

**Module Title Database Design and Development**

**Assignment Title Shwe Li Restaurant**

**Examination Cycle Spring 2020**

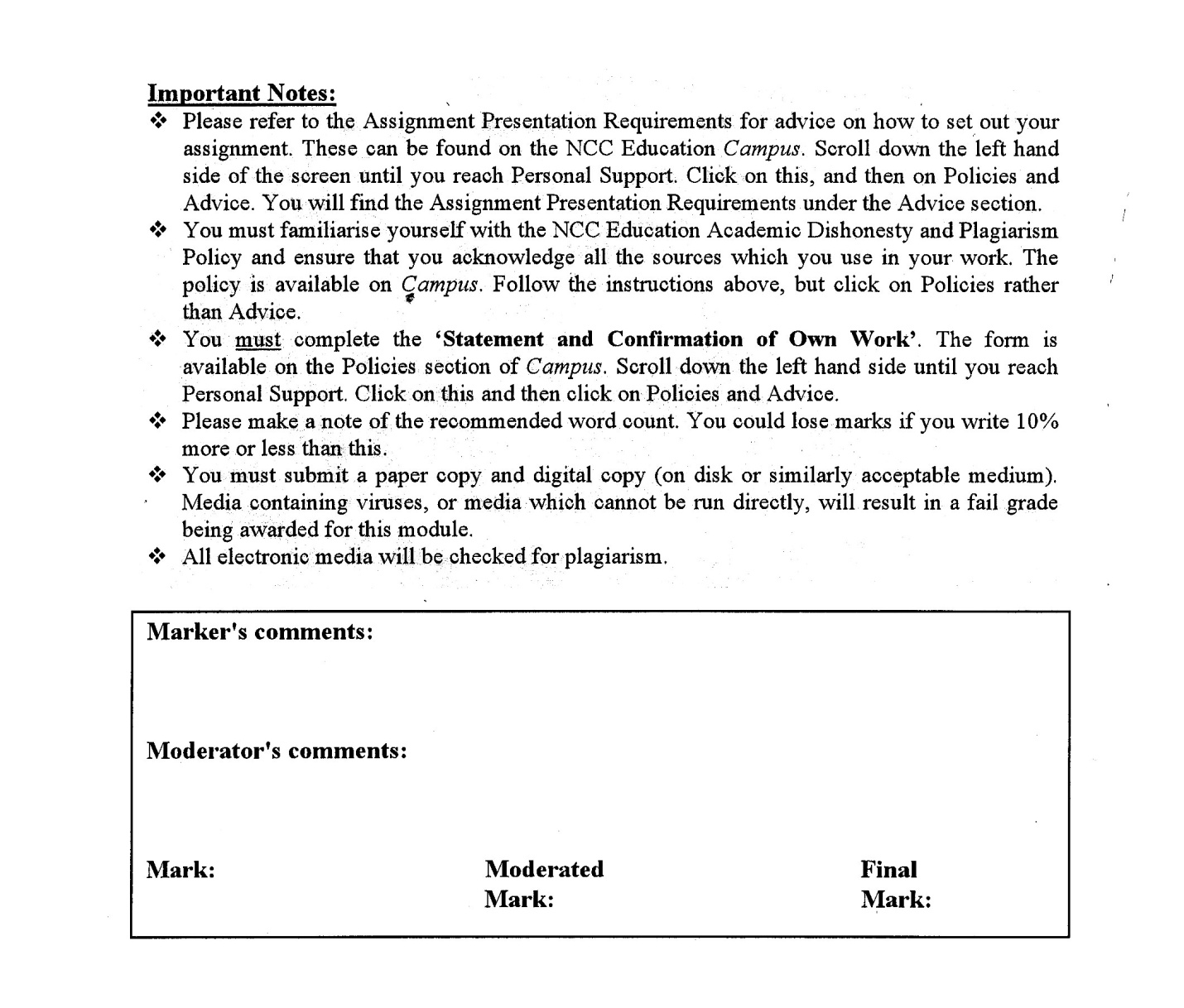
**Candidate Name PYAE THUTA**

**Candidate No**

**00185308**

**Centre Name KMD Computer Centre (Yangon)**

**Submission Date: 31.1.2020**

<https://www.facebook.com/pg/HTML5CSS3TUTORIALS/about/>

****

**Statement and Confirmation of Own Work**

**Programmed/Qualification name: Level 5 Diploma in Computing IT**

*All NCC Education assessed assignments submitted by students must have this statement as the cover page or it will not be accepted for marking. Please ensure that this statement is either firmly attached to the cover of the assignment or electronically inserted into the front of the assignment.*

**Student declaration**

I have read and understood NCC Education’s Policy on Academic Dishonesty and Plagiarism.

I can confirm the following details:

**Student ID/Registration number : 00185308**

**Name : PYAE THUTA**

**Centre Name : KMD Computer Centre (Yangon)**

**Module Name : Database Design and Development**

**Module Leader : TAH MLAH MOO**

**Number of words : (2280) words**

I confirm that this is my own work and that I have not plagiarized any part of it. I have also noted the assessment criteria and pass mark for assignments.

**Due Date : 31.1.2020**

**Student Signature : thuta**

**& Submitted Date : 31.1.2020**

**Table of Contents**

Contents

[Introduction 4](#_Toc30439458)

[Task 1 5](#_Toc30439459)

[Description of Business 6](#_Toc30439460)

[Task 2 11](#_Toc30439461)

[Entity Relationship Diagram 12](#_Toc30439462)

[Purpose of normalization 13](#_Toc30439463)

[How Normalization solves the problems of update anomalies 16](#_Toc30439464)

[Data Dictionary 17](#_Toc30439465)

[The integrity constraints 17](#_Toc30439466)

[Task 3 20](#_Toc30439467)

[Scripts To Create table structures 21](#_Toc30439468)

[Coding And View 21](#_Toc30439469)

[Summary 33](#_Toc30439470)

[Task 4 34](#_Toc30439471)

[Data Population 35](#_Toc30439472)

[Summary 47](#_Toc30439473)

[Task 5 48](#_Toc30439474)

[Enhancement 49](#_Toc30439475)

[Task 6 56](#_Toc30439476)

[SQL reports 57](#_Toc30439477)

[Query 1 57](#_Toc30439478)

[Query 2 58](#_Toc30439479)

[Query 3 59](#_Toc30439480)

[Query 4 60](#_Toc30439481)

[Query 5 61](#_Toc30439482)

[Task 7 62](#_Toc30439483)

[Task 8 66](#_Toc30439484)

[References: 68](#_Toc30439485)

[Candidate Checklist 69](#_Toc30439486)

# 

# Introduction

Firstly, An organization has to be chosen to research and base this database project. The analysis about the transaction between that organization has to be made.

The restaurant system is chosen for this assignment because it is more familiar than other system so that the analysis about the business is done easily. The entity relationship diagram has to be produced after normalization. Then, the SQL scripts have to be written to a DBMS for chosen system.

“Shwe **Li**”, a popular Chinese restaurant based in Thingangyun, Yangon, Myanmar, is chosen to complete the assignment. The restaurant is providing various Chinese foods to their customers.

The research on this restaurant will need to be made to analyze the daily processes of restaurant and data flow between these processes.

# Task 1

## Description of Business

**Scenario**

“**Shwe Li**” is a popular Chinese restaurant based in Thingangyun, Yangon, Myanmar. The restaurant is providing various Chinese foods to their customers. Being a popular Chinese restaurant, it operates in an environment in competition with other Chinese restaurants in Yangon.

There are 7 chefs in “Shwe Li” restaurant. A chef has to cook at least 3 kinds of meal. A meal is cooked only by a particular chef. Meals are specified by types : Rice, Pork, Chicken, Seafood and soup.

There are table number one to thirty in the restaurant. Customers can book tables by phone or online. A customer can book more than one table while a table can be booked by only one customer. Tables are specified by table size. When a customer books table, customer name, phone number and date must be recorded as customer data.

The orders made by customers are specified by table numbers not by customer names. A table can have more than one order. But an order can be made from only one table.

An order can have one or more meals and a meal can be ordered many times. Whenever an order is made, the total price is calculated, and payment must be made according to the calculated cost.

The restaurant used to process with paper-based system to store the detail information in their business. Using paper-based system, they faced with data loss and delays in processing their business . Now, the administrators of “**Shwe Li**” want a database system that can help them in managing their meals, staffs and orders for their daily processes in the restaurant. The database system must be able to process the works mentioned above.

**Document 1. Chef Allocated to Meals**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Meal ID** | **Meal Name** | **Meal Type** | **Chef ID** | **Chef Name** |
| M1 | Sweet and Sour Pork | Pork | C1 | Dave Kuma |
| M2 | [Spicy Hoisin Pork Stir-Fry](https://www.thespruceeats.com/spicy-hoisin-pork-stir-fry-recipe-694832) | Pork | C1 | Dave Kuma |
| M3 | Pork Onion | Pork | C3 | Robert Wayne |
| M4 | Chicken Masala | Chicken | C3 | Robert Wayne |
| M5 | Fried Pork Soup | Soup | C2 | Anton |
| M6 | Seafood Tom Yum | Soup | C4 | Caroline |
| M7 | Chicken with Black Sesame | Soup | C4 | Caroline |
| M8 | Lobster Sauce | Seafood | C6 | Anna |
| M9 | Garlic Scallops | Seafood | C5 | Scott Lang |
| M10 | Honey Prawns | Seafood | C5 | Scott Lang |
| M11 | Plain Rice | Rice | C7 | Smith |
| M12 | Fried Rice | Rice | C7 | Smith |

From Document 1, it is possible to derive following tables.

* Meal
* Meal Type
* Chef

**Document 2. Managers Allocated to Staffs**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Staff ID** | **Staff Type** | **Staff name** | **Manager ID** | **Manager name** |
| S1 | Waiter | Harry Wood | M1 | Jimmy Brown |
| S2 | Waiter | Alan Wayne | M1 | Jimmy Brown |
| S3 | Waiter | Louis Lang | M1 | Jimmy Brown |
| S4 | Chef Assistant | Bryan Lee | M2 | David Jones |
| S5 | Chef Assistant | James | M2 | David Jones |
| S6 | Cleaner | Sam Lavine | M3 | Carson |
| S7 | Cleaner | Mary Ling | M3 | Carson |
| S8 | Cleaner | Suzan Holland | M3 | Carson |
| S9 | Accountant | Ivan Coral | M4 | Thomas Hills |
| S10 | Accountant | Emily | M4 | Thomas Hills |
| S11 | Accountant | Walter | M4 | Thomas Hills |
| S12 | Waiter | Jonny Dave | M1 | Jimmy Brown |
| S13 | Waiter | Carter Smith | M1 | Jimmy Brown |
| S14 | Chef Assistant | Antony | M2 | David Jones |
| S15 | Chef Assistant | Alex Moore | M2 | David Jones |

From Document 2. it is possible to derive following tables.

* Staff
* Staff Type
* Manager

**Document 3. Customers booking tables**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Customer Name** | **Customer Phone No** | **Table number** | **Date** | **Time** |
| Chris Daniel | 09223166580 | 3 | 23.12.2019 | 6:00 PM |
| Alva Max | 01455123646 | 6 | 23.12.2019 | 7:30 PM |
| Alva Max | 01455123642 | 7 | 23.12.2019 | 7:30 PM |
| Autumn Smith | 09781211341 | 4 | 23.12.2019 | 5:30 PM |
| Scott Biddle | 09443145218 | 8 | 24.12.2019 | 7:30 PM |
| Yasir Chowdhury | 09784673739 | 2 | 24.12.2019 | 6:30 PM |
| Victor | 09783737781 | 1 | 24.12.2019 | 8:00 PM |
| Victor | 09783737781 | 1 | 24.12.2019 | 8:00 PM |
| Anton Carter | 09254121199 | 3 | 24.12.2019 | 7:30 PM |
| Matthew | 09422138765 | 9 | 25.12.2019 | 7:00 PM |
| Matthew | 09422138765 | 3 | 25.12.2019 | 7:00 PM |
| Bryan Baker | 09211345889 | 7 | 25.12.2019 | 7:30 PM |
| Jonny Miller | 09796029284 | 5 | 26.12.2019 | 7:00 PM |
| Jonny Miller | 09796029284 | 6 | 26.12.2019 | 7:00 PM |
| Jonny Miller | 09796029284 | 7 | 26.12.2019 | 7:00 PM |

From Document 3. it is possible to derive following tables.

* Customer
* Table
* Book

**Document 4. Customer’s orders from tables**

|  |  |  |
| --- | --- | --- |
| **Meal name** | **Table No** | **Quantity** |
| Mu Shu Pork | 3 | 1 |
| Plain Rice | 3 | 4 |
| Boneless Chili Chicken | 3 | 1 |
| Spicy Prawn Soup | 3 | 1 |
| Seafood Tom Yum | 6 | 1 |
| Vegetables steam rice | 6 | 2 |
| Chicken Broccoli | 4 | 1 |
| Boneless chili chicken | 4 | 1 |
| Plain Rice | 4 | 3 |
| Vegetables steam rice | 4 | 1 |

From Document 4. it is possible to derive following table.

* Order

# Task 2

## Entity Relationship Diagram



## Purpose of normalization

Normalization is one of the processes of organizing the data in database system. Normalization is mainly used for minimizing the redundancy from a relation or sets of relations in database system. It converts the large tables in the database system into small tables for the more efficient account of inserting, updating, deleting and selecting the data stored in the database.

Normalization has two main objectives

1. Eliminating redundant data
2. Ensuring data dependencies to make sense

Eliminating redundant data is the process of storing the same data in one or more tables. Ensuring data dependencies is the process of storing only the related data in the table. By normalizing the data properly, it allows to use the storage space efficiently, to understand the transaction of business more clearly, to reduce the repetition of data and to have easier account of maintaining database. Moreover, a good normalization helps to process of searching, sorting and creating indexes faster.

A proper normalization can help to avoid data duplicity and to store the similar records in the same table in order to have the snow flake schema which make data more readable.

**3rd Normal Form**

MealID (PK)

MealName

Price

MealTypeID (FK)

ChefID (FK)

ChefID (PK)

ChefName

ChefPhone

ChefEmail

Salary

MealID (PK) (FK)

OrderID (PK) (FK)

OrerID (PK)

TableNumber (Fk)

Total Price

TableNo (PK)

NoOfPeople

TableID (PK)(FK)

BookID (PK)(FK)

BookingID (PK)

BookingDate

BookingTime

CustomerID (FK)

CustomerID (PK)

CustomerName

CusPhoneNo

StaffID (PK)

StaffName

StaffEmail

StaffPhone

StaffTypeID (FK)

ManagerID (FK)

SatffTypeID (PK)

StaffType

ManagerID (PK)

ManagerName

ManagerEmail

ManagerPhone

Salary

According to normalization, 12 entity is resulted.

**Entity Relationship Diagram**

Manager

Staff

StaffType

Order

Table

MealOrder

BookTable

Meal

MealType

Book

Chef

Customer

## How Normalization solves the problems of update anomalies

Update Anomalies : In this restaurant system, staff are divided by staff types. Each staff type has a respective manager. So, changing the manager of a staff type means changing the manager of every staffs in that staff type. If a manager for a staff type is needed to update, it is needed to be updated the manager of that staffs on every rows. To overcome this issue, the staff and staff type are split into two separates tables so that the twos data are stored separately in their respective tables. This two tables are clearly resulted from normalization. The manager of every staff type can be easily updated without the need of updating in every row of staff table.

In this way, normalization solves the problem of update anomalies in this restaurant system.

## Data **Dictionary**

## The integrity constraints

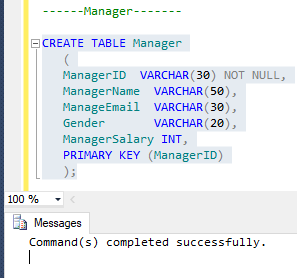
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute Name/ Key** | **Data Type** | **Length** | **Constraints** | **Description** |
| **Meal** |  |  |  |  |
| MealID | Varchar | 30 | PRIMARY KEY | Unique ID for Meal |
| MealName | Varchar | 50 |  | Meal’s name |
| Price | Integer |  |  | Price of meal |
| MealTypeID | Varchar | 30 | FOREIGN KEY | Unique ID for Meal Type |
| ChefID | Varchar | 30 | FOREIGN KEY | Unique ID for chef |
|  |  |  |  |  |
| **MealType** |  |  |  |  |
| MealTypeID | Varchar | 30 | PRIMARY KEY | Unique ID for meal |
| MealTypeName | Varchar | 50 |  | Name of Meal |
|  |  |  |  |  |
| **Chef** |  |  |  |  |
| ChefID | Varchar | 30 | PRIMARY KEY | Unique ID for chef |
| ChefName | Varchar | 50 |  | Chef’s name |
| ChefEmail | Varchar | 30 |  | Chef’s phone number |
| Gender | Varchar | 20 |  | Chef’s email address |
| ChefSalary | Integer |  |  | Salary of staff |
|  |  |  |  |  |
| **MealOrder** |  |  |  |  |
| MealID | Varchar | 30 | PRIMARY KEY / FOREIGN KRY | Unique ID for Meal |
| OrderID | Varchar | 30 | PRIMARY KEY/ FOREIGN KEY | Unique ID for Order |
| MealQuantity | Integer |  |  |  |
|  |  |  |  |  |
| **Order\_** |  |  |  |  |
| OrderID | Varchar | 30 | PRIMARY KEY | Unique ID for Order |
| OrderDate | DATE |  |  | The date when order is made |
| StaffID | Varchar | 30 | FOREIGN KEY | Unique ID for Staff |
| TableNumber | Integer |  | FOREIGN KEY | Unique number for table |
|  |  |  |  |  |
| **Table\_** |  |  |  |  |
| TableNo | Integer |  | PRIMARY KEY | Unique number for table |
| NoOfPeople | Integer |  |  | The no of people in the table |
|  |  |  |  |  |
| **BookTable** |  |  |  |  |
| TableNumber | Integer |  | PRIMARY KEY/ FOREIGN KEY | Unique number for table |
| BookID | Varchar | 30 | PRIMARY KEY/ FOREIGN KEY | Unique ID for booking |
|  |  |  |  |  |
| **Book** |  |  |  |  |
| BookingID | Varchar | 30 | PRIMARY KEY | Unique ID for booking |
| BookingDate | DATE |  |  | Date of booking |
| BookingTime | TIME |  |  | Time of booking |
| CustomerID | Varchar | 30 | FOREIGN KEY | Unique ID for Customer |
|  |  |  |  |  |
| **Customer** |  |  |  |  |
| CustomerID | Varchar | 30 | PRIMARY KEY | Unique ID for customer |
| CustomerName | Varchar | 50 |  | Customer’s name |
| CustomerPhone | Varchar | 30 |  | Customer’s Phone number |
|  |  |  |  |  |
| **Staff** |  |  |  |  |
| StaffID | Varchar | 30 | PRIMARY KEY | Unique ID for Staff |
| StaffName | Varchar | 50 |  | Staff’s name |
| StaffPhone | Varchar | 30 |  | Staff’s Phone number |
| StaffGender | Varchar | 20 |  | Gender of Staff |
| StaffTypeID | Varchar | 30 | FOREIGN KEY | Unique ID for Staff Type |
| ManagerID | Varchar | 30 | FOREIGN KEY | Unique ID for Manager |
|  |  |  |  |  |
| **StaffType** |  |  |  |  |
| StaffTypeID | Varchar | 30 | PRIMARY KEY | Unique ID for Staff Type |
| StaffTypeName | Varchar | 50 |  | Name of staff type |
| StaffTypeSalary | Integer |  |  |  |
|  |  |  |  |  |
| **Manager** |  |  |  |  |
| ManagerID | Varchar | 30 | PRIMARY KEY | Unique ID for Manager |
| ManagerName | Varchar | 50 |  | Name of manager |
| ManagerEmail | Varchar | 30 |  | Email of manager |
| Gender | Varchar | 20 |  | Manager’s Gender |
| ManagerSalary | Integer |  |  | Manager’s salary |

# Task 3

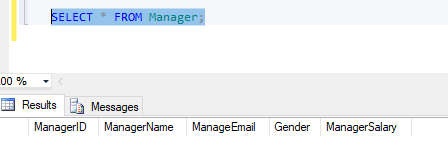
## Scripts To Create table structures

### Coding And View

**Manager Table**

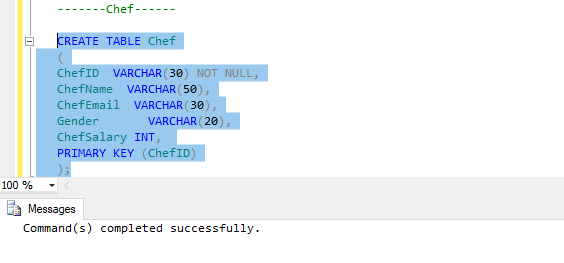


Script to create **Manager** table and result.

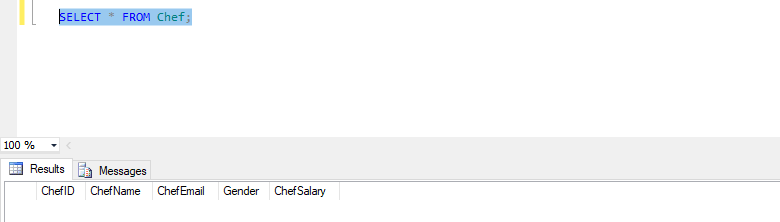


Checking whether the create script work successfully.

**Chef Table**

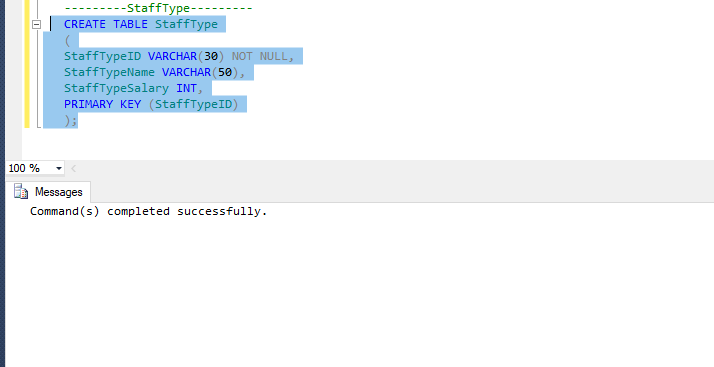


Script to create **Chef** table and result.

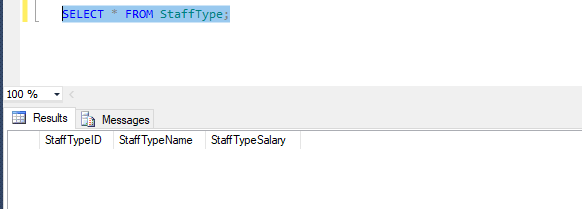


Checking whether the create script work successfully.

**StaffType Table**

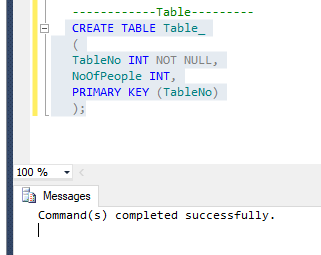


Script to create **StaffType** table and result.



Checking whether the create script work successfully.

**Table\_**



Script to create **table\_** table and result.

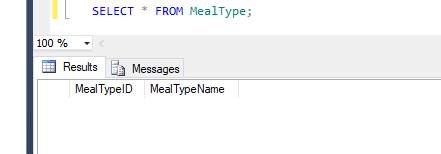


Checking whether the create script work successfully.

**MealType**

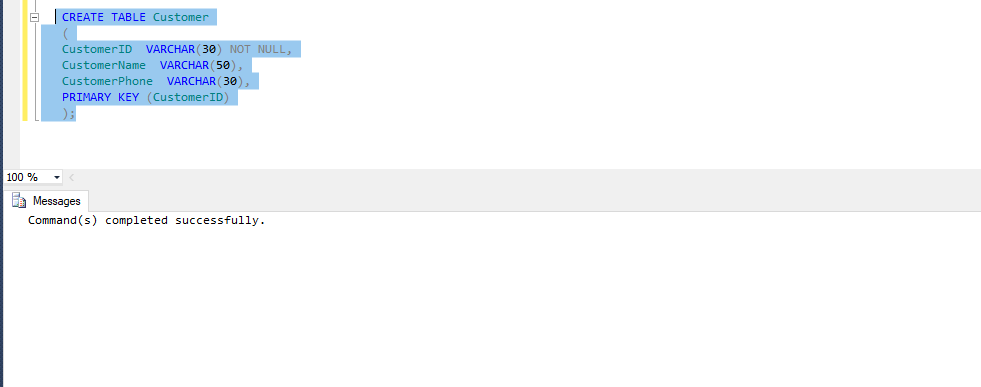


Script to create **MealType** table and result.



Checking whether the create script work successfully.

**Customer**

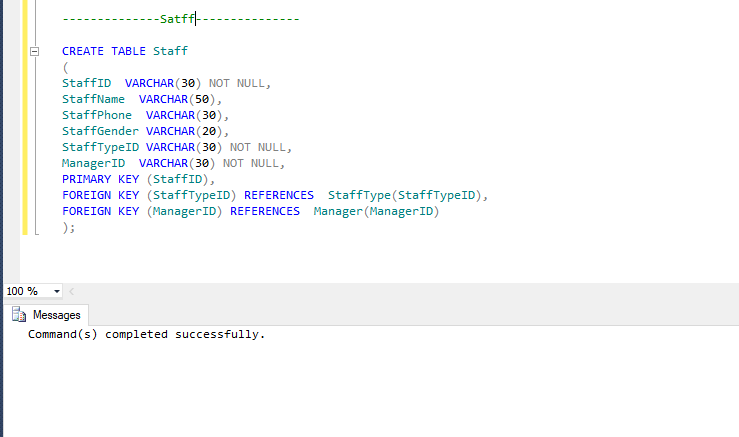


Script to create **Customer** table and result.

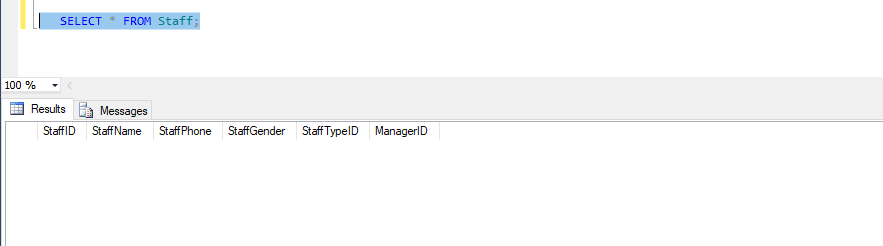


Checking whether the create script work successfully.

**Staff Table**

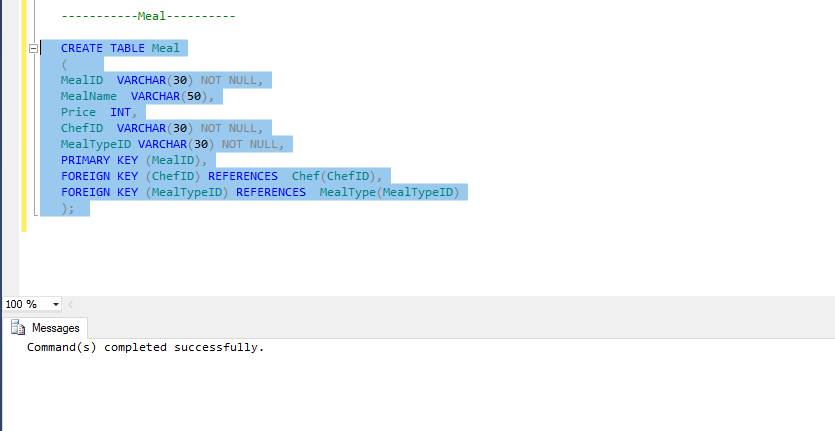


Script to create **Staff** table and result.

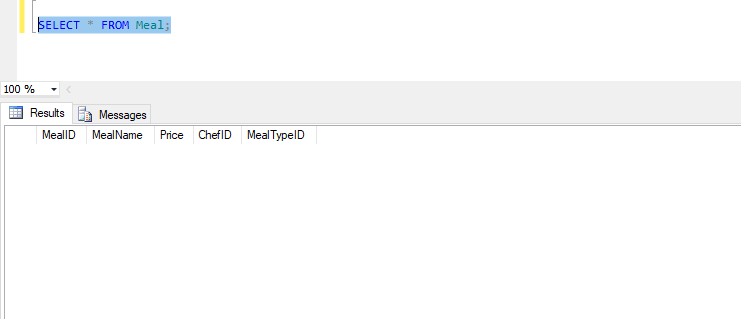


Checking whether the create script work successfully.

**Meal Table**

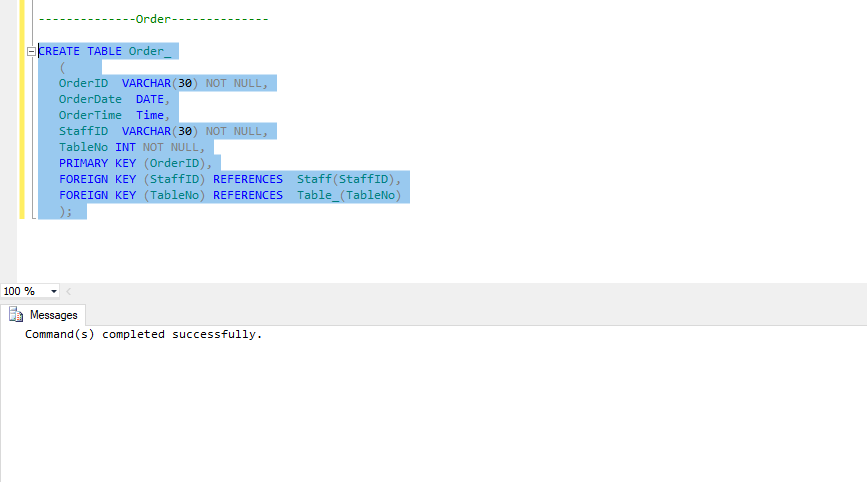


Script to create **Meal** table and result.

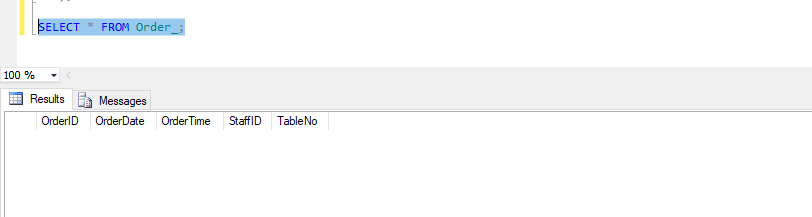


Checking whether the create script work successfully.

**Order Table**

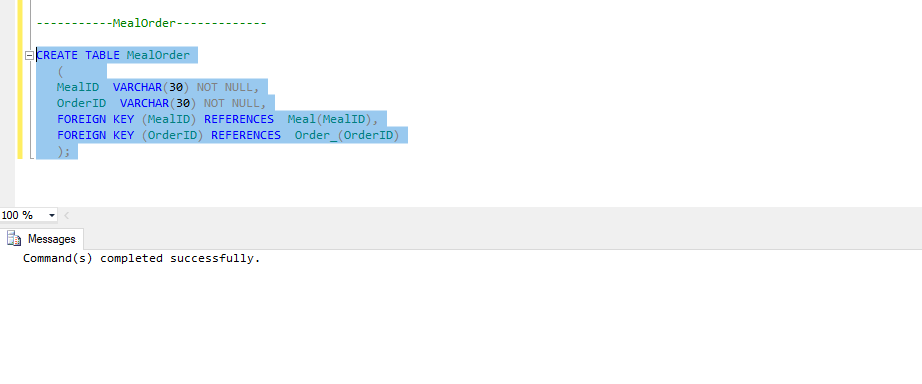


Script to create **Order\_** table and result.



Checking whether the create script work successfully.

**MealOrder Table**

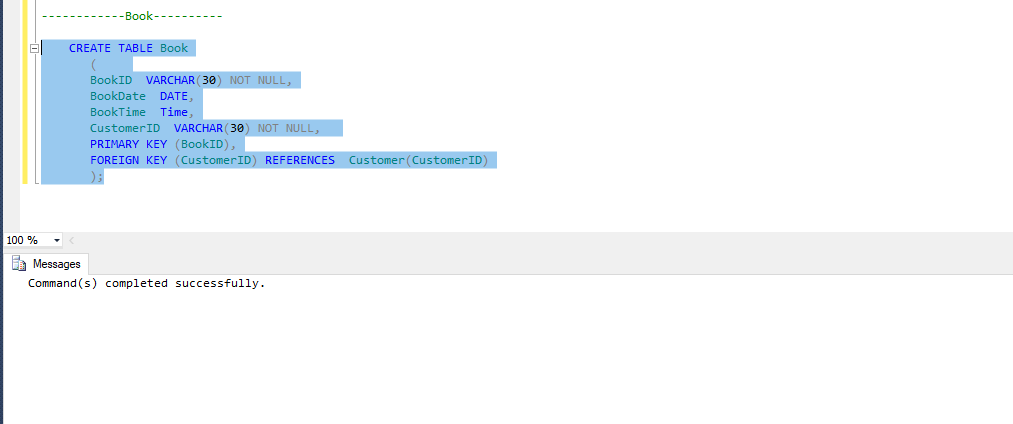


Script to create **MealOrder** table and result.

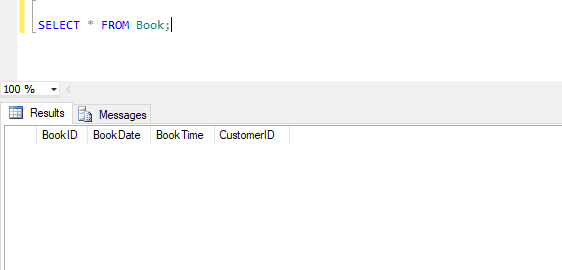
a

Checking whether the create script work successfully.

**Book Table**

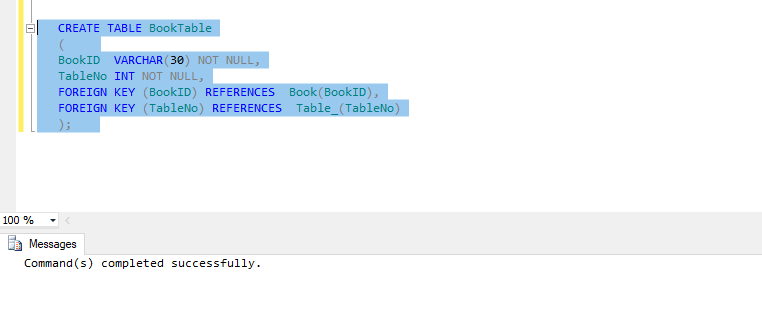


Script to create **Book** table and result.

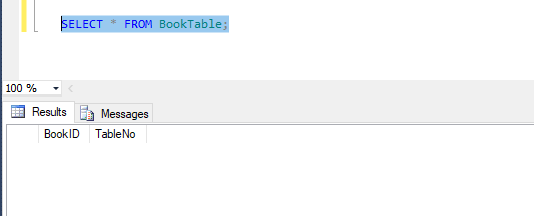


Checking whether the create script work successfully.

**BookTable Table**



Script to create **BookTable** table and result.



Checking whether the create script work successfully.

### Summary

To develop these create scripts, a new database for the restaurant is created first. The tables names, columns and data types had to be written down so that the integrity constraints can be understand easily. After being understand the tables their relationships, the create scripts are written down. There was an error when the tables with foreign keys were run first. Therefore, the scripts for tables without foreign keys are run first when running the create scripts. For example, Staff table has a foreign key, StaffTypeID, from StaffType Table. When the create script for staff table is run first, an error is occurred. So, Staff table script is run first to fix this error.

Then, the scripts for tables with foreign keys are run in order.

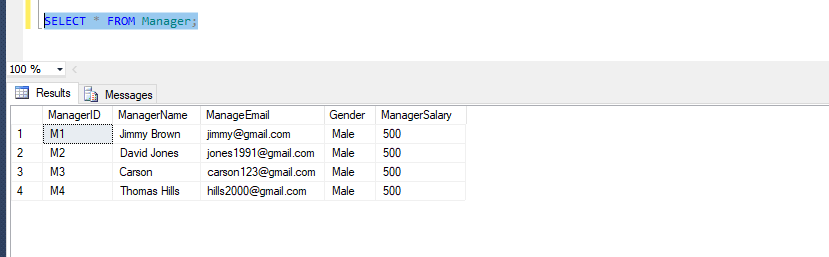
# Task 4

## Data Population

**Insert script and Result for Manager Table**



Running the script to insert data into Manager Table.

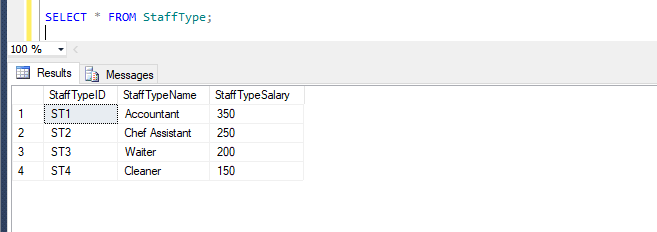


Checking whether the **insert script** works successfully.

**Insert script and Result for StaffType Table**

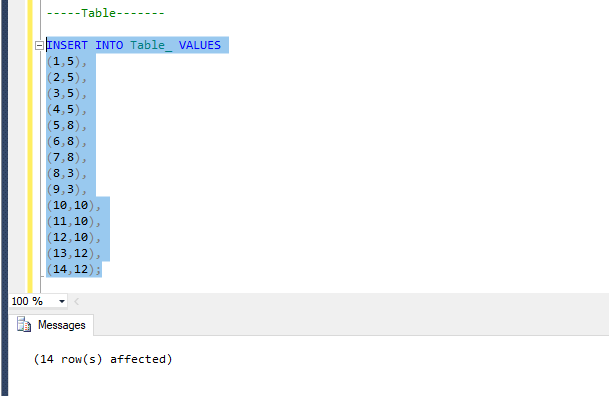


Running the script to insert data into SatffType Table.

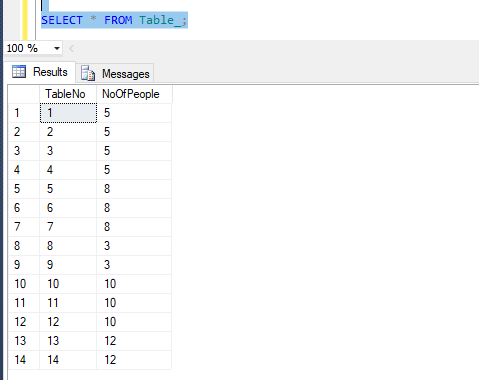


Checking whether the **insert script** works successfully.

**Insert script and Result for Table\_ Table**

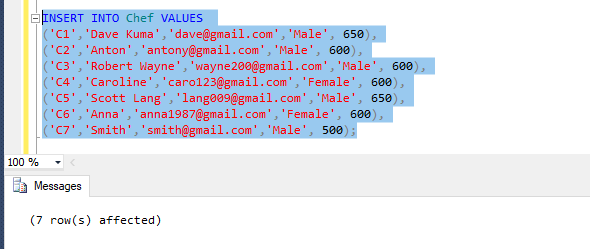


Running the script to insert data into Table\_ Table.

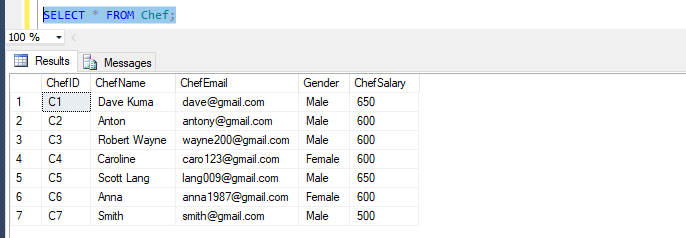


Checking whether the **insert script** works successfully.

**Insert script and Result for Chef Table**

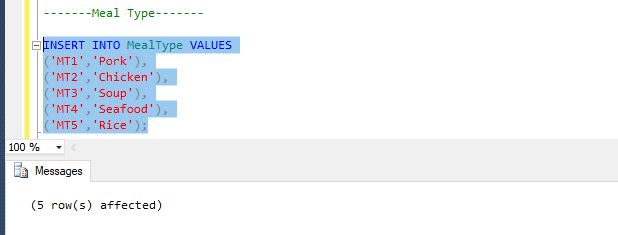


Running the script to insert data into Chef Table.

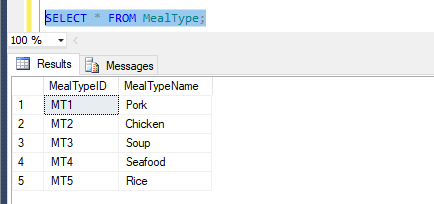


Checking whether the **insert script** works successfully.

**Insert script and Result for MealType Table**

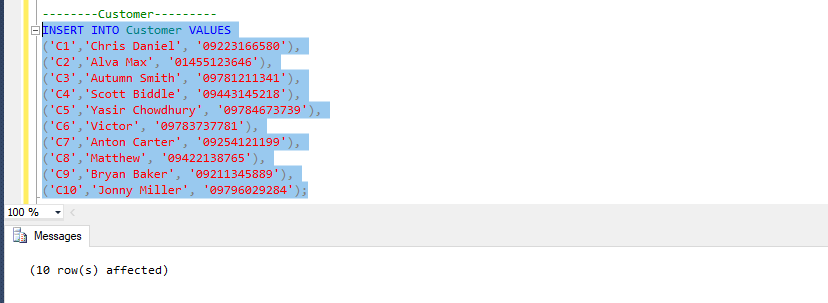


Running the script to insert data into Chef Table.

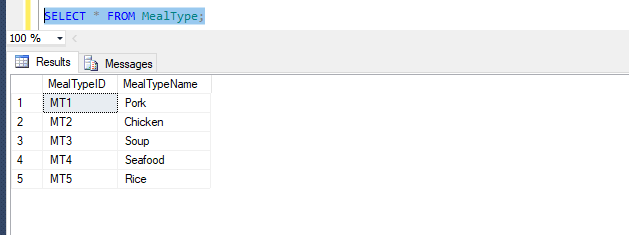


Checking whether the **insert script** works successfully.

**Insert script and Result for Customer Table**



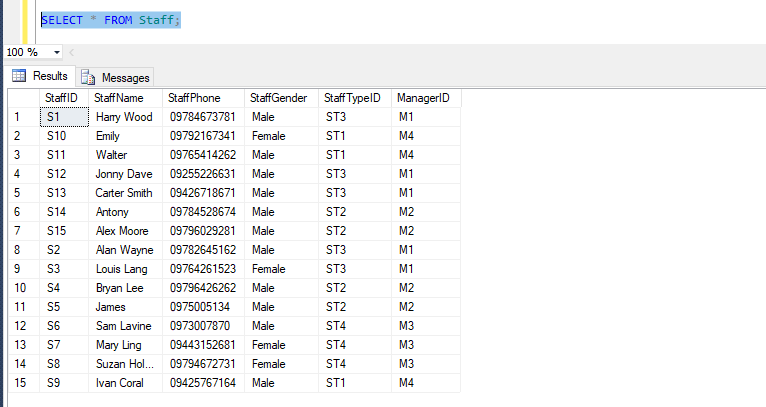
Running the script to insert data into Chef Table.



Checking whether the **insert script** works successfully.

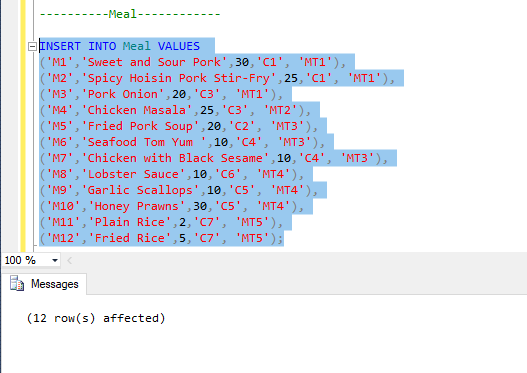
**Insert script and Result for Staff Table**

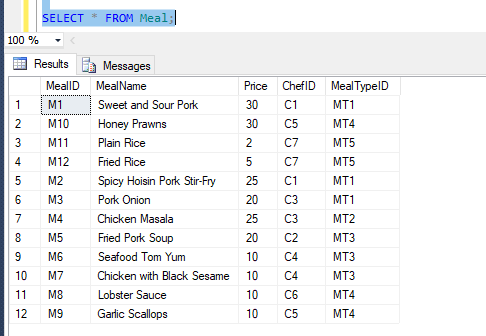




Checking whether the **insert script** works successfully.

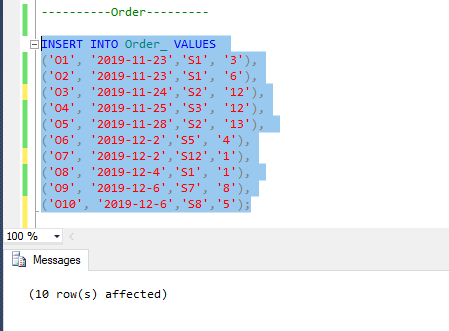
**Insert script and Result for Meal Table**



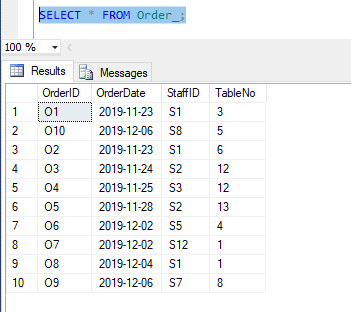


Checking whether the **insert script** works successfully.

**Insert script and Result for Order\_Table**

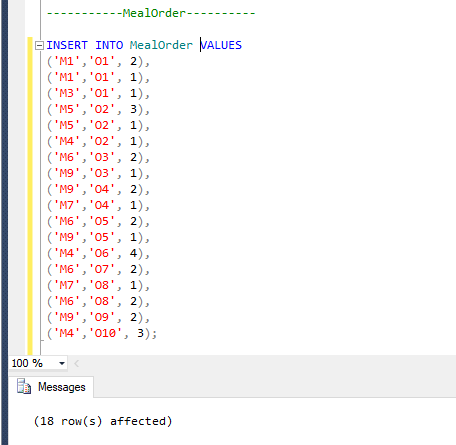


Running the script to insert data into Order\_ Table.

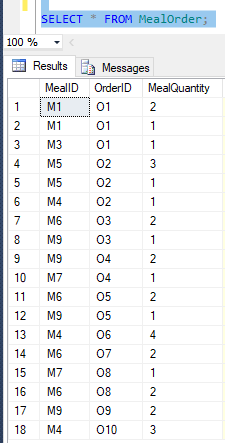


Checking whether the **insert script** works successfully.

**Insert script and Result for MealOrder Table**

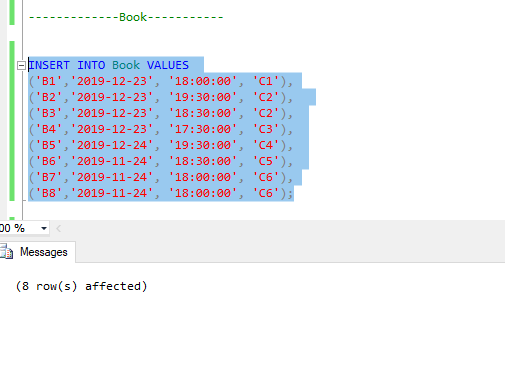


Running the script to insert data into MealOrder Table.

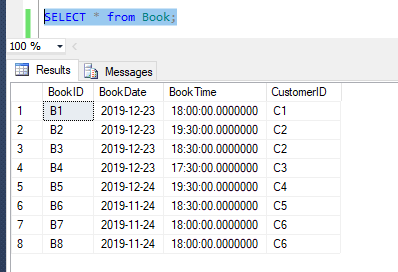


Checking whether the **insert script** works successfully.

**Insert script and Result for Book Table**

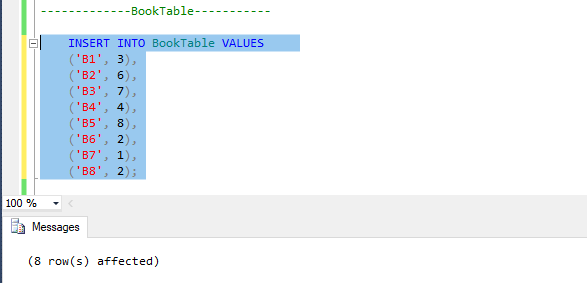


Running the script to insert data into Book Table.

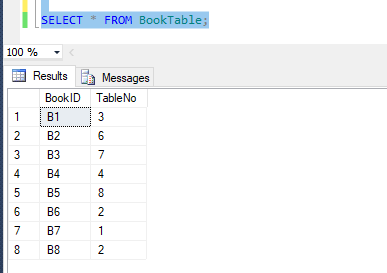


Checking whether the **insert script** works successfully.

**Insert script and Result for BookTable Table**



Running the script to insert data into BookTable Table.



Checking whether the **insert script** works successfully.

### Summary

To populate the database, the data are analyzed form the documents from Task (1). When the perfect and appropriate data are got, the insert scripts are written down. In the case of writing the scripts, it must be careful that the inserted data meet with the data type of that columns. When a varchar value is inserted into the column with integer data type, an error is found. For example, data type of the column TableNo from Table\_ Table is given Integer. There was an error when the varchar data “T1” is inserted.

In the case of running the scripts, the data are inserted into the table without foreign keys first. When inserting data into foreign key columns, the data must be existed in the related table first. If not, there was an error.

# 

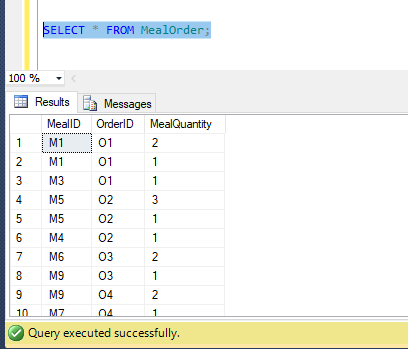
# Task 5

## Enhancement

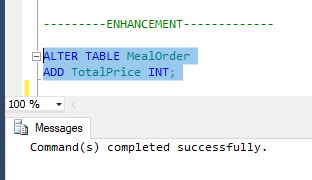
Enhancement 1

* There should be a column to store total price of each order in MealOrder table so that purchasing will be done easily. This can be done by adding a column into MealOrder table using alter script. The new column added into MealOrder table must be filled with data. This can be done by multiplication of MealQuantity and price.

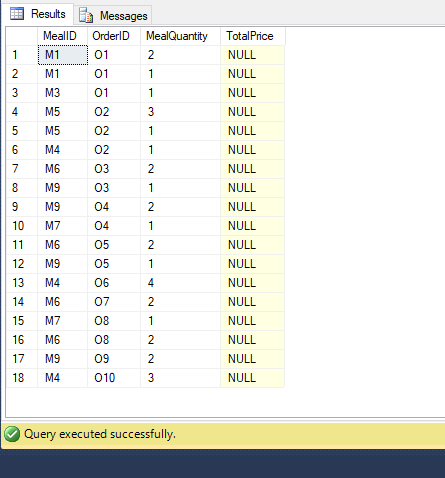
**SQL Scripts**



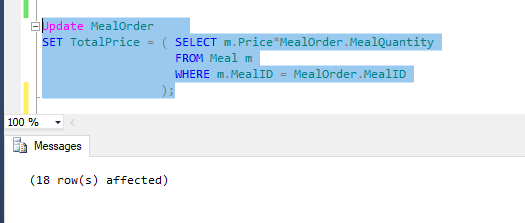
Before running the script.



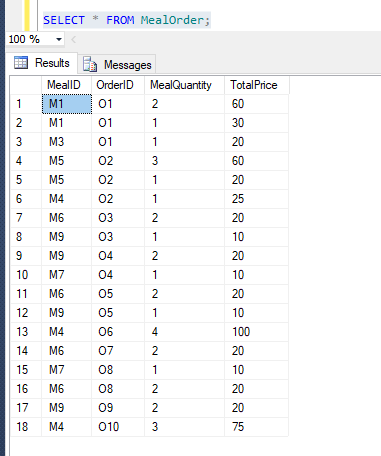
The script is running.



After the alter script is run.



Updating the new column script and result.

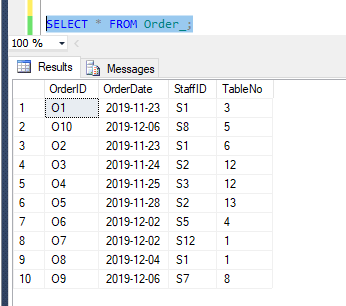


After running the update script.

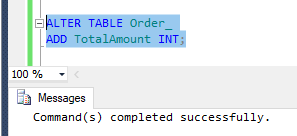
Enhancement 2

* A new column naming “TotalAmount” is added to Order\_ table. This column will store total amount of price for each order helping in faster purchasing process. Adding this new column can be done by using alter script. Then, the column will be filled with data by summation of total price for every same OrderID with the use of SUM and by joining two tables.

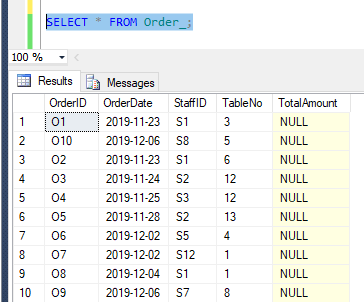
**SQL Scripts**



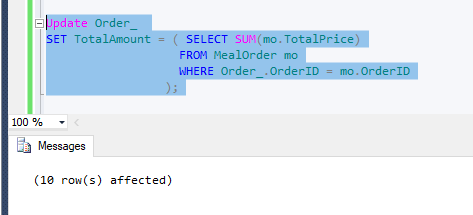
Before adding new column.



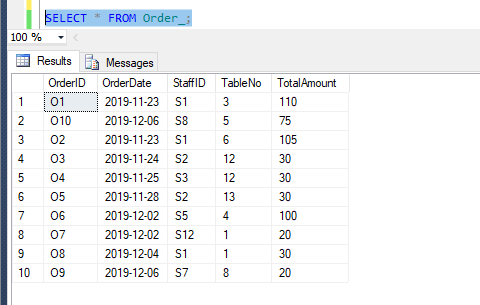
Running alter script and result.



After adding new column.



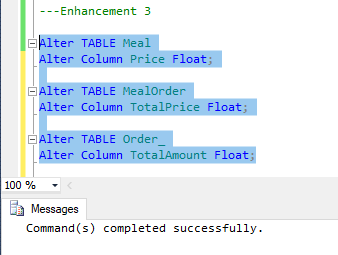
Running Update script to populate new column with data.



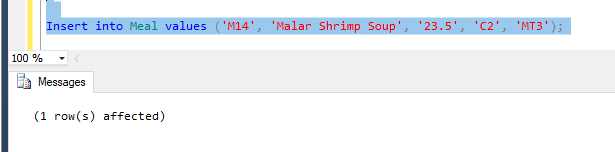
After updating data.

Enhancement 3

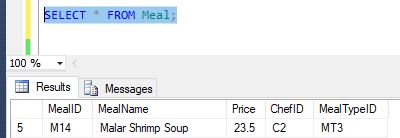
* The columns storing price, total price and total amount are created in integer data type first. The prices of meals may be with decimal in future. So, the data type of these columns is changed into decimal. This is done by using alter script.



Running Alter scripts to change the data type of columns.



A new meal with the price of decimal data type is inserted into Meal table to check whether the alter script completed successfully or not.



The inserted price is stored as float data type successfully.

# 

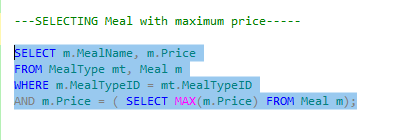
# Task 6

# SQL reports

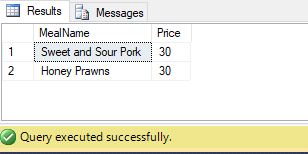
## Query 1

**Purpose** – customers may want to know the meal with maximum price. This query retrieve the name of meal that has maximum price by using sub query and joining two tables Meal and MealType.

**SQL script**



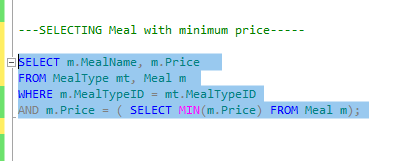
**Result**



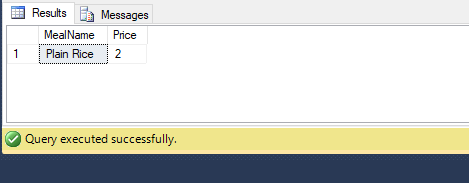
## Query 2

**Purpose** – customers may want to know the meal with minimum price. This query retrieves the name of meal that has minimum price by using sub query and joining two tables Meal and MealType.

**SQL script**



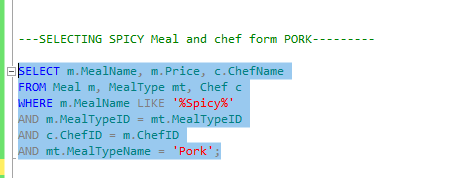
**Result**



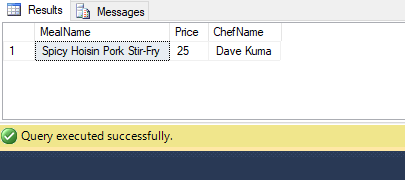
## Query 3

**Purpose** – there may be customers who want to order the spicy meal in the meal type “Pork” and want to know the name of chef who cook that meal. This query retrieves meal which name contain “spicy”, price of that meal and name of the chef who cook that meal by joining three tables and using LIKE function.

**SQL script**



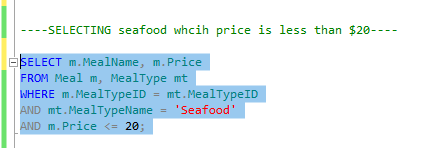
**Result**



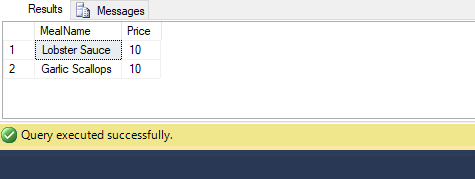
## Query 4

**Purpose** – there may be customers who want to order the seafood which price is less than $20. This query retrieves seafood name which price is less than $20 by joining two tables and < operator.

**SQL script**



**Result**

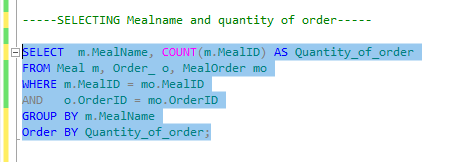


## 

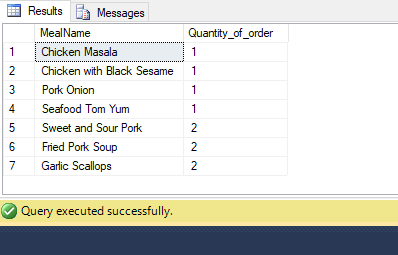
## Query 5

**Purpose** – the administrators of restaurant will need to check which meal was order most and number of ordering for each meal. This query retrieves meal name in the ascending order of ordering number by joining three tables and using count function, group by method and order by method.

**SQL script**



**Result**



# Task 7

## Distributed Database Option

In the future, this restaurant will need to merge with other companies and make expansion of the business. These expansions can be done easily by the use of a distributed database. A distributed database is a kind of database system which is store in many difference computers i.e. the database is not stored by attaching all storage devices to a common processor. These many different computers are located in same place and are connected over internet. Distributed database helps the business to expand more.

**Advantages of distributed database**

* Managing data with transparency of different level
* Improve availability and reliability
* Make expansion easier
* Improve performance of executing
* Greater data control

1. **Managing data with transparency of different level**

Distributed database system basically helps in network transparency, replication transparency and fragmentation transparency. This restaurant system may need higher level of transparency and higher level of complexity in the future. So, it is better to operate with a distributed database system in the future.

1. **Improve availability and reliability**

Since the DBMS software are distributed to many different sites, one or more sites can be able to operate the data if one of the site is broken down. This increases availability because users can process through data every time. Customers can make booking everything to the restaurant. By using distributed database system, this process can be done without any error. Moreover, the processing time of the data will be faster since the data is processed by many sites raising the reliability of database. In this way, customers will not face with any delay.

1. **Make expansion easier**

When using a distributed database system, it is easier to add more data, to increase database size or adding more processors. This make expansion of the business easier. This restaurant will need to expand in the future. At this time, using distributed database system will help the business a lot.

1. **Improve performance of executing**

Using distributed database system enable business to execute multiples queries at different sites in the same time. This is done by breaking up a query into numerous subqueries which are basically executed in parallel. Thus, distributed database system improves the performance of the business. This will avoid the booking system and order system form delays and errors.

1. **Greater data control**

In a distributed database, data is distributed to respective sites. So, a site has to control the information related to its location instead of controlling by one central database. In this way, data is easier to control and manipulate.

(Akashkumar17, 2019)

There are three types of distributed database.

* Homogeneous
* Heterogeneous
* Federated

**Recommended types for this system**

Heterogeneous Distributed database is suitable for this restaurant system. Heterogeneous database type allows separate sites of the organization to develop without the need of an overall central plan based on a single DBMS. Different operating systems and databases are contained in each site.

**Advantages**

* Global data center can store huge data form different data centers

The restaurant can have main restaurant and other small restaurants. All the data form restaurants are stored in the one main restaurant. So, using heterogeneous system can help the organization to store large data amount.

* Remote access is done by the use of global schema

The organization can access their data remotely by the use of global schema. So, the data is easy to be retrieve and increase availability.

(Qaisrani, 2014)

**Factors why distributed database is increasingly popular nowadays**

Nowadays, large organization with numerous offices are using distributed database. Distributed database helps business to develop new sites not only within a country but also to other countries. So, expansion is easier for the business with the use of distributed database. In a distributed database, each locations of the company can directly interact with its database. It us also useful in spreading network through the business. According to these advantages, distributed database is becoming more popular nowadays.

(Raja, 2014)

**SUMMARY**

For this restaurant system, a distributed database should be implemented. This restaurant will have to merge with other food companies or distributes food to other places in the future. In this case, distributed database will help a lot. Heterogeneous Distributed database is recommended for this system because it make data storage larger, better availability and faster data retrieval. This business will have better security, higher performance and reduce delay making more friendly to customers. According to these factors, this business should be implemented a distributed database.

# 

# Task 8

## 

## Evaluation

Rolfe, G., Freshwater, D. and Jasper, M.(2001) model is a kind of reflective writing model. It is simply based on three questions: What? So what? Now what? (Anon., n.d.)The following report is a reflection critically reviewed the learning that have been undertaken to complete this assignment by using this model.

**What ?**

In this assignment, an appropriate organization has to be chosen first to make research and analyze about database system in that organization. The data flows in that organization and the types of everyday transactions must be investigated. A description which include overview of day-to-day operations of the organization and data transaction has to be written. This should also include entity relationship diagram and normalization of the entities in the DBMS. When the normalization is done, the scripts to create tables structures and populate these tables with data have to be written. Then, the created database must be enhanced, and retrieval of data has to be made. It is little bit difficult to choose an organization which must be familiar. Before starting this assignment, it is important to understand SQL language and to be able to use SQL scripts well. Each tasks have to be included brief explanation. So, it is necessary to have a good writing skill of reports. The restaurant system is chosen for this assignment because it is more familiar than other system. There is so much new experience form the tasks.

**So what ?**

Making analysis about the restaurant is not as easy as being thought before. It is not easy to make interview or research about the restaurant because the staff and administrators are always busy. During analysis of the transactions, all the key tasks and data flows have to be written down in order to make normalization easily. Distributed database option means a bit difficulty because of the less knowledge about it. But that problem is solved later by reading online articles about it. Microsoft SQL server is chosen to run database and write SQL scripts. It is quite interesting to do research about a real business system. Most problems and difficulties which happens running the scripts are solved by asking lecturers and reading online articles. Since administrators of the organization did not allow to use the real name of their staffs and chefs, their names have to be imagined. When creating tables with integrity constraints, the scripts were run again and again because of errors. When making enhancements and reporting about distributed database option, some online articles had to be read in order to complete these tasks successfully. Being able to use SQL language well, writing the scripts is not difficult but getting the idea is. All these issues and difficulties become experience later. All the tasks give different experience and allows to learn new skills. The advantages and importance of distributed database are known as new knowledge. New SQL scripts and functions are learned from the tasks. Since the database system is created for a real organization, a great practical experience is got.

**Now What ?**

Ultimately, the successful outcome is achieved for this project. But it is better to learn more about distributed database and SQL operators to improve this process. The database system will be more functionable if more data transactions are implemented. For further improvements, there should be a better data analysis so that the scripts will not be needed to run again and again and save time consuming.

**2280**

## 

## References:

Akashkumar17, 2019. *Advantages of Distributed database.* [Online]   
Available at: https://www.geeksforgeeks.org/advantages-of-distributed-database/  
[Accessed 20 January 2020].

Anon., 2020. *Reflective Writing.* [Online]   
Available at: https://www.op.ac.nz/assets/LearningAdvice/d0a1fac268/Reflective-writing.pdf

Anon., n.d. *Reflective Writing.* [Online]   
Available at: https://www.op.ac.nz/assets/LearningAdvice/d0a1fac268/Reflective-writing.pdf  
[Accessed 25 January 2020].

Qaisrani, F., 2014. *TAHAROC/types-of-data.* [Online]   
Available at: https://www.slideshare.net/TAHAROC/types-of-data  
[Accessed 21 July 2020].

Raja, A., 2014. *What Is a Distributed Database, and What Are Distributed Data Systems For?.* [Online]   
Available at: https://www.atlantic.net/cloud-hosting/about-distributed-databases-and-distributed-data-systems/  
[Accessed 20 January 2020].

## Candidate Checklist

Please use the following checklist to ensure that your work is ready for submission.



Have you read the NCC Education documents 'What is Academic Misconduct? Guidance for Candidates' and 'Avoiding Plagiarism and Collusion: Guidance for Candidates' and ensured that you have acknowledge all the sources that you have used in your work?

Have you completed the 'Statement and Confirmation of Own Work' form and attached it to your assignment? You must do this.



Have you ensured that your work has not gone over or under the recommended word count by more than 10%?



Have you ensured that your work does not contain viruses and can be run directly?

