycle', 'T006'),

('5342050010', 'T0072398U', 'Skye Lim Ke Bei', '95372087', '20, Jln Membina, Singapore 164020', '2000-11-20', 'Bicycle', 'T001'),

('5342050011', 'S6874292H', 'Zac Chen Ke Lian', '96811989', '294, Punggol Central, Singapore 289763', '1968-11-29', 'Motorcycle', 'T007'),

('5342050012', 'S7947291P', 'Jesus Wilder', '95274152', '61, Bishan Street 21, Singapore 570443', '1979-05-15', 'Motorcycle', 'T005'),

('5342050013', 'S7937202Q', 'Matthew Lye Hong Jun', '89625144', '285, Choa Chu Kang, Singapore 680295', '1979-04-18', 'Bicycle', 'T008'),

('5342050014', 'T0182920R', 'Timothy Smith', '81996927', '106, Simei Street 1, Singapore 520106', '2001-02-21', 'Bicycle', 'T005'),

('5342050015', 'S8323930T', 'Kok Bi Hiam', '83790245', '56, Lim Chu Kang Lane 6, Singapore 719614', '1983-07-18', 'Motorcycle', 'T003'),

('5342050016', 'S8673292B', 'Pearlyn Goh Si Yu', '93891129', '20, Eastwood Rd, Singapore 486442', '1986-05-12', 'Motorcycle', 'T009'),

('5342050017', 'S8474922C', 'Travis Teo Che Zhuang', '81759795', '344, Ang Mo Kio Ave 3, Singapore 560382', '1984-08-17', 'Motorcycle', 'T001'),

('5342050018', 'T0148202X', 'Chea Tim Yeok', '98715401', '766, Choa Chu Kang North 5, Singapore 789212', '2001-04-19', 'Bicycle', 'T010'),

('5342050019', 'S7584208G', 'Justin Gan Er Zi', '89301332', '32, Hougang Ave 6, Singapore 782102', '1975-01-21', 'Motorcycle', 'T009'),

('5342050020', 'S7932820F', 'Aaron Lam Jing Yu', '87387101', '67, Yishun Ave 5, Singapore 261782', '1979-09-17', 'Motorcycle', 'T003');

UPDATE Team SET leaderID='5342050017' WHERE teamID='T001'

UPDATE Team SET leaderID='5342050008' WHERE teamid='T002'

UPDATE Team SET leaderID='5342050015' WHERE teamid='T003'

UPDATE Team SET leaderID='5342050005' WHERE teamid='T004'

UPDATE Team SET leaderID='5342050014' WHERE teamid='T005'

UPDATE Team SET leaderID='5342050009' WHERE teamid='T006'

UPDATE Team SET leaderID='5342050011' WHERE teamid='T007'

UPDATE Team SET leaderID='5342050013' WHERE teamid='T008'

UPDATE Team SET leaderID='5342050019' WHERE teamid='T009'

UPDATE Team SET leaderID='5342050018' WHERE teamid='T010'

**--Delivery**

INSERT INTO Delivery

VALUES

('5402050002', '17, Kim Keat View #14-17, Singapore 128301', '2021-01-01 15:07:37', '5342050009'),

('5402050003', '67, Pasir Panjang Rd #02-21, Singapore 139241', '2021-02-01 15:18:39', '5342050010'),

('5402050004', '23, Hougang Ave 1 #10-44, Singapore 171935', '2021-02-01 14:31:43', '5342050015'),

('5402050007', '15, Sengkang Central #25-23, Singapore 162910', '2021-02-01 15:13:57', '5342050004'),

('5402050009', '78, Sembawang Dr #06-73, Singapore 129921', '2021-12-29 16:49:11', '5342050014'),

('5402050011', '54,Yishun Street 21 #05-89, Singapore 123456', '2021-02-01 14:53:35', '5342050009'),

('5402050013', '9, Choa Chu Kang Loop #08-33, Singapore 168215', '2021-02-1 15:08:19', '5342050003'),

('5402050015', 'Bishan Street 12 #03-85, Singapore 112010', '2020-11-24 12:16:25', '5342050006'),

('5402050018', '72, Clementi Ave 3 #04-25, Singapore 156712', '2021-02-01 14:54:56', '5342050013'),

('5402050019', '69, Ang Mo Kio Ave 6 #12-69, Singapore 126929', '2021-01-07 11:46:03', '5342050007');

**--Equipment**

INSERT INTO Equipment

VALUES

('5482050001', 'B Long Sleeve T', 15.00, '5482050001'),

('5482050002', 'V Long Sleeve T', 15.00, '5482050002'),

('5482050003', 'E Long Sleeve T', 15.00, '5482050003'),

('5482050004', 'B Short Sleeve T', 15.00, '5482050001'),

('5482050005', 'V Short Sleeve T', 15.00, '5482050002'),

('5482050006', 'E Short Sleeve T', 15.00, '5482050003'),

('5482050007', 'B Normal Bag', 20.00, '5482050001'),

('5482050008', 'V Normal Bag', 20.00, '5482050002'),

('5482050009', 'E Normal Bag', 20.00, '5482050003'),

('5482050010', 'B Halal Bag', 20.00, '5482050001'),

('5482050011', 'V Halal Bag', 20.00, '5482050002'),

('5482050012', 'E Halal Bag', 20.00, '5482050003'),

('5482050013', 'V Backpack', 30.00, '5482050002'),

('5482050014', 'E Backpack', 30.00, '5482050003'),

('5482050015', 'Rain Coat', 30.00, '5482050003' );

**--Item**

INSERT INTO Item

VALUES

(1,'Cheese burger','A classic burger with a beef patty and a slice of cheese between two slices of burger bread',2.00),

(2,'Double Chicken filet burger', 'A burger with two chicken filet, a slice of cheese, mayo and lettuce in between two slices of burger bread',4.50),

(3, 'New York Fish and Chips', 'Two filets of dory fish that are fried in New York style', 7.00),

(4, 'Classic Fish Burger', 'Fried fish patty with a slice of cheese and lettuce between two slices of burger bread',2.00),

(5, 'Seafood Fried Rice', 'Fried rice accompanied with prawns, sting rays, and sotongs',5.00),

(6, 'Chilli Crab','Fresh crabs mixed with the spiciest of chilli spices', 20.00),

(7, 'Egg Prata', 'Prata with egg',1.50),

(8,'Pearl Milk Tea','Milk Tea with the addition of fan-favourite black pearls',3.00),

(9,'Sweet and Sour Pork','Pork fried with batter mixed with a sauce that is both sweet and sour', 5.50),

(10,'Lemon Fish','Fried fish filet a sweet lemon sauce',5.50),

(11,'Rice with 5 toppings', 'Rice with soup with a choice of 5 toppings for consideration',4.50),

(12,'Ayam-penyet', 'Chicken with malay rice and curry',4.00),

(13,'Nasi-Lemak Chicken wing','Coconut rice with a side of chicken wings, pickles and nuts',2.00),

(14,'Prawn Noodles','Yellow Noodles cooked with fresh prawns',2.50),

(15,'Mee goreng', 'Fried noodles with sweet sauce',2.50);

**--OrderItem**

INSERT INTO OrderItem

VALUES

('5402050001',2,3,13.50),

('5402050002',4,2,4.00),

('5402050003',3,1,7.00),

('5402050004',1,2,4.00),

('5402050005',7,4,6.00),

('5402050006',5,1,5.00),

('5402050007',7,2,3.00),

('5402050008',1,2,4.00),

('5402050009',8,2,6.00),

('5402050010',9,1,5.50),

('5402050011',10,2,11.00),

('5402050012',2,2,9.00),

('5402050013',14,2,5.00),

('5402050014',11,1,4.50),

('5402050015',13,2,4.00),

('5402050016',12,1,4.00),

('5402050017',14,1,2.50),

('5402050018',15,2,5.00),

('5402050019',6,1,20.00),

('5402050020',2,2,9.00);

**--Menu**

INSERT INTO Menu

VALUES

('4372050001', '6692050002', 'Ronald Donald"s Menu'),

('4372050001', '6692050009', 'Ronald Donald"s Menu'),

('4372050001', '6692050017', 'Ronald Donald"s Menu'),

('4372050002', '6692050016', 'Long Schlong Silver"s Menu'),

('4372050002', '6692050025', 'Long Schlong Silver"s Menu'),

('4372050003', '6692050013', 'Kai Xing"s Menu'),

('4372050003', '6692050020', 'Kai Xing"s Menu'),

('4372050004', '6692050006', 'Ho Jiak"s Menu'),

('4372050004', '6692050021', 'Ho Jiak"s Menu'),

('4372050005', '6692050001', 'Vasantham Prata"s Menu'),

('4372050006', '6692050015', 'HuoLong BubbleTea"s Menu'),

('4372050006', '6692050018', 'HuoLong BubbleTea"s Menu'),

('4372050007', '6692050005', 'Burger Queen"s Menu'),

('4372050008', '6692050012', 'Earthly Goodness"s Menu'),

('4372050009', '6692050008', 'Pui Pui Burger"s Menu'),

('4372050010', '6692050003', 'Fish & Chics" Menu'),

('4372050011', '6692050004', 'Ah Bee Yong Tow Foo"s Menu'),

('4372050011', '6692050023', 'Ah Bee Yong Tow Foo"s Menu'),

('4372050012', '6692050014', 'Yee Haw"s Menu'),

('4372050013', '6692050010', 'A-Yum Penyet"s Menu'),

('4372050013', '6692050019', 'A-Yum Penyet"s Menu'),

('4372050014', '6692050007', 'Makcik Nasi Lemak"s Menu'),

('4372050014', '6692050024', 'Makcik Nasi Lemak"s Menu'),

('4372050015', '6692050011', 'Oishi Raja"s Menu'),

('4372050015', '6692050022', 'Oishi Raja"s Menu');

**--Cuisine**

INSERT INTO Cuisine

VALUES

('7522050001', 'Halal'),

('7522050002', 'Chinese'),

('7522050003', 'Western'),

('7522050004', 'Indian'),

('7522050005', 'Malay'),

('7522050006', 'Taiwanese'),

('7522050007', 'Hainanese'),

('7522050008', 'Local'),

('7522050009', 'Fast Food'),

('7522050010', 'Healthy'),

('7522050011', 'Vegetarian'),

('7522050012', 'Non-Halal');

**--Promotion**

INSERT INTO Promotion

VALUES

('P5630001', '11:11', '11:11 Singles Day Offer', 0.11, 'Yes'),

('P5630002', 'VDay', 'Valentine"s Day Offer', 0.20, 'Yes'),

('P5630003', 'National Day', 'National Day Offer', 0.30, 'No'),

('P5630004', '88 CNY Promo', 'Chinese New Year Offer', 0.12, 'Yes'),

('P5630005', 'Celebrate Christmas', 'Christmas Day Offer', 0.20, 'Yes'),

('P5630006', 'Riding with Mum', 'Mother"s Day Offer', 0.25, 'Yes'),

('P5630007', '12:12 Sales', '12:12 Online Revolution Offer', 0.15, 'Yes'),

('P5630008', 'Hari Raya Sales', 'Hari Raya Offer', 0.20, 'No'),

('P5630009', 'Riding with Dad', 'Father"s Day Offer', 0.25, 'Yes'),

('P5630010', 'Vesak Sales', 'Vesak Day Offer', 0.20, 'No'),

('P5630011', 'Deepavali Sales', 'Deepavali Day Offer', 0.20, 'No'),

('P5630012', 'Happy Labour Day', 'Labour Day Offer', 0.25, 'Yes');

**--Enjoys**

INSERT INTO Enjoys

VALUES

('5402050001','P5630001'),

('5402050002','P5630003'),

('5402050003','P5630004'),

('5402050004','P5630004'),

('5402050005','P5630002'),

('5402050006','P5630006'),

('5402050007','P5630010'),

('5402050008','P5630008'),

('5402050009','P5630005'),

('5402050010','P5630009'),

('5402050011','P5630011'),

('5402050012','P5630002'),

('5402050013','P5630004'),

('5402050014','P5630005'),

('5402050015','P5630004'),

('5402050016','P5630004'),

('5402050017','P5630007'),

('5402050018','P5630007'),

('5402050019', 'P5630011'),

('5402050020','P5630012');

**--Runs**

INSERT INTO Runs

VALUES

('P5630011','6692050001',200),

('P5630004','6692050002',200),

('P5630005','6692050003',200),

('P5630004','6692050004',200),

('P5630005','6692050005',200),

('P5630004','6692050006',200),

('P5630008','6692050007',200),

('P5630012','6692050008',200),

('P5630004','6692050009',200),

('P5630008','6692050010',200),

('P5630011','6692050011',200),

('P5630004','6692050012',200),

('P5630004','6692050013',200),

('P5630012','6692050014',200),

('P5630003','6692050015',200),

('P5630009','6692050016',200),

('P5630004','6692050017',200),

('P5630003','6692050018',200),

('P5630008','6692050019',200),

('P5630004','6692050020',200),

('P5630004','6692050021',200),

('P5630011','6692050022',200),

('P5630004','6692050023',200),

('P5630008','6692050024', 200),

('P5630009','6692050025',200);

**--Belongs**

INSERT INTO Belongs

VALUES

('6692050001', '7522050001'),

('6692050001', '7522050004'),

('6692050001', '7522050008'),

('6692050002', '7522050001'),

('6692050002', '7522050003'),

('6692050002', '7522050009'),

('6692050003', '7522050001'),

('6692050003', '7522050003'),

('6692050004', '7522050002'),

('6692050004', '7522050008'),

('6692050004', '7522050012'),

('6692050005', '7522050001'),

('6692050005', '7522050003'),

('6692050005', '7522050009'),

('6692050006', '7522050002'),

('6692050006', '7522050012'),

('6692050007', '7522050001'),

('6692050007', '7522050005'),

('6692050007', '7522050008'),

('6692050008', '7522050001'),

('6692050008', '7522050003'),

('6692050008', '7522050009'),

('6692050009', '7522050001'),

('6692050009', '7522050003'),

('6692050009', '7522050009'),

('6692050010', '7522050001'),

('6692050010', '7522050005'),

('6692050010', '7522050008'),

('6692050011', '7522050001'),

('6692050011', '7522050010'),

('6692050011', '7522050011'),

('6692050012', '7522050001'),

('6692050012', '7522050010'),

('6692050012', '7522050011'),

('6692050013', '7522050001'),

('6692050013', '7522050002'),

('6692050013', '7522050007'),

('6692050013', '7522050008'),

('6692050014', '7522050001'),

('6692050014', '7522050003'),

('6692050014', '7522050008'),

('6692050015', '7522050006'),

('6692050016', '7522050001'),

('6692050016', '7522050003'),

('6692050016', '7522050009'),

('6692050017', '7522050001'),

('6692050017', '7522050003'),

('6692050017', '7522050009'),

('6692050018', '7522050006'),

('6692050019', '7522050001'),

('6692050019', '7522050005'),

('6692050019', '7522050008'),

('6692050020', '7522050001'),

('6692050020', '7522050002'),

('6692050020', '7522050007'),

('6692050020', '7522050008'),

('6692050021', '7522050002'),

('6692050021', '7522050012'),

('6692050022', '7522050001'),

('6692050022', '7522050010'),

('6692050022', '7522050011'),

('6692050023', '7522050002'),

('6692050023', '7522050008'),

('6692050023', '7522050012'),

('6692050024', '7522050001'),

('6692050024', '7522050005'),

('6692050024', '7522050008'),

('6692050025', '7522050001'),

('6692050025', '7522050003'),

('6692050025', '7522050009');

**--Has**

INSERT INTO Has

VALUES

('4372050005','6692050001',7),

('4372050001','6692050002',1),

('4372050010','6692050003',3),

('4372050011','6692050004',11),

('4372050007','6692050005',2),

('4372050004','6692050006',6),

('4372050014','6692050007',13),

('4372050009','6692050008',4),

('4372050001','6692050009',1),

('4372050013','6692050010',12),

('4372050015','6692050011',15),

('4372050008','6692050012',9),

('4372050003','6692050013',5),

('4372050012','6692050014',14),

('4372050006','6692050015',8),

('4372050002','6692050016',10),

('4372050001','6692050017',1),

('4372050006','6692050018',8),

('4372050013','6692050019',12),

('4372050003','6692050020',5),

('4372050004','6692050021',6),

('4372050015','6692050022',15),

('4372050011','6692050023',11),

('4372050014','6692050024',13),

('4372050002','6692050025',10);

**--Purchase**

INSERT INTO Purchase

VALUES

('5342050001', '2021/1/11', '5482050001',2),

('5342050002', '2021/1/11', '5482050003',1),

('5342050003', '2021/1/20', '5482050007',1),

('5342050004', '2021/1/15', '5482050009',1),

('5342050005', '2021/1/14', '5482050001',3),

('5342050006', '2021/1/15', '5482050010',1),

('5342050007', '2021/1/13', '5482050013',1),

('5342050008', '2021/1/12', '5482050007',1),

('5342050009', '2021/1/19', '5482050015',2),

('5342050010', '2021/1/18', '5482050015',2),

('5342050011', '2021/1/17', '5482050004',3),

('5342050012', '2021/1/16', '5482050001',2),

('5342050013', '2021/1/20', '5482050013',1),

('5342050014', '2021/1/12', '5482050001',4),

('5342050015', '2021/1/16', '5482050012',1),

('5342050016', '2021/1/18', '5482050011',1),

('5342050017', '2021/1/21', '5482050014',1),

('5342050018', '2021/1/10', '5482050005',3),

('5342050019', '2021/1/15', '5482050006',2),

('5342050020', '2021/1/22', '5482050015',2);

**--Achieve**

INSERT INTO Achieve

VALUES

('5342050006', '2020/12/29', '5272050001'),

('5342050006', '2020/12/29', '5272050007'),

('5342050006', '2020/12/29', '5272050005'),

('5342050013', '2020/12/29', '5272050003'),

('5342050018', '2020/12/29', '5272050003'),

('5342050017', '2020/12/29', '5272050003'),

('5342050009', '2020/12/29', '5272050009'),

('5342050001', '2020/12/29', '5272050010'),

('5342050003', '2020/12/29', '5272050005'),

('5342050019', '2020/12/29', '5272050005');

**--Assign**

INSERT INTO Assign

VALUES

('5402050001','5342050002', 'Delivering'),

('5402050004','5342050004', 'Delivered'),

('5402050006', '5342050005', 'Delivering'),

('5402050009','5342050013', 'Delivering'),

('5402050010', '5342050014', 'Delivering'),

('5402050011', '5342050006', 'Delivered'),

('5402050014', '5342050003', 'Delivering'),

('5402050015','5342050018', 'Delivering'),

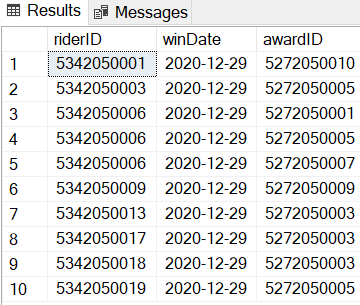
('5402050017','5342050017', 'Delivering'),

('5402050018','5342050006', 'Delivered'),

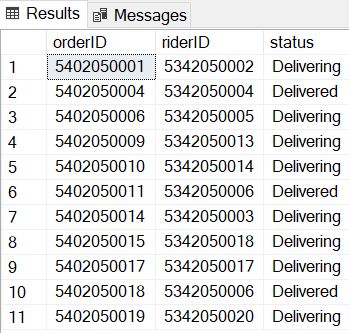
('5402050019','5342050020', 'Delivering');

# **Display all tables**

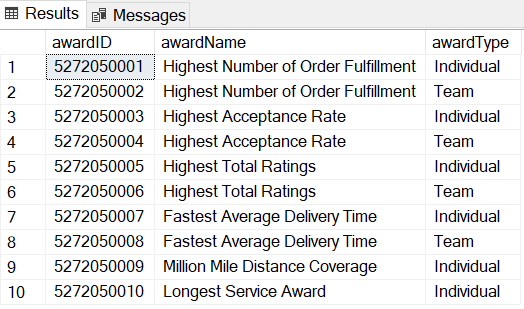
**SELECT \* FROM Achieve**

****

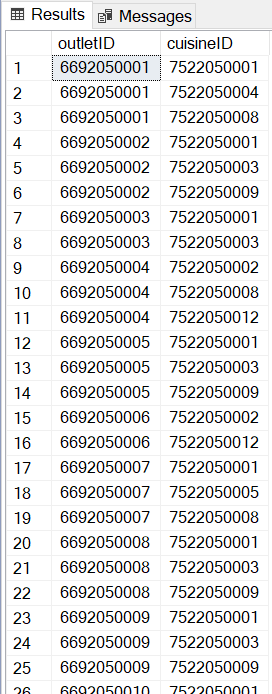
**SELECT \* FROM Assign**

****

**SELECT \* FROM Award**

****

**SELECT \* FROM Belongs**

****

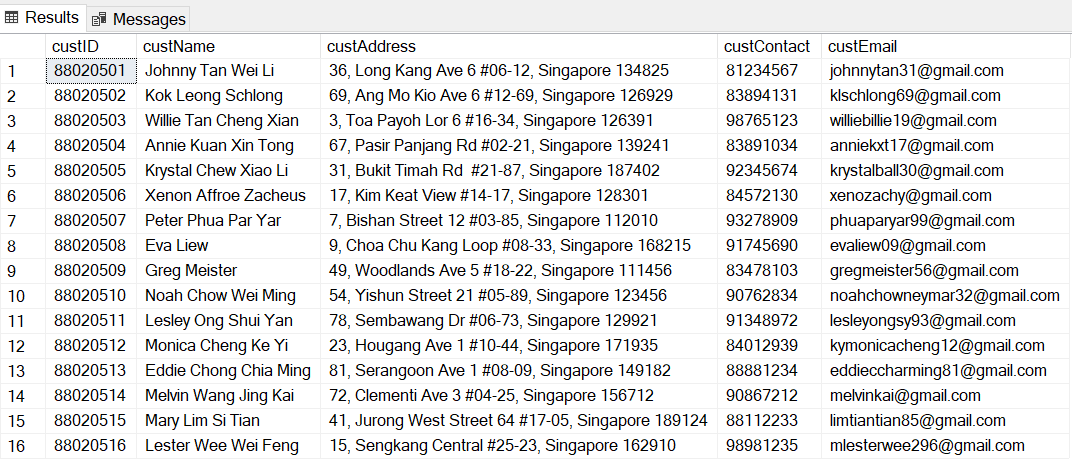
**SELECT \* FROM Business**

****

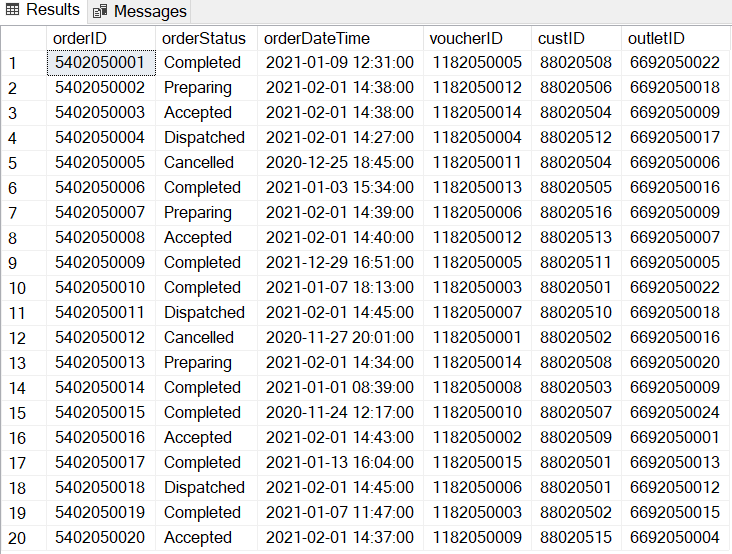
**SELECT \* FROM Cuisine**

****

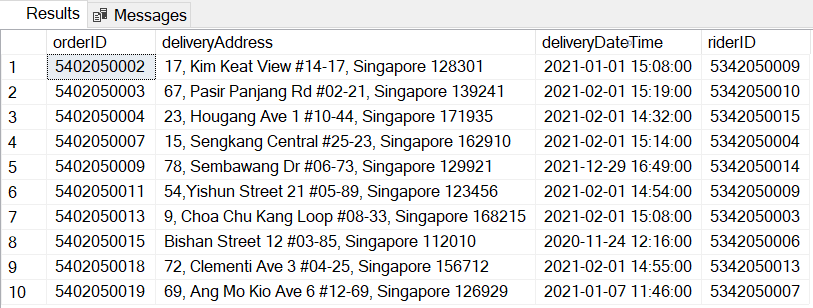
**SELECT \* FROM Customer**

****

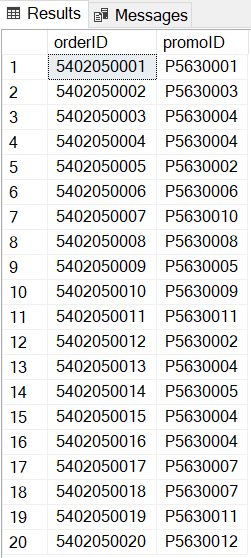
**SELECT \* FROM CustOrder**

****

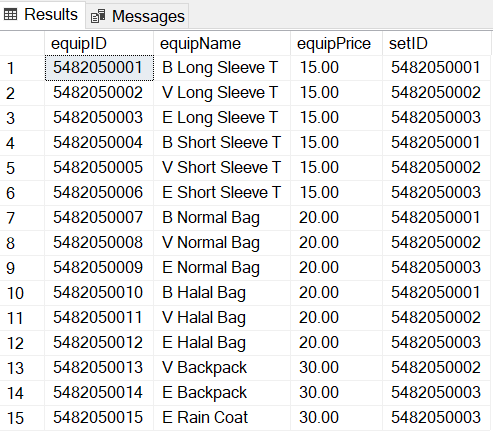
**SELECT \* FROM Delivery**

****

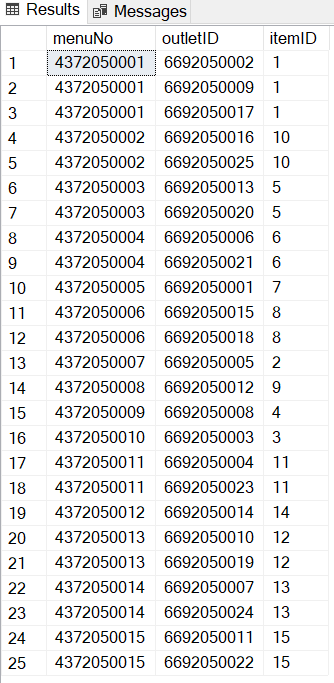
**SELECT \* FROM Enjoys**

****

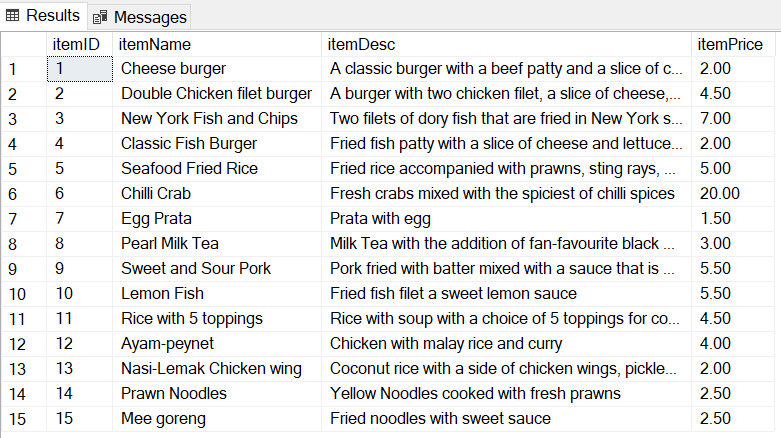
**SELECT \* FROM Equipment**

****

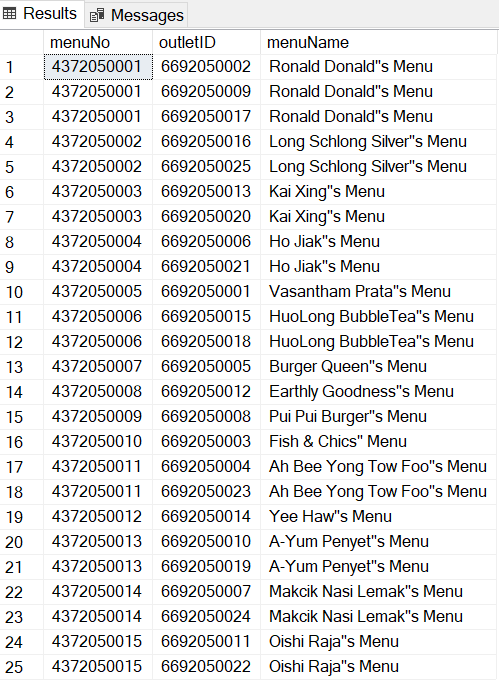
**SELECT \* FROM Has**

****

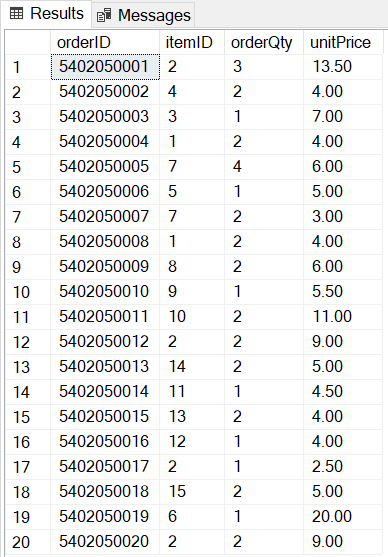
**SELECT \* FROM Items**

****

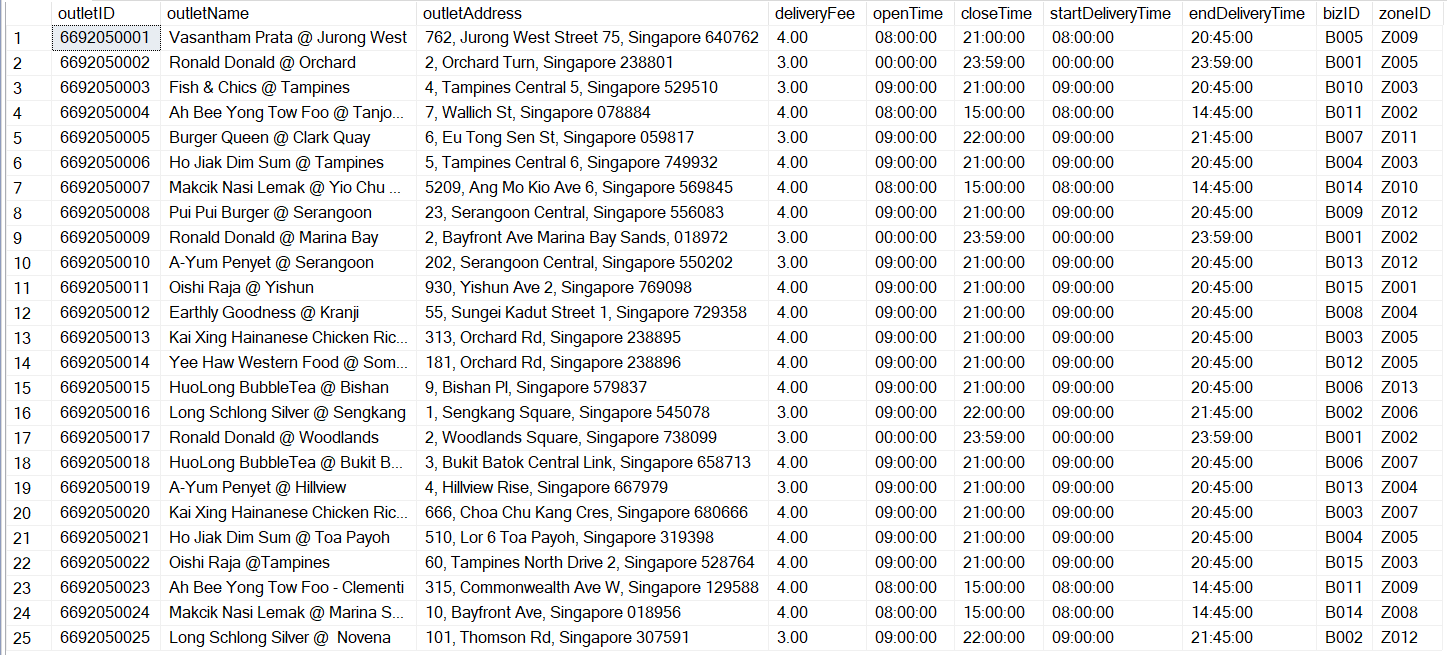
**SELECT \* FROM Menu**

****

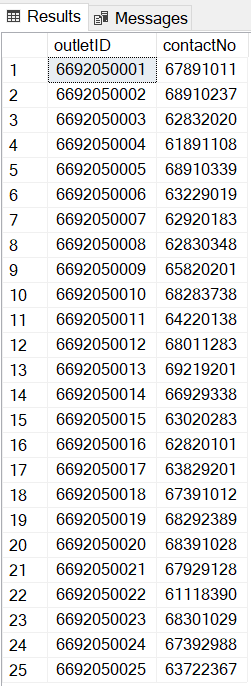
**SELECT \* FROM OrderItem**

****

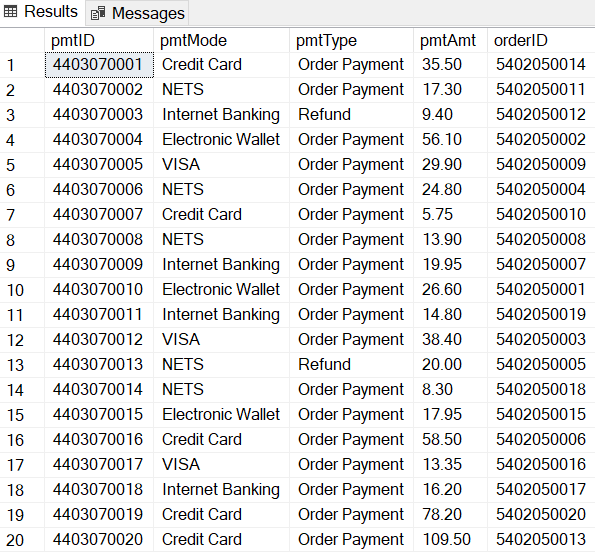
**SELECT \* FROM Outlet**

****

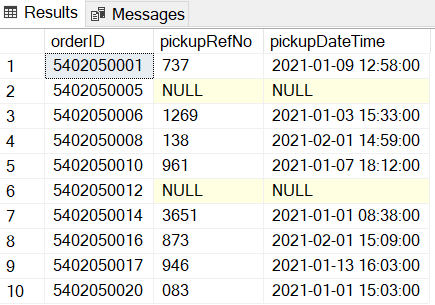
**SELECT \* FROM OutletContact**

****

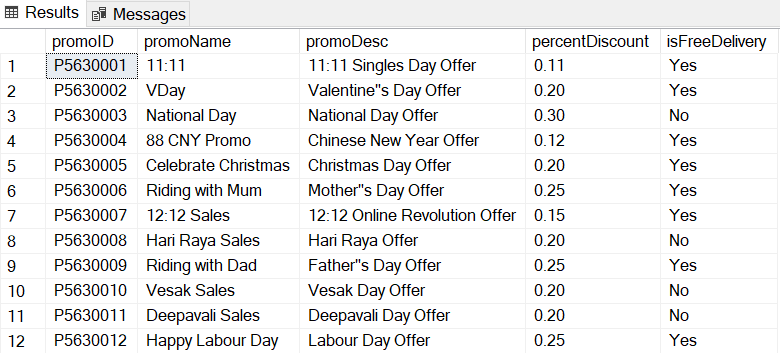
**SELECT \* FROM Payment**

****

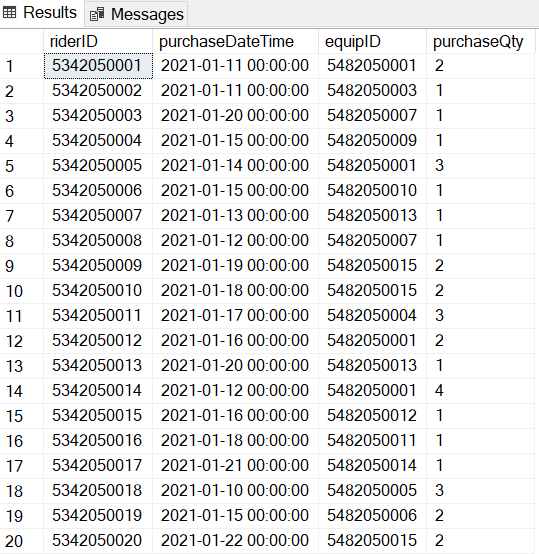
**SELECT \* FROM PickUp**

****

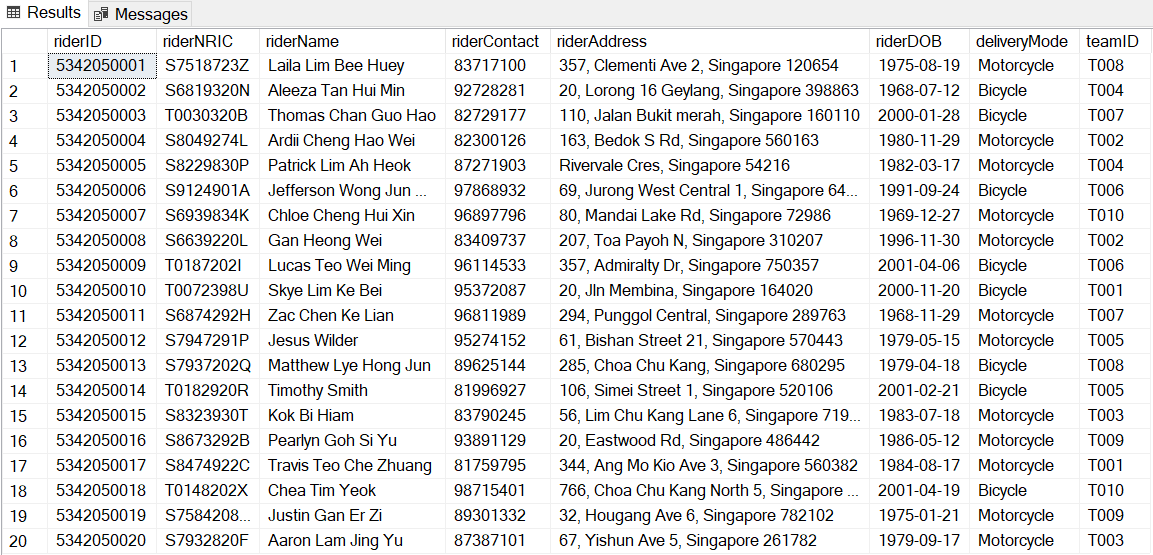
**SELECT \* FROM Promotions**

****

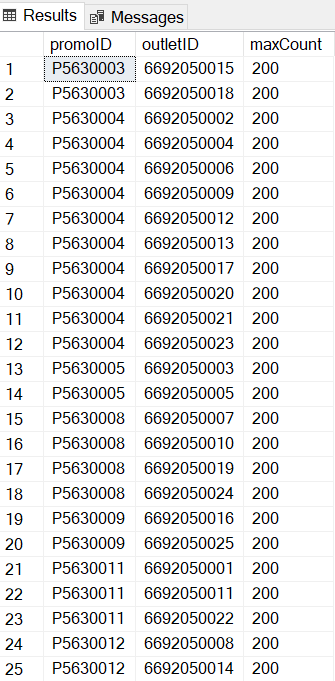
**SELECT \* FROM Purchase**

****

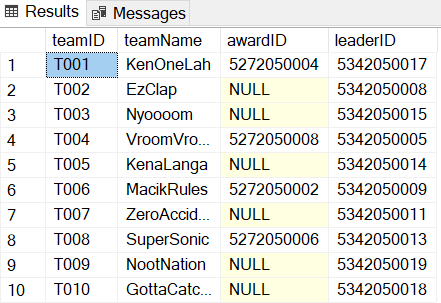
**SELECT \* FROM Rider**

****

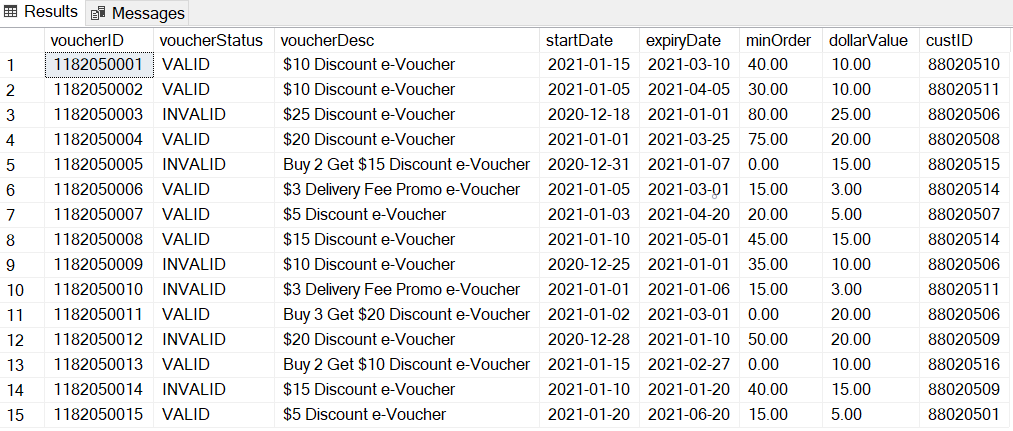
**SELECT \* FROM Runs**

****

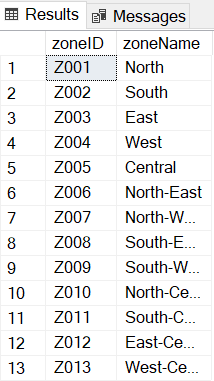
**SELECT \* FROM Team**

****

**SELECT \* FROM Voucher**

****

**SELECT \* FROM Zone**

****

# **SQL Queries**

## Edward’s Queries:

1. How many Individuals in team''T006'' won individual awards

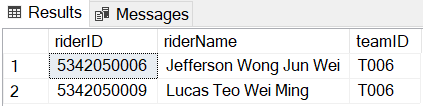
* SELECT riderID, riderName,teamID FROM Rider

WHERE riderID in

(SELECT a.riderID FROM Achieve a INNER JOIN rider r

ON a.riderID = r.riderID

WHERE teamID = 'T006')



1. Out of all the orders, which item was ordered more than or equals to 4 times.

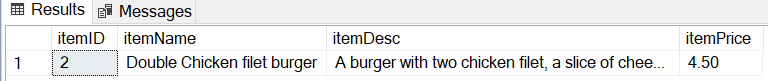
* SELECT \* FROM Item i

WHERE itemID in

( SELECT itemID FROM OrderItem

GROUP BY itemID

HAVING COUNT(itemID)>=4)



1. Out of all the outlets, which Cuisine belongs to less than 3 Outlets

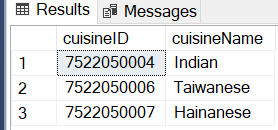
* SELECT \* FROM Cuisine

WHERE cuisineID in

(SELECT cuisineID FROM Belongs

GROUP BY cuisineID

HAVING COUNT(cuisineID) < 3)



## Molei’s Queries:

1. List the orderID, Item name and Order Quantity for all orders made by the customer 'Eva Liew' and display the results in ascending order of order quantity.  
   SELECT o.orderID, itemName, orderQty

FROM OrderItem o

INNER JOIN CustOrder co

ON co.orderID=o.orderID

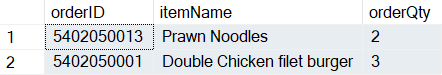
INNER JOIN Item i

ON o.itemID=i.itemID

INNER JOIN Customer c

ON co.custID=c.custID

WHERE custName = 'Eva Liew'

ORDER BY orderQty ASC  


1. List the riderID, rider name and delivery address of riders who made delivery in January 2021.

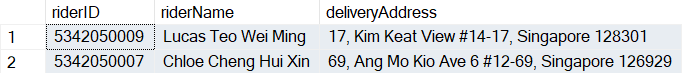
SELECT r.riderID,riderName,deliveryAddress

FROM Rider r

INNER JOIN Delivery d

ON r.riderID=d.riderID

WHERE deliveryDateTime between '2021-01-01 00:00:00' and '2021-01-31 23:59:59'



1. List the delivery mode and the number of orders delivered by each delivery mode in February 2021.

SELECT deliveryMode, COUNT(orderID) as "Number of orders delivered"

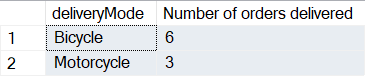
FROM Rider r

INNER JOIN Delivery d

ON r.riderID=d.riderID

WHERE deliveryDateTime between '2021-01-01 00:00:00' and '2021-12-31 23:59:59'

GROUP BY deliveryMode



## Jeremy’s Queries:

1. How much has each team spent on equipment in total?

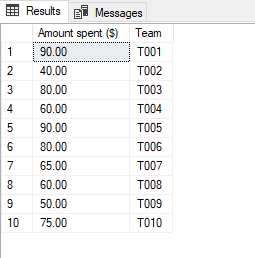
SELECT SUM(e.equipPrice\*p.PurchaseQty) AS 'Amount spent ($)',r.teamID AS 'Team'

FROM Equipment e

INNER JOIN Purchase p ON e.equipID = p.equipID

INNER JOIN Rider r ON r.RiderID = p.riderID

GROUP BY r.teamID



1. Which region has the least pickups?

SELECT COUNT(p.orderID) AS 'No. of pickups', z.zoneName AS 'Region' FROM Pickup p

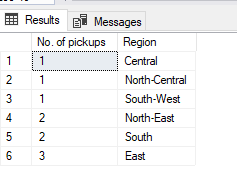
INNER JOIN CustOrder cO ON p.OrderID = cO.OrderID

INNER JOIN Outlet o ON cO.outletID = o.outletID

INNER JOIN Zone z ON o.ZoneID = z.ZoneID

GROUP BY z.zoneName

ORDER BY COUNT(p.orderID) ASC



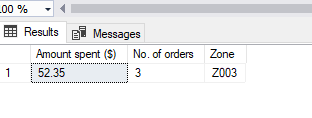
1. What is the total amount spent and number of orders from outlets in zone “Z003”?

SELECT SUM(p.pmtAmt) AS 'Amount spent ($)', COUNT(p.pmtID) AS 'No. of orders', o.zoneID AS 'Zone' FROM Payment p

INNER JOIN CustOrder cO ON p.orderID = cO.orderID

INNER JOIN Outlet o ON o.outletID = cO.outletID

WHERE o.zoneID = 'Z003' GROUP BY o.zoneID



## Jun Han’s Queries:

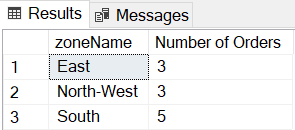
1. Which food outlet has at least 1 order in January 2021?

* SELECT o.outletName, COUNT(c.orderID) AS 'Number of Orders' FROM Outlet o INNER JOIN CustOrder c ON c.outletID = o.outletID WHERE DATEPART(YEAR,c.orderDateTime) = 2021 AND DATEPART(MONTH,c.orderDateTime) = 1 GROUP BY o.outletName



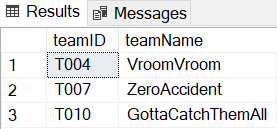
1. Which zone has 3 or more orders and how many orders are there for that zone?

* SELECT z.zoneName, COUNT(c.orderID) AS 'Number of Orders' FROM Zone z INNER JOIN Outlet o ON z.zoneID = o.zoneID INNER JOIN CustOrder c ON o.outletID = c.outletID GROUP BY z.zoneName HAVING COUNT(c.orderID)>=3



1. Which team has riders that are above the age of 50?

* SELECT DISTINCT(t.teamID), t.teamName FROM Team t INNER JOIN Rider ON t.teamID = r.teamID WHERE DATEDIFF(YEAR,r.riderDOB, GETDATE()) >50



## Evan’s Queries:

1. How many orders are delivered by riders at the age of 25 and below?

* Select r.riderName, DATEDIFF(YEAR, r.riderDOB, GETDATE()) AS 'Age', COUNT(d.orderID) AS 'Number of Orders' FROM Rider r INNER JOIN Delivery d

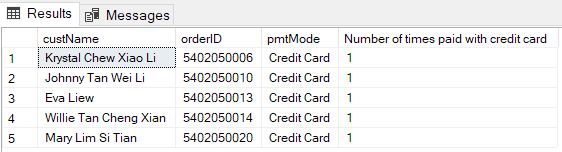
ON r.riderID = d.riderID WHERE DATEDIFF(YEAR, r.riderDOB, GETDATE()) <= 25 GROUP BY r.riderName, DATEDIFF(YEAR, r.riderDOB, GETDATE())



2. Which customers made payment using ‘Credit Card’ and how many times did they pay with it?

* SELECT c.custName, o.orderID, p.pmtMode, COUNT(p.pmtID) AS 'Number of times paid with credit card' FROM Customer c INNER JOIN CustOrder o

ON c.custID = o.custID INNER JOIN Payment p ON o.orderID = p.orderID WHERE p.pmtMode = 'Credit Card' GROUP BY c.custName, o.orderID, p.pmtMode



3. Display the total number of customers and orders MokeFood had and their total earnings for the month of January 2021

* SELECT COUNT(DISTINCT(c.custName)) AS 'Number of customers', COUNT(co.orderID) AS 'Number of orders', SUM(p.pmtAmt) AS 'Total Earnings ($)'

FROM Customer c INNER JOIN CustOrder co ON c.custID = co.custID

INNER JOIN Payment p ON co.orderID = p.orderID WHERE DATEPART(MONTH, co.orderDateTime) = '1' AND DATEPART(YEAR, co.orderDateTime) = '2021'

AND p.pmtType = 'Order Payment'

