UNIVERSITY OF MALAYA

Database (WIA2001)

Semester 1 2021/22

Tutorial Group : K1 T4

Lecturer : DR. FARIZA HANUM BINTI MD NASARUDDIN

Group : 6

Group Members :

Name	Matric ID
FRANKIE LIM QI QUAN (Leader)	U2005263/1
PUA JING YI	U2005396/1
GOH CHEE LAM	U2005382/1
KELVIN CHEAH	U2005394/1
FARISAH ATHIRA BINTI MD ZAMRI	17206915/2

Table of Contents

Report 1

1.0 Introduction

A database design for a cafe called Ke Nina Cafe that sells drinks and food to its customers and is located at Bandar Baru, Kuala Kangsar, Perak. Its operation hours are from 10 am to 9 pm daily. The cafe focuses on milk tea drinks and western foods. Customers are allowed to order either through deliveries or walk-in. There are 4 job positions which are the cashier, chef, worker who prepares drinks and delivery man plus one Manager that manages the workers. The workers work 1 shift per day (during MCO) with 1 holiday per week. Then, the cafe uses a point of sales system, which allows them to identify the most popular food in the cafe. It also allows them to generate reports containing the revenue and net profit earned to facilitate their business decisions making. Hence, they use cloud servers to store business data.

The problems currently exist by the cafe are they do not have a proper database that records their workers working hours and can automatically calculate total salary per week. They still use manual calculation where it may lead to miscalculation and is a tedious process. Other than that, they are using spreadsheets like Excel to keep track of the food ingredients left. This means the workers will need to count the ingredients left every day and update it in the spreadsheet where it may lead to data anomalies. Plus, they are planning to open up a new branch in the upcoming time, so that they can reach more customers to expand their business. This is one of the reasons they need a better system to help them manage all the data.

The solution to the problems stated above is to create a proper relational database for them to record every worker's working hours and the food ingredients. After implementing a proper database, the total salary of each worker and the number of ingredients left can be checked with just one click.

2.0 Issues in Managing Data

1. The cafe does not have a proper database system to record their working hours and the one that can automatically calculate the total salary per week.
2. Using spreadsheets like Excel to keep track of the food ingredients left, where the worker will need to count the ingredients every day and update it in the spreadsheet where it may lead to data anomalies.
3. The workers record the customers’ orders on a piece of paper. During the peak hours, the handwriting of the workers becomes ugly and hard to read by the other workers. Besides that, there is the possibility that the workers miss some of the orders.

3.0 Business Rules and Assumptions

Business Rules:

- One customer can place one or many orders.
- One order is placed by one and only one customer.
- One employee can handle zero orders.
- One employee can handle many orders.
- One order can be handled by one and only one employee.
- One order has one or many Menu.
- One Menu can be found in zero orders.
- One Menu can be found in many orders.
- One Menu has one or many ingredients.
- One ingredient belongs to one or many Menu.
- One supplier can supply one or many ingredients.
- One ingredient is supplied by one and many suppliers.
- One order belongs to one and only one payment.
- One payment has one and only one order.
- One payment uses zero voucher.
- One payment uses one and only one voucher.
- One voucher is used in one and only one payment.
- One payment is processed by one and only one employee.
- One employee can process zero payments.
- One employee can process many payments.
- One delivery has one and only one order.
- One order is found in zero delivery.
- One order is found in one and only one delivery.
- One delivery is delivered by one and only one employee.
- One employee can deliver zero deliveries.
- One employee can deliver many deliveries.

Assumptions:

- The customer must place at least one order in the restaurant.
- The delivery address is the same as the customer address.
- The customer who places the order also makes the payment at the counter.

- The currency used is Ringgit Malaysia(RM).
- The customer must give basic information before placing the order.
- The unit price of ingredients is calculated using the supply total price divided by the supply quantity.
- There is no rounding of the payment amount for the cash payment method.

4.0 System Objectives and Scope

Ke Nina Cafe was a cafe located in No.12, Persiaran Bendahara 1, Bandar Baru, 33000 Kuala Kangsar, Perak. The cafe currently has only one main restaurant but is expecting to open a new branch soon at a strategic location to expand their business.

The cafe has about 4 staff working as the cashier, chef, the worker who prepares drinks and a delivery man, with 1 shift per day during Movement Control Order(MCO) from 9 am to 10 pm. There will be 2 workers assigned as chefs in the kitchen, 2 workers serving the customers and preparing the drink and another 2 workers assigned to deliver food to the customer. Also, a manager is assigned and responsible for the restaurant's daily operations. Usually, the manager will supervise other workers in the restaurant and order ingredients from their suppliers weekly.

Then, each customer that comes in will be given a paper to write the food and drinks they ordered by writing down the code. This method is used by the cafe to record and keep track of the order. Ke Nina Cafe has each food and drinks to be uniquely identified by assigning different codes for different types of food. Besides that, before ordering food, a customer needs to provide their basic information like their name so that the system can store the data for future reference. Plus, there are also deliveries provided for the customers via the cafe delivery, FoodPanda and GrabFood so that the user can choose any purchase method they want.

Other than that, customers can choose either using cash, e-wallet, debit or credit cards for the payment process. The cafe hopes that its customers can pay conveniently and have the best dine-in or take-out experiences without worrying about the payment. Furthermore, the cafe also provides 15% discounts (drinks only) for customers who had loyalty cards or vouchers. Each voucher has a unique code to be identified and can be used for one time only with no minimum spend requirements for each purchase. The coupon discount rate is fixed at 15 %, but on certain national holidays like Hari Kebangsaan, the discount rate may increase.

Next, moving on to the management of the ingredients, the worker needs to count how many ingredients are left every day and update it in the Excel spreadsheet. This is considered to be time-consuming and may lead to data anomalies, where the cafe manager hopes they have a proper database system to take care of it. The cafe has around 6 suppliers that will supply the ingredients weekly. The manager is the one in charge of the order and the amount will depend on the quantity of the ingredients left.

The cafe is still using a punch card system to keep track of the workers working hours. However, the salary is calculated manually by referring to the record on the punch card. The process is so tedious, where the owner of the cafe hopes that it can be handled automatically by a system, as it is easy to make mistakes by manual calculation. Then, the cafe is using a cloud server to generate reports about the revenue and net profit earned by the business. In brief, as Ke Nina Cafe may grow in the future, they may have difficulties in managing the increasing amount of data used and generated by the cafe. To ensure the continued success of the cafe, a database system is needed to help solve the increasing problems of data management.

5.0 User Requirements

Employer has the highest access to the database. Maintaining data can be done by entering, updating and deleting data. The employer can maintain data and records on the restaurant and employees. Employers can maintain data and records on the menu, ingredients and suppliers. The employees can record the data of payments, vouchers, customers, orders and delivery.

Searching data and records can be done by the employer and employees. The employer can perform searches on employees, menu, ingredients and suppliers. The employee can perform searches on payments, vouchers, customers, orders and delivery.

Tracking data and records can be done by the employer, manager and employees. The employer is able to check employees and the salary of employees according to their job type and working hours. The employer is also able to track the information of ingredients and the information of suppliers. Besides, the employer is able to check their restaurant's revenue, expenditures and information about the payment methods. Next, the employee is able to track the status of the payments, the status of vouchers, the status of delivery and check the status of orders.

Generating reports can be done only by the employer. The employer can generate reports on sales of the restaurant and the salary of employees. For example, the employer can calculate the total payments by using each payment method that is made by the customers and calculate the number of payments using each payment method. The employer can also generate reports on payments, the status of ingredients and orders from customers. By checking the number of orders made by each customer, the employer can get insights from the reports about the loyalty of the customers and some of the most popular menu items.

6.0 Entities and Attributes Before Normalization

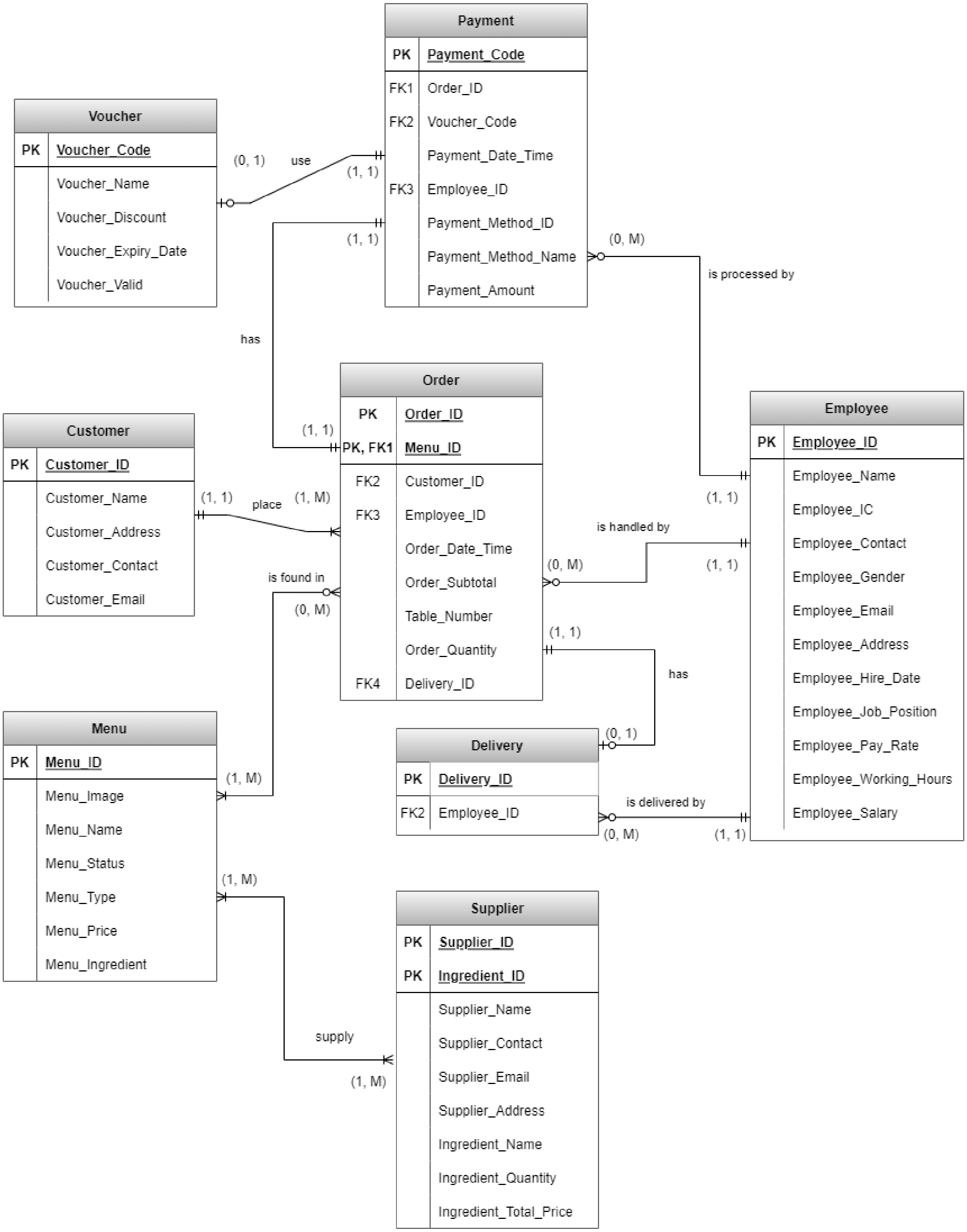
Entity	**Attributes**
Customer	**Customer_ID** Customer_Name Customer_Address Customer_Contact Customer_Email
Order	**Order_ID** **Menu_ID** Customer_ID Employee_ID Order_Date_Time Order_Subtotal Table_No Order_Quantity Delivery_ID
Menu	**Menu_ID** Menu_Image Menu_Name Menu_Status Menu_Type Menu_Price Menu_Ingredient
Supplier	**Supplier_ID** **Ingredient_ID** Supplier_Name Supplier_Contact Supplier_Email Supplier_Address Ingredient_Name Ingredient_Quantity

	Ingredient_Total_Price
Payment	**Payment_Code** Order_ID Voucher_Code Payment_Date_Time Employee_ID Payment_Method_ID Payment_Method_Name Payment_Amount
Voucher	**Voucher_Code** Voucher_Name Voucher_Discount Voucher_Expiry_Date Voucher_Valid
Employee	**Employee_ID** Employee_Name Employee_IC Employee_Contact Employee_Gender Employee_Email Employee_Address Employee_Hire_Date Employee_Job_Position Employee_Pay_Rate Employee_Working_Hours Employee_Salary
Delivery	**Delivery_ID** Employee_ID

7.0 Relationship, Cardinalities And Constraints

- One employee can have only one IC.
- The contact number does not contain a country code.
- The ingredient expiry date is not tracked.
- The customer needs to order something in the restaurant.
- The customer must use only one payment method for each order payment.
- Vouchers can be used only once for payment.
- Only one person will order ingredients from the suppliers.
- The order will be delivered to the customer address.

8.0 Conceptual Entity Relationship Diagram



9.0 Normalization Process

9.1 Table Schemas:

EMPLOYEE(**Employee_ID**, Employee_Name, Employee_IC, Employee_Contact, Employee_Gender, Employee_Email, Employee_Address, Employee_Hire_Date, Employee_Job_Name, Employee_Pay_Rate, Employee_Working_Hours, Employee_Salary)

CUSTOMER(**Customer_ID**, Customer_Name, Customer_Contact, Customer_Address, Customer_Email)

MENU(**Menu_ID**, Menu_Name, Menu_Image, Menu_Status, Menu_Type, Menu_Price, Menu_Ingredient)

ORDER(**Order_ID**, Order_Date_Time. Customer_ID, **Menu_ID**, Order_Quantity, Order_Subtotal, Table_No, Employee_ID, Delivery_ID)

PAYMENT(**Payment_Code**, Order_ID, Payment_Date_Time, Employee_ID, Payment_Method, Voucher_Code)

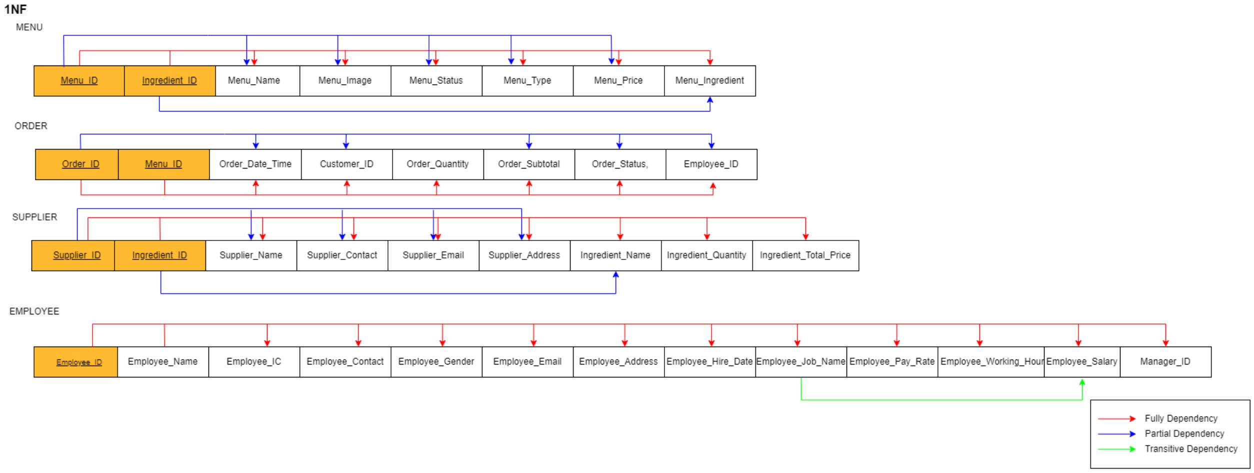
SUPPLIER(**Supplier_ID**, Supplier_Name, Supplier_Contact, Supplier_Email, Supplier_Address, **Ingredient_ID**, Ingredient_Name, Ingredient_Quantity, Ingredient_Total_Price)

VOUCHER(**Voucher_Code**, Voucher_Name, Voucher_Discount, Voucher_Expiry_Date)

DELIVERY(**Delivery_ID**, Employee_ID)

9.2 UNNORMALIZED FORM (UNF)

According to the Entity-Relationship Diagram (ERD) above, it is still in an Unnormalized Form because it contains repeating groups. The tables that contain the repeating groups are MENU, ORDER and SUPPLIER. Below are the examples of the repeating group.



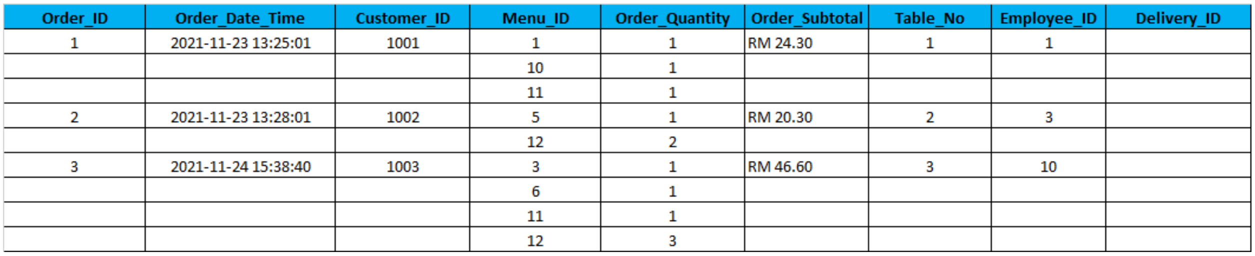
UNF Menu Table:

MENU(**<u>Menu_ID</u>**, Menu_Name, Menu_Image, Menu_Status, Menu_Type, Menu_Price, Menu_Ingredient)

Menu_ID	Menu_Name	Menu_Image	Menu_Status	Menu_Type	Menu_Price	Menu_Ingredient
1	Brown Sugar Boba Milk (H)	Drink Image 1	Available	Drink	RM10.30	Fresh Milk
						Brown Sugar Boba
2	Brown Sugar Boba Milk (C)	Drink Image 2	Not Available	Drink	Rm13.30	Fresh Milk
						Brown Sugar Boba
3	Brown Sugar Boba Milk with Grass Jelly (H)	Drink Image 3	Available	Drink	RM10.30	Fresh Milk
						Grass Jelly
4	Brown Sugar Boba Milk with Grass Jelly (C)	Drink Image 4	Available	Drink	RM13.30	Fresh Milk
						Grass Jelly
5	Brown Sugar Boba Milk Tea (H)	Drink Image 5	Available	Drink	RM10.30	Condensed Milk
						Black Tea Leaves
						Brown Sugar Boba
6	Brown Sugar Boba Milk Tea (C)	Drink Image 6	Not available	Drink	RM13.30	Condensed Milk
						Black Tea Leaves
						Brown Sugar Boba
7	Damascus Rose Tea (H)	Drink Image 7	Available	Drink	RM 8.00	Rose Syrup
						Rose Pedal
						Pink Boba
8	Jasmine Tea with Honey (H)	Drink Image 8	Available	Drink	RM8.00	Natural Honey
						Jasmine Tea Powder
						Pink Boba
9	Fries	Food Image 1	Available	Food	RM5.00	Fries
						Himalayan salt
10	Taiwanese Hotdog	Food Image 2	Available	Food	RM6.00	Taiwanese Hotdog
						Cucumber
11	Golden Chicken Chop	Food Image 3	Available	Food	RM8.00	Chicken thigh
						Seasoning Powder
12	Fried Popiah	Food Image 6	Available	Food	RM5.00	Popiah
						Cucumber

UNF Order Table:

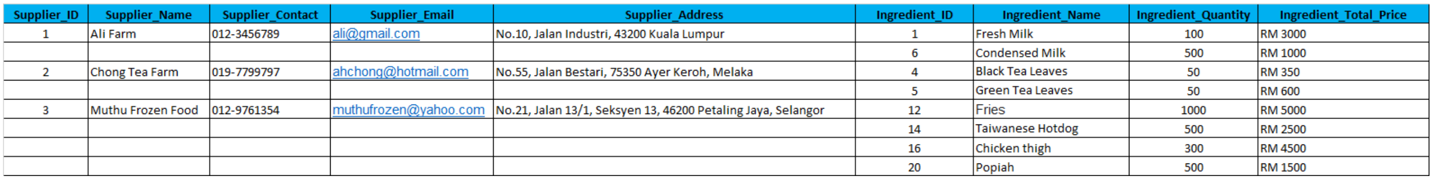
ORDER(**<u>Order_ID</u>**, Order_Date_Time, Customer_ID, **<u>Menu_ID</u>**, Order_Quantity, Order_Subtotal, Table_No, Employee_ID, Delivery_ID)



Order_ID	Order_Date_Time	Customer_ID	Menu_ID	Order_Quantity	Order_Subtotal	Table_No	Employee_ID	Delivery_ID
1	2021-11-23 13:25:01	1001	1	1	RM 24.30	1	1	
			10	1				
			11	1				
2	2021-11-23 13:28:01	1002	5	1	RM 20.30	2	3	
			12	2				
3	2021-11-24 15:38:40	1003	3	1	RM 46.60	3	10	
			6	1				
			11	1				
			12	3				

UNF Supplier Table:

SUPPLIER(**<u>Supplier_ID</u>**, Supplier_Name, Supplier_Contact, Supplier_Email, Supplier_Address, **<u>Ingredient_ID</u>**, Ingredient_Name, Ingredient_Quantity, Ingredient_Total_Price)



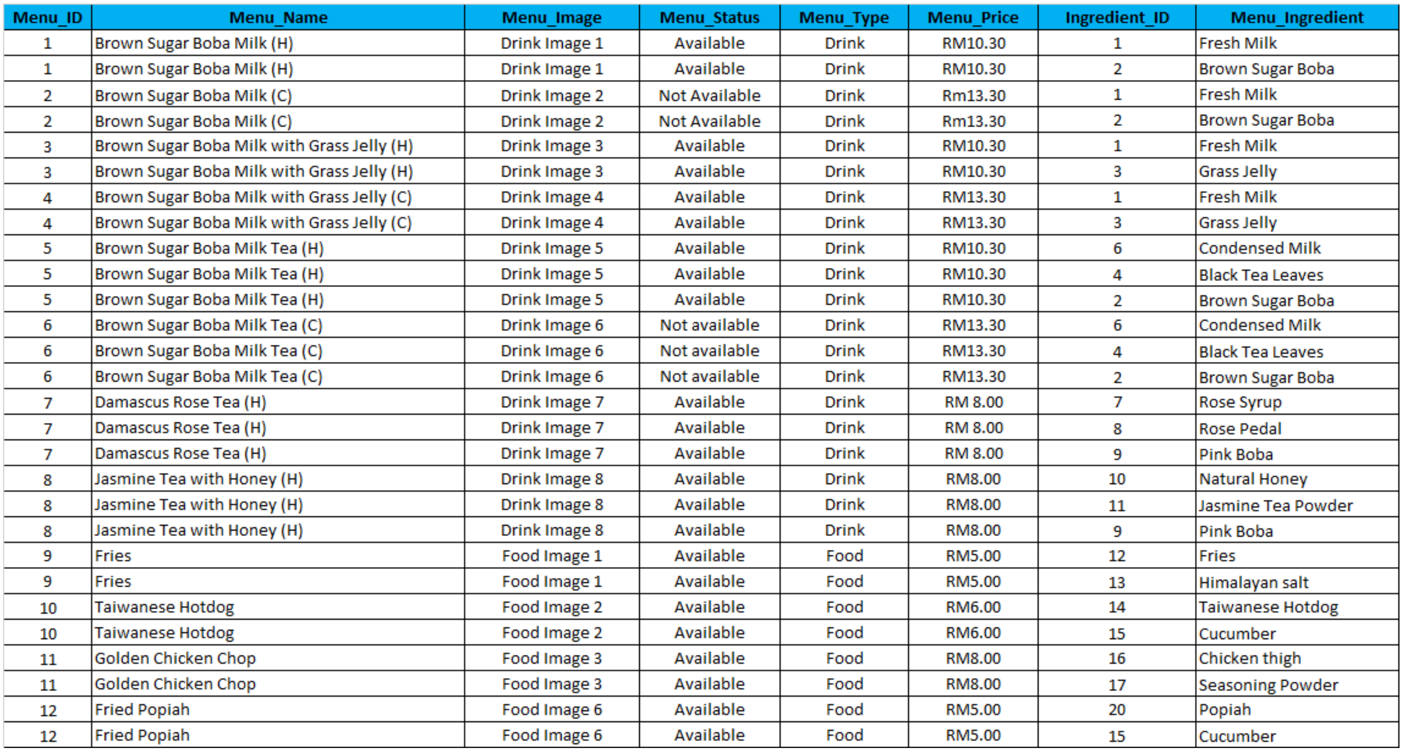
Supplier_ID	Supplier_Name	Supplier_Contact	Supplier_Email	Supplier_Address	Ingredient_ID	Ingredient_Name	Ingredient_Quantity	Ingredient_Total_Price
1	Ali Farm	012-3456789	ali@gmail.com	No.10, Jalan Industri, 43200 Kuala Lumpur	1	Fresh Milk	100	RM 3000
					6	Condensed Milk	500	RM 1000
2	Chong Tea Farm	019-7799797	ahchong@hotmail.com	No.55, Jalan Bestari, 75350 Ayer Keroh, Melaka	4	Black Tea Leaves	50	RM 350
					5	Green Tea Leaves	50	RM 600
3	Muthu Frozen Food	012-9761354	muthufrozen@yahoo.com	No.21, Jalan 13/1, Seksyen 13, 46200 Petaling Jaya, Selangor	12	Fries	1000	RM 5000
					14	Taiwanese Hotdog	500	RM 2500
					16	Chicken thigh	300	RM 4500
					20	Popiah	500	RM 1500

9.3 FIRST NORMAL FORM (1NF)

To convert UNF into 1NF, we need to identify the primary key and all dependencies. We also need to eliminate the repeating groups.
To eliminate the above repetition, each table must be split into multiple tables.

1NF Menu Table:

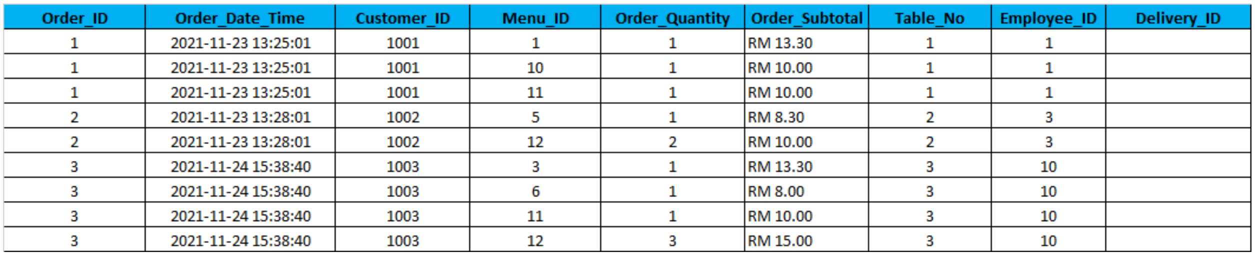
MENU(**Menu_ID**, Menu_Name, Menu_Image, Menu_Status, Menu_Type, Menu_Price, **Ingredient_ID**, Menu_Ingredient)



Menu_ID	Menu_Name	Menu_Image	Menu_Status	Menu_Type	Menu_Price	Ingredient_ID	Menu_Ingredient
1	Brown Sugar Boba Milk (H)	Drink Image 1	Available	Drink	RM10.30	1	Fresh Milk
1	Brown Sugar Boba Milk (H)	Drink Image 1	Available	Drink	RM10.30	2	Brown Sugar Boba
2	Brown Sugar Boba Milk (C)	Drink Image 2	Not Available	Drink	Rm13.30	1	Fresh Milk
2	Brown Sugar Boba Milk (C)	Drink Image 2	Not Available	Drink	Rm13.30	2	Brown Sugar Boba
3	Brown Sugar Boba Milk with Grass Jelly (H)	Drink Image 3	Available	Drink	RM10.30	1	Fresh Milk
3	Brown Sugar Boba Milk with Grass Jelly (H)	Drink Image 3	Available	Drink	RM10.30	3	Grass Jelly
4	Brown Sugar Boba Milk with Grass Jelly (C)	Drink Image 4	Available	Drink	RM13.30	1	Fresh Milk
4	Brown Sugar Boba Milk with Grass Jelly (C)	Drink Image 4	Available	Drink	RM13.30	3	Grass Jelly
5	Brown Sugar Boba Milk Tea (H)	Drink Image 5	Available	Drink	RM10.30	6	Condensed Milk
5	Brown Sugar Boba Milk Tea (H)	Drink Image 5	Available	Drink	RM10.30	4	Black Tea Leaves
5	Brown Sugar Boba Milk Tea (H)	Drink Image 5	Available	Drink	RM10.30	2	Brown Sugar Boba
6	Brown Sugar Boba Milk Tea (C)	Drink Image 6	Not available	Drink	RM13.30	6	Condensed Milk
6	Brown Sugar Boba Milk Tea (C)	Drink Image 6	Not available	Drink	RM13.30	4	Black Tea Leaves
6	Brown Sugar Boba Milk Tea (C)	Drink Image 6	Not available	Drink	RM13.30	2	Brown Sugar Boba
7	Damascus Rose Tea (H)	Drink Image 7	Available	Drink	RM 8.00	7	Rose Syrup
7	Damascus Rose Tea (H)	Drink Image 7	Available	Drink	RM 8.00	8	Rose Pedal
7	Damascus Rose Tea (H)	Drink Image 7	Available	Drink	RM 8.00	9	Pink Boba
8	Jasmine Tea with Honey (H)	Drink Image 8	Available	Drink	RM8.00	10	Natural Honey
8	Jasmine Tea with Honey (H)	Drink Image 8	Available	Drink	RM8.00	11	Jasmine Tea Powder
8	Jasmine Tea with Honey (H)	Drink Image 8	Available	Drink	RM8.00	9	Pink Boba
9	Fries	Food Image 1	Available	Food	RM5.00	12	Fries
9	Fries	Food Image 1	Available	Food	RM5.00	13	Himalayan salt
10	Taiwanese Hotdog	Food Image 2	Available	Food	RM6.00	14	Taiwanese Hotdog
10	Taiwanese Hotdog	Food Image 2	Available	Food	RM6.00	15	Cucumber
11	Golden Chicken Chop	Food Image 3	Available	Food	RM8.00	16	Chicken thigh
11	Golden Chicken Chop	Food Image 3	Available	Food	RM8.00	17	Seasoning Powder
12	Fried Popiah	Food Image 6	Available	Food	RM5.00	20	Popiah
12	Fried Popiah	Food Image 6	Available	Food	RM5.00	15	Cucumber

1NF Order Table:

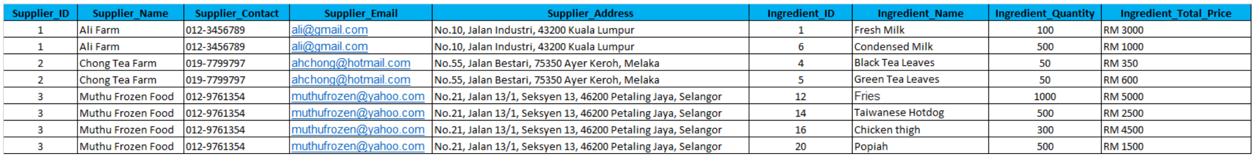
ORDER(**<u>Order_ID</u>**, Order_Date_Time, Customer_ID, **<u>Menu_ID</u>**, Order_Quantity, Order_Subtotal, Table_No, Employee_ID, Delivery_ID)



Order_ID	Order_Date_Time	Customer_ID	Menu_ID	Order_Quantity	Order_Subtotal	Table_No	Employee_ID	Delivery_ID
1	2021-11-23 13:25:01	1001	1	1	RM 13.30	1	1	
1	2021-11-23 13:25:01	1001	10	1	RM 10.00	1	1	
1	2021-11-23 13:25:01	1001	11	1	RM 10.00	1	1	
2	2021-11-23 13:28:01	1002	5	1	RM 8.30	2	3	
2	2021-11-23 13:28:01	1002	12	2	RM 10.00	2	3	
3	2021-11-24 15:38:40	1003	3	1	RM 13.30	3	10	
3	2021-11-24 15:38:40	1003	6	1	RM 8.00	3	10	
3	2021-11-24 15:38:40	1003	11	1	RM 10.00	3	10	
3	2021-11-24 15:38:40	1003	12	3	RM 15.00	3	10	

1NF Supplier Table:

SUPPLIER(**<u>Supplier_ID</u>**, Supplier_Name, Supplier_Contact, Supplier_Email, Supplier_Address, **<u>Ingredient_ID</u>**, Ingredient_Name, Ingredient_Quantity, Ingredient_Total_Price)



Supplier_ID	Supplier_Name	Supplier_Contact	Supplier_Email	Supplier_Address	Ingredient_ID	Ingredient_Name	Ingredient_Quantity	Ingredient_Total_Price
1	Ali Farm	012-3456789	ali@gmail.com	No.10, Jalan Industri, 43200 Kuala Lumpur	1	Fresh Milk	100	RM 3000
1	Ali Farm	012-3456789	ali@gmail.com	No.10, Jalan Industri, 43200 Kuala Lumpur	6	Condensed Milk	500	RM 1000
2	Chong Tea Farm	019-7799797	ahchong@hotmail.com	No.55, Jalan Bestari, 75350 Ayer Keroh, Melaka	4	Black Tea Leaves	50	RM 350
2	Chong Tea Farm	019-7799797	ahchong@hotmail.com	No.55, Jalan Bestari, 75350 Ayer Keroh, Melaka	5	Green Tea Leaves	50	RM 600
3	Muthu Frozen Food	012-9761354	muthufrozen@yahoo.com	No.21, Jalan 13/1, Seksyen 13, 46200 Petaling Jaya, Selangor	12	Fries	1000	RM 5000
3	Muthu Frozen Food	012-9761354	muthufrozen@yahoo.com	No.21, Jalan 13/1, Seksyen 13, 46200 Petaling Jaya, Selangor	14	Taiwanese Hotdog	500	RM 2500
3	Muthu Frozen Food	012-9761354	muthufrozen@yahoo.com	No.21, Jalan 13/1, Seksyen 13, 46200 Petaling Jaya, Selangor	16	Chicken thigh	300	RM 4500
3	Muthu Frozen Food	012-9761354	muthufrozen@yahoo.com	No.21, Jalan 13/1, Seksyen 13, 46200 Petaling Jaya, Selangor	20	Popiah	500	RM 1500

9.4 SECOND NORMAL FORM (2NF)

To convert 1NF to 2NF, we need to eliminate partial dependencies and reassign corresponding dependent attributes in new tables.

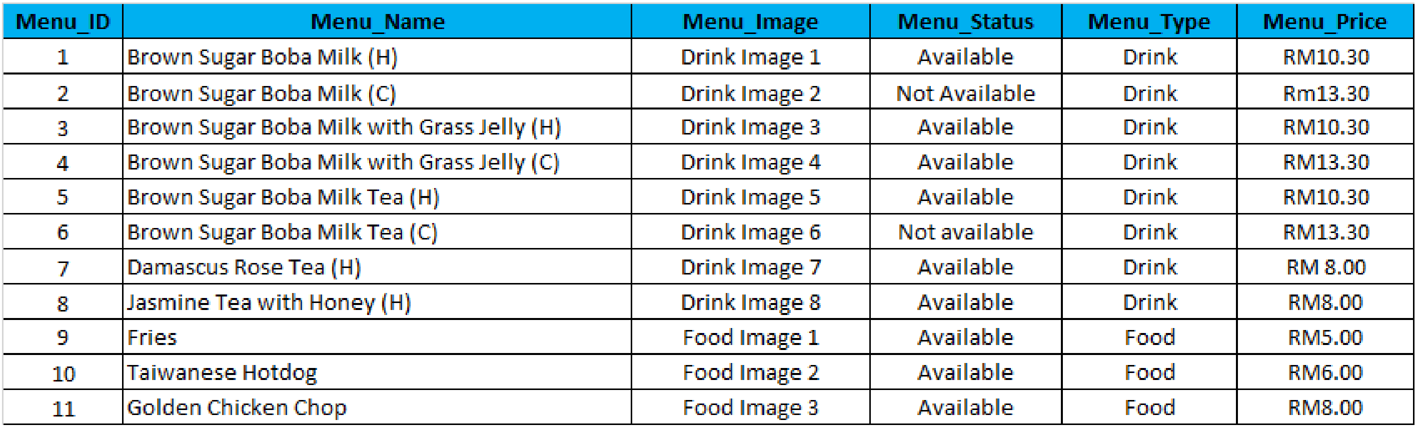
Partial Dependencies:

Menu_ID -> Menu_Name, Menu_Image, Menu_Status, Menu_Type, Menu_Price

Ingredient_ID -> Menu_Ingredient

2NF Menu Table:

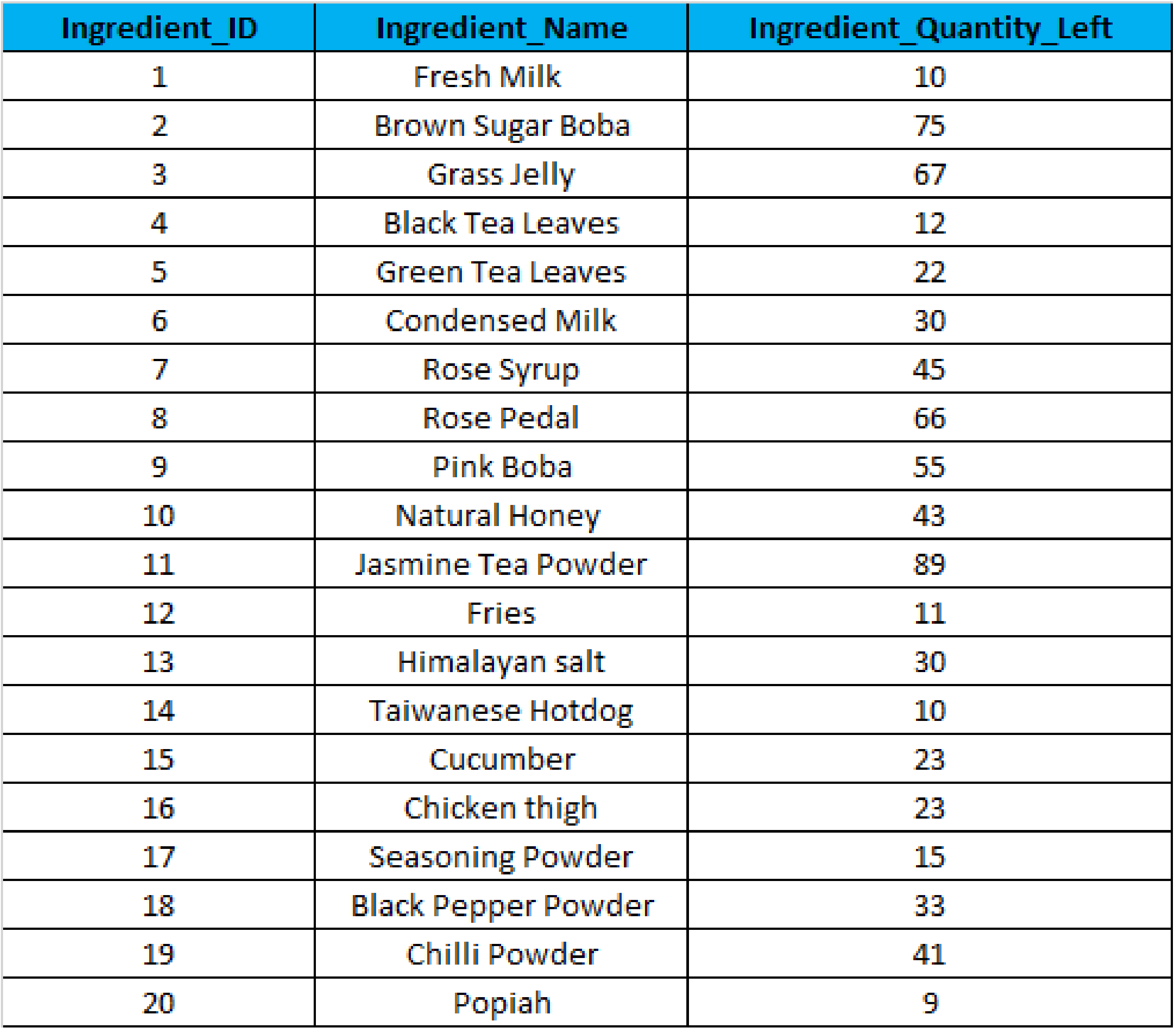
MENU(**Menu_ID**, Menu_Name, Menu_Image, Menu_Status, Menu_Type, Menu_Price)



Menu_ID	Menu_Name	Menu_Image	Menu_Status	Menu_Type	Menu_Price
1	Brown Sugar Boba Milk (H)	Drink Image 1	Available	Drink	RM10.30
2	Brown Sugar Boba Milk (C)	Drink Image 2	Not Available	Drink	Rm13.30
3	Brown Sugar Boba Milk with Grass Jelly (H)	Drink Image 3	Available	Drink	RM10.30
4	Brown Sugar Boba Milk with Grass Jelly (C)	Drink Image 4	Available	Drink	RM13.30
5	Brown Sugar Boba Milk Tea (H)	Drink Image 5	Available	Drink	RM10.30
6	Brown Sugar Boba Milk Tea (C)	Drink Image 6	Not available	Drink	RM13.30
7	Damascus Rose Tea (H)	Drink Image 7	Available	Drink	RM 8.00
8	Jasmine Tea with Honey (H)	Drink Image 8	Available	Drink	RM8.00
9	Fries	Food Image 1	Available	Food	RM5.00
10	Taiwanese Hotdog	Food Image 2	Available	Food	RM6.00
11	Golden Chicken Chop	Food Image 3	Available	Food	RM8.00

2NF Ingredient Table:

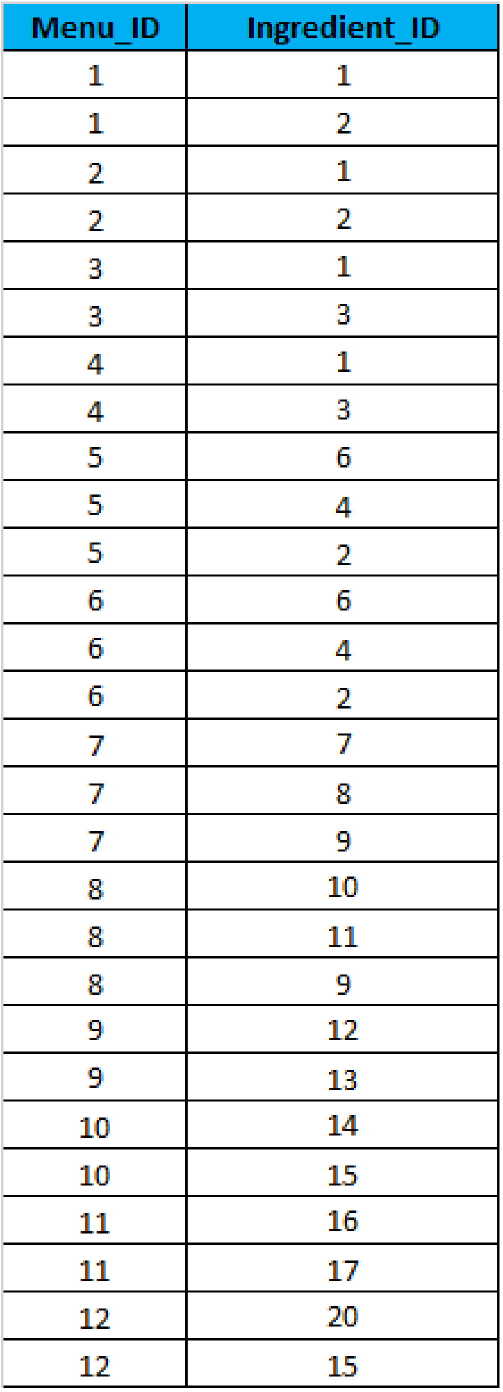
INGREDIENT(**Ingredient_ID**, Ingredient_Name, Ingredient_Quantity_Left)



Ingredient_ID	Ingredient_Name	Ingredient_Quantity_Left
1	Fresh Milk	10
2	Brown Sugar Boba	75
3	Grass Jelly	67
4	Black Tea Leaves	12
5	Green Tea Leaves	22
6	Condensed Milk	30
7	Rose Syrup	45
8	Rose Pedal	66
9	Pink Boba	55
10	Natural Honey	43
11	Jasmine Tea Powder	89
12	Fries	11
13	Himalayan salt	30
14	Taiwanese Hotdog	10
15	Cucumber	23
16	Chicken thigh	23
17	Seasoning Powder	15
18	Black Pepper Powder	33
19	Chilli Powder	41
20	Popiah	9

2NF Menu_Ingredient Table:

MENU_INGREDIENT(**Menu_ID**, **Ingredient_ID**)



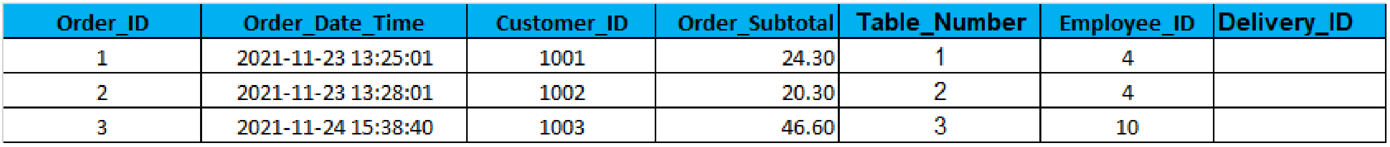
Menu_ID	Ingredient_ID
1	1
1	2
2	1
2	2
3	1
3	3
4	1
4	3
5	6
5	4
5	2
6	6
6	4
6	2
7	7
7	8
7	9
8	10
8	11
8	9
9	12
9	13
10	14
10	15
11	16
11	17
12	20
12	15

Partial dependency:

<u>Order_ID</u> -> Order_Date_Time, Customer_ID, Order_Subtotal, Table_No, Employee_ID, Delivery_ID

2NF Order Table:

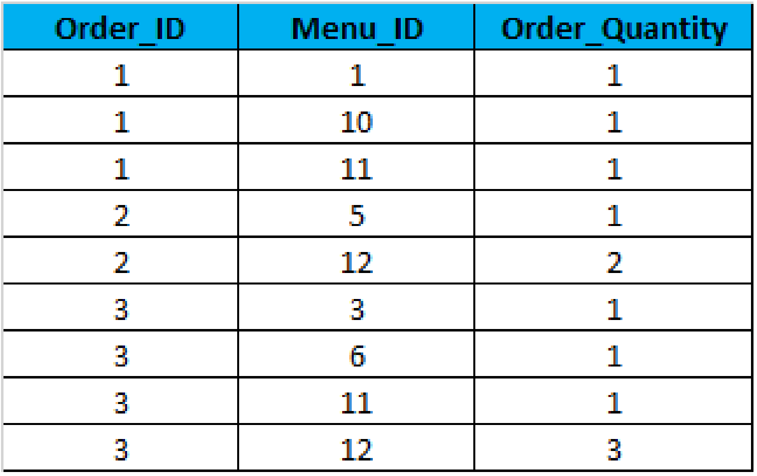
ORDER(**<u>Order_ID</u>**, Order_Date_Time, Customer_ID, Order_Subtotal, Table_No, Employee_ID, Delivery_ID)



Order_ID	Order_Date_Time	Customer_ID	Order_Subtotal	Table_Number	Employee_ID	Delivery_ID
1	2021-11-23 13:25:01	1001	24.30	1	4	
2	2021-11-23 13:28:01	1002	20.30	2	4	
3	2021-11-24 15:38:40	1003	46.60	3	10	

2NF Order_Menu Table:

ORDER_MENU(**<u>Order_ID</u>**, **<u>Menu_ID</u>**, Order_Quantity)



Order_ID	Menu_ID	Order_Quantity
1	1	1
1	10	1
1	11	1
2	5	1
2	12	2
3	3	1
3	6	1
3	11	1
3	12	3

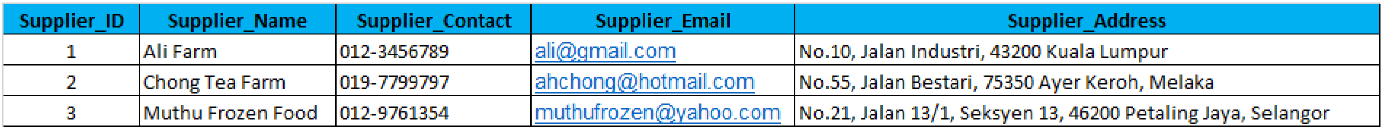
Partial Dependencies:

<u>Supplier_ID</u> -> Supplier_Name, Supplier_Contact, Supplier_Email, Supplier_Address

<u>Ingredient_ID</u> -> Ingredient_Name

2NF Supplier Table:

SUPPLIER(**Supplier_ID**, Supplier_Name, Supplier_Contact, Supplier_Email, Supplier_Address)

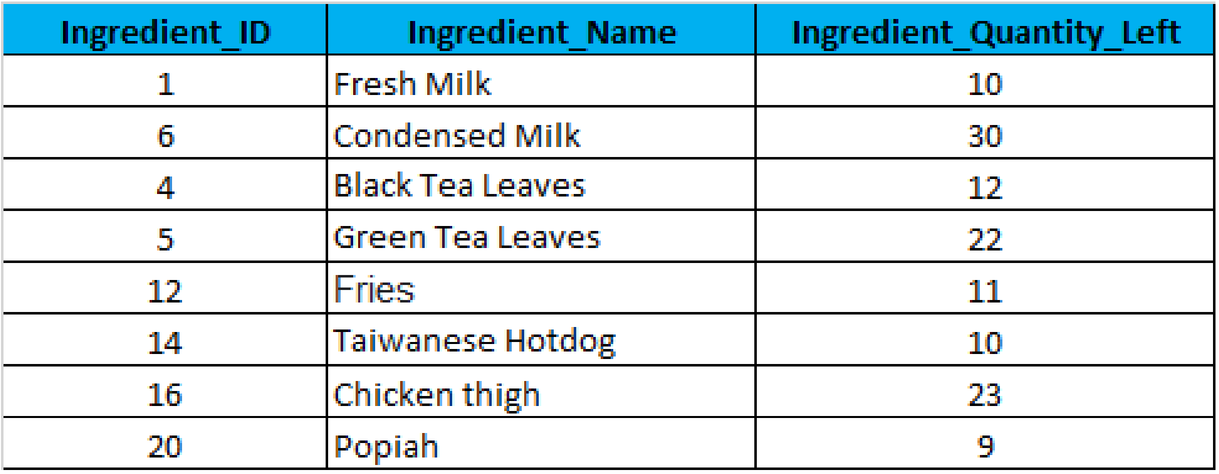


Supplier_ID	Supplier_Name	Supplier_Contact	Supplier_Email	Supplier_Address
1	Ali Farm	012-3456789	ali@gmail.com	No.10, Jalan Industri, 43200 Kuala Lumpur
2	Chong Tea Farm	019-7799797	ahchong@hotmail.com	No.55, Jalan Bestari, 75350 Ayer Keroh, Melaka
3	Muthu Frozen Food	012-9761354	muthufrozen@yahoo.com	No.21, Jalan 13/1, Seksyen 13, 46200 Petaling Jaya, Selangor

2NF Ingredient Table:

Add new attribute called Ingredient_Quantity_Left

INGREDIENT(**Ingredient_ID**, Ingredient_Name, Ingredient_Quantity_Left)

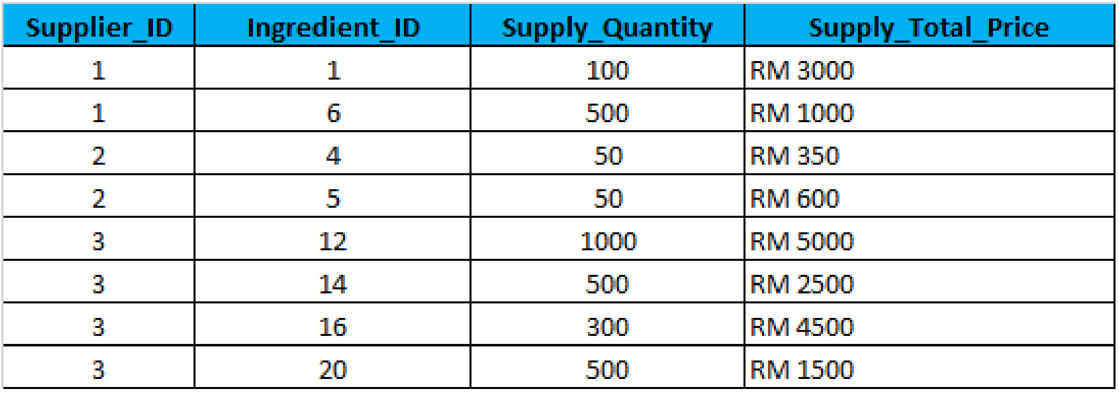


Ingredient_ID	Ingredient_Name	Ingredient_Quantity_Left
1	Fresh Milk	10
6	Condensed Milk	30
4	Black Tea Leaves	12
5	Green Tea Leaves	22
12	Fries	11
14	Taiwanese Hotdog	10
16	Chicken thigh	23
20	Popiah	9

2NF Supply Table:

Rename Ingredient_Quantity to Supply_Quantity

SUPPLY(**Supplier_ID**, **Ingredient_ID**, Supply_Quantity, Supply_Total_Price)



Supplier_ID	Ingredient_ID	Supply_Quantity	Supply_Total_Price
1	1	100	RM 3000
1	6	500	RM 1000
2	4	50	RM 350
2	5	50	RM 600
3	12	1000	RM 5000
3	14	500	RM 2500
3	16	300	RM 4500
3	20	500	RM 1500

9.5 THIRD NORMAL FORM (3NF)

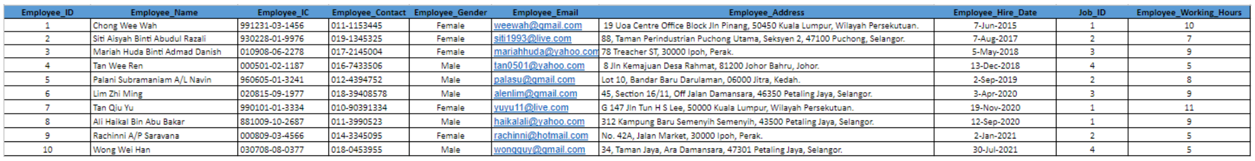
To convert 2NF to 3NF, we need to eliminate transitive dependencies and reassign corresponding dependent attributes in new tables.

Transitive Dependency:
Employee_Job_Name -> Employee_Pay_Rate

Derived Attribute:
Employee_Salary = Employee_Pay_Rate x Employee_Working_Hours

3NF Employee Table:

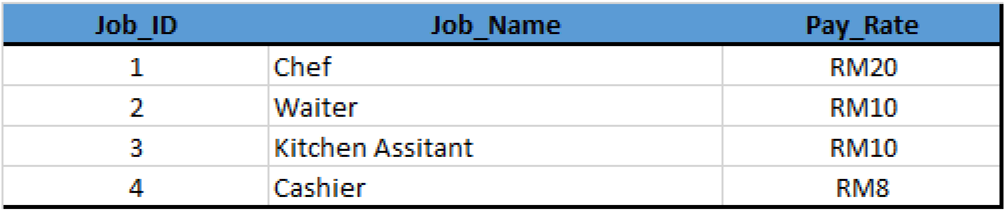
EMPLOYEE(**Employee_ID**, Employee_Name, Employee_IC, Employee_Contact, Employee_Gender, Employee_Email, Employee_Address, Employee_Hire_Date, Job_ID, Employee_Working_Hours)



Employee_ID	Employee_Name	Employee_IC	Employee_Contact	Employee_Gender	Employee_Email	Employee_Address	Employee_Hire_Date	Job_ID	Employee_Working_Hours
1	Chong Wee Wah	991231-03-1456	011-1153445	Female	weewah@gmail.com	19 Uoa Centre Office Block Jln Pinang, 50450 Kuala Lumpur, Wilayah Persekutuan.	7-Jun-2015	1	10
2	Siti Aisyah Binti Abudul Razali	930228-01-9976	019-1345325	Female	siti1993@live.com	88, Taman Perindustrian Puchong Utama, Seksyen 2, 47100 Puchong, Selangor.	7-Aug-2017	2	7
3	Mariah Huda Binti Admad Danish	010908-06-2278	017-2145004	Female	mariahhuda@yahoo.com	78 Treacher ST, 30000 Ipoh, Perak.	5-May-2018	3	9
4	Tan Wee Ren	000501-02-1187	016-7433506	Male	tan0501@yahoo.com	8 Jln Kemajuan Desa Rahmat, 81200 Johor Bahru, Johor.	13-Dec-2018	4	5
5	Palani Subramaniam A/L Navin	960605-01-3241	012-4394752	Male	palasu@gmail.com	Lot 10, Bandar Baru Darulaman, 06000 Jitra, Kedah.	2-Sep-2019	2	8
6	Lim Zhi Ming	020815-09-1977	018-39408578	Male	alenlim@gmail.com	45, Section 16/11, Off Jalan Damansara, 46350 Petaling Jaya, Selangor.	3-Apr-2020	3	9
7	Tan Qiu Yu	990101-01-3334	010-90391334	Female	yuyu11@live.com	G 147 Jln Tun H S Lee, 50000 Kuala Lumpur, Wilayah Persekutuan.	19-Nov-2020	1	11
8	Ali Haikal Bin Abu Bakar	881009-10-2687	011-3990523	Male	haikalali@yahoo.com	312 Kampung Baru Semenyih Semenyih, 43500 Petaling Jaya, Selangor.	12-Sep-2020	1	9
9	Rachinni A/P Saravana	000809-03-4566	014-3345095	Female	rachinni@hotmail.com	No. 42A, Jalan Market, 30000 Ipoh, Perak.	2-Jan-2021	2	5
10	Wong Wei Han	030708-08-0377	018-0453955	Male	wongguy@gmail.com	34, Taman Jaya, Ara Damansara, 47301 Petaling Jaya, Selangor.	30-Jul-2021	4	5

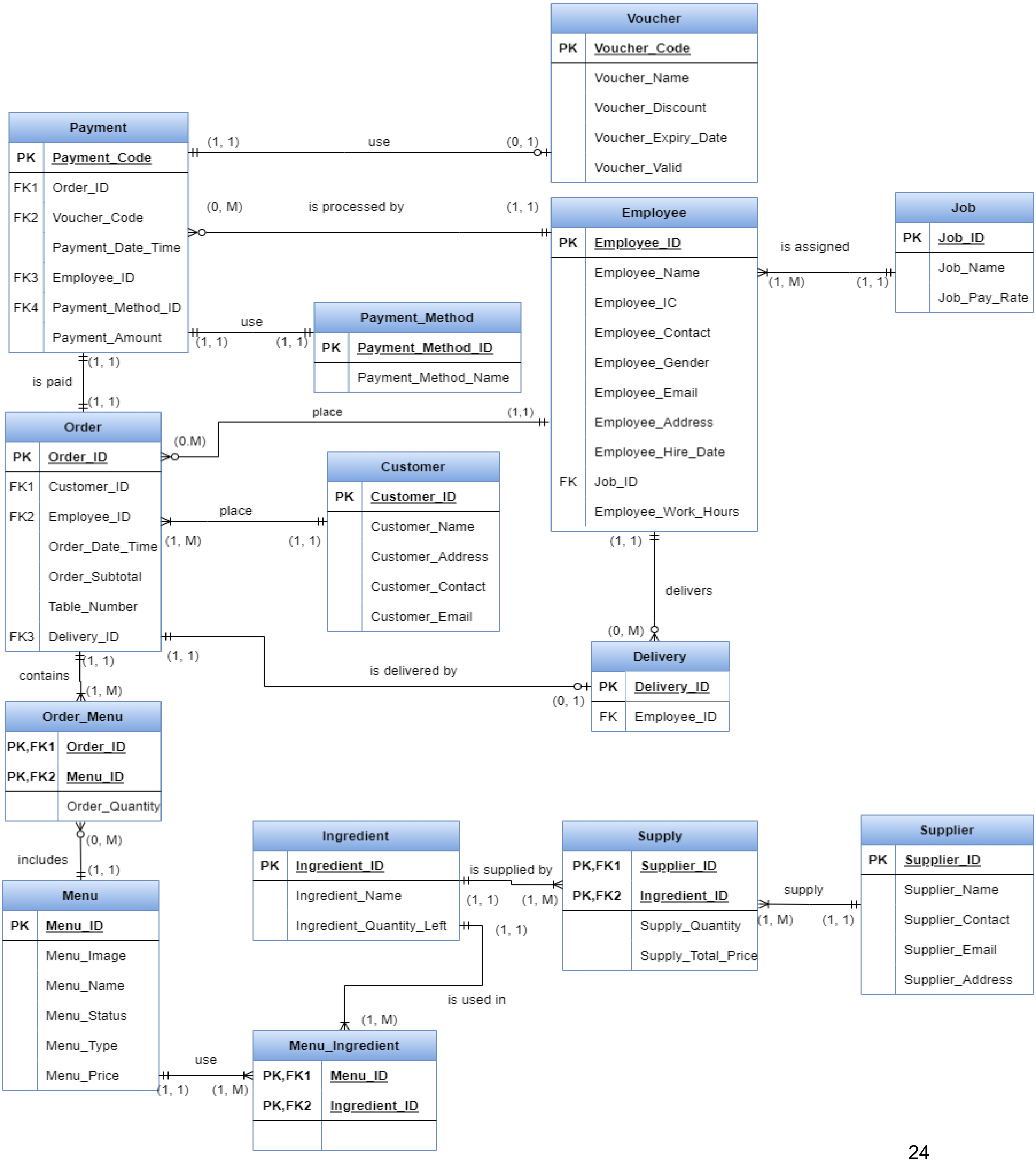
3NF Job table:

JOB(**<u>Job_ID</u>**, Job_Name, Job_Pay_Rate)



Job_ID	Job_Name	Pay_Rate
1	Chef	RM20
2	Waiter	RM10
3	Kitchen Assitant	RM10
4	Cashier	RM8

10.0 Implementable Entity Relationship Diagram



	Voucher
PK	Voucher_Code
	Voucher_Name
	Voucher_Discount
	Voucher_Expiry_Date
	Voucher_Valid

	Payment
PK	Payment_Code
FK1	Order_ID
FK2	Voucher_Code
	Payment_Date_Time
FK3	Employee_ID
FK4	Payment_Method_ID
	Payment_Amount

	Employee
PK	Employee_ID
	Employee_Name
	Employee_IC
	Employee_Contact
	Employee_Gender
	Employee_Email
	Employee_Address
	Employee_Hire_Date
FK	Job_ID
	Employee_Work_Hours

	Job
PK	Job_ID
	Job_Name
	Job_Pay_Rate

	Payment_Method
PK	Payment_Method_ID
	Payment_Method_Name

	Order
PK	Order_ID
FK1	Customer_ID
FK2	Employee_ID
	Order_Date_Time
	Order_Subtotal
	Table_Number
FK3	Delivery_ID

	Customer
PK	Customer_ID
	Customer_Name
	Customer_Address
	Customer_Contact
	Customer_Email

	Delivery
PK	Delivery_ID
FK	Employee_ID

	Order_Menu
PK,FK1	Order_ID
PK,FK2	Menu_ID
	Order_Quantity

	Ingredient
PK	Ingredient_ID
	Ingredient_Name
	Ingredient_Quantity_Left

	Supply
PK,FK1	Supplier_ID
PK,FK2	Ingredient_ID
	Supply_Quantity
	Supply_Total_Price

	Supplier
PK	Supplier_ID
	Supplier_Name
	Supplier_Contact
	Supplier_Email
	Supplier_Address

	Menu
PK	Menu_ID
	Menu_Image
	Menu_Name
	Menu_Status
	Menu_Type
	Menu_Price

	Menu_Ingredient
PK,FK1	Menu_ID
PK,FK2	Ingredient_ID

11.0 Entities and Attributes After Normalization

Entities	**Attributes**
Customer	**Customer_ID** Customer_Name Customer_Address Customer_Contact Customer_Email
Order	**Order_ID** Customer_ID Employee_ID Order_Date_Time Order_Subtotal Table_No Delivery_ID
Menu	**Menu_ID** Menu_Image Menu_Name Menu_Status Menu_Type Menu_Price
Supplier	**Supplier_ID** Supplier_Name Supplier_Contact Supplier_Email Supplier_Address
Payment	**Payment_Code** Order_ID Voucher_Code Payment_Date_Time

	Employee_ID Payment_Method_ID Payment_Amount
Voucher	**Voucher_Code** Voucher_Name Voucher_Discount Voucher_Expiry_Date Voucher_Valid
Employee	**Employee_ID** Employee_Name Employee_IC Employee_Contact Employee_Gender Employee_Email Employee_Address Employee_Hire_Date Job_ID Employee_Work_Hours
Delivery	**Delivery_ID** Employee_ID
Payment_Method	**Payment_Method_ID** Payment_Method_Name
Job	**Job_ID** Job_Name Job_Pay_Rate
Order_Menu	**Order_ID** **Menu_ID** Order_Quantity
Menu_Ingredient	**Menu_ID**

	Ingredient_ID
Ingredient	**Ingredient_ID** Ingredient_Name Ingredient_Quantity_Left
Supply	**Supplier_ID** **Ingredient_ID** Supply_Quantity Supply_Total_Price

12.0 Challenges Faced During Data Collection Process and How To Overcome

1. The data of business processes and business rules collected are not completed. We solved this problem by conducting several meetings with the manager of Ke Nina Cafe to get more information needed.

2. The information we collected by interviewing the Manager of Ke Nina Cafe is scattered and not in proper form. We need to rephrase the sentences and analyze each of them accordingly through group discussion to extract business rules.

3. It is difficult for us to conduct interviews with the manager of Ke Nina Cafe because she is busy. We do not want to bother her too much by suggesting different time slots during non-peak hours so that her working performance is not affected.

4. It is hard for all of us to have a physical meeting with the manager of Ke Nina Cafe due to distance. We use google meet to conduct the online interview session.

5. Hard to communicate with the manager of the restaurant using email. To make sure our communication more effectively, we go to the restaurant directly to discuss with the manager before interviewing

13.0 Conclusion

A proper database for Ke Nina Cafe to record every worker's working hours and the food ingredients is designed to solve the problem of the database they have now. The total salary of each worker and the number of ingredients left can be checked easily using the database that we design.

14.0 Proof of The Interview/Data Collection

14. 1 Cop:

KE NINA CAFE ★ KE NINA CAFE ★

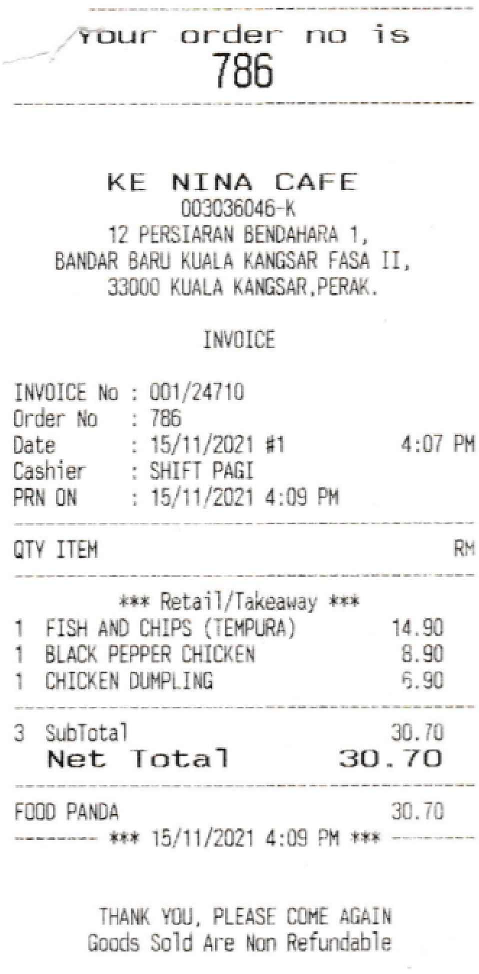
201903296314
(003036046-K)

14.2 Online Interview With Manager On Google Meet:

https://drive.google.com/file/d/1iO0J7SyT6PZVqfSfPGIfXyoK3kKWDma0/view?usp=sharing

14.3 Photos with Employer and Manager:

14.4 Sample Receipt:



Your order no is
786

KE NINA CAFE
003036046-K
12 PERSIARAN BENDAHARA 1,
BANDAR BARU KUALA KANGSAR FASA II,
33000 KUALA KANGSAR,PERAK.

INVOICE

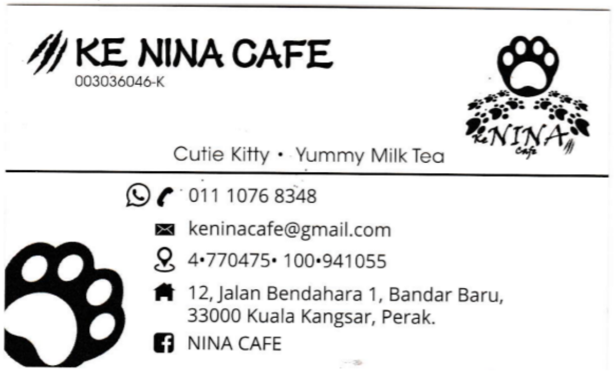
INVOICE No : 001/24710
Order No : 786
Date : 15/11/2021 #1 4:07 PM
Cashier : SHIFT PAGI
PRN ON : 15/11/2021 4:09 PM

QTY	ITEM	RM
	*** Retail/Takeaway ***	
1	FISH AND CHIPS (TEMPURA)	14.90
1	BLACK PEPPER CHICKEN	8.90
1	CHICKEN DUMPLING	6.90
3	SubTotal	30.70
	Net Total	**30.70**
	FOOD PANDA	30.70

-------- *** 15/11/2021 4:09 PM *** --------

THANK YOU, PLEASE COME AGAIN
Goods Sold Are Non Refundable

14.5 Name Card:



KE NINA CAFE
003036036046-K

Cutie Kitty • Yummy Milk Tea

Ke NINA cafe

☎ 011 1076 8348
keninacafe@gmail.com
4•770475• 100•941055
12, Jalan Bendahara 1, Bandar Baru,
33000 Kuala Kangsar, Perak.
NINA CAFE

<u>Report 2</u>

1.0 Database Design

1.1 Introduction

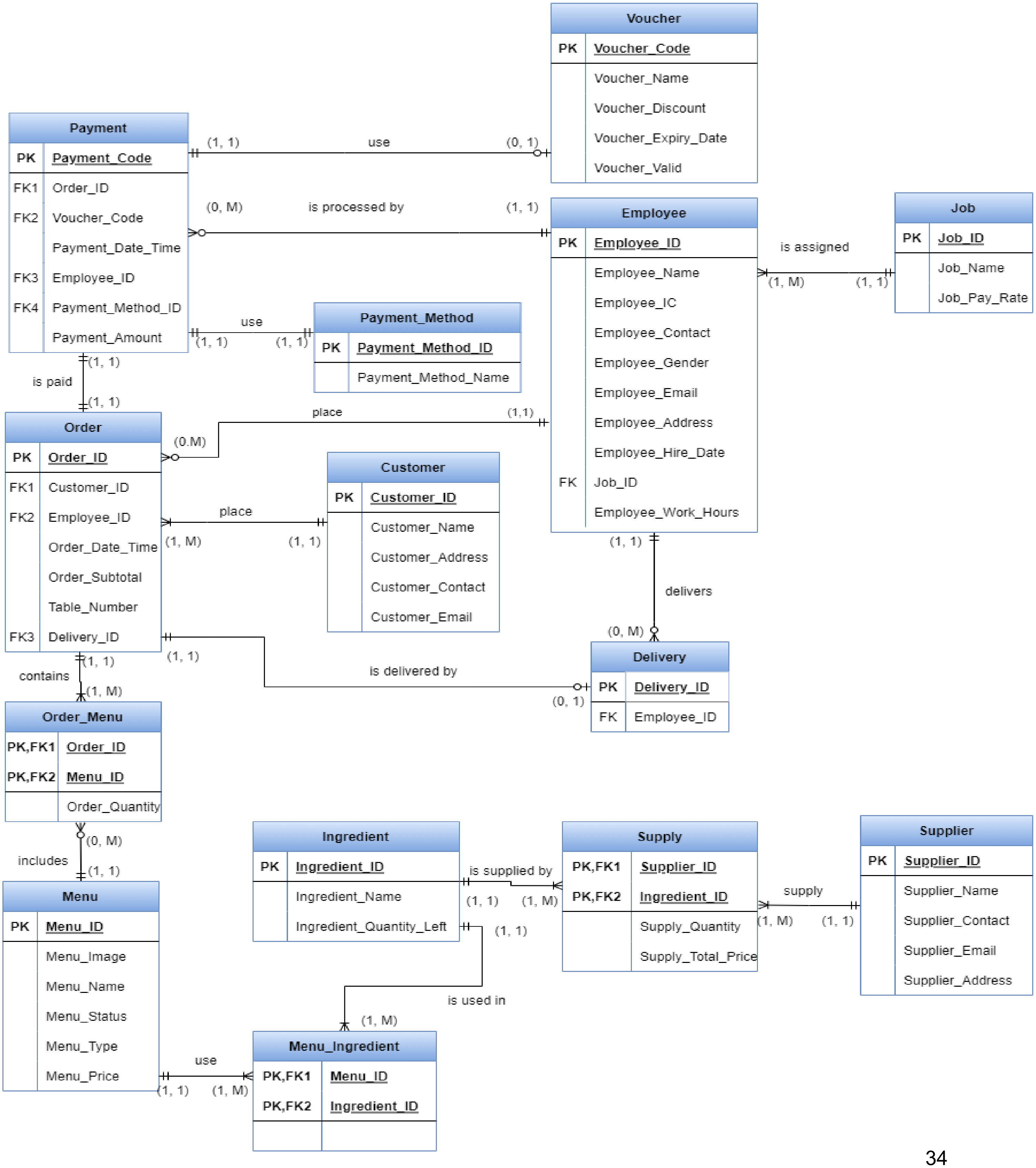
Database is a shared, integrated computer structure that stores a collection of end-user data and metadata. End-user data is the raw facts of interest to the end-user and metadata is the data about data, which the end-user data are integrated and managed. Metadata is used to describe data characteristics and relationships. In addition, databases are used by an organization as a method of storing, managing and retrieving information.

From the database management for the restaurant that we have created, there are 14 entities in total which are Vouchers, Payments, Payment_Methods, Employees, Jobs, Customers, Orders, Delivery, Order Menu, Menu, Menu_Ingredients, Ingredients, Supply and Suppliers. Each entity in the database has its own primary key whose values uniquely identify entity instances or each row in its table. The primary key attribute cannot be a null value and cannot have repeating values. Besides that, there are some of the entities in the database that have foreign keys. A foreign key is an attribute or combination of attributes in one table whose values must either match the primary key in another table or be null. The foreign keys can be applied to minimize data redundancy and establish relationships between entities. This is known as a relational database.

The main purpose of the database that we have created is for storing, managing and retrieving information about the restaurant. This database can help the employer and employees of the restaurant to do their work more efficiently. The employer can use this database to store the information of their orders, customers, employees, suppliers, the quantity of the ingredients left and more. If the employer uses spreadsheets like Excel to keep track of the food ingredients left, where the worker will need to count the ingredients every day and update them in the spreadsheet, it may lead to data anomalies. Next, the employees can also use this database to record the menu ID that is ordered by the customers and do not need to record on a piece of paper. During the peak hours, the handwriting of the workers becomes ugly and hard to read by the other workers. Besides that, there is the possibility that the workers miss some of the orders.

There are some future improvements to this restaurant database to make it better. If the employer expects to open more new branches at strategic locations to expand their business, we can improve this database so that this database can store the information of each branch of the restaurant. Besides, the managers are needed to help the employer to manage each branch. Each branch is managed by one manager so that we can improve this database to record the data of all the managers. Next, we can also improve this database to help the employer or managers to record the ingredients expiry date so that they will not need to remember or record using spreadsheets like Excel. In conclusion, we hope that we can do more improvement on this restaurant database and provide solutions for complicated and repetitive tasks to suit future needs. Hence, this database is able to meet the business standards and compete with other similar database management systems for restaurants.

1.2 Overview of ERD



1.3 Code

1.3.1 Customers

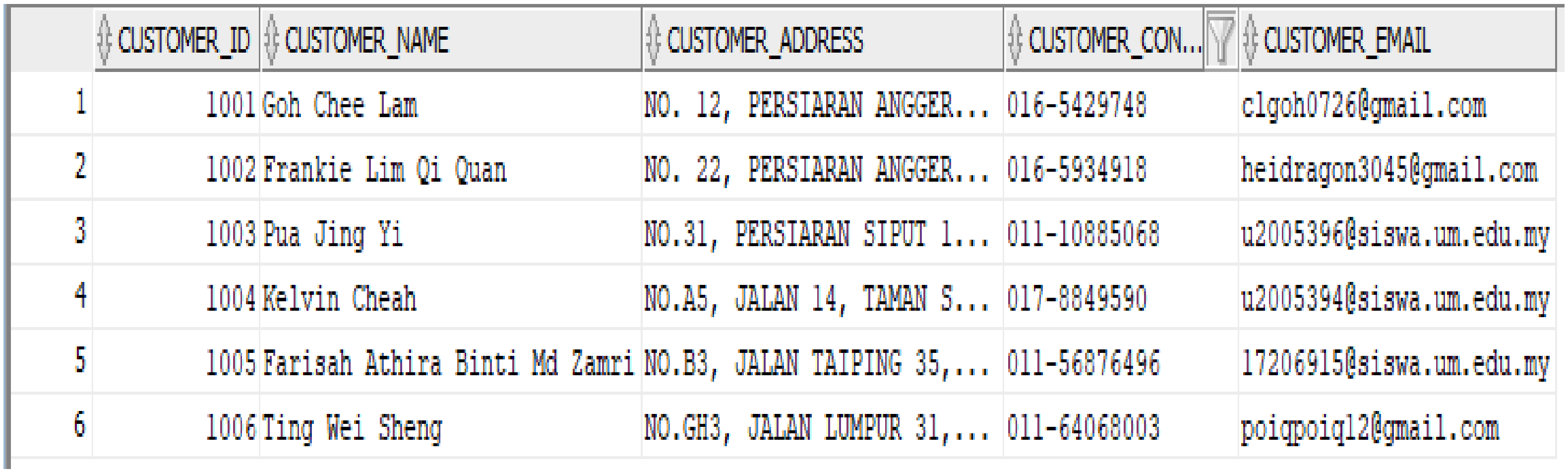
Code:

```
CREATE TABLE Customers(
        Customer_ID         INT,
        Customer_Name       VARCHAR(50)         NOT NULL,
        Customer_Address    VARCHAR(255)        NOT NULL,
        Customer_Contact    VARCHAR(20)         NOT NULL,
        Customer_Email      VARCHAR(255),
        CONSTRAINT Customers_PK
                PRIMARY KEY(Customer_ID),
        CONSTRAINT Customer_Email_CHK
                CHECK (Customer_Email LIKE '%___@___%.__%')
);
INSERT INTO CUSTOMERS (CUSTOMER_ID, CUSTOMER_NAME, CUSTOMER_ADDRESS, CUSTOMER_CONTACT, CUSTOMER_EMAIL)
VALUES (1001, 'Goh Chee Lam', 'NO. 12, PERSIARAN ANGGERIK 4, TAMAN ANGGERIK, 33000 KUALA KANGSAR, PERAK', '016-5429748', 'clgoh0726@gmail.com');


INSERT INTO CUSTOMERS (CUSTOMER_ID, CUSTOMER_NAME, CUSTOMER_ADDRESS, CUSTOMER_CONTACT, CUSTOMER_EMAIL)
VALUES (1002, 'Frankie Lim Qi Quan', 'NO. 22, PERSIARAN ANGGERIK 12, TAMAN ANGGERIK BIRU, 33000 KUALA KANGSAR, PERAK', '016-5934918', 'heidragon3045@gmail.com');


INSERT INTO CUSTOMERS (CUSTOMER_ID, CUSTOMER_NAME, CUSTOMER_ADDRESS, CUSTOMER_CONTACT, CUSTOMER_EMAIL)
VALUES (1003, 'Pua Jing Yi', 'NO.31, PERSIARAN SIPUT 10, TAMAN SIPUT, 33000 KUALA KANGSAR, PERAK', '011-10885068', 'u2005396@siswa.um.edu.my');
```

```
INSERT INTO CUSTOMERS (CUSTOMER_ID, CUSTOMER_NAME,
CUSTOMER_ADDRESS, CUSTOMER_CONTACT, CUSTOMER_EMAIL)

VALUES (1004, 'Kelvin Cheah', 'NO.A5, JALAN 14, TAMAN SUNGAI, 33000 KUALA
KANGSAR, PERAK', '017-8849590', 'u2005394@siswa.um.edu.my');
```

```
INSERT INTO CUSTOMERS (CUSTOMER_ID, CUSTOMER_NAME,
CUSTOMER_ADDRESS, CUSTOMER_CONTACT, CUSTOMER_EMAIL)

VALUES (1005, 'Farisah Athira Binti Md Zamri', 'NO.B3, JALAN TAIPING 35, TAMAN
TAIPING, 33000 KUALA KANGSAR, PERAK', '011-56876496',
'17206915@siswa.um.edu.my');
```

```
INSERT INTO CUSTOMERS (CUSTOMER_ID, CUSTOMER_NAME,
CUSTOMER_ADDRESS, CUSTOMER_CONTACT, CUSTOMER_EMAIL)

VALUES (1006, 'Ting Wei Sheng', 'NO.GH3, JALAN LUMPUR 31, TAMAN LUMPUR,
43200 DAMANSARA, KUALA LUMPUR', '011-64068003', 'poiqpoiq12@gmail.com');
```

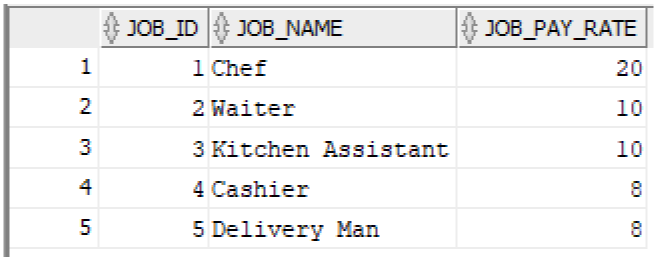


	CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_ADDRESS	CUSTOMER_CON...	CUSTOMER_EMAIL
1	1001	Goh Chee Lam	NO. 12, PERSIARAN ANGGER...	016-5429748	clgoh0726@gmail.com
2	1002	Frankie Lim Qi Quan	NO. 22, PERSIARAN ANGGER...	016-5934918	heidragon3045@gmail.com
3	1003	Pua Jing Yi	NO.31, PERSIARAN SIPUT 1...	011-10885068	u2005396@siswa.um.edu.my
4	1004	Kelvin Cheah	NO.A5, JALAN 14, TAMAN S...	017-8849590	u2005394@siswa.um.edu.my
5	1005	Farisah Athira Binti Md Zamri	NO.B3, JALAN TAIPING 35,...	011-56876496	17206915@siswa.um.edu.my
6	1006	Ting Wei Sheng	NO.GH3, JALAN LUMPUR 31,...	011-64068003	poiqpoiq12@gmail.com

1.3.2 Jobs

Code:

```
CREATE TABLE Jobs(
        Job_ID                  INT,
        Job_Name                VARCHAR(50)         NOT NULL,
        Job_Pay_Rate            NUMBER(4, 2)        NOT NULL,
        CONSTRAINT Jobs_PK
                PRIMARY KEY(Job_ID)
);
INSERT INTO JOBS (JOB_ID, JOB_NAME, JOB_PAY_RATE)
VALUES (1, 'Chef', 20);

INSERT INTO JOBS (JOB_ID, JOB_NAME, JOB_PAY_RATE)
VALUES (2, 'Waiter', 10);

INSERT INTO JOBS (JOB_ID, JOB_NAME, JOB_PAY_RATE)
VALUES (3, 'Kitchen Assistant', 10);

INSERT INTO JOBS (JOB_ID, JOB_NAME, JOB_PAY_RATE)
VALUES (4, 'Cashier', 8);

INSERT INTO JOBS (JOB_ID, JOB_NAME, JOB_PAY_RATE)
VALUES (5, 'Delivery Man', 8);
```



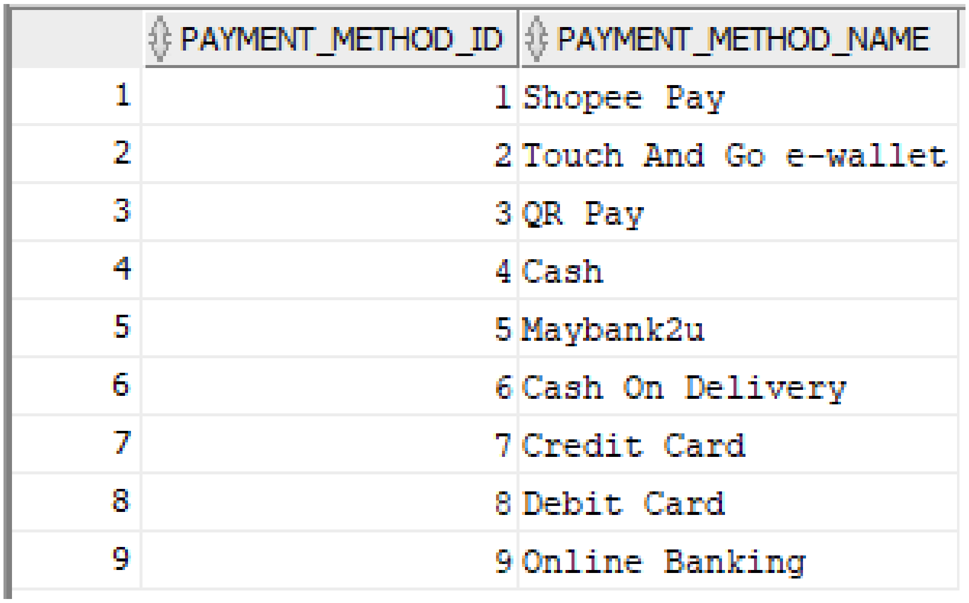
	JOB_ID	JOB_NAME	JOB_PAY_RATE
1	1	Chef	20
2	2	Waiter	10
3	3	Kitchen Assistant	10
4	4	Cashier	8
5	5	Delivery Man	8

1.3.3 Payment_Methods

Code:

```
CREATE TABLE Payment_Methods(
        Payment_Method_ID           INT,
        Payment_Method_Name         VARCHAR(100)    NOT NULL,
        CONSTRAINT Payment_Methods_PK
                PRIMARY KEY(Payment_Method_ID)
);
INSERT INTO PAYMENT_METHODS (PAYMENT_METHOD_ID,
PAYMENT_METHOD_NAME)
VALUES (1, 'Shopee Pay');


INSERT INTO PAYMENT_METHODS (PAYMENT_METHOD_ID,
PAYMENT_METHOD_NAME)
VALUES (2, 'Touch And Go e-wallet');


INSERT INTO PAYMENT_METHODS (PAYMENT_METHOD_ID,
PAYMENT_METHOD_NAME)
VALUES (3, 'QR Pay');


INSERT INTO PAYMENT_METHODS (PAYMENT_METHOD_ID,
PAYMENT_METHOD_NAME)
VALUES (4, 'Cash');


INSERT INTO PAYMENT_METHODS (PAYMENT_METHOD_ID,
PAYMENT_METHOD_NAME)
VALUES (5, 'Maybank2u');


INSERT INTO PAYMENT_METHODS (PAYMENT_METHOD_ID,
PAYMENT_METHOD_NAME)
```

```
VALUES (6, 'Cash On Delivery');

INSERT INTO PAYMENT_METHODS (PAYMENT_METHOD_ID,
PAYMENT_METHOD_NAME)
VALUES (7, 'Credit Card');

INSERT INTO PAYMENT_METHODS (PAYMENT_METHOD_ID,
PAYMENT_METHOD_NAME)
VALUES (8, 'Debit Card');

INSERT INTO PAYMENT_METHODS (PAYMENT_METHOD_ID,
PAYMENT_METHOD_NAME)
VALUES (9, 'Online Banking');
```



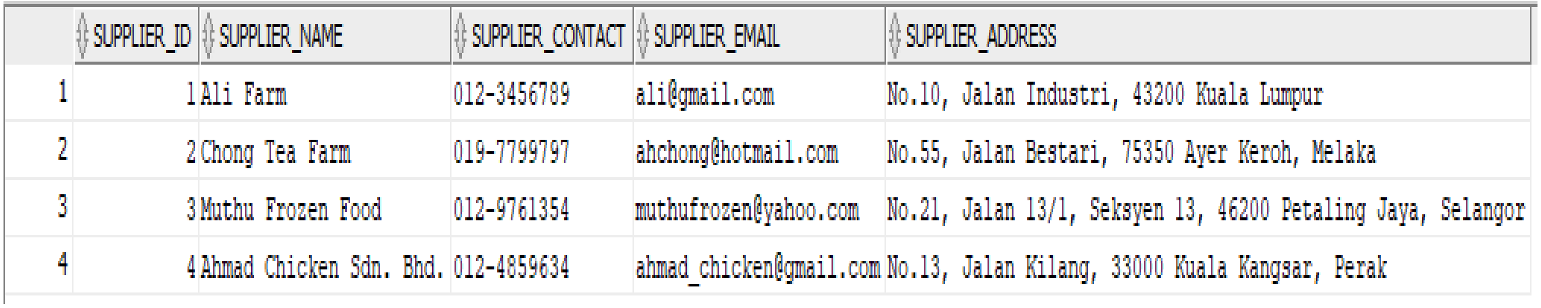
	PAYMENT_METHOD_ID	PAYMENT_METHOD_NAME
1	1	Shopee Pay
2	2	Touch And Go e-wallet
3	3	QR Pay
4	4	Cash
5	5	Maybank2u
6	6	Cash On Delivery
7	7	Credit Card
8	8	Debit Card
9	9	Online Banking

1.3.4 Supplier

Code:

```
CREATE TABLE Suppliers(
        Supplier_ID             INT,
        Supplier_Name           VARCHAR(100)        NOT NULL,
        Supplier_Contact        VARCHAR(20)         NOT NULL,
        Supplier_Email          VARCHAR(100)        NOT NULL,
        Supplier_Address        VARCHAR(255)        NOT NULL,
        CONSTRAINT Suppliers_PK
                PRIMARY KEY(Supplier_ID),
        CONSTRAINT Supplier_Email_CHK
                CHECK (Supplier_Email LIKE '%___@___%.__%')
);
INSERT INTO SUPPLIERS (SUPPLIER_ID, SUPPLIER_NAME, SUPPLIER_CONTACT,
SUPPLIER_EMAIL, SUPPLIER_ADDRESS)
VALUES (1, 'Ali Farm', '012-3456789', 'ali@gmail.com', 'No.10, Jalan Industri, 43200
Kuala Lumpur');


INSERT INTO SUPPLIERS (SUPPLIER_ID, SUPPLIER_NAME, SUPPLIER_CONTACT,
SUPPLIER_EMAIL, SUPPLIER_ADDRESS)
VALUES (2, 'Chong Tea Farm', '019-7799797', 'ahchong@hotmail.com', 'No.55, Jalan
Bestari, 75350 Ayer Keroh, Melaka');


INSERT INTO SUPPLIERS (SUPPLIER_ID, SUPPLIER_NAME, SUPPLIER_CONTACT,
SUPPLIER_EMAIL, SUPPLIER_ADDRESS)
VALUES (3, 'Muthu Frozen Food', '012-9761354', 'muthufrozen@yahoo.com', 'No.21,
Jalan 13/1, Seksyen 13, 46200 Petaling Jaya, Selangor');


INSERT INTO SUPPLIERS (SUPPLIER_ID, SUPPLIER_NAME, SUPPLIER_CONTACT,
SUPPLIER_EMAIL, SUPPLIER_ADDRESS)
```

```
VALUES (4, 'Ahmad Chicken Sdn. Bhd.', '012-4859634', 'ahmad_chicken@gmail.com',
'No.13, Jalan Kilang, 33000 Kuala Kangsar, Perak');
```



	SUPPLIER_ID	SUPPLIER_NAME	SUPPLIER_CONTACT	SUPPLIER_EMAIL	SUPPLIER_ADDRESS
1	1	Ali Farm	012-3456789	ali@gmail.com	No.10, Jalan Industri, 43200 Kuala Lumpur
2	2	Chong Tea Farm	019-7799797	ahchong@hotmail.com	No.55, Jalan Bestari, 75350 Ayer Keroh, Melaka
3	3	Muthu Frozen Food	012-9761354	muthufrozen@yahoo.com	No.21, Jalan 13/1, Seksyen 13, 46200 Petaling Jaya, Selangor
4	4	Ahmad Chicken Sdn. Bhd.	012-4859634	ahmad_chicken@gmail.com	No.13, Jalan Kilang, 33000 Kuala Kangsar, Perak

1.3.5 Ingredients

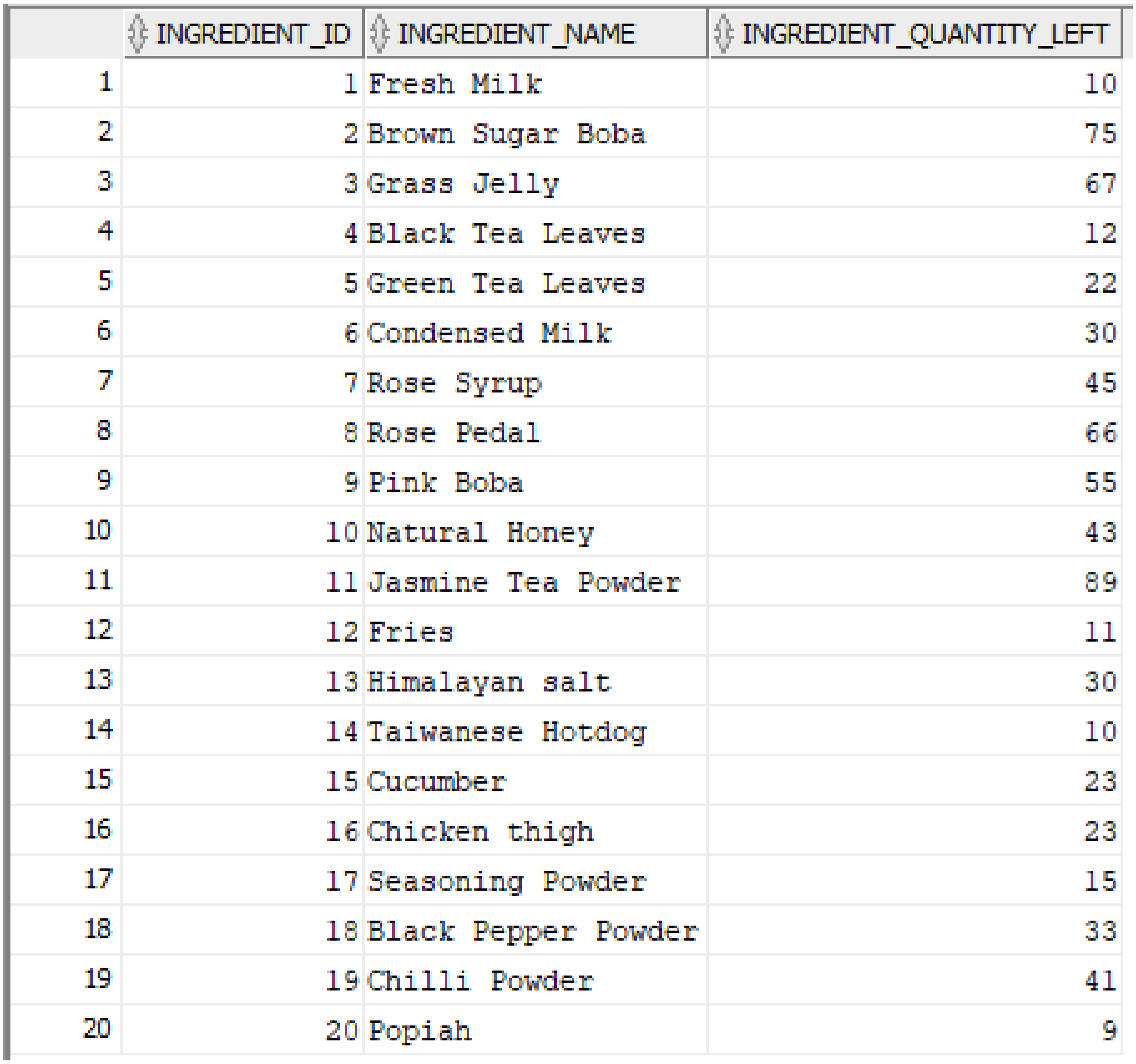
Code:

```
CREATE TABLE Ingredients(
        Ingredient_ID                INT,
        Ingredient_Name              VARCHAR(100)        NOT NULL,
        Ingredient_Quantity_Left     INT                 NOT NULL,
        CONSTRAINT Ingredients_PK
        PRIMARY KEY(Ingredient_ID)
);
INSERT INTO INGREDIENTS (INGREDIENT_ID, INGREDIENT_NAME,
INGREDIENT_QUANTITY_LEFT)
VALUES (1, 'Fresh Milk    ', 10);

INSERT INTO INGREDIENTS (INGREDIENT_ID, INGREDIENT_NAME,
INGREDIENT_QUANTITY_LEFT)
VALUES (2, 'Brown Sugar Boba', 75);

INSERT INTO INGREDIENTS (INGREDIENT_ID, INGREDIENT_NAME,
INGREDIENT_QUANTITY_LEFT)
VALUES (3, 'Grass Jelly', 67);
```

```
INSERT INTO INGREDIENTS (INGREDIENT_ID, INGREDIENT_NAME,
INGREDIENT_QUANTITY_LEFT)

VALUES (4, 'Black Tea Leaves', 12);


INSERT INTO INGREDIENTS (INGREDIENT_ID, INGREDIENT_NAME,
INGREDIENT_QUANTITY_LEFT)

VALUES (5, 'Green Tea Leaves', 22);


INSERT INTO INGREDIENTS (INGREDIENT_ID, INGREDIENT_NAME,
INGREDIENT_QUANTITY_LEFT)

VALUES (6, 'Condensed Milk', 30);


INSERT INTO INGREDIENTS (INGREDIENT_ID, INGREDIENT_NAME,
INGREDIENT_QUANTITY_LEFT)

VALUES (7, 'Rose Syrup', 45);


INSERT INTO INGREDIENTS (INGREDIENT_ID, INGREDIENT_NAME,
INGREDIENT_QUANTITY_LEFT)

VALUES (8, 'Rose Pedal', 66);


INSERT INTO INGREDIENTS (INGREDIENT_ID, INGREDIENT_NAME,
INGREDIENT_QUANTITY_LEFT)

VALUES (9, 'Pink Boba', 55);


INSERT INTO INGREDIENTS (INGREDIENT_ID, INGREDIENT_NAME,
INGREDIENT_QUANTITY_LEFT)

VALUES (10, 'Natural Honey ', 43);


INSERT INTO INGREDIENTS (INGREDIENT_ID, INGREDIENT_NAME,
INGREDIENT_QUANTITY_LEFT)

VALUES (11, 'Jasmine Tea Powder', 89);
```

```
INSERT INTO INGREDIENTS (INGREDIENT_ID, INGREDIENT_NAME,
INGREDIENT_QUANTITY_LEFT)

VALUES (12, 'Fries', 11);


INSERT INTO INGREDIENTS (INGREDIENT_ID, INGREDIENT_NAME,
INGREDIENT_QUANTITY_LEFT)

VALUES (13, 'Himalayan salt', 30);


INSERT INTO INGREDIENTS (INGREDIENT_ID, INGREDIENT_NAME,
INGREDIENT_QUANTITY_LEFT)

VALUES (14, 'Taiwanese Hotdog', 10);


INSERT INTO INGREDIENTS (INGREDIENT_ID, INGREDIENT_NAME,
INGREDIENT_QUANTITY_LEFT)

VALUES (15, 'Cucumber', 23);


INSERT INTO INGREDIENTS (INGREDIENT_ID, INGREDIENT_NAME,
INGREDIENT_QUANTITY_LEFT)

VALUES (16, 'Chicken thigh', 23);


INSERT INTO INGREDIENTS (INGREDIENT_ID, INGREDIENT_NAME,
INGREDIENT_QUANTITY_LEFT)

VALUES (17, 'Seasoning Powder', 15);


INSERT INTO INGREDIENTS (INGREDIENT_ID, INGREDIENT_NAME,
INGREDIENT_QUANTITY_LEFT)

VALUES (18, 'Black Pepper Powder', 33);


INSERT INTO INGREDIENTS (INGREDIENT_ID, INGREDIENT_NAME,
INGREDIENT_QUANTITY_LEFT)

VALUES (19, 'Chilli Powder', 41);
```

```
INSERT INTO INGREDIENTS (INGREDIENT_ID, INGREDIENT_NAME, INGREDIENT_QUANTITY_LEFT)

VALUES (20, 'Popiah', 9);
```



	INGREDIENT_ID	INGREDIENT_NAME	INGREDIENT_QUANTITY_LEFT
1	1	Fresh Milk	10
2	2	Brown Sugar Boba	75
3	3	Grass Jelly	67
4	4	Black Tea Leaves	12
5	5	Green Tea Leaves	22
6	6	Condensed Milk	30
7	7	Rose Syrup	45
8	8	Rose Pedal	66
9	9	Pink Boba	55
10	10	Natural Honey	43
11	11	Jasmine Tea Powder	89
12	12	Fries	11
13	13	Himalayan salt	30
14	14	Taiwanese Hotdog	10
15	15	Cucumber	23
16	16	Chicken thigh	23
17	17	Seasoning Powder	15
18	18	Black Pepper Powder	33
19	19	Chilli Powder	41
20	20	Popiah	9

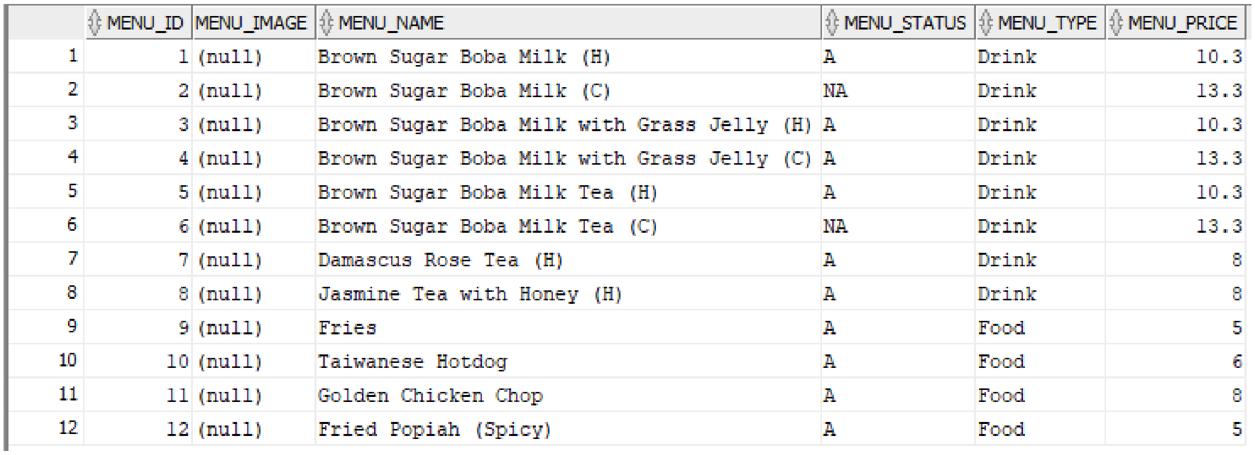
1.3.6 Menu

Code:

```
CREATE TABLE Menu(
        Menu_ID             INT,
        Menu_Image          BLOB,
        Menu_Name           VARCHAR(255)        NOT NULL,
        Menu_Status         VARCHAR(2)          NOT NULL,
        Menu_Type           VARCHAR(20)         NOT NULL,
        Menu_Price          NUMBER(4, 2)        NOT NULL,
        CONSTRAINT Menu_PK
                PRIMARY KEY(Menu_ID),
        CONSTRAINT Menu_Status_CHK
                CHECK (Menu_Status IN ('A', 'NA')),
        CONSTRAINT Menu_Price_CHK
                CHECK (Menu_Price > 0)
);
INSERT INTO MENU (MENU_ID, MENU_IMAGE, MENU_NAME, MENU_STATUS,
MENU_TYPE, MENU_PRICE)
VALUES (1, '', 'Brown Sugar Boba Milk (H)', 'A', 'Drink', 10.30);


INSERT INTO MENU (MENU_ID, MENU_IMAGE, MENU_NAME, MENU_STATUS,
MENU_TYPE, MENU_PRICE)
VALUES (2, '', 'Brown Sugar Boba Milk (C)', 'NA', 'Drink', 13.30);


INSERT INTO MENU (MENU_ID, MENU_IMAGE, MENU_NAME, MENU_STATUS,
MENU_TYPE, MENU_PRICE)
VALUES (3, '', 'Brown Sugar Boba Milk with Grass Jelly (H)', 'A', 'Drink', 10.30);


INSERT INTO MENU (MENU_ID, MENU_IMAGE, MENU_NAME, MENU_STATUS,
MENU_TYPE, MENU_PRICE)
```

```
VALUES (4, '', 'Brown Sugar Boba Milk with Grass Jelly (C)', 'A', 'Drink', 13.30);

INSERT INTO MENU (MENU_ID, MENU_IMAGE, MENU_NAME, MENU_STATUS,
MENU_TYPE, MENU_PRICE)
VALUES (5, '', 'Brown Sugar Boba Milk Tea (H)', 'A', 'Drink', 10.30);

INSERT INTO MENU (MENU_ID, MENU_IMAGE, MENU_NAME, MENU_STATUS,
MENU_TYPE, MENU_PRICE)
VALUES (6, '', 'Brown Sugar Boba Milk Tea (C)', 'NA', 'Drink', 13.30);

INSERT INTO MENU (MENU_ID, MENU_IMAGE, MENU_NAME, MENU_STATUS,
MENU_TYPE, MENU_PRICE)
VALUES (7, '', 'Damascus Rose Tea (H)', 'A', 'Drink', 8.00);

INSERT INTO MENU (MENU_ID, MENU_IMAGE, MENU_NAME, MENU_STATUS,
MENU_TYPE, MENU_PRICE)
VALUES (8, '', 'Jasmine Tea with Honey (H)', 'A', 'Drink', 8.00);

INSERT INTO MENU (MENU_ID, MENU_IMAGE, MENU_NAME, MENU_STATUS,
MENU_TYPE, MENU_PRICE)
VALUES (9, '', 'Fries', 'A', 'Food', 5.00);

INSERT INTO MENU (MENU_ID, MENU_IMAGE, MENU_NAME, MENU_STATUS,
MENU_TYPE, MENU_PRICE)
VALUES (10, '', 'Taiwanese Hotdog', 'A', 'Food', 6.00);

INSERT INTO MENU (MENU_ID, MENU_IMAGE, MENU_NAME, MENU_STATUS,
MENU_TYPE, MENU_PRICE)
VALUES (11, '', 'Golden Chicken Chop ', 'A', 'Food', 8.00);

INSERT INTO MENU (MENU_ID, MENU_IMAGE, MENU_NAME, MENU_STATUS,
MENU_TYPE, MENU_PRICE)
```

```
VALUES (12, '', 'Fried Popiah (Spicy)', 'A', 'Food', 5.00);
```



	MENU_ID	MENU_IMAGE	MENU_NAME	MENU_STATUS	MENU_TYPE	MENU_PRICE
1	1	(null)	Brown Sugar Boba Milk (H)	A	Drink	10.3
2	2	(null)	Brown Sugar Boba Milk (C)	NA	Drink	13.3
3	3	(null)	Brown Sugar Boba Milk with Grass Jelly (H)	A	Drink	10.3
4	4	(null)	Brown Sugar Boba Milk with Grass Jelly (C)	A	Drink	13.3
5	5	(null)	Brown Sugar Boba Milk Tea (H)	A	Drink	10.3
6	6	(null)	Brown Sugar Boba Milk Tea (C)	NA	Drink	13.3
7	7	(null)	Damascus Rose Tea (H)	A	Drink	8
8	8	(null)	Jasmine Tea with Honey (H)	A	Drink	8
9	9	(null)	Fries	A	Food	5
10	10	(null)	Taiwanese Hotdog	A	Food	6
11	11	(null)	Golden Chicken Chop	A	Food	8
12	12	(null)	Fried Popiah (Spicy)	A	Food	5

1.3.7 Vouchers

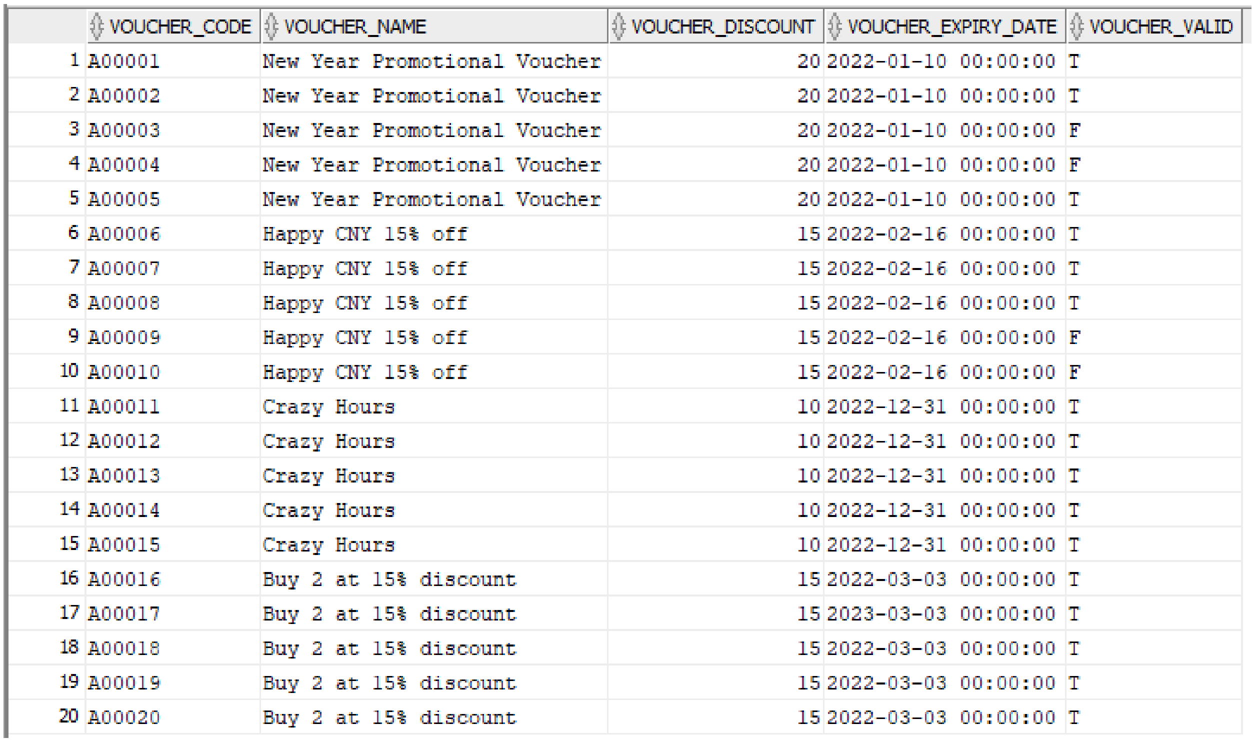
Code:

```
CREATE TABLE Vouchers(
        Voucher_Code              VARCHAR(255),
        Voucher_Name              VARCHAR(100)       NOT NULL,
        Voucher_Discount          INT                NOT NULL,
        Voucher_Expiry_Date       DATE               NOT NULL,
        Voucher_Valid             CHAR               DEFAULT 'T',
        CONSTRAINT Vouchers_PK
                PRIMARY KEY(Voucher_Code),
        CONSTRAINT Voucher_Valid_CHK
                CHECK (Voucher_Valid IN ('T', 'F'))
);
INSERT INTO VOUCHERS (VOUCHER_CODE, VOUCHER_NAME,
VOUCHER_DISCOUNT, VOUCHER_EXPIRY_DATE, VOUCHER_VALID)
VALUES ('A00001', 'New Year Promotional Voucher', 20, to_date('2022-01-10 00:00:00',
'YYYY-MM-DD:HH24:MI:SS'), 'T');
```

```
INSERT INTO VOUCHERS (VOUCHER_CODE, VOUCHER_NAME,
VOUCHER_DISCOUNT, VOUCHER_EXPIRY_DATE, VOUCHER_VALID)

VALUES ('A00002', 'New Year Promotional Voucher', 20, to_date('2022-01-10 00:00:00',
'YYYY-MM-DD:HH24:MI:SS'), 'T');


INSERT INTO VOUCHERS (VOUCHER_CODE, VOUCHER_NAME,
VOUCHER_DISCOUNT, VOUCHER_EXPIRY_DATE, VOUCHER_VALID)

VALUES ('A00003', 'New Year Promotional Voucher', 20, to_date('2022-01-10 00:00:00',
'YYYY-MM-DD:HH24:MI:SS'), 'F');


INSERT INTO VOUCHERS (VOUCHER_CODE, VOUCHER_NAME,
VOUCHER_DISCOUNT, VOUCHER_EXPIRY_DATE, VOUCHER_VALID)

VALUES ('A00004', 'New Year Promotional Voucher', 20, to_date('2022-01-10 00:00:00',
'YYYY-MM-DD:HH24:MI:SS'), 'F');


INSERT INTO VOUCHERS (VOUCHER_CODE, VOUCHER_NAME,
VOUCHER_DISCOUNT, VOUCHER_EXPIRY_DATE, VOUCHER_VALID)

VALUES ('A00005', 'New Year Promotional Voucher', 20, to_date('2022-01-10 00:00:00',
'YYYY-MM-DD:HH24:MI:SS'), 'T');


INSERT INTO VOUCHERS (VOUCHER_CODE, VOUCHER_NAME,
VOUCHER_DISCOUNT, VOUCHER_EXPIRY_DATE, VOUCHER_VALID)

VALUES ('A00006', 'Happy CNY 15% off', 15, to_date('2022-02-16 00:00:00',
'YYYY-MM-DD:HH24:MI:SS'), 'T');


INSERT INTO VOUCHERS (VOUCHER_CODE, VOUCHER_NAME,
VOUCHER_DISCOUNT, VOUCHER_EXPIRY_DATE, VOUCHER_VALID)

VALUES ('A00007', 'Happy CNY 15% off', 15, to_date('2022-02-16 00:00:00',
'YYYY-MM-DD:HH24:MI:SS'), 'T');


INSERT INTO VOUCHERS (VOUCHER_CODE, VOUCHER_NAME,
VOUCHER_DISCOUNT, VOUCHER_EXPIRY_DATE, VOUCHER_VALID)
```

```
VALUES ('A00008', 'Happy CNY 15% off', 15, to_date('2022-02-16 00:00:00',
'YYYY-MM-DD:HH24:MI:SS'), 'T');


INSERT INTO VOUCHERS (VOUCHER_CODE, VOUCHER_NAME,
VOUCHER_DISCOUNT, VOUCHER_EXPIRY_DATE, VOUCHER_VALID)
VALUES ('A00009', 'Happy CNY 15% off', 15, to_date('2022-02-16 00:00:00',
'YYYY-MM-DD:HH24:MI:SS'), 'F');


INSERT INTO VOUCHERS (VOUCHER_CODE, VOUCHER_NAME,
VOUCHER_DISCOUNT, VOUCHER_EXPIRY_DATE, VOUCHER_VALID)
VALUES ('A00010', 'Happy CNY 15% off', 15, to_date('2022-02-16 00:00:00',
'YYYY-MM-DD:HH24:MI:SS'), 'F');


INSERT INTO VOUCHERS (VOUCHER_CODE, VOUCHER_NAME,
VOUCHER_DISCOUNT, VOUCHER_EXPIRY_DATE, VOUCHER_VALID)
VALUES ('A00011', 'Crazy Hours', 10, to_date('2022-12-31 00:00:00',
'YYYY-MM-DD:HH24:MI:SS'), 'T');


INSERT INTO VOUCHERS (VOUCHER_CODE, VOUCHER_NAME,
VOUCHER_DISCOUNT, VOUCHER_EXPIRY_DATE, VOUCHER_VALID)
VALUES ('A00012', 'Crazy Hours', 10, to_date('2022-12-31 00:00:00',
'YYYY-MM-DD:HH24:MI:SS'), 'T');


INSERT INTO VOUCHERS (VOUCHER_CODE, VOUCHER_NAME,
VOUCHER_DISCOUNT, VOUCHER_EXPIRY_DATE, VOUCHER_VALID)
VALUES ('A00013', 'Crazy Hours', 10, to_date('2022-12-31 00:00:00',
'YYYY-MM-DD:HH24:MI:SS'), 'T');


INSERT INTO VOUCHERS (VOUCHER_CODE, VOUCHER_NAME,
VOUCHER_DISCOUNT, VOUCHER_EXPIRY_DATE, VOUCHER_VALID)
VALUES ('A00014', 'Crazy Hours', 10, to_date('2022-12-31 00:00:00',
'YYYY-MM-DD:HH24:MI:SS'), 'T');
```

```
INSERT INTO VOUCHERS (VOUCHER_CODE, VOUCHER_NAME,
VOUCHER_DISCOUNT, VOUCHER_EXPIRY_DATE, VOUCHER_VALID)

VALUES ('A00015', 'Crazy Hours', 10, to_date('2022-12-31 00:00:00',
'YYYY-MM-DD:HH24:MI:SS'), 'T');


INSERT INTO VOUCHERS (VOUCHER_CODE, VOUCHER_NAME,
VOUCHER_DISCOUNT, VOUCHER_EXPIRY_DATE, VOUCHER_VALID)

VALUES ('A00016', 'Buy 2 at 15% discount', 15, to_date('2022-03-03 00:00:00',
'YYYY-MM-DD:HH24:MI:SS'), 'T');


INSERT INTO VOUCHERS (VOUCHER_CODE, VOUCHER_NAME,
VOUCHER_DISCOUNT, VOUCHER_EXPIRY_DATE, VOUCHER_VALID)

VALUES ('A00017', 'Buy 2 at 15% discount', 15, to_date('2023-03-03 00:00:00',
'YYYY-MM-DD:HH24:MI:SS'), 'T');


INSERT INTO VOUCHERS (VOUCHER_CODE, VOUCHER_NAME,
VOUCHER_DISCOUNT, VOUCHER_EXPIRY_DATE, VOUCHER_VALID)

VALUES ('A00018', 'Buy 2 at 15% discount', 15, to_date('2022-03-03 00:00:00',
'YYYY-MM-DD:HH24:MI:SS'), 'T');


INSERT INTO VOUCHERS (VOUCHER_CODE, VOUCHER_NAME,
VOUCHER_DISCOUNT, VOUCHER_EXPIRY_DATE, VOUCHER_VALID)

VALUES ('A00019', 'Buy 2 at 15% discount', 15, to_date('2022-03-03 00:00:00',
'YYYY-MM-DD:HH24:MI:SS'), 'T');


INSERT INTO VOUCHERS (VOUCHER_CODE, VOUCHER_NAME,
VOUCHER_DISCOUNT, VOUCHER_EXPIRY_DATE, VOUCHER_VALID)

VALUES ('A00020', 'Buy 2 at 15% discount', 15, to_date('2022-03-03 00:00:00',
'YYYY-MM-DD:HH24:MI:SS'), 'T');
```



	VOUCHER_CODE	VOUCHER_NAME	VOUCHER_DISCOUNT	VOUCHER_EXPIRY_DATE	VOUCHER_VALID
1	A00001	New Year Promotional Voucher	20	2022-01-10 00:00:00	T
2	A00002	New Year Promotional Voucher	20	2022-01-10 00:00:00	T
3	A00003	New Year Promotional Voucher	20	2022-01-10 00:00:00	F
4	A00004	New Year Promotional Voucher	20	2022-01-10 00:00:00	F
5	A00005	New Year Promotional Voucher	20	2022-01-10 00:00:00	T
6	A00006	Happy CNY 15% off	15	2022-02-16 00:00:00	T
7	A00007	Happy CNY 15% off	15	2022-02-16 00:00:00	T
8	A00008	Happy CNY 15% off	15	2022-02-16 00:00:00	T
9	A00009	Happy CNY 15% off	15	2022-02-16 00:00:00	F
10	A00010	Happy CNY 15% off	15	2022-02-16 00:00:00	F
11	A00011	Crazy Hours	10	2022-12-31 00:00:00	T
12	A00012	Crazy Hours	10	2022-12-31 00:00:00	T
13	A00013	Crazy Hours	10	2022-12-31 00:00:00	T
14	A00014	Crazy Hours	10	2022-12-31 00:00:00	T
15	A00015	Crazy Hours	10	2022-12-31 00:00:00	T
16	A00016	Buy 2 at 15% discount	15	2022-03-03 00:00:00	T
17	A00017	Buy 2 at 15% discount	15	2023-03-03 00:00:00	T
18	A00018	Buy 2 at 15% discount	15	2022-03-03 00:00:00	T
19	A00019	Buy 2 at 15% discount	15	2022-03-03 00:00:00	T
20	A00020	Buy 2 at 15% discount	15	2022-03-03 00:00:00	T

1.3.8 Menu_Ingredients

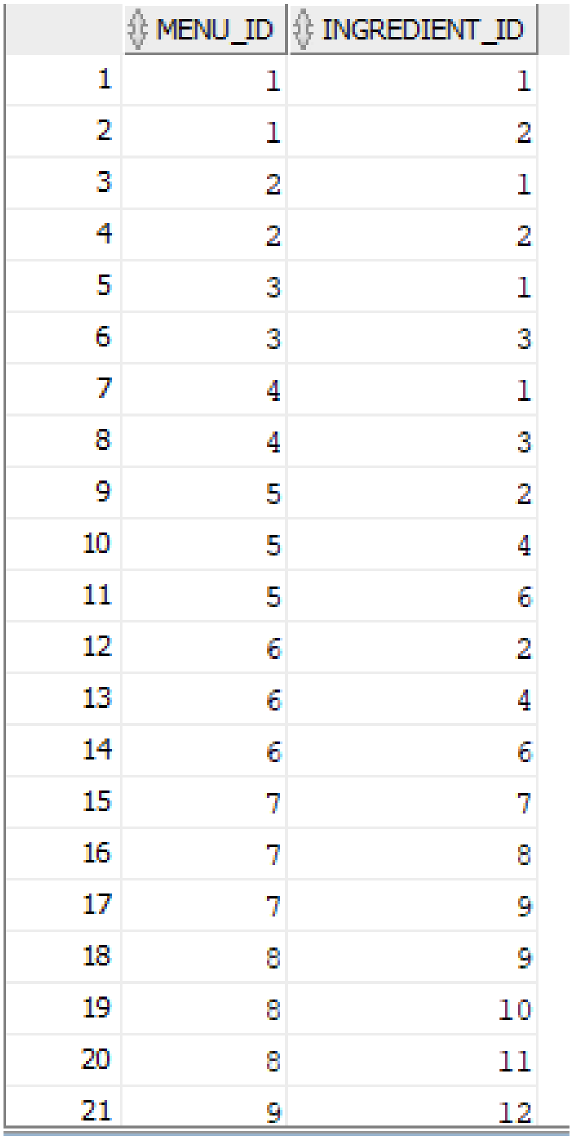
Code:

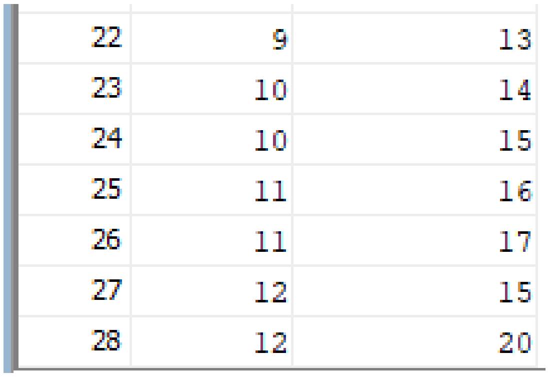
```
CREATE TABLE Menu_Ingredients(
        Menu_ID                 INT,
        Ingredient_ID           INT,
        CONSTRAINT Menu_Ingredients_PK
                PRIMARY KEY(Menu_ID, Ingredient_ID),
        CONSTRAINT MenuIngredient_Menu_FK
                FOREIGN KEY (Menu_ID)
                REFERENCES Menu(Menu_ID),
        CONSTRAINT Menu_Ingredient_Ingredient_FK
                FOREIGN KEY (Ingredient_ID)
                REFERENCES Ingredients(Ingredient_ID)
);
INSERT INTO MENU_INGREDIENTS (MENU_ID, INGREDIENT_ID)
VALUES (1, 1);

INSERT INTO MENU_INGREDIENTS (MENU_ID, INGREDIENT_ID)
VALUES (1, 2);

INSERT INTO MENU_INGREDIENTS (MENU_ID, INGREDIENT_ID)
VALUES (2, 1);

INSERT INTO MENU_INGREDIENTS (MENU_ID, INGREDIENT_ID)
VALUES (2, 2);

INSERT INTO MENU_INGREDIENTS (MENU_ID, INGREDIENT_ID)
VALUES (3, 1);
```

```
INSERT INTO MENU_INGREDIENTS (MENU_ID, INGREDIENT_ID)
VALUES (3, 3);

INSERT INTO MENU_INGREDIENTS (MENU_ID, INGREDIENT_ID)
VALUES (4, 1);

INSERT INTO MENU_INGREDIENTS (MENU_ID, INGREDIENT_ID)
VALUES (4, 3);

INSERT INTO MENU_INGREDIENTS (MENU_ID, INGREDIENT_ID)
VALUES (5, 6);

INSERT INTO MENU_INGREDIENTS (MENU_ID, INGREDIENT_ID)
VALUES (5, 4);

INSERT INTO MENU_INGREDIENTS (MENU_ID, INGREDIENT_ID)
VALUES (5, 2);

INSERT INTO MENU_INGREDIENTS (MENU_ID, INGREDIENT_ID)
VALUES (6, 6);

INSERT INTO MENU_INGREDIENTS (MENU_ID, INGREDIENT_ID)
VALUES (6, 4);

INSERT INTO MENU_INGREDIENTS (MENU_ID, INGREDIENT_ID)
VALUES (6, 2);

INSERT INTO MENU_INGREDIENTS (MENU_ID, INGREDIENT_ID)
```

```
VALUES (7, 7);

INSERT INTO MENU_INGREDIENTS (MENU_ID, INGREDIENT_ID)
VALUES (7, 8);

INSERT INTO MENU_INGREDIENTS (MENU_ID, INGREDIENT_ID)
VALUES (7, 9);

INSERT INTO MENU_INGREDIENTS (MENU_ID, INGREDIENT_ID)
VALUES (8, 10);

INSERT INTO MENU_INGREDIENTS (MENU_ID, INGREDIENT_ID)
VALUES (8, 11);

INSERT INTO MENU_INGREDIENTS (MENU_ID, INGREDIENT_ID)
VALUES (8, 9);

INSERT INTO MENU_INGREDIENTS (MENU_ID, INGREDIENT_ID)
VALUES (9, 12);

INSERT INTO MENU_INGREDIENTS (MENU_ID, INGREDIENT_ID)
VALUES (9, 13);

INSERT INTO MENU_INGREDIENTS (MENU_ID, INGREDIENT_ID)
VALUES (10, 14);

INSERT INTO MENU_INGREDIENTS (MENU_ID, INGREDIENT_ID)
VALUES (10, 15);
```

```
INSERT INTO MENU_INGREDIENTS (MENU_ID, INGREDIENT_ID)
VALUES (11, 16);

INSERT INTO MENU_INGREDIENTS (MENU_ID, INGREDIENT_ID)
VALUES (11, 17);

INSERT INTO MENU_INGREDIENTS (MENU_ID, INGREDIENT_ID)
VALUES (12, 20);

INSERT INTO MENU_INGREDIENTS (MENU_ID, INGREDIENT_ID)
VALUES (12, 15);
```





	MENU_ID	INGREDIENT_ID
1	1	1
2	1	2
3	2	1
4	2	2
5	3	1
6	3	3
7	4	1
8	4	3
9	5	2
10	5	4
11	5	6
12	6	2
13	6	4
14	6	6
15	7	7
16	7	8
17	7	9
18	8	9
19	8	10
20	8	11
21	9	12
22	9	13
23	10	14
24	10	15
25	11	16
26	11	17
27	12	15
28	12	20

1.3.9 Employees

Code:

```
CREATE TABLE Employees(
        Employee_ID                INT,
        Employee_Name              VARCHAR(50)        NOT NULL,
        Employee_IC                VARCHAR(20)        NOT NULL,
        Employee_Contact           VARCHAR(20)        NOT NULL,
        Employee_Gender            VARCHAR(1)         NOT NULL,
        Employee_Email             VARCHAR(30)        NOT NULL,
        Employee_Address           VARCHAR(100),
        Employee_Hire_Date         DATE               NOT NULL,
        Job_ID                     INT                NOT NULL,
        Employee_Work_Hours        NUMBER(4,2)        NOT NULL,
        CONSTRAINT Employees_PK
                PRIMARY KEY(Employee_ID),
        CONSTRAINT Employee_Job_FK
                FOREIGN KEY (Job_ID)
                REFERENCES Jobs(Job_ID),
        CONSTRAINT Employee_Gender_CHK
                CHECK (Employee_Gender IN ('M', 'F')),
        CONSTRAINT Employee_Email_CHK
                CHECK (Employee_Email LIKE '%___@___%.__%')
);
INSERT INTO EMPLOYEES (EMPLOYEE_ID, EMPLOYEE_NAME, EMPLOYEE_IC,
EMPLOYEE_CONTACT, EMPLOYEE_GENDER, EMPLOYEE_EMAIL,
EMPLOYEE_ADDRESS, EMPLOYEE_HIRE_DATE, JOB_ID,
EMPLOYEE_WORK_HOURS)
VALUES (1, 'Chong Wee Wah', '991231-03-1456', '011-1153445', 'F',
'weewah@gmail.com', ' 19 Uoa Centre Office Block Jln Pinang, 50450 Kuala Lumpur,
Wilayah Persekutuan.', to_date('2015-06-07', 'RRRR-MM-DD'), 1, 10);
```

```
INSERT INTO EMPLOYEES (EMPLOYEE_ID, EMPLOYEE_NAME, EMPLOYEE_IC,
EMPLOYEE_CONTACT, EMPLOYEE_GENDER, EMPLOYEE_EMAIL,
EMPLOYEE_ADDRESS, EMPLOYEE_HIRE_DATE, JOB_ID,
EMPLOYEE_WORK_HOURS)

VALUES (2, 'Siti Aisyah Binti Abudul Razali', '930228-01-9976', '019-1345325', 'F',
'siti1993@live.com', '88, Taman Perindustrian Puchong Utama, Seksyen 2, 47100
Puchong, Selangor.', to_date('2017-08-07', 'RRRR-MM-DD'), 2, 7);


INSERT INTO EMPLOYEES (EMPLOYEE_ID, EMPLOYEE_NAME, EMPLOYEE_IC,
EMPLOYEE_CONTACT, EMPLOYEE_GENDER, EMPLOYEE_EMAIL,
EMPLOYEE_ADDRESS, EMPLOYEE_HIRE_DATE, JOB_ID,
EMPLOYEE_WORK_HOURS)

VALUES (3, 'Mariah Huda Binti Admad Danish', '010908-06-2278', '017-2145004', 'F',
'mariahhuda@yahoo.com', '78 Treacher ST, 30000 Ipoh, Perak.', to_date('2018-05-05',
'RRRR-MM-DD'), 3, 9);


INSERT INTO EMPLOYEES (EMPLOYEE_ID, EMPLOYEE_NAME, EMPLOYEE_IC,
EMPLOYEE_CONTACT, EMPLOYEE_GENDER, EMPLOYEE_EMAIL,
EMPLOYEE_ADDRESS, EMPLOYEE_HIRE_DATE, JOB_ID,
EMPLOYEE_WORK_HOURS)

VALUES (4, 'Tan Wee Ren', '000501-02-1187', '016-7433506', 'M',
'tan0501@yahoo.com', ' 8 Jln Kemajuan Desa Rahmat, 81200 Johor Bahru, Johor.',
to_date('2018-12-13', 'RRRR-MM-DD'), 4, 5);


INSERT INTO EMPLOYEES (EMPLOYEE_ID, EMPLOYEE_NAME, EMPLOYEE_IC,
EMPLOYEE_CONTACT, EMPLOYEE_GENDER, EMPLOYEE_EMAIL,
EMPLOYEE_ADDRESS, EMPLOYEE_HIRE_DATE, JOB_ID,
EMPLOYEE_WORK_HOURS)

VALUES (5, 'Palani Subramaniam A/L Navin ', '960605-01-3241', '012-4394752', 'M',
'palasu@gmail.com', 'Lot 10, Bandar Baru Darulaman, 06000 Jitra, Kedah.',
to_date('2019-09-02', 'RRRR-MM-DD'), 2, 8);


INSERT INTO EMPLOYEES (EMPLOYEE_ID, EMPLOYEE_NAME, EMPLOYEE_IC,
EMPLOYEE_CONTACT, EMPLOYEE_GENDER, EMPLOYEE_EMAIL,
EMPLOYEE_ADDRESS, EMPLOYEE_HIRE_DATE, JOB_ID,
EMPLOYEE_WORK_HOURS)
```

```
VALUES (6, 'Lim Zhi Ming', '020815-09-1977', '018-39408578', 'M', 'alenlim@gmail.com',
'45, Section 16/11, Off Jalan Damansara, 46350 Petaling Jaya, Selangor.',
to_date('2020-04-03', 'RRRR-MM-DD'), 3, 9);
```

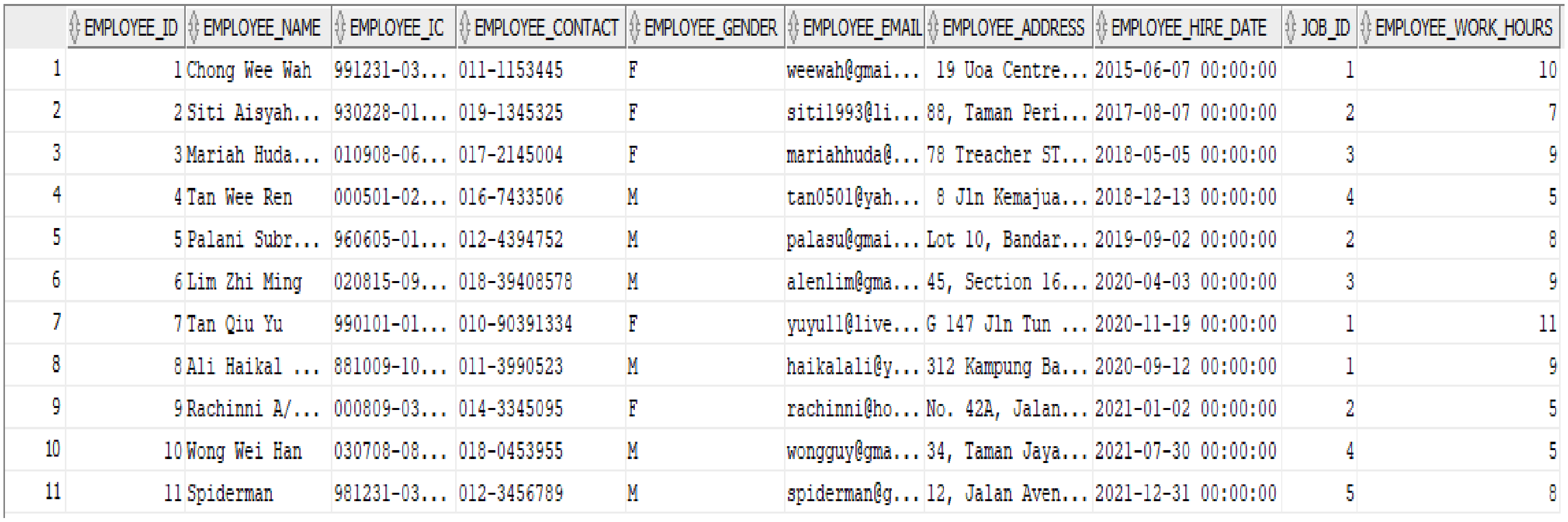
```
INSERT INTO EMPLOYEES (EMPLOYEE_ID, EMPLOYEE_NAME, EMPLOYEE_IC,
EMPLOYEE_CONTACT, EMPLOYEE_GENDER, EMPLOYEE_EMAIL,
EMPLOYEE_ADDRESS, EMPLOYEE_HIRE_DATE, JOB_ID,
EMPLOYEE_WORK_HOURS)

VALUES (7, 'Tan Qiu Yu', '990101-01-3334', '010-90391334', 'F', 'yuyu11@live.com', 'G
147 Jln Tun H S Lee, 50000 Kuala Lumpur, Wilayah Persekutuan.', to_date('2020-11-19',
'RRRR-MM-DD'), 1, 11);
```

```
INSERT INTO EMPLOYEES (EMPLOYEE_ID, EMPLOYEE_NAME, EMPLOYEE_IC,
EMPLOYEE_CONTACT, EMPLOYEE_GENDER, EMPLOYEE_EMAIL,
EMPLOYEE_ADDRESS, EMPLOYEE_HIRE_DATE, JOB_ID,
EMPLOYEE_WORK_HOURS)

VALUES (8, 'Ali Haikal Bin Abu Bakar', '881009-10-2687', '011-3990523', 'M',
'haikalali@yahoo.com', '312 Kampung Baru Semenyih Semenyih, 43500 Petaling Jaya,
Selangor.', to_date('2020-09-12', 'RRRR-MM-DD'), 1, 9);
```

```
INSERT INTO EMPLOYEES (EMPLOYEE_ID, EMPLOYEE_NAME, EMPLOYEE_IC,
EMPLOYEE_CONTACT, EMPLOYEE_GENDER, EMPLOYEE_EMAIL,
EMPLOYEE_ADDRESS, EMPLOYEE_HIRE_DATE, JOB_ID,
EMPLOYEE_WORK_HOURS)

VALUES (9, 'Rachinni A/P Saravana ', '000809-03-4566', '014-3345095', 'F',
'rachinni@hotmail.com', 'No. 42A, Jalan Market, 30000 Ipoh, Perak.',
to_date('2021-01-02', 'RRRR-MM-DD'), 2, 5);
```

```
INSERT INTO EMPLOYEES (EMPLOYEE_ID, EMPLOYEE_NAME, EMPLOYEE_IC,
EMPLOYEE_CONTACT, EMPLOYEE_GENDER, EMPLOYEE_EMAIL,
EMPLOYEE_ADDRESS, EMPLOYEE_HIRE_DATE, JOB_ID,
EMPLOYEE_WORK_HOURS)

VALUES (10, 'Wong Wei Han', '030708-08-0377', '018-0453955', 'M',
'wongguy@gmail.com', '34, Taman Jaya, Ara Damansara, 47301 Petaling Jaya,
Selangor.', to_date('2021-07-30', 'RRRR-MM-DD'), 4, 5);
```

```
INSERT INTO EMPLOYEES (EMPLOYEE_ID, EMPLOYEE_NAME, EMPLOYEE_IC,
EMPLOYEE_CONTACT, EMPLOYEE_GENDER, EMPLOYEE_EMAIL,
EMPLOYEE_ADDRESS, EMPLOYEE_HIRE_DATE, JOB_ID,
EMPLOYEE_WORK_HOURS)

VALUES (11, 'Spiderman', '981231-03-0001', '012-3456789', 'M',
'spiderman@gmail.com', '12, Jalan Avengers, Taman Marvel, 33000 Kuala Kangsar,
Perak', to_date('2021-12-31', 'RRRR-MM-DD'), 5, 8);
```

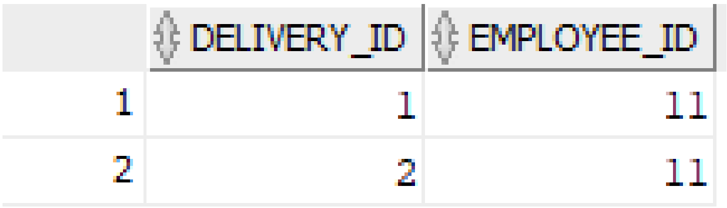


	EMPLOYEE_ID	EMPLOYEE_NAME	EMPLOYEE_IC	EMPLOYEE_CONTACT	EMPLOYEE_GENDER	EMPLOYEE_EMAIL	EMPLOYEE_ADDRESS	EMPLOYEE_HIRE_DATE	JOB_ID	EMPLOYEE_WORK_HOURS
1	1	Chong Wee Wah	991231-03...	011-1153445	F	weewah@gmai...	19 Uoa Centre...	2015-06-07 00:00:00	1	10
2	2	Siti Aisyah...	930228-01...	019-1345325	F	siti1993@li...	88, Taman Peri...	2017-08-07 00:00:00	2	7
3	3	Mariah Huda...	010908-06...	017-2145004	F	mariahhuda@...	78 Treacher ST...	2018-05-05 00:00:00	3	9
4	4	Tan Wee Ren	000501-02...	016-7433506	M	tan0501@yah...	8 Jln Kemajua...	2018-12-13 00:00:00	4	5
5	5	Palani Subr...	960605-01...	012-4394752	M	palasu@gmai...	Lot 10, Bandar...	2019-09-02 00:00:00	2	8
6	6	Lim Zhi Ming	020815-09...	018-39408578	M	alenlim@gma...	45, Section 16...	2020-04-03 00:00:00	3	9
7	7	Tan Qiu Yu	990101-01...	010-90391334	F	yuyu11@live...	G 147 Jln Tun ...	2020-11-19 00:00:00	1	11
8	8	Ali Haikal ...	881009-10...	011-3990523	M	haikalali@y...	312 Kampung Ba...	2020-09-12 00:00:00	1	9
9	9	Rachinni A/...	000809-03...	014-3345095	F	rachinni@ho...	No. 42A, Jalan...	2021-01-02 00:00:00	2	5
10	10	Wong Wei Han	030708-08...	018-0453955	M	wongguy@gma...	34, Taman Jaya...	2021-07-30 00:00:00	4	5
11	11	Spiderman	981231-03...	012-3456789	M	spiderman@g...	12, Jalan Aven...	2021-12-31 00:00:00	5	8

1.3.10 Delivery

Code:

```
CREATE TABLE Delivery(
        Delivery_ID             INT,
        Employee_ID             INT             NOT NULL,
        CONSTRAINT Delivery_PK
                PRIMARY KEY(Delivery_ID),
        CONSTRAINT Delivery_Employee_FK
                FOREIGN KEY(Employee_ID)
                REFERENCES Employees(Employee_ID)
);
INSERT INTO DELIVERY (DELIVERY_ID, EMPLOYEE_ID)
VALUES (1, 11);

INSERT INTO DELIVERY (DELIVERY_ID, EMPLOYEE_ID)
VALUES (2, 11);
```



	DELIVERY_ID	EMPLOYEE_ID
1	1	11
2	2	11

1.3.11 Orders

Code:

```
CREATE TABLE Orders(
        Order_ID            INT,
        Order_Date_Time     DATE                NOT NULL,
        Customer_ID         INT                 NOT NULL,
        Table_Number        INT,
        Delivery_ID         INT,
        Employee_ID         INT                 NOT NULL,
        Order_Subtotal      NUMBER(8, 2)        NOT NULL,
        CONSTRAINT Orders_PK
                PRIMARY KEY (Order_ID),
        CONSTRAINT Order_Customer_FK
                FOREIGN KEY (Customer_ID)
                REFERENCES Customers(Customer_ID),
        CONSTRAINT Order_Employee_FK
                FOREIGN KEY (Employee_ID)
                REFERENCES Employees(Employee_ID),
        CONSTRAINT Order_Delivery_FK
                FOREIGN KEY (Delivery_ID)
                REFERENCES Delivery(Delivery_ID)
);
INSERT INTO ORDERS (ORDER_ID, ORDER_DATE_TIME, CUSTOMER_ID,
TABLE_NUMBER, DELIVERY_ID, EMPLOYEE_ID, ORDER_SUBTOTAL)
VALUES (1, to_date('2021-11-23 13:25:01', 'YYYY-MM-DD:HH24:MI:SS'), 1001, 1, '', 4,
24.30);


INSERT INTO ORDERS (ORDER_ID, ORDER_DATE_TIME, CUSTOMER_ID,
TABLE_NUMBER, DELIVERY_ID, EMPLOYEE_ID, ORDER_SUBTOTAL)
```

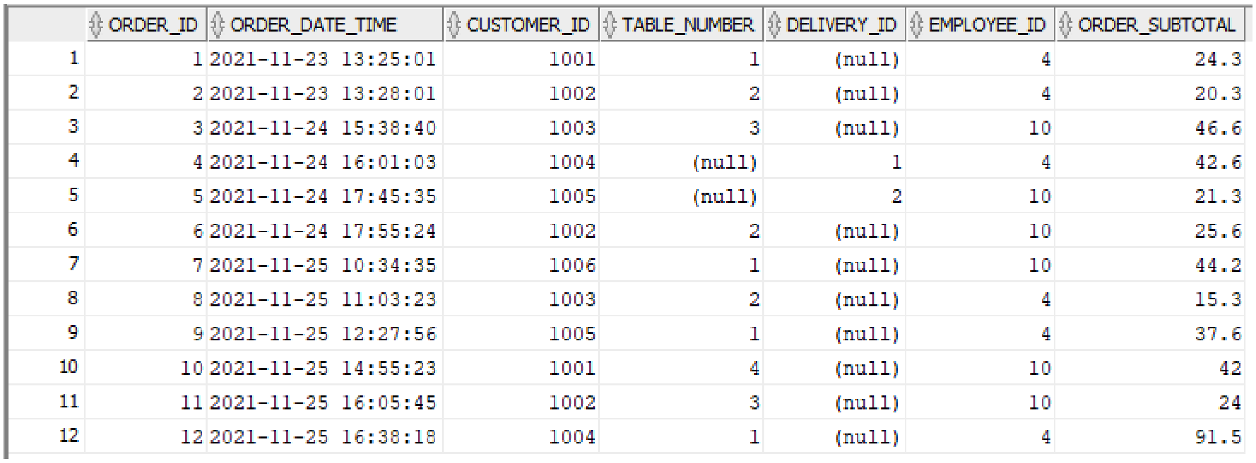
```
VALUES (2, to_date('2021-11-23 13:28:01', 'YYYY-MM-DD:HH24:MI:SS'), 1002, 2, '', 4,
20.30);


INSERT INTO ORDERS (ORDER_ID, ORDER_DATE_TIME, CUSTOMER_ID,
TABLE_NUMBER, DELIVERY_ID, EMPLOYEE_ID, ORDER_SUBTOTAL)

VALUES (3, to_date('2021-11-24 15:38:40', 'YYYY-MM-DD:HH24:MI:SS'), 1003, 3, '', 10,
46.60);


INSERT INTO ORDERS (ORDER_ID, ORDER_DATE_TIME, CUSTOMER_ID,
TABLE_NUMBER, DELIVERY_ID, EMPLOYEE_ID, ORDER_SUBTOTAL)

VALUES (4, to_date('2021-11-24 16:01:03', 'YYYY-MM-DD:HH24:MI:SS'), 1004, '', 1, 4,
42.60);


INSERT INTO ORDERS (ORDER_ID, ORDER_DATE_TIME, CUSTOMER_ID,
TABLE_NUMBER, DELIVERY_ID, EMPLOYEE_ID, ORDER_SUBTOTAL)

VALUES (5, to_date('2021-11-24 17:45:35', 'YYYY-MM-DD:HH24:MI:SS'), 1005, '', 2, 10,
21.30);


INSERT INTO ORDERS (ORDER_ID, ORDER_DATE_TIME, CUSTOMER_ID,
TABLE_NUMBER, DELIVERY_ID, EMPLOYEE_ID, ORDER_SUBTOTAL)

VALUES (6, to_date('2021-11-24 17:55:24', 'YYYY-MM-DD:HH24:MI:SS'), 1002, 2, '', 10,
25.6);


INSERT INTO ORDERS (ORDER_ID, ORDER_DATE_TIME, CUSTOMER_ID,
TABLE_NUMBER, DELIVERY_ID, EMPLOYEE_ID, ORDER_SUBTOTAL)

VALUES (7, to_date('2021-11-25 10:34:35', 'YYYY-MM-DD:HH24:MI:SS'), 1006, 1, '', 10,
44.20);


INSERT INTO ORDERS (ORDER_ID, ORDER_DATE_TIME, CUSTOMER_ID,
TABLE_NUMBER, DELIVERY_ID, EMPLOYEE_ID, ORDER_SUBTOTAL)

VALUES (8, to_date('2021-11-25 11:03:23', 'YYYY-MM-DD:HH24:MI:SS'), 1003, 2, '', 4,
15.30);
```

```
INSERT INTO ORDERS (ORDER_ID, ORDER_DATE_TIME, CUSTOMER_ID,
TABLE_NUMBER, DELIVERY_ID, EMPLOYEE_ID, ORDER_SUBTOTAL)

VALUES (9, to_date('2021-11-25 12:27:56', 'YYYY-MM-DD:HH24:MI:SS'), 1005, 1, ", 4,
37.60);
```

```
INSERT INTO ORDERS (ORDER_ID, ORDER_DATE_TIME, CUSTOMER_ID,
TABLE_NUMBER, DELIVERY_ID, EMPLOYEE_ID, ORDER_SUBTOTAL)

VALUES (10, to_date('2021-11-25 14:55:23', 'YYYY-MM-DD:HH24:MI:SS'), 1001, 4, ",
10, 42.00);
```

```
INSERT INTO ORDERS (ORDER_ID, ORDER_DATE_TIME, CUSTOMER_ID,
TABLE_NUMBER, DELIVERY_ID, EMPLOYEE_ID, ORDER_SUBTOTAL)

VALUES (11, to_date('2021-11-25 16:05:45', 'YYYY-MM-DD:HH24:MI:SS'), 1002, 3, ",
10, 24.00);
```

```
INSERT INTO ORDERS (ORDER_ID, ORDER_DATE_TIME, CUSTOMER_ID,
TABLE_NUMBER, DELIVERY_ID, EMPLOYEE_ID, ORDER_SUBTOTAL)

VALUES (12, to_date('2021-11-25 16:38:18', 'YYYY-MM-DD:HH24:MI:SS'), 1004, 1, ", 4,
91.50);
```



	ORDER_ID	ORDER_DATE_TIME	CUSTOMER_ID	TABLE_NUMBER	DELIVERY_ID	EMPLOYEE_ID	ORDER_SUBTOTAL
1	1	2021-11-23 13:25:01	1001	1	(null)	4	24.3
2	2	2021-11-23 13:28:01	1002	2	(null)	4	20.3
3	3	2021-11-24 15:38:40	1003	3	(null)	10	46.6
4	4	2021-11-24 16:01:03	1004	(null)	1	4	42.6
5	5	2021-11-24 17:45:35	1005	(null)	2	10	21.3
6	6	2021-11-24 17:55:24	1002	2	(null)	10	25.6
7	7	2021-11-25 10:34:35	1006	1	(null)	10	44.2
8	8	2021-11-25 11:03:23	1003	2	(null)	4	15.3
9	9	2021-11-25 12:27:56	1005	1	(null)	4	37.6
10	10	2021-11-25 14:55:23	1001	4	(null)	10	42
11	11	2021-11-25 16:05:45	1002	3	(null)	10	24
12	12	2021-11-25 16:38:18	1004	1	(null)	4	91.5

1.3.12 Order_Menu

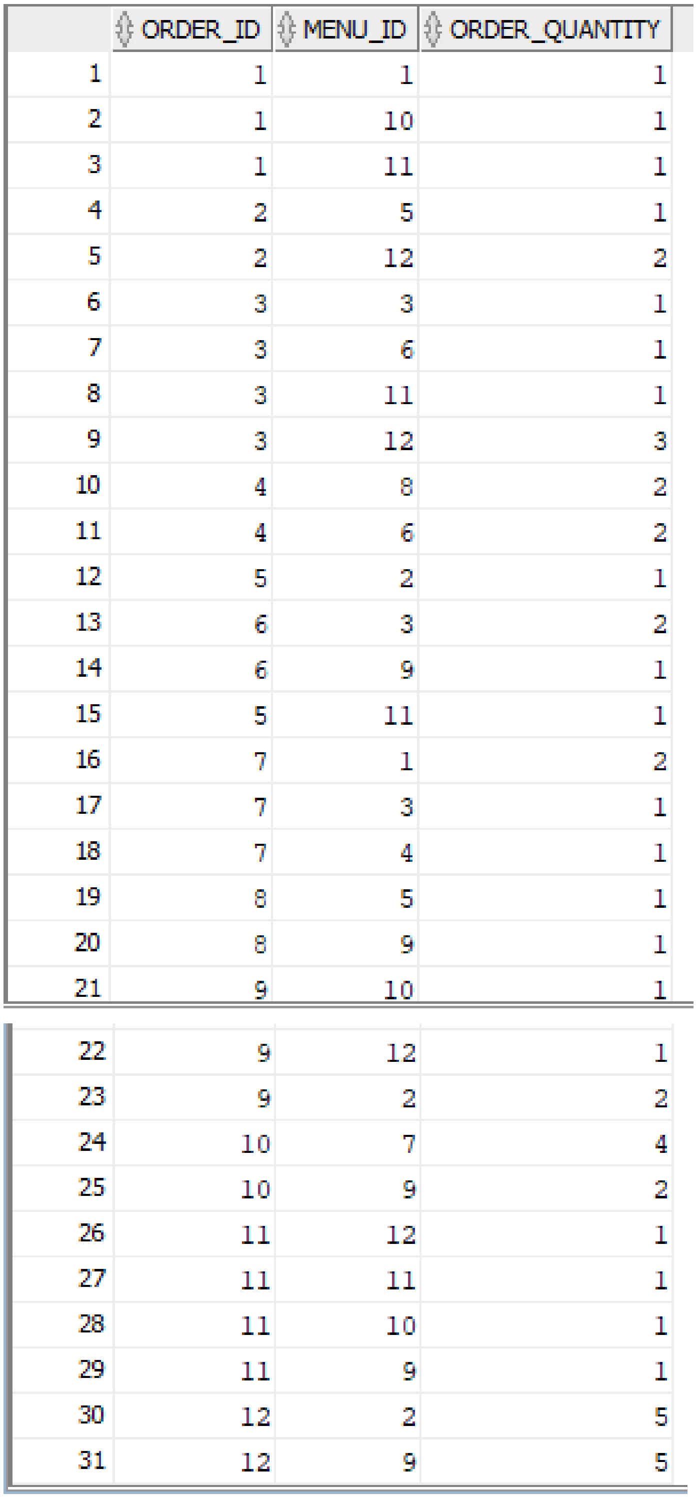
Code:

```
CREATE TABLE Order_Menu(
        Order_ID                INT         NOT NULL,
        Menu_ID                 INT         NOT NULL,
        Order_Quantity          INT         NOT NULL,
        CONSTRAINT Order_Menu_PK
                PRIMARY KEY(Order_ID, Menu_ID),
        CONSTRAINT OrderMenu_Order_FK
                FOREIGN KEY(Order_ID)
                REFERENCES Orders(Order_ID),
        CONSTRAINT OrderMenu_Menu_FK
                FOREIGN KEY(Menu_ID)
                REFERENCES Menu(Menu_ID),
        CONSTRAINT Order_Quantity_CHK
                CHECK (Order_Quantity > 0)
);
INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (1, 1, 1);

INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (1, 10, 1);

INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (1, 11, 1);

INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (2, 5, 1);
```

```
INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (2, 12, 2);

INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (3, 3, 1);

INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (3, 6, 1);

INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (3, 11, 1);

INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (3, 12, 3);

INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (4, 8, 2);

INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (4, 6, 2);

INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (5, 2, 1);

INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (6, 3, 2);

INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
```

```
VALUES (6, 9, 1);

INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (5, 11, 1);

INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (7, 1, 2);

INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (7, 3, 1);

INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (7, 4, 1);

INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (8, 5, 1);

INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (8, 9, 1);

INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (9, 10, 1);

INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (9, 12, 1);

INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (9, 2, 2);
```

```
INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (10, 7, 4);

INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (10, 9, 2);

INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (11, 12, 1);

INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (11, 11, 1);

INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (11, 10, 1);

INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (11, 9, 1);

INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (12, 2, 5);

INSERT INTO ORDER_MENU (ORDER_ID, MENU_ID, ORDER_QUANTITY)
VALUES (12, 9, 5);
```



	ORDER_ID	MENU_ID	ORDER_QUANTITY
1	1	1	1
2	1	10	1
3	1	11	1
4	2	5	1
5	2	12	2
6	3	3	1
7	3	6	1
8	3	11	1
9	3	12	3
10	4	8	2
11	4	6	2
12	5	2	1
13	6	3	2
14	6	9	1
15	5	11	1
16	7	1	2
17	7	3	1
18	7	4	1
19	8	5	1
20	8	9	1
21	9	10	1
22	9	12	1
23	9	2	2
24	10	7	4
25	10	9	2
26	11	12	1
27	11	11	1
28	11	10	1
29	11	9	1
30	12	2	5
31	12	9	5

1.3.13 Supply

Code:

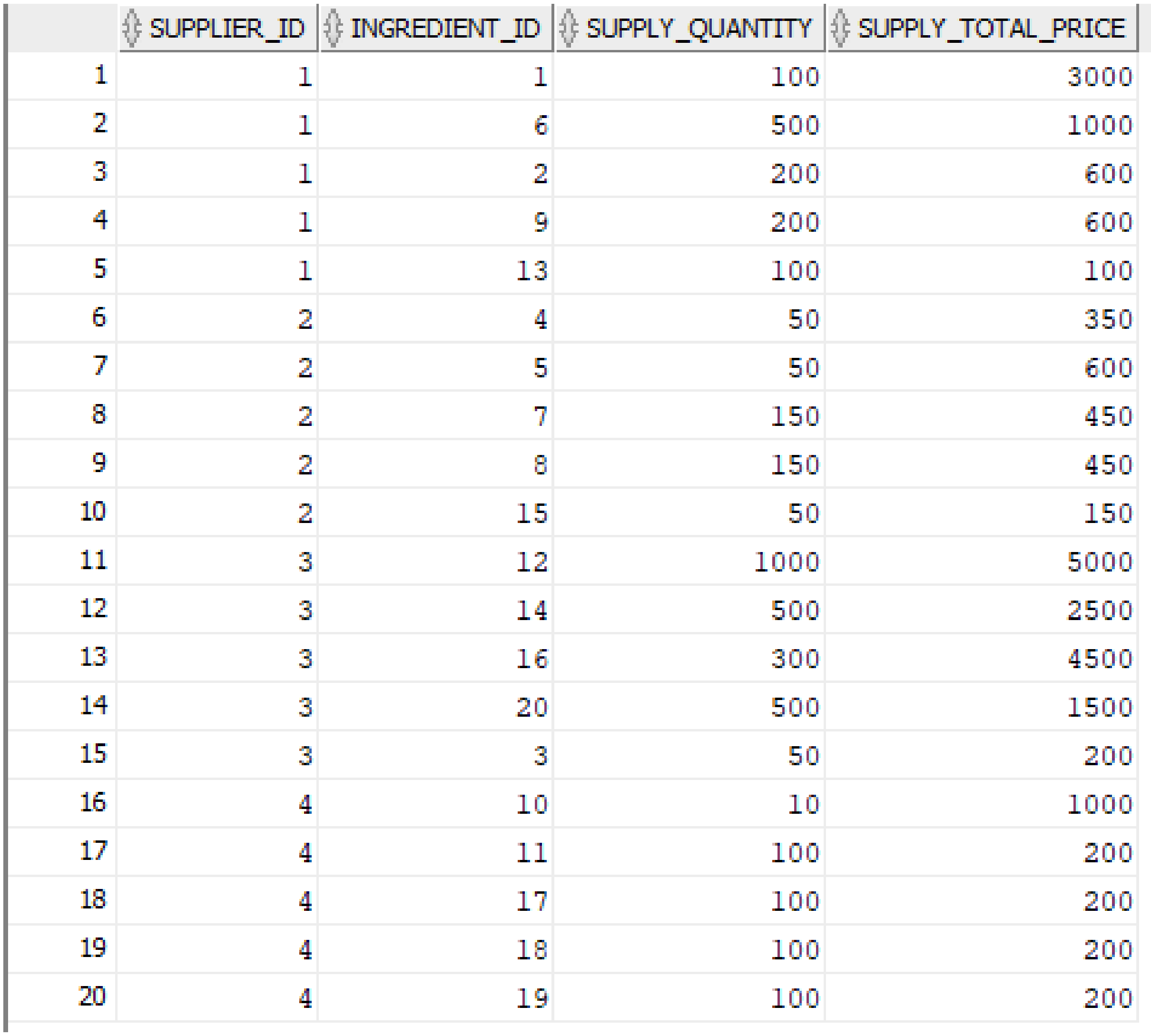
```
CREATE TABLE Supply(
    Supplier_ID         INT,
    Ingredient_ID       INT,
    Supply_Quantity     INT                 NOT NULL,
    Supply_Total_Price  NUMBER(8, 2)        NOT NULL,
    CONSTRAINT Supply_PK
        PRIMARY KEY (Supplier_ID, Ingredient_ID),
    CONSTRAINT Supply_Supplier_FK
        FOREIGN KEY (Supplier_ID)
        REFERENCES Suppliers(Supplier_ID),
    CONSTRAINT Supply_Ingredient_FK
        FOREIGN KEY (Ingredient_ID)
        REFERENCES Ingredients(Ingredient_ID),
    CONSTRAINT Supply_Quantity_CHK
        CHECK (Supply_Quantity > 0),
    CONSTRAINT Supply_Total_Price_CHK
        CHECK (Supply_Total_Price > 0)
);
INSERT INTO SUPPLY (SUPPLIER_ID, INGREDIENT_ID, SUPPLY_QUANTITY,
SUPPLY_TOTAL_PRICE)
VALUES (1, 1, 100, 3000);


INSERT INTO SUPPLY (SUPPLIER_ID, INGREDIENT_ID, SUPPLY_QUANTITY,
SUPPLY_TOTAL_PRICE)
VALUES (1, 6, 500, 1000);
```

```
INSERT INTO SUPPLY (SUPPLIER_ID, INGREDIENT_ID, SUPPLY_QUANTITY,
SUPPLY_TOTAL_PRICE)

VALUES (1, 2, 200, 600);


INSERT INTO SUPPLY (SUPPLIER_ID, INGREDIENT_ID, SUPPLY_QUANTITY,
SUPPLY_TOTAL_PRICE)

VALUES (1, 9, 200, 600);


INSERT INTO SUPPLY (SUPPLIER_ID, INGREDIENT_ID, SUPPLY_QUANTITY,
SUPPLY_TOTAL_PRICE)

VALUES (1, 13, 100, 100);


INSERT INTO SUPPLY (SUPPLIER_ID, INGREDIENT_ID, SUPPLY_QUANTITY,
SUPPLY_TOTAL_PRICE)

VALUES (2, 4, 50, 350);


INSERT INTO SUPPLY (SUPPLIER_ID, INGREDIENT_ID, SUPPLY_QUANTITY,
SUPPLY_TOTAL_PRICE)

VALUES (2, 5, 50, 600);


INSERT INTO SUPPLY (SUPPLIER_ID, INGREDIENT_ID, SUPPLY_QUANTITY,
SUPPLY_TOTAL_PRICE)

VALUES (2, 7, 150, 450);


INSERT INTO SUPPLY (SUPPLIER_ID, INGREDIENT_ID, SUPPLY_QUANTITY,
SUPPLY_TOTAL_PRICE)

VALUES (2, 8, 150, 450);


INSERT INTO SUPPLY (SUPPLIER_ID, INGREDIENT_ID, SUPPLY_QUANTITY,
SUPPLY_TOTAL_PRICE)

VALUES (2, 15, 50, 150);
```

```
INSERT INTO SUPPLY (SUPPLIER_ID, INGREDIENT_ID, SUPPLY_QUANTITY,
SUPPLY_TOTAL_PRICE)

VALUES (3, 12, 1000, 5000);


INSERT INTO SUPPLY (SUPPLIER_ID, INGREDIENT_ID, SUPPLY_QUANTITY,
SUPPLY_TOTAL_PRICE)

VALUES (3, 14, 500, 2500);


INSERT INTO SUPPLY (SUPPLIER_ID, INGREDIENT_ID, SUPPLY_QUANTITY,
SUPPLY_TOTAL_PRICE)

VALUES (3, 16, 300, 4500);


INSERT INTO SUPPLY (SUPPLIER_ID, INGREDIENT_ID, SUPPLY_QUANTITY,
SUPPLY_TOTAL_PRICE)

VALUES (3, 20, 500, 1500);


INSERT INTO SUPPLY (SUPPLIER_ID, INGREDIENT_ID, SUPPLY_QUANTITY,
SUPPLY_TOTAL_PRICE)

VALUES (3, 3, 50, 200);


INSERT INTO SUPPLY (SUPPLIER_ID, INGREDIENT_ID, SUPPLY_QUANTITY,
SUPPLY_TOTAL_PRICE)

VALUES (4, 10, 10, 1000);


INSERT INTO SUPPLY (SUPPLIER_ID, INGREDIENT_ID, SUPPLY_QUANTITY,
SUPPLY_TOTAL_PRICE)

VALUES (4, 11, 100, 200);


INSERT INTO SUPPLY (SUPPLIER_ID, INGREDIENT_ID, SUPPLY_QUANTITY,
SUPPLY_TOTAL_PRICE)

VALUES (4, 17, 100, 200);
```

```
INSERT INTO SUPPLY (SUPPLIER_ID, INGREDIENT_ID, SUPPLY_QUANTITY,
SUPPLY_TOTAL_PRICE)
VALUES (4, 18, 100, 200);
```

```
INSERT INTO SUPPLY (SUPPLIER_ID, INGREDIENT_ID, SUPPLY_QUANTITY,
SUPPLY_TOTAL_PRICE)
VALUES (4, 19, 100, 200);
```



	SUPPLIER_ID	INGREDIENT_ID	SUPPLY_QUANTITY	SUPPLY_TOTAL_PRICE
1	1	1	100	3000
2	1	6	500	1000
3	1	2	200	600
4	1	9	200	600
5	1	13	100	100
6	2	4	50	350
7	2	5	50	600
8	2	7	150	450
9	2	8	150	450
10	2	15	50	150
11	3	12	1000	5000
12	3	14	500	2500
13	3	16	300	4500
14	3	20	500	1500
15	3	3	50	200
16	4	10	10	1000
17	4	11	100	200
18	4	17	100	200
19	4	18	100	200
20	4	19	100	200

1.3.14 Payments

Code:

```
CREATE TABLE Payments(
        Payment_Code            INT,
        Order_ID                INT             NOT NULL,
        Voucher_Code            VARCHAR(255),
        Payment_Date_Time       DATE            NOT NULL,
        Employee_ID             INT             NOT NULL,
        Payment_Method_ID       INT             NOT NULL,
        Payment_Amount          NUMBER(8, 2)    NOT NULL,
        CONSTRAINT Payments_PK
                PRIMARY KEY(Payment_Code),
        CONSTRAINT Payment_Order_FK
                FOREIGN KEY(Order_ID)
                REFERENCES Orders(Order_ID),
        CONSTRAINT Payment_Voucher_FK
                FOREIGN KEY(Voucher_Code)
                REFERENCES Vouchers(Voucher_Code),
        CONSTRAINT Payment_Employee_FK
                FOREIGN KEY(Employee_ID)
                REFERENCES Employees(Employee_ID),
        CONSTRAINT Payment_Method_FK
                FOREIGN KEY(Payment_Method_ID)
                REFERENCES Payment_Methods(Payment_Method_ID)
);
INSERT INTO PAYMENTS (PAYMENT_CODE, ORDER_ID, VOUCHER_CODE,
PAYMENT_DATE_TIME, EMPLOYEE_ID, PAYMENT_METHOD_ID,
PAYMENT_AMOUNT)
```

```
VALUES (1, 1, 'A00003', to_date('2021-11-23 14:10:20', 'YYYY-MM-DD:HH24:MI:SS'), 4,
4, 19.45);


INSERT INTO PAYMENTS (PAYMENT_CODE, ORDER_ID, VOUCHER_CODE,
PAYMENT_DATE_TIME, EMPLOYEE_ID, PAYMENT_METHOD_ID,
PAYMENT_AMOUNT)

VALUES (2, 2, 'A00004', to_date('2021-11-23 15:00:03', 'YYYY-MM-DD:HH24:MI:SS'), 4,
4, 16.25);


INSERT INTO PAYMENTS (PAYMENT_CODE, ORDER_ID, VOUCHER_CODE,
PAYMENT_DATE_TIME, EMPLOYEE_ID, PAYMENT_METHOD_ID,
PAYMENT_AMOUNT)

VALUES (3, 3, 'A00009', to_date('2021-11-24 17:18:30', 'YYYY-MM-DD:HH24:MI:SS'), 4,
1, 39.60);


INSERT INTO PAYMENTS (PAYMENT_CODE, ORDER_ID, VOUCHER_CODE,
PAYMENT_DATE_TIME, EMPLOYEE_ID, PAYMENT_METHOD_ID,
PAYMENT_AMOUNT)

VALUES (4, 4, 'A00010', to_date('2021-11-24 16:30:03', 'YYYY-MM-DD:HH24:MI:SS'), 4,
5, 36.21);


INSERT INTO PAYMENTS (PAYMENT_CODE, ORDER_ID, VOUCHER_CODE,
PAYMENT_DATE_TIME, EMPLOYEE_ID, PAYMENT_METHOD_ID,
PAYMENT_AMOUNT)

VALUES (5, 5, '', to_date('2021-11-24 18:30:31', 'YYYY-MM-DD:HH24:MI:SS'), 4, 6,
21.30);


INSERT INTO PAYMENTS (PAYMENT_CODE, ORDER_ID, VOUCHER_CODE,
PAYMENT_DATE_TIME, EMPLOYEE_ID, PAYMENT_METHOD_ID,
PAYMENT_AMOUNT)

VALUES (6, 6, '', to_date('2021-11-24 19:02:24', 'YYYY-MM-DD:HH24:MI:SS'), 10, 4,
25.60);
```

```
INSERT INTO PAYMENTS (PAYMENT_CODE, ORDER_ID, VOUCHER_CODE,
PAYMENT_DATE_TIME, EMPLOYEE_ID, PAYMENT_METHOD_ID,
PAYMENT_AMOUNT)

VALUES (7, 7, '', to_date('2021-11-25 11:23:56', 'YYYY-MM-DD:HH24:MI:SS'), 10, 2,
44.20);
```

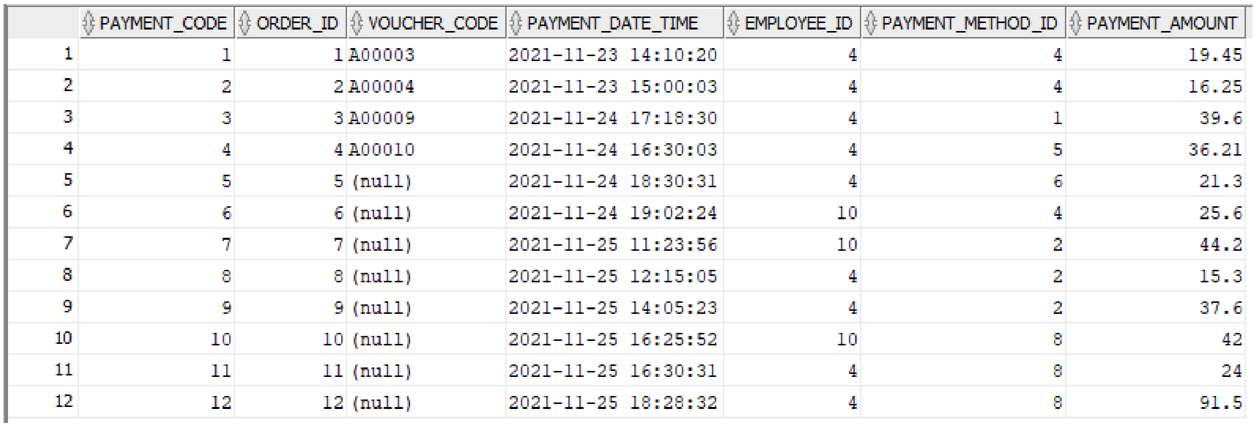
```
INSERT INTO PAYMENTS (PAYMENT_CODE, ORDER_ID, VOUCHER_CODE,
PAYMENT_DATE_TIME, EMPLOYEE_ID, PAYMENT_METHOD_ID,
PAYMENT_AMOUNT)

VALUES (8, 8, '', to_date('2021-11-25 12:15:05', 'YYYY-MM-DD:HH24:MI:SS'), 4, 2,
15.30);
```

```
INSERT INTO PAYMENTS (PAYMENT_CODE, ORDER_ID, VOUCHER_CODE,
PAYMENT_DATE_TIME, EMPLOYEE_ID, PAYMENT_METHOD_ID,
PAYMENT_AMOUNT)

VALUES (9, 9, '', to_date('2021-11-25 14:05:23', 'YYYY-MM-DD:HH24:MI:SS'), 4, 2,
37.60);
```

```
INSERT INTO PAYMENTS (PAYMENT_CODE, ORDER_ID, VOUCHER_CODE,
PAYMENT_DATE_TIME, EMPLOYEE_ID, PAYMENT_METHOD_ID,
PAYMENT_AMOUNT)

VALUES (10, 10, '', to_date('2021-11-25 16:25:52', 'YYYY-MM-DD:HH24:MI:SS'), 10, 8,
42.00);
```

```
INSERT INTO PAYMENTS (PAYMENT_CODE, ORDER_ID, VOUCHER_CODE,
PAYMENT_DATE_TIME, EMPLOYEE_ID, PAYMENT_METHOD_ID,
PAYMENT_AMOUNT)

VALUES (11, 11, '', to_date('2021-11-25 16:30:31', 'YYYY-MM-DD:HH24:MI:SS'), 4, 8,
24.00);
```

```
INSERT INTO PAYMENTS (PAYMENT_CODE, ORDER_ID, VOUCHER_CODE,
PAYMENT_DATE_TIME, EMPLOYEE_ID, PAYMENT_METHOD_ID,
PAYMENT_AMOUNT)

VALUES (12, 12, '', to_date('2021-11-25 18:28:32', 'YYYY-MM-DD:HH24:MI:SS'), 4, 8,
91.50);
```



	PAYMENT_CODE	ORDER_ID	VOUCHER_CODE	PAYMENT_DATE_TIME	EMPLOYEE_ID	PAYMENT_METHOD_ID	PAYMENT_AMOUNT
1	1	1	A00003	2021-11-23 14:10:20	4	4	19.45
2	2	2	A00004	2021-11-23 15:00:03	4	4	16.25
3	3	3	A00009	2021-11-24 17:18:30	4	1	39.6
4	4	4	A00010	2021-11-24 16:30:03	4	5	36.21
5	5	5	(null)	2021-11-24 18:30:31	4	6	21.3
6	6	6	(null)	2021-11-24 19:02:24	10	4	25.6
7	7	7	(null)	2021-11-25 11:23:56	10	2	44.2
8	8	8	(null)	2021-11-25 12:15:05	4	2	15.3
9	9	9	(null)	2021-11-25 14:05:23	4	2	37.6
10	10	10	(null)	2021-11-25 16:25:52	10	8	42
11	11	11	(null)	2021-11-25 16:30:31	4	8	24
12	12	12	(null)	2021-11-25 18:28:32	4	8	91.5

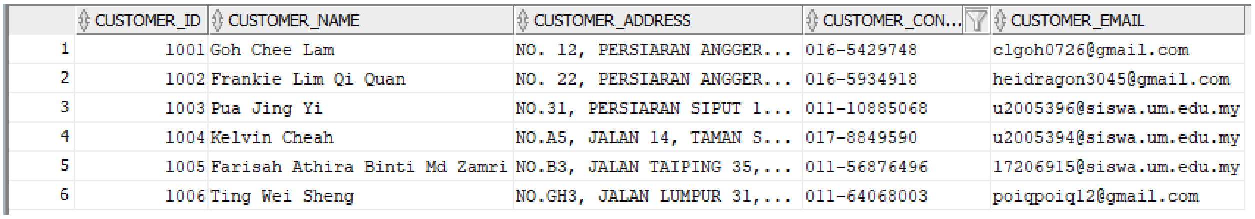
2.0 SQL Statement And Description

2.1 ALTER

Description: The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.

Situation 1: The employee wants to alter the data(column) and information in the table customers. They want to add a new column named Birthday in the Customers table.

CUSTOMERS TABLE

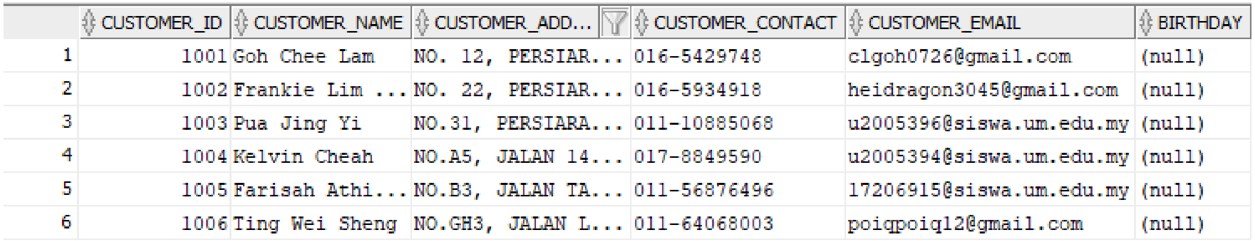


	CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_ADDRESS	CUSTOMER_CON...	CUSTOMER_EMAIL
1	1001	Goh Chee Lam	NO. 12, PERSIARAN ANGGER...	016-5429748	clgoh0726@gmail.com
2	1002	Frankie Lim Qi Quan	NO. 22, PERSIARAN ANGGER...	016-5934918	heidragon3045@gmail.com
3	1003	Pua Jing Yi	NO.31, PERSIARAN SIPUT 1...	011-10885068	u2005396@siswa.um.edu.my
4	1004	Kelvin Cheah	NO.A5, JALAN 14, TAMAN S...	017-8849590	u2005394@siswa.um.edu.my
5	1005	Farisah Athira Binti Md Zamri	NO.B3, JALAN TAIPING 35,...	011-56876496	17206915@siswa.um.edu.my
6	1006	Ting Wei Sheng	NO.GH3, JALAN LUMPUR 31,...	011-64068003	poiqpoiq12@gmail.com

<u>**SQL Statement**</u>

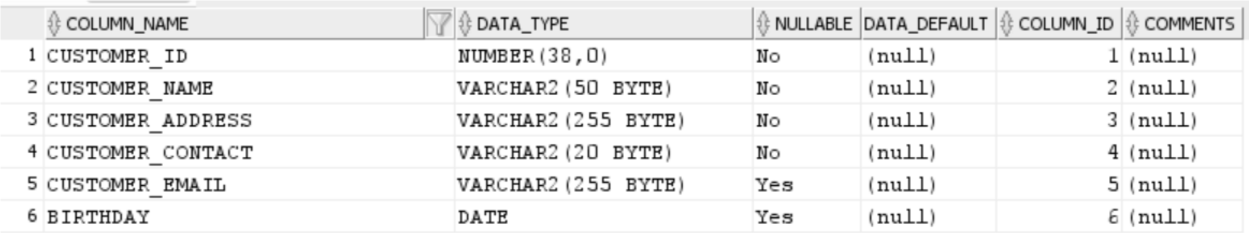
ALTER TABLE Customers

ADD Birthday date;



	CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_ADD...	CUSTOMER_CONTACT	CUSTOMER_EMAIL	BIRTHDAY
1	1001	Goh Chee Lam	NO. 12, PERSIAR...	016-5429748	clgoh0726@gmail.com	(null)
2	1002	Frankie Lim ...	NO. 22, PERSIAR...	016-5934918	heidragon3045@gmail.com	(null)
3	1003	Pua Jing Yi	NO.31, PERSIARA...	011-10885068	u2005396@siswa.um.edu.my	(null)
4	1004	Kelvin Cheah	NO.A5, JALAN 14...	017-8849590	u2005394@siswa.um.edu.my	(null)
5	1005	Farisah Athi...	NO.B3, JALAN TA...	011-56876496	17206915@siswa.um.edu.my	(null)
6	1006	Ting Wei Sheng	NO.GH3, JALAN L...	011-64068003	poiqpoiq12@gmail.com	(null)

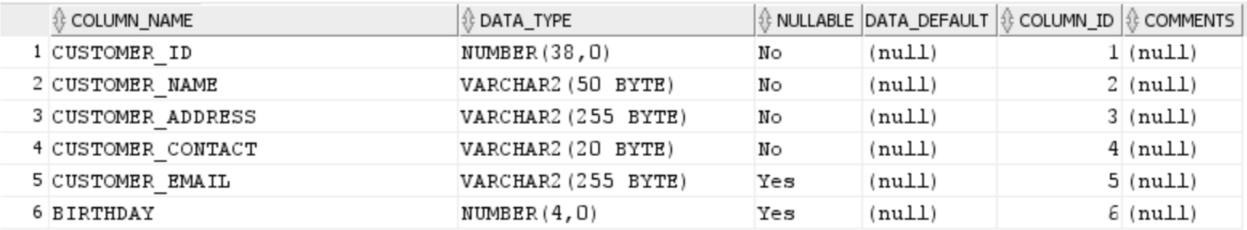
Situation 2: If the employee wants to alter the data(column) and information in the table customers. They can modify the data type if they thought the data type needed some changes.



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	CUSTOMER_ID	NUMBER(38,0)	No	(null)	1	(null)
2	CUSTOMER_NAME	VARCHAR2(50 BYTE)	No	(null)	2	(null)
3	CUSTOMER_ADDRESS	VARCHAR2(255 BYTE)	No	(null)	3	(null)
4	CUSTOMER_CONTACT	VARCHAR2(20 BYTE)	No	(null)	4	(null)
5	CUSTOMER_EMAIL	VARCHAR2(255 BYTE)	Yes	(null)	5	(null)
6	BIRTHDAY	DATE	Yes	(null)	6	(null)

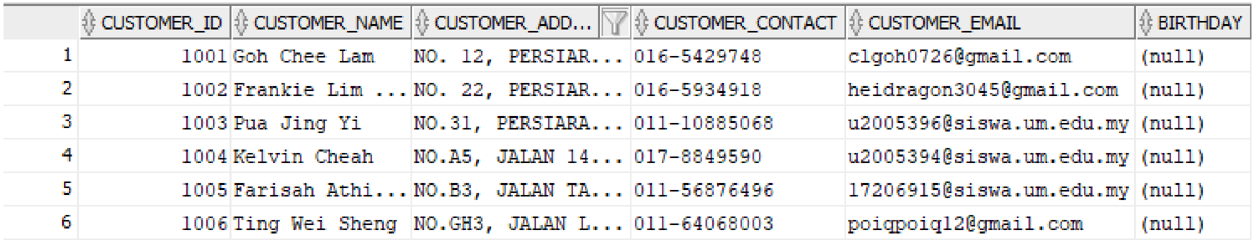
<u>SQL Statement</u>

ALTER TABLE Customers
MODIFY Birthday NUMBER(4, 0);



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	CUSTOMER_ID	NUMBER(38,0)	No	(null)	1	(null)
2	CUSTOMER_NAME	VARCHAR2(50 BYTE)	No	(null)	2	(null)
3	CUSTOMER_ADDRESS	VARCHAR2(255 BYTE)	No	(null)	3	(null)
4	CUSTOMER_CONTACT	VARCHAR2(20 BYTE)	No	(null)	4	(null)
5	CUSTOMER_EMAIL	VARCHAR2(255 BYTE)	Yes	(null)	5	(null)
6	BIRTHDAY	NUMBER(4,0)	Yes	(null)	6	(null)

Situation 3: If the employee wants to alter the data(column) and information in the table customers. They want to delete the column name Birthday in the Customers table.

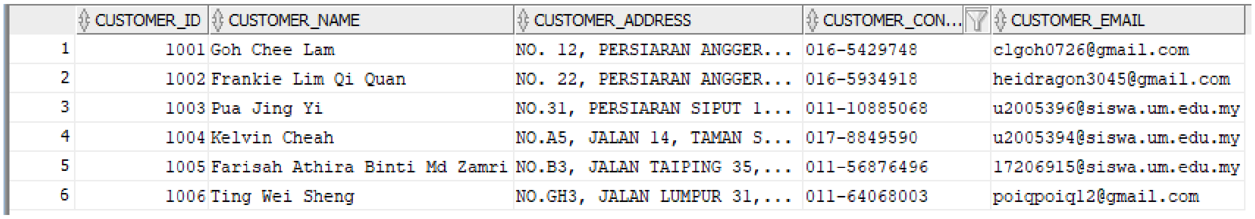


	CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_ADD...	CUSTOMER_CONTACT	CUSTOMER_EMAIL	BIRTHDAY
1	1001	Goh Chee Lam	NO. 12, PERSIAR...	016-5429748	clgoh0726@gmail.com	(null)
2	1002	Frankie Lim ...	NO. 22, PERSIAR...	016-5934918	heidragon3045@gmail.com	(null)
3	1003	Pua Jing Yi	NO.31, PERSIARA...	011-10885068	u2005396@siswa.um.edu.my	(null)
4	1004	Kelvin Cheah	NO.A5, JALAN 14...	017-8849590	u2005394@siswa.um.edu.my	(null)
5	1005	Farisah Athi...	NO.B3, JALAN TA...	011-56876496	17206915@siswa.um.edu.my	(null)
6	1006	Ting Wei Sheng	NO.GH3, JALAN L...	011-64068003	poiqpoiq12@gmail.com	(null)

SQL Statement

ALTER TABLE Customers

DROP COLUMN Birthday;



	CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_ADDRESS	CUSTOMER_CON...	CUSTOMER_EMAIL
1	1001	Goh Chee Lam	NO. 12, PERSIARAN ANGGER...	016-5429748	clgoh0726@gmail.com
2	1002	Frankie Lim Qi Quan	NO. 22, PERSIARAN ANGGER...	016-5934918	heidragon3045@gmail.com
3	1003	Pua Jing Yi	NO.31, PERSIARAN SIPUT 1...	011-10885068	u2005396@siswa.um.edu.my
4	1004	Kelvin Cheah	NO.A5, JALAN 14, TAMAN S...	017-8849590	u2005394@siswa.um.edu.my
5	1005	Farisah Athira Binti Md Zamri	NO.B3, JALAN TAIPING 35,...	011-56876496	17206915@siswa.um.edu.my
6	1006	Ting Wei Sheng	NO.GH3, JALAN LUMPUR 31,...	011-64068003	poiqpoiq12@gmail.com

2.2 LIKE, WHERE

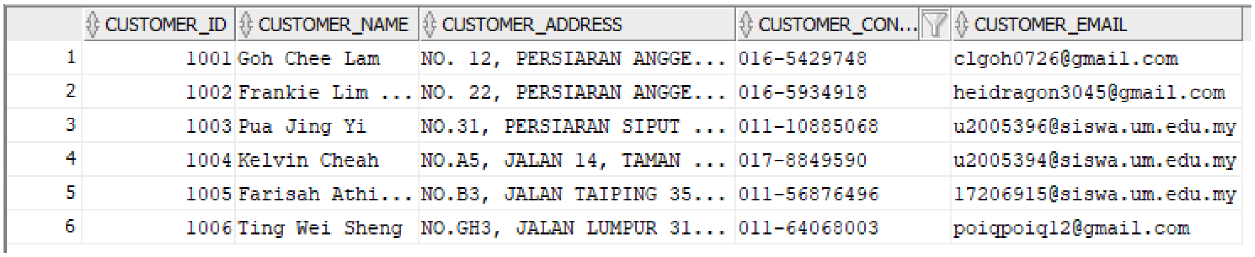
Description:

LIKE: The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.

WHERE: The WHERE clause is used to filter records. It is used to extract only those records that fulfil a specified condition.

Situation: The employer wants to search all customers with CUSTOMER_NAME starting with "Pua".

CUSTOMERS TABLE



	CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_ADDRESS	CUSTOMER_CON...	CUSTOMER_EMAIL
1	1001	Goh Chee Lam	NO. 12, PERSIARAN ANGGE...	016-5429748	clgoh0726@gmail.com
2	1002	Frankie Lim ...	NO. 22, PERSIARAN ANGGE...	016-5934918	heidragon3045@gmail.com
3	1003	Pua Jing Yi	NO.31, PERSIARAN SIPUT ...	011-10885068	u2005396@siswa.um.edu.my
4	1004	Kelvin Cheah	NO.A5, JALAN 14, TAMAN ...	017-8849590	u2005394@siswa.um.edu.my
5	1005	Farisah Athi...	NO.B3, JALAN TAIPING 35...	011-56876496	17206915@siswa.um.edu.my
6	1006	Ting Wei Sheng	NO.GH3, JALAN LUMPUR 31...	011-64068003	poiqpoiq12@gmail.com

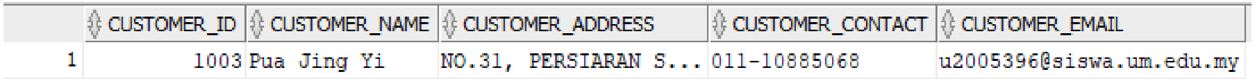
SQL Statement

SELECT *

FROM CUSTOMERS

WHERE Customer_Name

LIKE '%Pua%';



	CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_ADDRESS	CUSTOMER_CONTACT	CUSTOMER_EMAIL
1	1003	Pua Jing Yi	NO.31, PERSIARAN S...	011-10885068	u2005396@siswa.um.edu.my

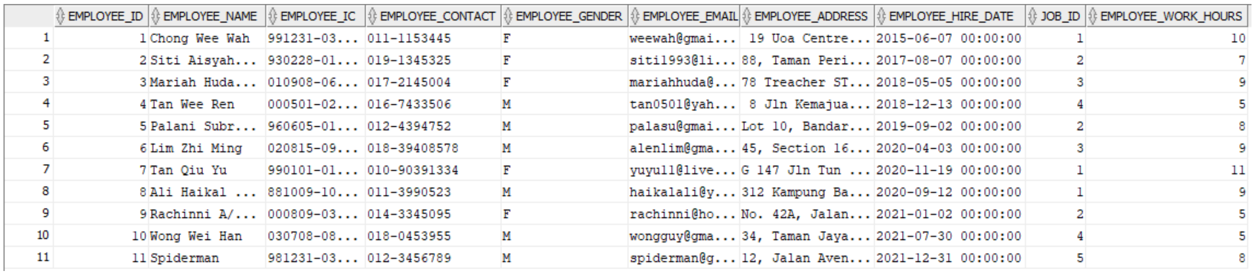
2.3 IN

Description:

IN: The IN operator is used to specify multiple values in a WHERE clause.

Situation: The employer wants to select employees who work 9, 10 and 11 hours from the EMPLOYEES table.

EMPLOYEES TABLE



	EMPLOYEE_ID	EMPLOYEE_NAME	EMPLOYEE_IC	EMPLOYEE_CONTACT	EMPLOYEE_GENDER	EMPLOYEE_EMAIL	EMPLOYEE_ADDRESS	EMPLOYEE_HIRE_DATE	JOB_ID	EMPLOYEE_WORK_HOURS
1	1	Chong Wee Wah	991231-03...	011-1153445	F	weewah@gmai...	19 Uoa Centre...	2015-06-07 00:00:00	1	10
2	2	Siti Aisyah...	930228-01...	019-1345325	F	siti1993@li...	88, Taman Peri...	2017-08-07 00:00:00	2	7
3	3	Mariah Huda...	010908-06...	017-2145004	F	mariahhuda@...	78 Treacher ST...	2018-05-05 00:00:00	3	9
4	4	Tan Wee Ren	000501-02...	016-7433506	M	tan0501@yah...	8 Jln Kemajua...	2018-12-13 00:00:00	4	5
5	5	Palani Subr...	960605-01...	012-4394752	M	palasu@gmai...	Lot 10, Bandar...	2019-09-02 00:00:00	2	8
6	6	Lim Zhi Ming	020815-09...	018-39408578	M	alenlim@gma...	45, Section 16...	2020-04-03 00:00:00	3	9
7	7	Tan Qiu Yu	990101-01...	010-90391334	F	yuyu11@live...	G 147 Jln Tun ...	2020-11-19 00:00:00	1	11
8	8	Ali Haikal ...	881009-10...	011-3990523	M	haikalali@y...	312 Kampung Ba...	2020-09-12 00:00:00	1	9
9	9	Rachinni A/...	000809-03...	014-3345095	F	rachinni@ho...	No. 42A, Jalan...	2021-01-02 00:00:00	2	5
10	10	Wong Wei Han	030708-08...	018-0453955	M	wongguy@gma...	34, Taman Jaya...	2021-07-30 00:00:00	4	5
11	11	Spiderman	981231-03...	012-3456789	M	spiderman@g...	12, Jalan Aven...	2021-12-31 00:00:00	5	8

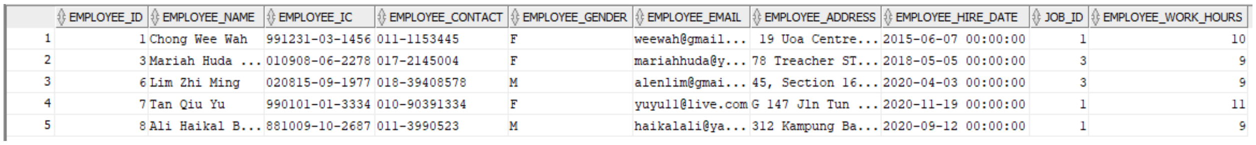
SQL Statement

SELECT *

FROM EMPLOYEES

WHERE Employee_Wrk_Hours

IN (9, 10, 11);



	EMPLOYEE_ID	EMPLOYEE_NAME	EMPLOYEE_IC	EMPLOYEE_CONTACT	EMPLOYEE_GENDER	EMPLOYEE_EMAIL	EMPLOYEE_ADDRESS	EMPLOYEE_HIRE_DATE	JOB_ID	EMPLOYEE_WORK_HOURS
1	1	Chong Wee Wah	991231-03-1456	011-1153445	F	weewah@gmail...	19 Uoa Centre...	2015-06-07 00:00:00	1	10
2	3	Mariah Huda ...	010908-06-2278	017-2145004	F	mariahhuda@y...	78 Treacher ST...	2018-05-05 00:00:00	3	9
3	6	Lim Zhi Ming	020815-09-1977	018-39408578	M	alenlim@gmai...	45, Section 16...	2020-04-03 00:00:00	3	9
4	7	Tan Qiu Yu	990101-01-3334	010-90391334	F	yuyu11@live.com	G 147 Jln Tun ...	2020-11-19 00:00:00	1	11
5	8	Ali Haikal B...	881009-10-2687	011-3990523	M	haikalali@ya...	312 Kampung Ba...	2020-09-12 00:00:00	1	9

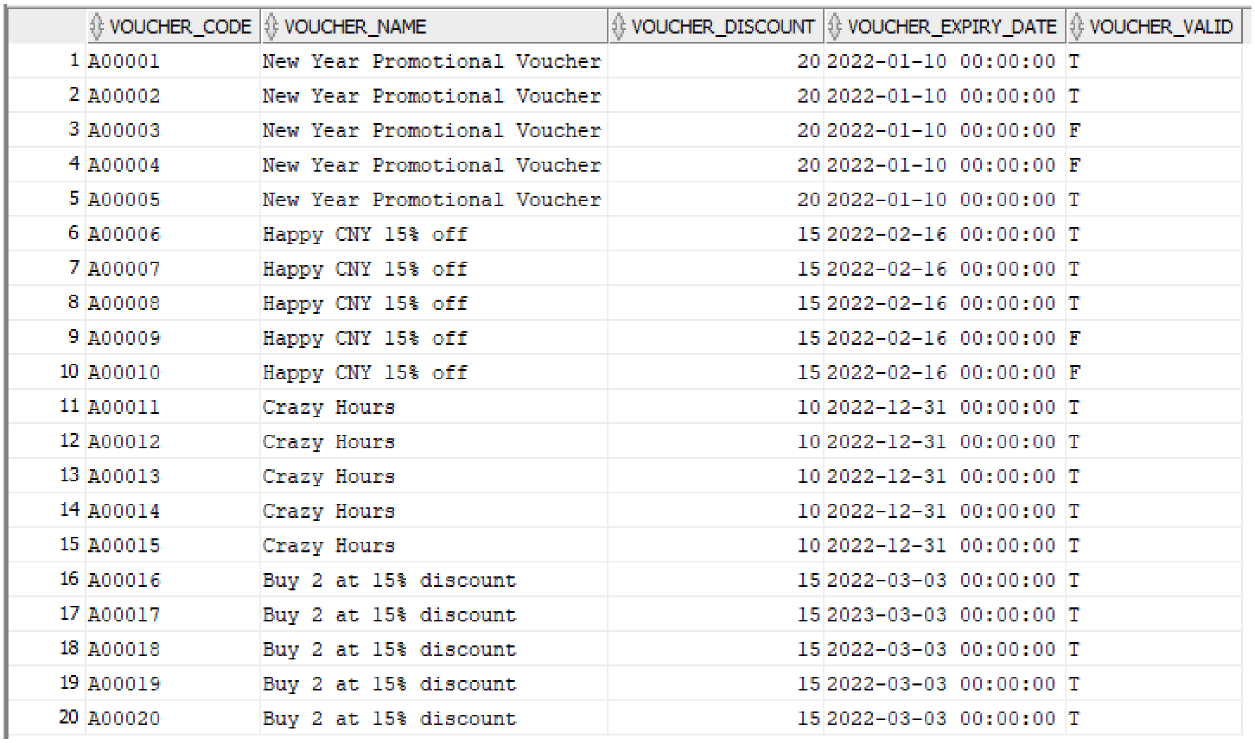
2.4 DELETE

Description:

DELETE: The DELETE statement is used to delete existing records in a table.

Situation: The employer wants to delete existing records in the VOUCHERS table because the employer does not want to do ‘Buy 2 at 15% discount’ promotions.

VOUCHERS TABLE

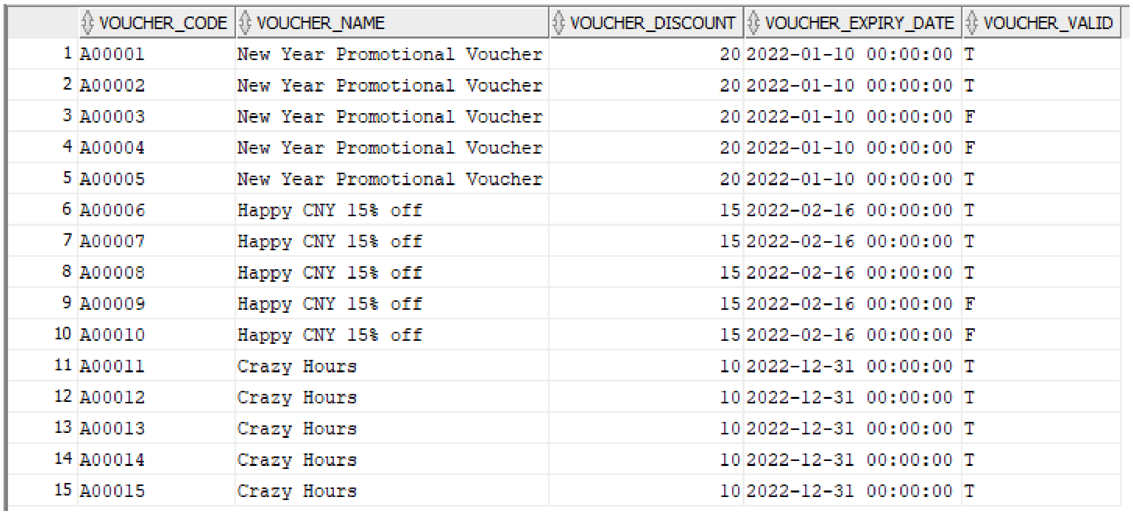


	VOUCHER_CODE	VOUCHER_NAME	VOUCHER_DISCOUNT	VOUCHER_EXPIRY_DATE	VOUCHER_VALID
1	A00001	New Year Promotional Voucher	20	2022-01-10 00:00:00	T
2	A00002	New Year Promotional Voucher	20	2022-01-10 00:00:00	T
3	A00003	New Year Promotional Voucher	20	2022-01-10 00:00:00	F
4	A00004	New Year Promotional Voucher	20	2022-01-10 00:00:00	F
5	A00005	New Year Promotional Voucher	20	2022-01-10 00:00:00	T
6	A00006	Happy CNY 15% off	15	2022-02-16 00:00:00	T
7	A00007	Happy CNY 15% off	15	2022-02-16 00:00:00	T
8	A00008	Happy CNY 15% off	15	2022-02-16 00:00:00	T
9	A00009	Happy CNY 15% off	15	2022-02-16 00:00:00	F
10	A00010	Happy CNY 15% off	15	2022-02-16 00:00:00	F
11	A00011	Crazy Hours	10	2022-12-31 00:00:00	T
12	A00012	Crazy Hours	10	2022-12-31 00:00:00	T
13	A00013	Crazy Hours	10	2022-12-31 00:00:00	T
14	A00014	Crazy Hours	10	2022-12-31 00:00:00	T
15	A00015	Crazy Hours	10	2022-12-31 00:00:00	T
16	A00016	Buy 2 at 15% discount	15	2022-03-03 00:00:00	T
17	A00017	Buy 2 at 15% discount	15	2023-03-03 00:00:00	T
18	A00018	Buy 2 at 15% discount	15	2022-03-03 00:00:00	T
19	A00019	Buy 2 at 15% discount	15	2022-03-03 00:00:00	T
20	A00020	Buy 2 at 15% discount	15	2022-03-03 00:00:00	T

SQL Statement

DELETE FROM VOUCHERS

WHERE Voucher_Name = 'Buy 2 at 15% discount';



	VOUCHER_CODE	VOUCHER_NAME	VOUCHER_DISCOUNT	VOUCHER_EXPIRY_DATE	VOUCHER_VALID
1	A00001	New Year Promotional Voucher	20	2022-01-10 00:00:00	T
2	A00002	New Year Promotional Voucher	20	2022-01-10 00:00:00	T
3	A00003	New Year Promotional Voucher	20	2022-01-10 00:00:00	F
4	A00004	New Year Promotional Voucher	20	2022-01-10 00:00:00	F
5	A00005	New Year Promotional Voucher	20	2022-01-10 00:00:00	T
6	A00006	Happy CNY 15% off	15	2022-02-16 00:00:00	T
7	A00007	Happy CNY 15% off	15	2022-02-16 00:00:00	T
8	A00008	Happy CNY 15% off	15	2022-02-16 00:00:00	T
9	A00009	Happy CNY 15% off	15	2022-02-16 00:00:00	F
10	A00010	Happy CNY 15% off	15	2022-02-16 00:00:00	F
11	A00011	Crazy Hours	10	2022-12-31 00:00:00	T
12	A00012	Crazy Hours	10	2022-12-31 00:00:00	T
13	A00013	Crazy Hours	10	2022-12-31 00:00:00	T
14	A00014	Crazy Hours	10	2022-12-31 00:00:00	T
15	A00015	Crazy Hours	10	2022-12-31 00:00:00	T

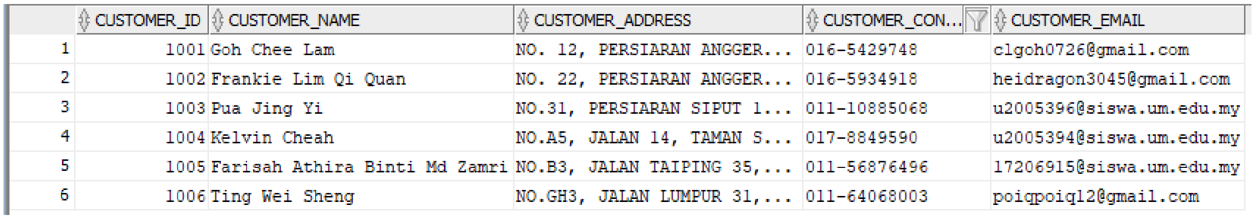
2.5 UPDATE

Description:

UPDATE: The UPDATE statement is used to modify the existing records in a table.

Situation: The employer wants to modify the existing records in a table. If the customer name is wrongly written, the employer wants to modify the record in the CUSTOMERS table.

CUSTOMERS TABLE



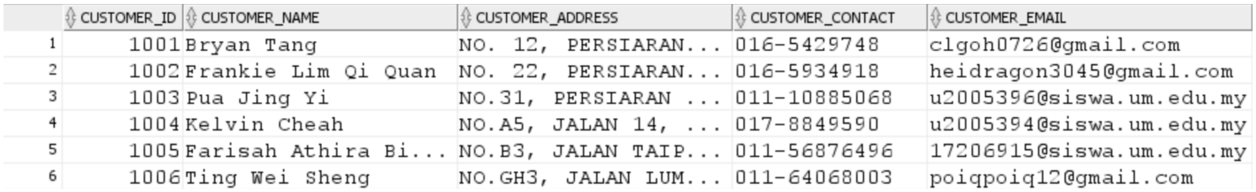
	CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_ADDRESS	CUSTOMER_CON...	CUSTOMER_EMAIL
1	1001	Goh Chee Lam	NO. 12, PERSIARAN ANGGER...	016-5429748	clgoh0726@gmail.com
2	1002	Frankie Lim Qi Quan	NO. 22, PERSIARAN ANGGER...	016-5934918	heidragon3045@gmail.com
3	1003	Pua Jing Yi	NO.31, PERSIARAN SIPUT 1...	011-10885068	u2005396@siswa.um.edu.my
4	1004	Kelvin Cheah	NO.A5, JALAN 14, TAMAN S...	017-8849590	u2005394@siswa.um.edu.my
5	1005	Farisah Athira Binti Md Zamri	NO.B3, JALAN TAIPING 35,...	011-56876496	17206915@siswa.um.edu.my
6	1006	Ting Wei Sheng	NO.GH3, JALAN LUMPUR 31,...	011-64068003	poiqpoiq12@gmail.com

<u>SQL Statement</u>

UPDATE CUSTOMERS

SET Customer_Name = 'Bryan Tang'

WHERE Customer_ID = 1001;



	CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_ADDRESS	CUSTOMER_CONTACT	CUSTOMER_EMAIL
1	1001	Bryan Tang	NO. 12, PERSIARAN...	016-5429748	clgoh0726@gmail.com
2	1002	Frankie Lim Qi Quan	NO. 22, PERSIARAN...	016-5934918	heidragon3045@gmail.com
3	1003	Pua Jing Yi	NO.31, PERSIARAN ...	011-10885068	u2005396@siswa.um.edu.my
4	1004	Kelvin Cheah	NO.A5, JALAN 14, ...	017-8849590	u2005394@siswa.um.edu.my
5	1005	Farisah Athira Bi...	NO.B3, JALAN TAIP...	011-56876496	17206915@siswa.um.edu.my
6	1006	Ting Wei Sheng	NO.GH3, JALAN LUM...	011-64068003	poiqpoiq12@gmail.com

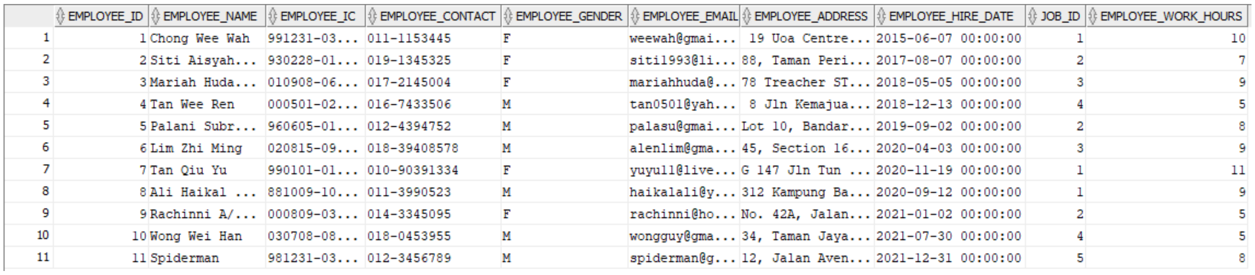
2.6 INSERT INTO

Description:

INSERT INTO: insert new data records into a table.

Situation: The employer hires a new employee to work at the restaurant and wants to insert new records containing employee information into the EMPLOYEES table.

EMPLOYEES TABLE

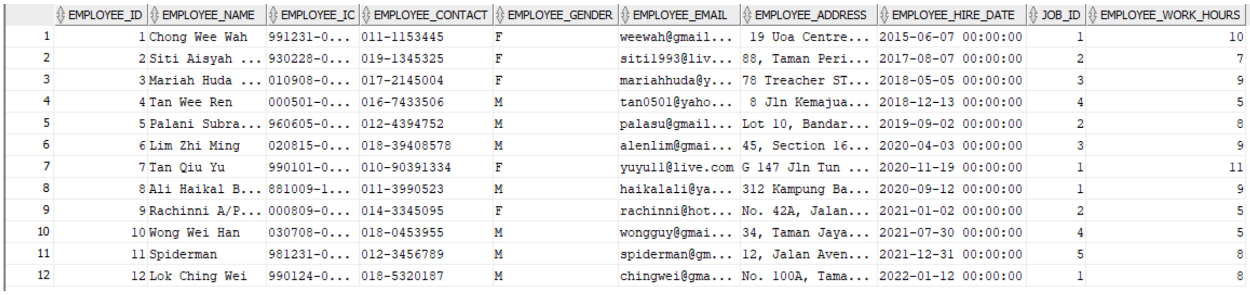


	EMPLOYEE_ID	EMPLOYEE_NAME	EMPLOYEE_IC	EMPLOYEE_CONTACT	EMPLOYEE_GENDER	EMPLOYEE_EMAIL	EMPLOYEE_ADDRESS	EMPLOYEE_HIRE_DATE	JOB_ID	EMPLOYEE_WORK_HOURS
1	1	Chong Wee Wah	991231-03...	011-1153445	F	weewah@gmai...	19 Uoa Centre...	2015-06-07 00:00:00	1	10
2	2	Siti Aisyah...	930228-01...	019-1345325	F	siti1993@li...	88, Taman Peri...	2017-08-07 00:00:00	2	7
3	3	Mariah Huda...	010908-06...	017-2145004	F	mariahhuda@...	78 Treacher ST...	2018-05-05 00:00:00	3	9
4	4	Tan Wee Ren	000501-02...	016-7433506	M	tan0501@yah...	8 Jln Kemajua...	2018-12-13 00:00:00	4	5
5	5	Palani Subr...	960605-01...	012-4394752	M	palasu@gmai...	Lot 10, Bandar...	2019-09-02 00:00:00	2	8
6	6	Lim Zhi Ming	020815-09...	018-39408578	M	alenlim@gma...	45, Section 16...	2020-04-03 00:00:00	3	9
7	7	Tan Qiu Yu	990101-01...	010-90391334	F	yuyu11@live...	G 147 Jln Tun ...	2020-11-19 00:00:00	1	11
8	8	Ali Haikal ...	881009-10...	011-3990523	M	haikalali@y...	312 Kampung Ba...	2020-09-12 00:00:00	1	9
9	9	Rachinni A/...	000809-03...	014-3345095	F	rachinni@ho...	No. 42A, Jalan...	2021-01-02 00:00:00	2	5
10	10	Wong Wei Han	030708-08...	018-0453955	M	wongguy@gma...	34, Taman Jaya...	2021-07-30 00:00:00	4	5
11	11	Spiderman	981231-03...	012-3456789	M	spiderman@g...	12, Jalan Aven...	2021-12-31 00:00:00	5	8

<u>SQL Statement</u>

INSERT INTO EMPLOYEES (EMPLOYEE_ID, EMPLOYEE_NAME, EMPLOYEE_IC, EMPLOYEE_CONTACT, EMPLOYEE_GENDER, EMPLOYEE_EMAIL, EMPLOYEE_ADDRESS, EMPLOYEE_HIRE_DATE, JOB_ID, EMPLOYEE_WORK_HOURS)

VALUES (12, 'Lok Ching Wei', '990124-08-9877', '018-5320187', 'M', 'chingwei@gmail.com', 'No. 100A, Taman Perindustrian Oxford, Seksyen 3, 47100 Puchong, Selangor.', to_date('2022-01-12', 'RRRR-MM-DD'), 1, 8);



	EMPLOYEE_ID	EMPLOYEE_NAME	EMPLOYEE_IC	EMPLOYEE_CONTACT	EMPLOYEE_GENDER	EMPLOYEE_EMAIL	EMPLOYEE_ADDRESS	EMPLOYEE_HIRE_DATE	JOB_ID	EMPLOYEE_WORK_HOURS
1	1	Chong Wee Wah	991231-0...	011-1153445	F	weewah@gmail...	19 Uoa Centre...	2015-06-07 00:00:00	1	10
2	2	Siti Aisyah ...	930228-0...	019-1345325	F	siti1993@liv...	88, Taman Peri...	2017-08-07 00:00:00	2	7
3	3	Mariah Huda ...	010908-0...	017-2145004	F	mariahhuda@y...	78 Treacher ST...	2018-05-05 00:00:00	3	9
4	4	Tan Wee Ren	000501-0...	016-7433506	M	tan0501@yaho...	8 Jln Kemajua...	2018-12-13 00:00:00	4	5
5	5	Palani Subra...	960605-0...	012-4394752	M	palasu@gmail...	Lot 10, Bandar...	2019-09-02 00:00:00	2	8
6	6	Lim Zhi Ming	020815-0...	018-39408578	M	alenlim@gmai...	45, Section 16...	2020-04-03 00:00:00	3	9
7	7	Tan Qiu Yu	990101-0...	010-90391334	F	yuyu11@live.com	G 147 Jln Tun ...	2020-11-19 00:00:00	1	11
8	8	Ali Haikal B...	881009-1...	011-3990523	M	haikalali@ya...	312 Kampung Ba...	2020-09-12 00:00:00	1	9
9	9	Rachinni A/P...	000809-0...	014-3345095	F	rachinni@hot...	No. 42A, Jalan...	2021-01-02 00:00:00	2	5
10	10	Wong Wei Han	030708-0...	018-0453955	M	wongguy@gmai...	34, Taman Jaya...	2021-07-30 00:00:00	4	5
11	11	Spiderman	981231-0...	012-3456789	M	spiderman@gm...	12, Jalan Aven...	2021-12-31 00:00:00	5	8
12	12	Lok Ching Wei	990124-0...	018-5320187	M	chingwei@gma...	No. 100A, Tama...	2022-01-12 00:00:00	1	8

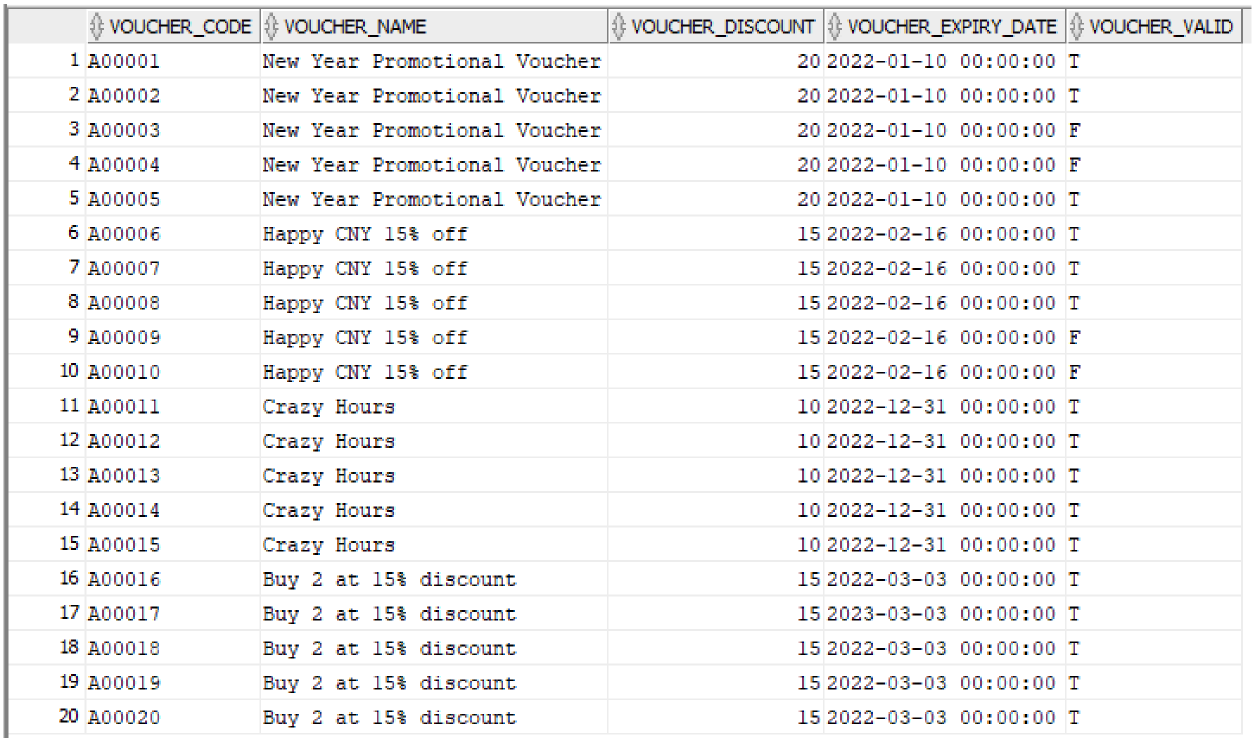
2.7 DISTINCT

Description:

SELECT DISTINCT: returns only distinct values from the selected table.

Situation: List out the unique vouchers that are released by the restaurant.

VOUCHERS TABLE

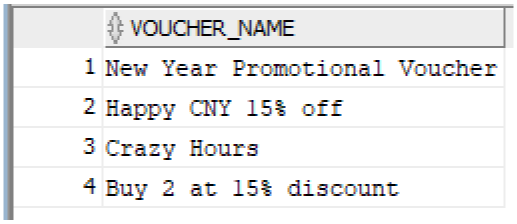


	VOUCHER_CODE	VOUCHER_NAME	VOUCHER_DISCOUNT	VOUCHER_EXPIRY_DATE	VOUCHER_VALID
1	A00001	New Year Promotional Voucher	20	2022-01-10 00:00:00	T
2	A00002	New Year Promotional Voucher	20	2022-01-10 00:00:00	T
3	A00003	New Year Promotional Voucher	20	2022-01-10 00:00:00	F
4	A00004	New Year Promotional Voucher	20	2022-01-10 00:00:00	F
5	A00005	New Year Promotional Voucher	20	2022-01-10 00:00:00	T
6	A00006	Happy CNY 15% off	15	2022-02-16 00:00:00	T
7	A00007	Happy CNY 15% off	15	2022-02-16 00:00:00	T
8	A00008	Happy CNY 15% off	15	2022-02-16 00:00:00	T
9	A00009	Happy CNY 15% off	15	2022-02-16 00:00:00	F
10	A00010	Happy CNY 15% off	15	2022-02-16 00:00:00	F
11	A00011	Crazy Hours	10	2022-12-31 00:00:00	T
12	A00012	Crazy Hours	10	2022-12-31 00:00:00	T
13	A00013	Crazy Hours	10	2022-12-31 00:00:00	T
14	A00014	Crazy Hours	10	2022-12-31 00:00:00	T
15	A00015	Crazy Hours	10	2022-12-31 00:00:00	T
16	A00016	Buy 2 at 15% discount	15	2022-03-03 00:00:00	T
17	A00017	Buy 2 at 15% discount	15	2023-03-03 00:00:00	T
18	A00018	Buy 2 at 15% discount	15	2022-03-03 00:00:00	T
19	A00019	Buy 2 at 15% discount	15	2022-03-03 00:00:00	T
20	A00020	Buy 2 at 15% discount	15	2022-03-03 00:00:00	T

<u>SQL Statement</u>

SELECT DISTINCT Voucher_Name

FROM VOUCHERS;



	VOUCHER_NAME
1	New Year Promotional Voucher
2	Happy CNY 15% off
3	Crazy Hours
4	Buy 2 at 15% discount

2.8 DISTINCT, COUNT(), ALIASES

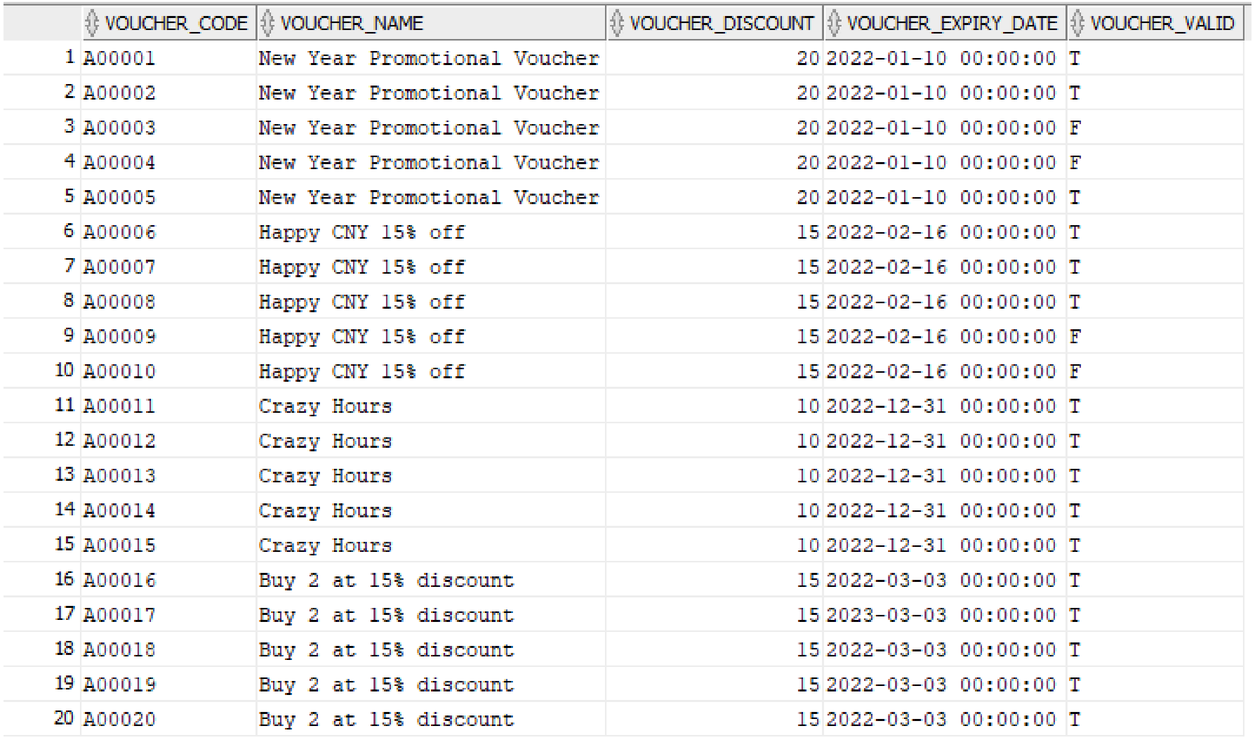
Description:

COUNT(): returns the total number of rows that meets a selected condition.

ALIASES: makes the column names to be more readable by using the “AS” keyword.

Situation: Calculate the number of unique vouchers that are released by the restaurant.

VOUCHERS TABLE

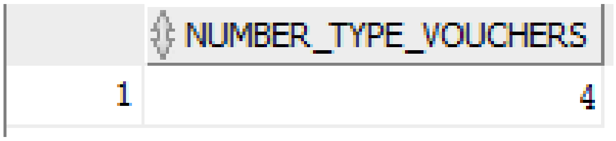


	VOUCHER_CODE	VOUCHER_NAME	VOUCHER_DISCOUNT	VOUCHER_EXPIRY_DATE	VOUCHER_VALID
1	A00001	New Year Promotional Voucher	20	2022-01-10 00:00:00	T
2	A00002	New Year Promotional Voucher	20	2022-01-10 00:00:00	T
3	A00003	New Year Promotional Voucher	20	2022-01-10 00:00:00	F
4	A00004	New Year Promotional Voucher	20	2022-01-10 00:00:00	F
5	A00005	New Year Promotional Voucher	20	2022-01-10 00:00:00	T
6	A00006	Happy CNY 15% off	15	2022-02-16 00:00:00	T
7	A00007	Happy CNY 15% off	15	2022-02-16 00:00:00	T
8	A00008	Happy CNY 15% off	15	2022-02-16 00:00:00	T
9	A00009	Happy CNY 15% off	15	2022-02-16 00:00:00	F
10	A00010	Happy CNY 15% off	15	2022-02-16 00:00:00	F
11	A00011	Crazy Hours	10	2022-12-31 00:00:00	T
12	A00012	Crazy Hours	10	2022-12-31 00:00:00	T
13	A00013	Crazy Hours	10	2022-12-31 00:00:00	T
14	A00014	Crazy Hours	10	2022-12-31 00:00:00	T
15	A00015	Crazy Hours	10	2022-12-31 00:00:00	T
16	A00016	Buy 2 at 15% discount	15	2022-03-03 00:00:00	T
17	A00017	Buy 2 at 15% discount	15	2023-03-03 00:00:00	T
18	A00018	Buy 2 at 15% discount	15	2022-03-03 00:00:00	T
19	A00019	Buy 2 at 15% discount	15	2022-03-03 00:00:00	T
20	A00020	Buy 2 at 15% discount	15	2022-03-03 00:00:00	T

<u>SQL Statement</u>

SELECT COUNT(DISTINCT Voucher_Name**) AS** NUMBER_TYPE_VOUCHERS

FROM VOUCHERS;



	NUMBER_TYPE_VOUCHERS
1	4

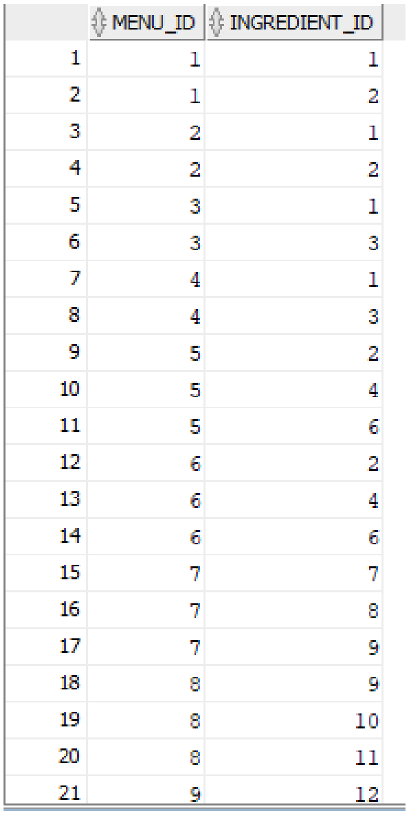
2.9 ANY()

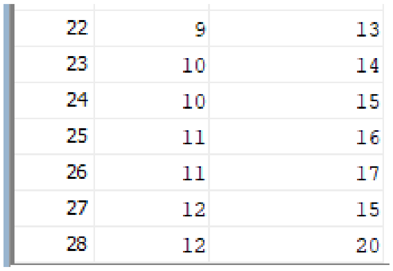
Description:

ANY(): An operator used to perform a comparison between a single column value and a range of other values. This operator will return true if any of the subquery values meet the stated condition.

Situation: If employees want to search the record of the menu that can be done if they still have ingredients less than 11

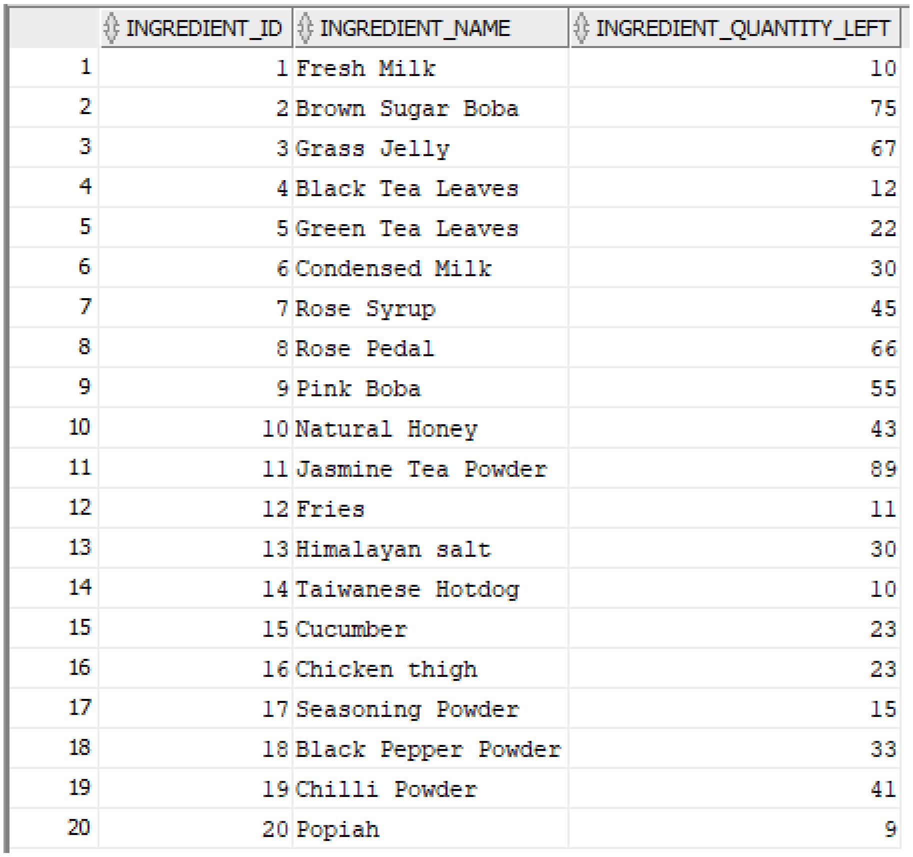
MENU_INGREDIENTS TABLE





	MENU_ID	INGREDIENT_ID
1	1	1
2	1	2
3	2	1
4	2	2
5	3	1
6	3	3
7	4	1
8	4	3
9	5	2
10	5	4
11	5	6
12	6	2
13	6	4
14	6	6
15	7	7
16	7	8
17	7	9
18	8	9
19	8	10
20	8	11
21	9	12
22	9	13
23	10	14
24	10	15
25	11	16
26	11	17
27	12	15
28	12	20

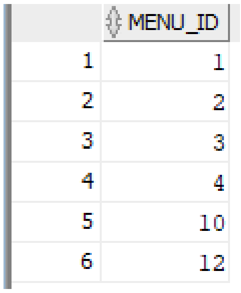
INGREDIENTS TABLE



	INGREDIENT_ID	INGREDIENT_NAME	INGREDIENT_QUANTITY_LEFT
1	1	Fresh Milk	10
2	2	Brown Sugar Boba	75
3	3	Grass Jelly	67
4	4	Black Tea Leaves	12
5	5	Green Tea Leaves	22
6	6	Condensed Milk	30
7	7	Rose Syrup	45
8	8	Rose Pedal	66
9	9	Pink Boba	55
10	10	Natural Honey	43
11	11	Jasmine Tea Powder	89
12	12	Fries	11
13	13	Himalayan salt	30
14	14	Taiwanese Hotdog	10
15	15	Cucumber	23
16	16	Chicken thigh	23
17	17	Seasoning Powder	15
18	18	Black Pepper Powder	33
19	19	Chilli Powder	41
20	20	Popiah	9

SQL Statement

```
SELECT Menu_ID
FROM MENU_INGREDIENTS
WHERE Ingredient_ID = ANY
        (SELECT Ingredient_ID
        FROM INGREDIENTS
        WHERE Ingredient_Quantity_Left < 11);
```



	MENU_ID
1	1
2	2
3	3
4	4
5	10
6	12

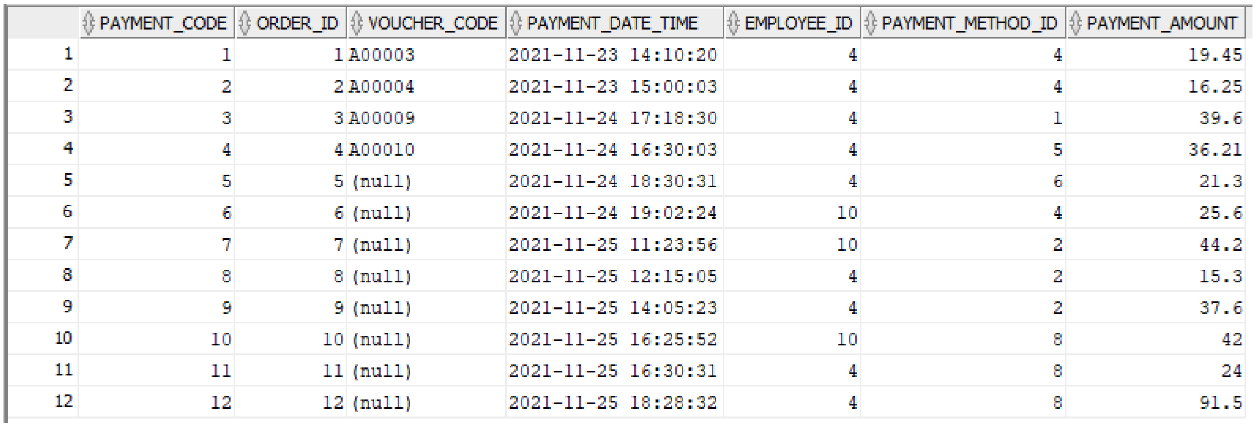
2.10 UNION

Description:

UNION: An operator that is used to combine the result-set of two or more SELECT statements. The column in the select statement within the union operator must have similar data types, the same number of columns picked, and in the same order.

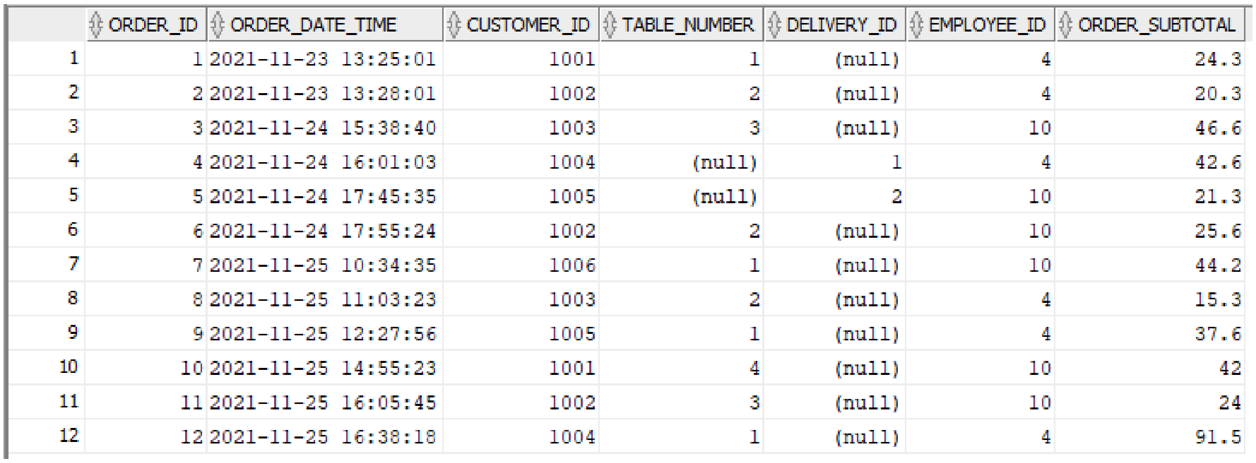
Situation: If the employee wants to search the information of all the order and payment date-time

PAYMENTS TABLE



	PAYMENT_CODE	ORDER_ID	VOUCHER_CODE	PAYMENT_DATE_TIME	EMPLOYEE_ID	PAYMENT_METHOD_ID	PAYMENT_AMOUNT
1	1	1	A00003	2021-11-23 14:10:20	4	4	19.45
2	2	2	A00004	2021-11-23 15:00:03	4	4	16.25
3	3	3	A00009	2021-11-24 17:18:30	4	1	39.6
4	4	4	A00010	2021-11-24 16:30:03	4	5	36.21
5	5	5	(null)	2021-11-24 18:30:31	4	6	21.3
6	6	6	(null)	2021-11-24 19:02:24	10	4	25.6
7	7	7	(null)	2021-11-25 11:23:56	10	2	44.2
8	8	8	(null)	2021-11-25 12:15:05	4	2	15.3
9	9	9	(null)	2021-11-25 14:05:23	4	2	37.6
10	10	10	(null)	2021-11-25 16:25:52	10	8	42
11	11	11	(null)	2021-11-25 16:30:31	4	8	24
12	12	12	(null)	2021-11-25 18:28:32	4	8	91.5

ORDERS TABLE



	ORDER_ID	ORDER_DATE_TIME	CUSTOMER_ID	TABLE_NUMBER	DELIVERY_ID	EMPLOYEE_ID	ORDER_SUBTOTAL
1	1	2021-11-23 13:25:01	1001	1	(null)	4	24.3
2	2	2021-11-23 13:28:01	1002	2	(null)	4	20.3
3	3	2021-11-24 15:38:40	1003	3	(null)	10	46.6
4	4	2021-11-24 16:01:03	1004	(null)	1	4	42.6
5	5	2021-11-24 17:45:35	1005	(null)	2	10	21.3
6	6	2021-11-24 17:55:24	1002	2	(null)	10	25.6
7	7	2021-11-25 10:34:35	1006	1	(null)	10	44.2
8	8	2021-11-25 11:03:23	1003	2	(null)	4	15.3
9	9	2021-11-25 12:27:56	1005	1	(null)	4	37.6
10	10	2021-11-25 14:55:23	1001	4	(null)	10	42
11	11	2021-11-25 16:05:45	1002	3	(null)	10	24
12	12	2021-11-25 16:38:18	1004	1	(null)	4	91.5

SQL Statement

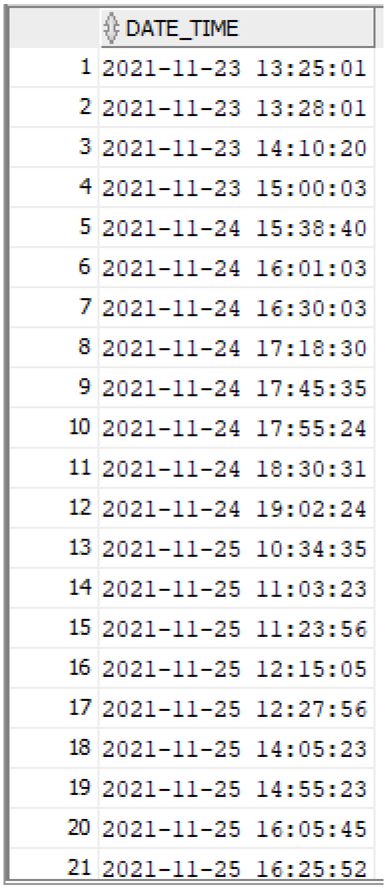
SELECT Payment_Date_Time **AS** Date_time

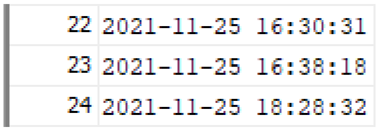
FROM PAYMENTS

UNION

(**SELECT** Order_Date_Time

FROM ORDERS);





	DATE_TIME
1	2021-11-23 13:25:01
2	2021-11-23 13:28:01
3	2021-11-23 14:10:20
4	2021-11-23 15:00:03
5	2021-11-24 15:38:40
6	2021-11-24 16:01:03
7	2021-11-24 16:30:03
8	2021-11-24 17:18:30
9	2021-11-24 17:45:35
10	2021-11-24 17:55:24
11	2021-11-24 18:30:31
12	2021-11-24 19:02:24
13	2021-11-25 10:34:35
14	2021-11-25 11:03:23
15	2021-11-25 11:23:56
16	2021-11-25 12:15:05
17	2021-11-25 12:27:56
18	2021-11-25 14:05:23
19	2021-11-25 14:55:23
20	2021-11-25 16:05:45
21	2021-11-25 16:25:52
22	2021-11-25 16:30:31
23	2021-11-25 16:38:18
24	2021-11-25 18:28:32

2.11 MIN(), BETWEEN

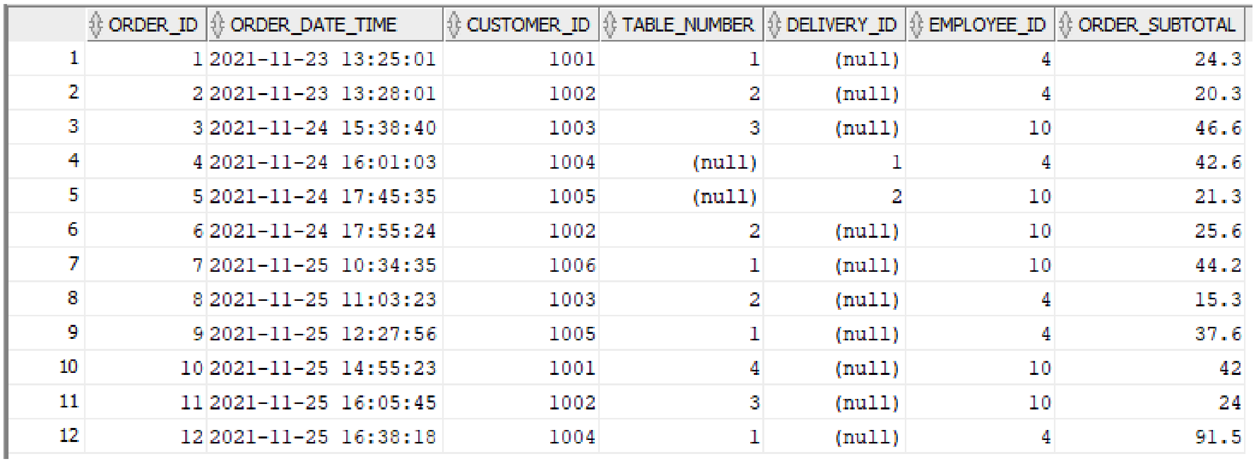
Description:

MIN(): Min() function returns the smallest value of the selected column.

BETWEEN: Between operator will select values within the specified range and those values can be numbers, text, or dates.

Situation: The employer wants to check the lowest order amount that is made by the customer between ‘2021-11-24 00:00:00’ and ‘2021-11-25 23:59:59’.

ORDERS TABLE



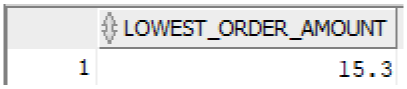
	ORDER_ID	ORDER_DATE_TIME	CUSTOMER_ID	TABLE_NUMBER	DELIVERY_ID	EMPLOYEE_ID	ORDER_SUBTOTAL
1	1	2021-11-23 13:25:01	1001	1	(null)	4	24.3
2	2	2021-11-23 13:28:01	1002	2	(null)	4	20.3
3	3	2021-11-24 15:38:40	1003	3	(null)	10	46.6
4	4	2021-11-24 16:01:03	1004	(null)	1	4	42.6
5	5	2021-11-24 17:45:35	1005	(null)	2	10	21.3
6	6	2021-11-24 17:55:24	1002	2	(null)	10	25.6
7	7	2021-11-25 10:34:35	1006	1	(null)	10	44.2
8	8	2021-11-25 11:03:23	1003	2	(null)	4	15.3
9	9	2021-11-25 12:27:56	1005	1	(null)	4	37.6
10	10	2021-11-25 14:55:23	1001	4	(null)	10	42
11	11	2021-11-25 16:05:45	1002	3	(null)	10	24
12	12	2021-11-25 16:38:18	1004	1	(null)	4	91.5

<u>**SQL Statement**</u>

SELECT MIN(Order_Subtotal**) AS** Lowest_Order_Amount

FROM ORDERS

WHERE Order_Date_Time **BETWEEN** '2021-11-24 00:00:00' **AND** '2021-11-25 23:59:59';



	LOWEST_ORDER_AMOUNT
1	15.3

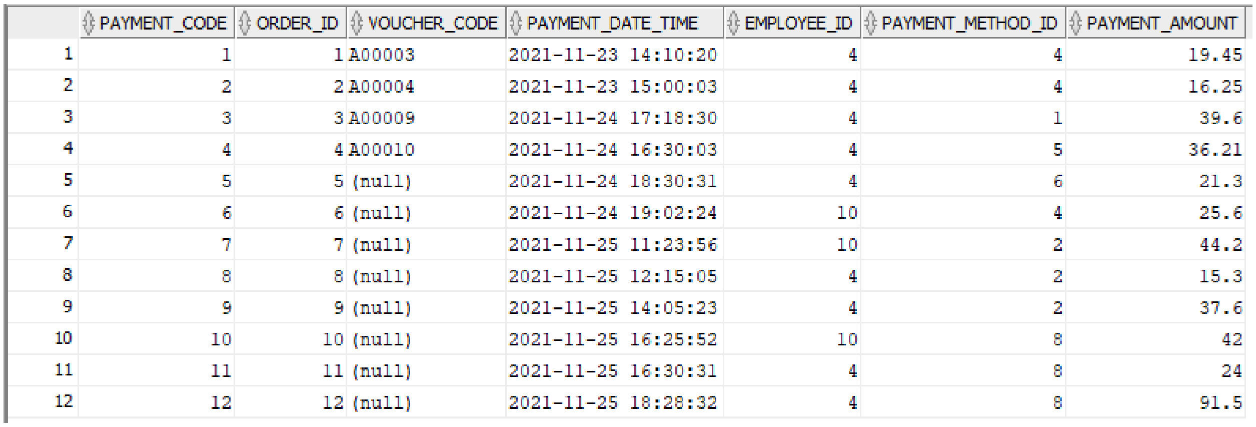
2.12 MAX(), BETWEEN

Description:

MAX(): returns the largest value of the selected column.

Situation: The employer wants to check the highest payment amount that is made by the customer between ‘2021-11-23 00:00:00’ and ‘2021-11-24 23:59:59’.

PAYMENTS TABLE



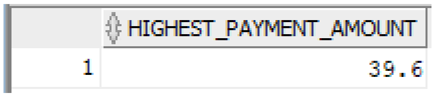
	PAYMENT_CODE	ORDER_ID	VOUCHER_CODE	PAYMENT_DATE_TIME	EMPLOYEE_ID	PAYMENT_METHOD_ID	PAYMENT_AMOUNT
1	1	1	A00003	2021-11-23 14:10:20	4	4	19.45
2	2	2	A00004	2021-11-23 15:00:03	4	4	16.25
3	3	3	A00009	2021-11-24 17:18:30	4	1	39.6
4	4	4	A00010	2021-11-24 16:30:03	4	5	36.21
5	5	5	(null)	2021-11-24 18:30:31	4	6	21.3
6	6	6	(null)	2021-11-24 19:02:24	10	4	25.6
7	7	7	(null)	2021-11-25 11:23:56	10	2	44.2
8	8	8	(null)	2021-11-25 12:15:05	4	2	15.3
9	9	9	(null)	2021-11-25 14:05:23	4	2	37.6
10	10	10	(null)	2021-11-25 16:25:52	10	8	42
11	11	11	(null)	2021-11-25 16:30:31	4	8	24
12	12	12	(null)	2021-11-25 18:28:32	4	8	91.5

<u>SQL Statement</u>

SELECT MAX(Payment_Amount**) AS** Highest_Payment_Amount

FROM PAYMENTS

WHERE Payment_Date_Time **BETWEEN** '2021-11-23 00:00:00' **AND** '2021-11-24 23:59:59';



	HIGHEST_PAYMENT_AMOUNT
1	39.6

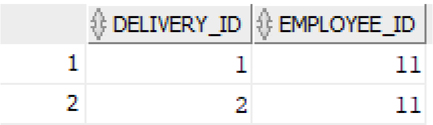
2.13 SELECT, INNER JOIN

Description:

INNER JOIN: select records that have matching values in both tables.

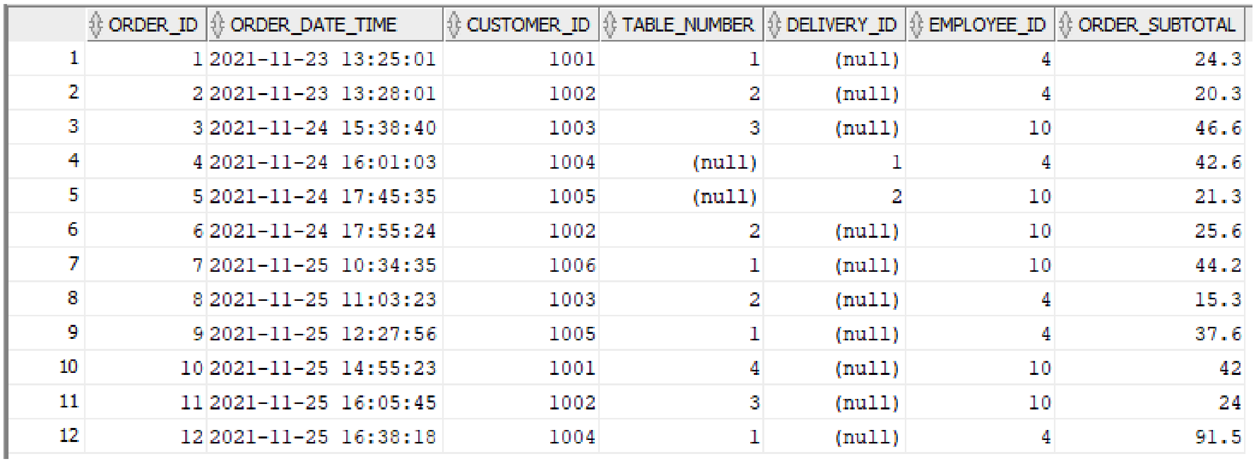
Situation: When the customers do the ordering online, the restaurant will provide the delivery service to the customers. In each delivery service, we will provide the information of the delivery man and customer.

DELIVERY TABLE



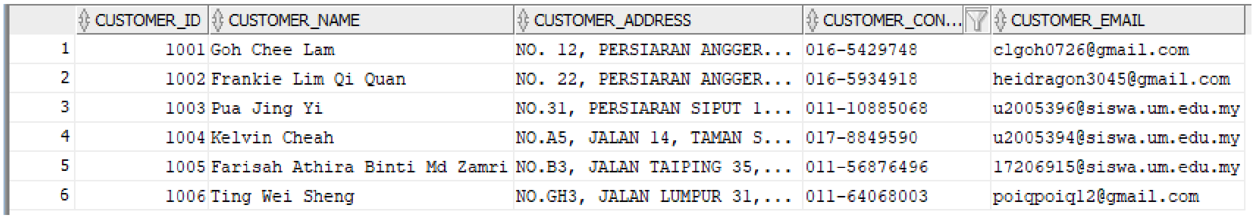
	DELIVERY_ID	EMPLOYEE_ID
1	1	11
2	2	11

ORDERS TABLE



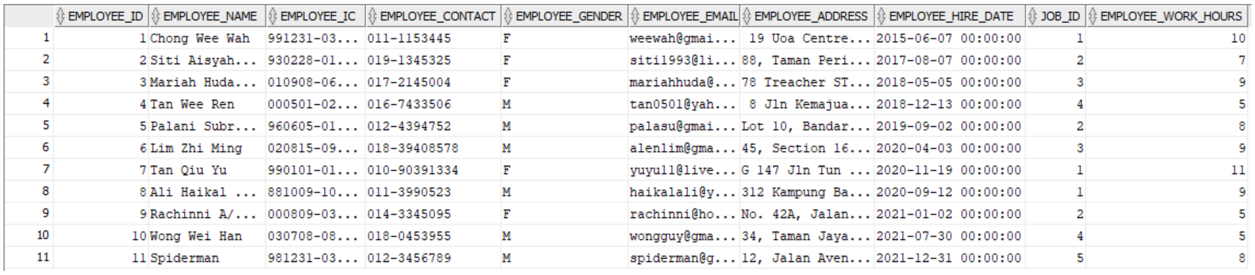
	ORDER_ID	ORDER_DATE_TIME	CUSTOMER_ID	TABLE_NUMBER	DELIVERY_ID	EMPLOYEE_ID	ORDER_SUBTOTAL
1	1	2021-11-23 13:25:01	1001	1	(null)	4	24.3
2	2	2021-11-23 13:28:01	1002	2	(null)	4	20.3
3	3	2021-11-24 15:38:40	1003	3	(null)	10	46.6
4	4	2021-11-24 16:01:03	1004	(null)	1	4	42.6
5	5	2021-11-24 17:45:35	1005	(null)	2	10	21.3
6	6	2021-11-24 17:55:24	1002	2	(null)	10	25.6
7	7	2021-11-25 10:34:35	1006	1	(null)	10	44.2
8	8	2021-11-25 11:03:23	1003	2	(null)	4	15.3
9	9	2021-11-25 12:27:56	1005	1	(null)	4	37.6
10	10	2021-11-25 14:55:23	1001	4	(null)	10	42
11	11	2021-11-25 16:05:45	1002	3	(null)	10	24
12	12	2021-11-25 16:38:18	1004	1	(null)	4	91.5

CUSTOMERS TABLE



	CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_ADDRESS	CUSTOMER_CON...	CUSTOMER_EMAIL
1	1001	Goh Chee Lam	NO. 12, PERSIARAN ANGGER...	016-5429748	clgoh0726@gmail.com
2	1002	Frankie Lim Qi Quan	NO. 22, PERSIARAN ANGGER...	016-5934918	heidragon3045@gmail.com
3	1003	Pua Jing Yi	NO.31, PERSIARAN SIPUT 1...	011-10885068	u2005396@siswa.um.edu.my
4	1004	Kelvin Cheah	NO.A5, JALAN 14, TAMAN S...	017-8849590	u2005394@siswa.um.edu.my
5	1005	Farisah Athira Binti Md Zamri	NO.B3, JALAN TAIPING 35,...	011-56876496	17206915@siswa.um.edu.my
6	1006	Ting Wei Sheng	NO.GH3, JALAN LUMPUR 31,...	011-64068003	poiqpoiq12@gmail.com

EMPLOYEES TABLE



	EMPLOYEE_ID	EMPLOYEE_NAME	EMPLOYEE_IC	EMPLOYEE_CONTACT	EMPLOYEE_GENDER	EMPLOYEE_EMAIL	EMPLOYEE_ADDRESS	EMPLOYEE_HIRE_DATE	JOB_ID	EMPLOYEE_WORK_HOURS
1	1	Chong Wee Wah	991231-03...	011-1153445	F	weewah@gmai...	19 Uoa Centre...	2015-06-07 00:00:00	1	10
2	2	Siti Aisyah...	930228-01...	019-1345325	F	siti1993@li...	88, Taman Peri...	2017-08-07 00:00:00	2	7
3	3	Mariah Huda...	010908-06...	017-2145004	F	mariahhuda@...	78 Treacher ST...	2018-05-05 00:00:00	3	9
4	4	Tan Wee Ren	000501-02...	016-7433506	M	tan0501@yah...	8 Jln Kemajua...	2018-12-13 00:00:00	4	5
5	5	Palani Subr...	960605-01...	012-4394752	M	palasu@gmai...	Lot 10, Bandar...	2019-09-02 00:00:00	2	8
6	6	Lim Zhi Ming	020815-09...	018-39408578	M	alenlim@gma...	45, Section 16...	2020-04-03 00:00:00	3	9
7	7	Tan Qiu Yu	990101-01...	010-90391334	F	yuyu11@live...	G 147 Jln Tun ...	2020-11-19 00:00:00	1	11
8	8	Ali Haikal ...	881009-10...	011-3990523	M	haikalali@y...	312 Kampung Ba...	2020-09-12 00:00:00	1	9
9	9	Rachinni A/...	000809-03...	014-3345095	F	rachinni@ho...	No. 42A, Jalan...	2021-01-02 00:00:00	2	5
10	10	Wong Wei Han	030708-08...	018-0453955	M	wongguy@gma...	34, Taman Jaya...	2021-07-30 00:00:00	4	5
11	11	Spiderman	981231-03...	012-3456789	M	spiderman@g...	12, Jalan Aven...	2021-12-31 00:00:00	5	8

<u>SQL Statement:</u>

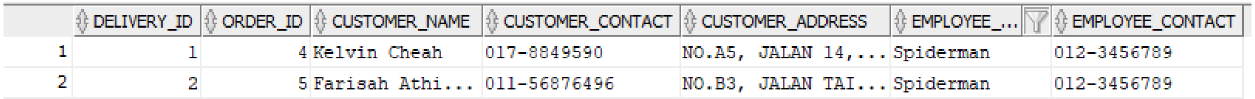
SELECT DELIVERY.Delivery_ID, ORDERS.Order_ID, CUSTOMERS.Customer_Name, CUSTOMERS.Customer_Contact, CUSTOMERS.Customer_Address, EMPLOYEES.Employee_Name, EMPLOYEES.Employee_Contact

FROM (((ORDERS

INNER JOIN CUSTOMERS **ON** ORDERS.Customer_ID = CUSTOMERS.Customer_ID)

INNER JOIN DELIVERY **ON** ORDERS.Delivery_ID = DELIVERY.Delivery_ID)

INNER JOIN EMPLOYEES **ON** Delivery.Employee_ID = EMPLOYEES.Employee_ID);



	DELIVERY_ID	ORDER_ID	CUSTOMER_NAME	CUSTOMER_CONTACT	CUSTOMER_ADDRESS	EMPLOYEE_...	EMPLOYEE_CONTACT
1	1	4	Kelvin Cheah	017-8849590	NO.A5, JALAN 14,...	Spiderman	012-3456789
2	2	5	Farisah Athi...	011-56876496	NO.B3, JALAN TAI...	Spiderman	012-3456789

2.14 EXIST, HAVING, COUNT(), GROUP BY

Description:

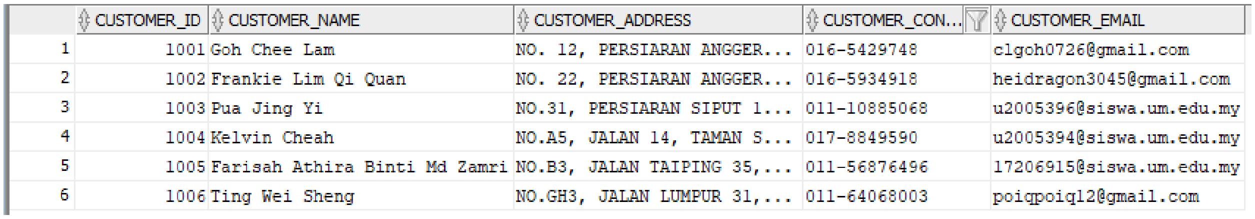
EXIST: used to test for the existence of any record in a subquery and returns TRUE if the subquery returns one or more records.

HAVING: The HAVING clause was added to SQL because the WHERE keyword cannot be used with aggregate functions.

GROUP BY: groups rows that have the same values into summary rows.

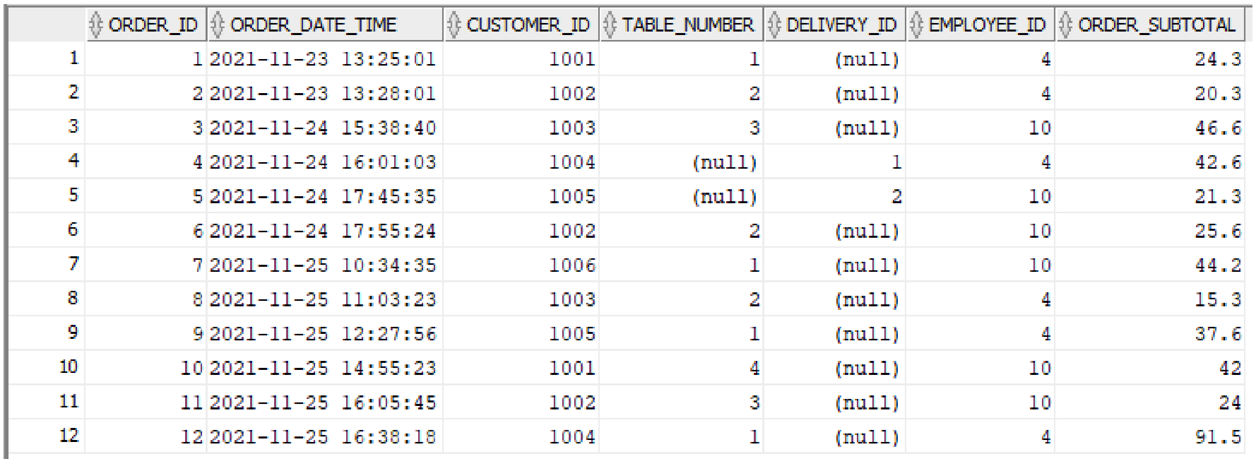
Situation: Employees want to know the customer that orders more than 2 orders (frequent customer).

CUSTOMERS TABLE



	CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_ADDRESS	CUSTOMER_CON...	CUSTOMER_EMAIL
1	1001	Goh Chee Lam	NO. 12, PERSIARAN ANGGER...	016-5429748	clgoh0726@gmail.com
2	1002	Frankie Lim Qi Quan	NO. 22, PERSIARAN ANGGER...	016-5934918	heidragon3045@gmail.com
3	1003	Pua Jing Yi	NO.31, PERSIARAN SIPUT 1...	011-10885068	u2005396@siswa.um.edu.my
4	1004	Kelvin Cheah	NO.A5, JALAN 14, TAMAN S...	017-8849590	u2005394@siswa.um.edu.my
5	1005	Farisah Athira Binti Md Zamri	NO.B3, JALAN TAIPING 35,...	011-56876496	17206915@siswa.um.edu.my
6	1006	Ting Wei Sheng	NO.GH3, JALAN LUMPUR 31,...	011-64068003	poiqpoiq12@gmail.com

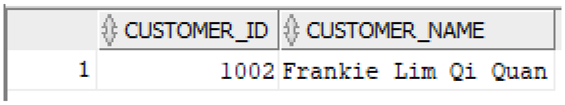
ORDERS TABLE



	ORDER_ID	ORDER_DATE_TIME	CUSTOMER_ID	TABLE_NUMBER	DELIVERY_ID	EMPLOYEE_ID	ORDER_SUBTOTAL
1	1	2021-11-23 13:25:01	1001	1	(null)	4	24.3
2	2	2021-11-23 13:28:01	1002	2	(null)	4	20.3
3	3	2021-11-24 15:38:40	1003	3	(null)	10	46.6
4	4	2021-11-24 16:01:03	1004	(null)	1	4	42.6
5	5	2021-11-24 17:45:35	1005	(null)	2	10	21.3
6	6	2021-11-24 17:55:24	1002	2	(null)	10	25.6
7	7	2021-11-25 10:34:35	1006	1	(null)	10	44.2
8	8	2021-11-25 11:03:23	1003	2	(null)	4	15.3
9	9	2021-11-25 12:27:56	1005	1	(null)	4	37.6
10	10	2021-11-25 14:55:23	1001	4	(null)	10	42
11	11	2021-11-25 16:05:45	1002	3	(null)	10	24
12	12	2021-11-25 16:38:18	1004	1	(null)	4	91.5

<u>SQL Statement</u>

```
SELECT Customer_ID, Customer_Name
FROM CUSTOMERS
WHERE EXISTS (
SELECT COUNT(Order_ID)
FROM ORDERS
WHERE ORDERS.Customer_ID = CUSTOMERS.Customer_ID
GROUP BY Customer_ID
HAVING COUNT(Order_ID)>2);
```



	CUSTOMER_ID	CUSTOMER_NAME
1	1002	Frankie Lim Qi Quan

2.15 SELECT, MULTIPLY (SQL Arithmetic Operator), ALIASES, LEFT JOIN, ORDER BY, SUM(), GROUP BY

Description:

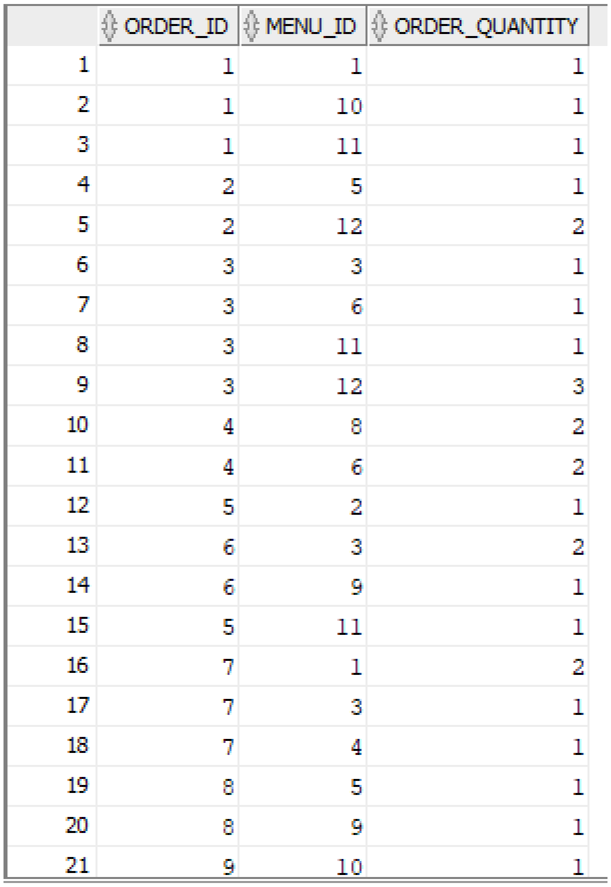
LEFT JOIN: returns all records from the left table (table1), and the matching records from the right table (table2). If there is no match, the result is 0 records from the right side.

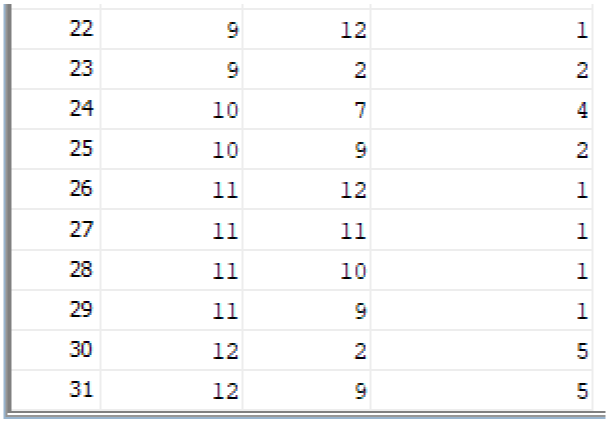
ORDER BY: used to sort the result-set in ascending or descending order.

SUM(): returns the total sum of a numeric column.

Situation 1: Calculate the total price of each menu listed in each order based on the quantity of the menu.

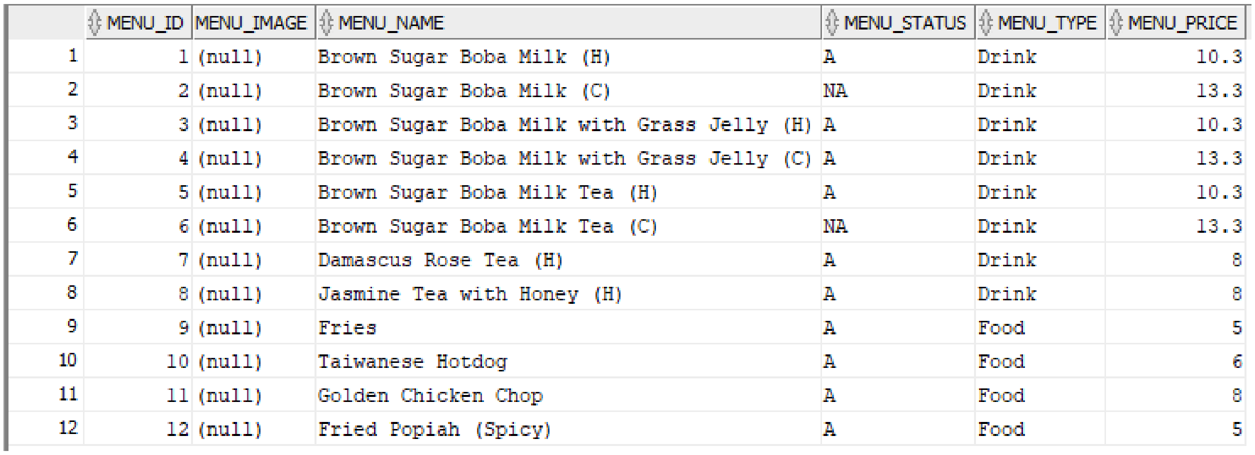
ORDER_MENU TABLE





	ORDER_ID	MENU_ID	ORDER_QUANTITY
1	1	1	1
2	1	10	1
3	1	11	1
4	2	5	1
5	2	12	2
6	3	3	1
7	3	6	1
8	3	11	1
9	3	12	3
10	4	8	2
11	4	6	2
12	5	2	1
13	6	3	2
14	6	9	1
15	5	11	1
16	7	1	2
17	7	3	1
18	7	4	1
19	8	5	1
20	8	9	1
21	9	10	1
22	9	12	1
23	9	2	2
24	10	7	4
25	10	9	2
26	11	12	1
27	11	11	1
28	11	10	1
29	11	9	1
30	12	2	5
31	12	9	5

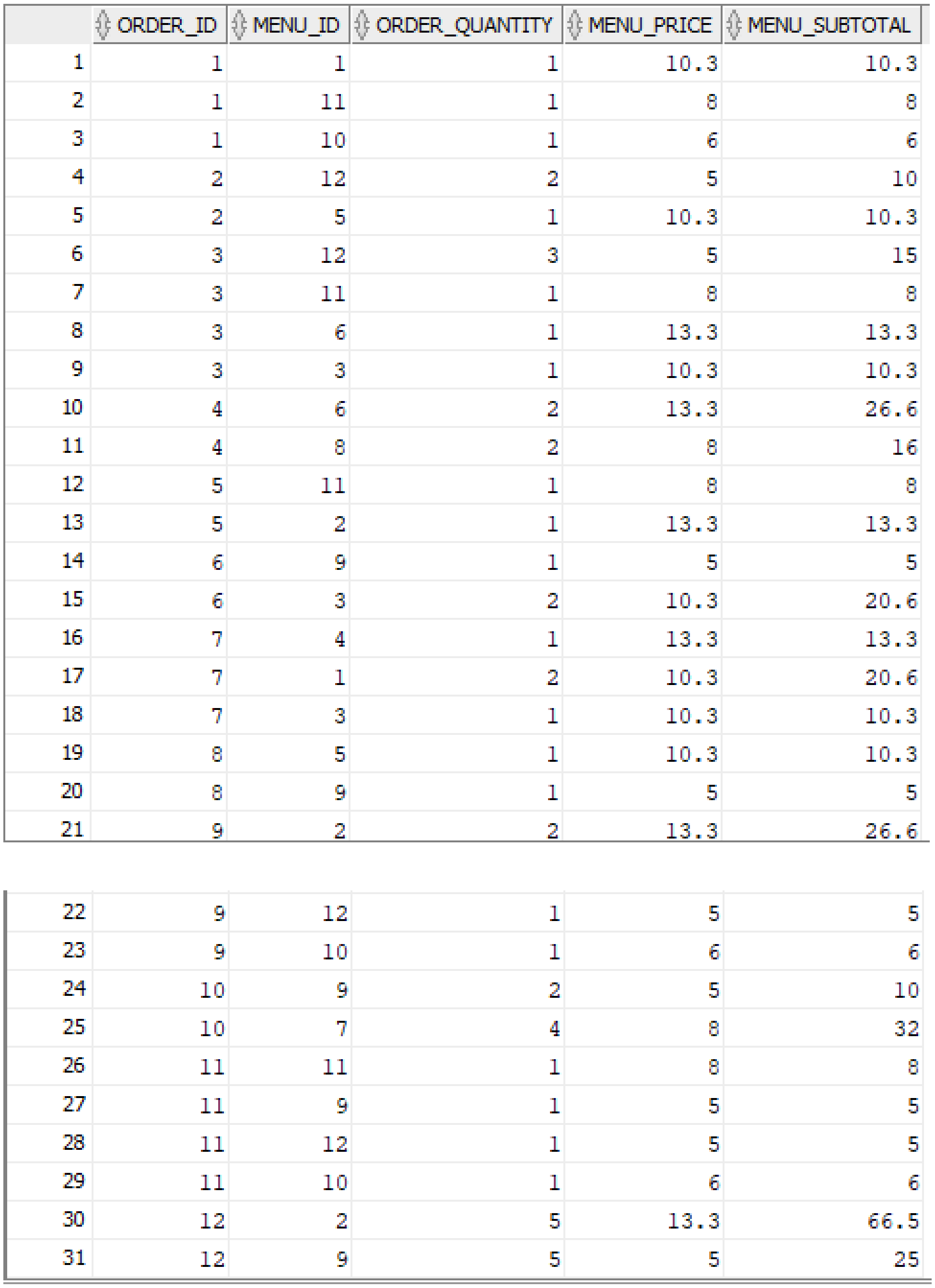
MENU TABLE



	MENU_ID	MENU_IMAGE	MENU_NAME	MENU_STATUS	MENU_TYPE	MENU_PRICE
1	1	(null)	Brown Sugar Boba Milk (H)	A	Drink	10.3
2	2	(null)	Brown Sugar Boba Milk (C)	NA	Drink	13.3
3	3	(null)	Brown Sugar Boba Milk with Grass Jelly (H)	A	Drink	10.3
4	4	(null)	Brown Sugar Boba Milk with Grass Jelly (C)	A	Drink	13.3
5	5	(null)	Brown Sugar Boba Milk Tea (H)	A	Drink	10.3
6	6	(null)	Brown Sugar Boba Milk Tea (C)	NA	Drink	13.3
7	7	(null)	Damascus Rose Tea (H)	A	Drink	8
8	8	(null)	Jasmine Tea with Honey (H)	A	Drink	8
9	9	(null)	Fries	A	Food	5
10	10	(null)	Taiwanese Hotdog	A	Food	6
11	11	(null)	Golden Chicken Chop	A	Food	8
12	12	(null)	Fried Popiah (Spicy)	A	Food	5

<u>SQL Statement</u>

```
SELECT ORDER_MENU.Order_ID, ORDER_MENU.Menu_ID, ORDER_MENU.Order_Quantity, MENU.Menu_Price, ORDER_MENU.Order_Quantity * MENU.Menu_Price AS Menu_Subtotal

FROM ORDER_MENU

LEFT JOIN MENU ON ORDER_MENU.Menu_ID = MENU.Menu_ID

ORDER BY ORDER_MENU.Order_ID;
```

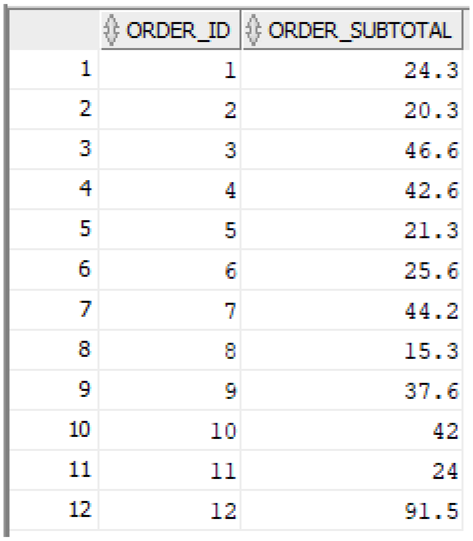


	ORDER_ID	MENU_ID	ORDER_QUANTITY	MENU_PRICE	MENU_SUBTOTAL
1	1	1	1	10.3	10.3
2	1	11	1	8	8
3	1	10	1	6	6
4	2	12	2	5	10
5	2	5	1	10.3	10.3
6	3	12	3	5	15
7	3	11	1	8	8
8	3	6	1	13.3	13.3
9	3	3	1	10.3	10.3
10	4	6	2	13.3	26.6
11	4	8	2	8	16
12	5	11	1	8	8
13	5	2	1	13.3	13.3
14	6	9	1	5	5
15	6	3	2	10.3	20.6
16	7	4	1	13.3	13.3
17	7	1	2	10.3	20.6
18	7	3	1	10.3	10.3
19	8	5	1	10.3	10.3
20	8	9	1	5	5
21	9	2	2	13.3	26.6
22	9	12	1	5	5
23	9	10	1	6	6
24	10	9	2	5	10
25	10	7	4	8	32
26	11	11	1	8	8
27	11	9	1	5	5
28	11	12	1	5	5
29	11	10	1	6	6
30	12	2	5	13.3	66.5
31	12	9	5	5	25

Situation 2: Calculate the total price of each order before payment or using vouchers.

<u>**SQL Statement**</u>

```
SELECT Order_ID, SUM(Menu_Subtotal) AS Order_Subtotal
FROM
        (SELECT ORDER_MENU.Order_ID, ORDER_MENU.Menu_ID, ORDER_MENU.Order_Quantity, MENU.Menu_Price, ORDER_MENU.Order_Quantity * MENU.Menu_Price AS Menu_Subtotal
        FROM ORDER_MENU
        LEFT JOIN MENU ON ORDER_MENU.Menu_ID = MENU.Menu_ID
        ORDER BY ORDER_MENU.Order_ID)
        GROUP BY Order_ID
        ORDER BY Order_ID;
```

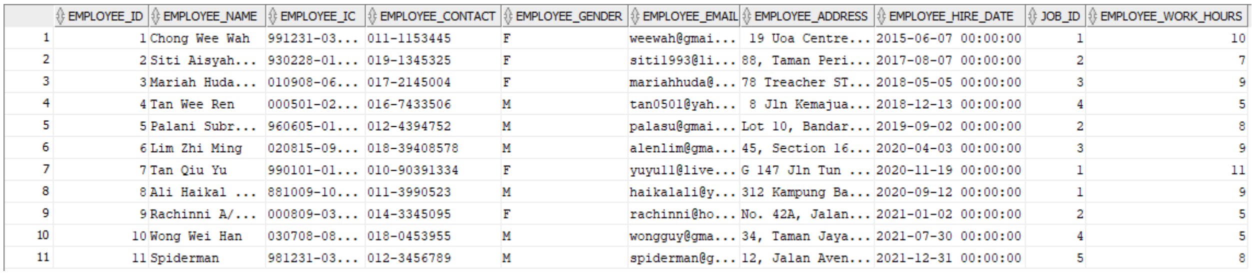


	ORDER_ID	ORDER_SUBTOTAL
1	1	24.3
2	2	20.3
3	3	46.6
4	4	42.6
5	5	21.3
6	6	25.6
7	7	44.2
8	8	15.3
9	9	37.6
10	10	42
11	11	24
12	12	91.5

2.16 SELECT, MULTIPLY (SQL Arithmetic Operator), ALIASES, INNER JOIN, ORDER BY, DESCENDING

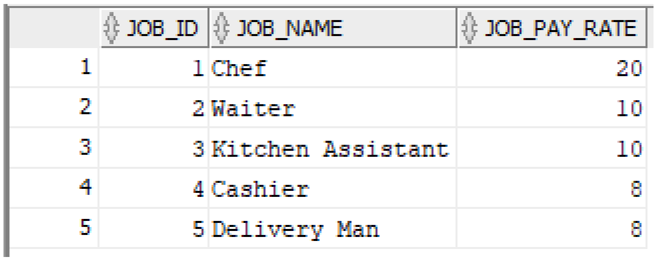
Situation: The employer is responsible to pay the salary to each employee once per week. Calculate each employees' salary per week according to their job type in descending order.

EMPLOYEES TABLE



	EMPLOYEE_ID	EMPLOYEE_NAME	EMPLOYEE_IC	EMPLOYEE_CONTACT	EMPLOYEE_GENDER	EMPLOYEE_EMAIL	EMPLOYEE_ADDRESS	EMPLOYEE_HIRE_DATE	JOB_ID	EMPLOYEE_WORK_HOURS
1	1	Chong Wee Wah	991231-03...	011-1153445	F	weewah@gmai...	19 Uoa Centre...	2015-06-07 00:00:00	1	10
2	2	Siti Aisyah...	930228-01...	019-1345325	F	siti1993@li...	88, Taman Peri...	2017-08-07 00:00:00	2	7
3	3	Mariah Huda...	010908-06...	017-2145004	F	mariahhuda@...	78 Treacher ST...	2018-05-05 00:00:00	3	9
4	4	Tan Wee Ren	000501-02...	016-7433506	M	tan0501@yah...	8 Jln Kemajua...	2018-12-13 00:00:00	4	5
5	5	Palani Subr...	960605-01...	012-4394752	M	palasu@gmai...	Lot 10, Bandar...	2019-09-02 00:00:00	2	8
6	6	Lim Zhi Ming	020815-09...	018-39408578	M	alenlim@gma...	45, Section 16...	2020-04-03 00:00:00	3	9
7	7	Tan Qiu Yu	990101-01...	010-90391334	F	yuyu11@live...	G 147 Jln Tun ...	2020-11-19 00:00:00	1	11
8	8	Ali Haikal ...	881009-10...	011-3990523	M	haikalali@y...	312 Kampung Ba...	2020-09-12 00:00:00	1	9
9	9	Rachinni A/...	000809-03...	014-3345095	F	rachinni@ho...	No. 42A, Jalan...	2021-01-02 00:00:00	2	5
10	10	Wong Wei Han	030708-08...	018-0453955	M	wongguy@gma...	34, Taman Jaya...	2021-07-30 00:00:00	4	5
11	11	Spiderman	981231-03...	012-3456789	M	spiderman@g...	12, Jalan Aven...	2021-12-31 00:00:00	5	8

JOBS TABLE



	JOB_ID	JOB_NAME	JOB_PAY_RATE
1	1	Chef	20
2	2	Waiter	10
3	3	Kitchen Assistant	10
4	4	Cashier	8
5	5	Delivery Man	8

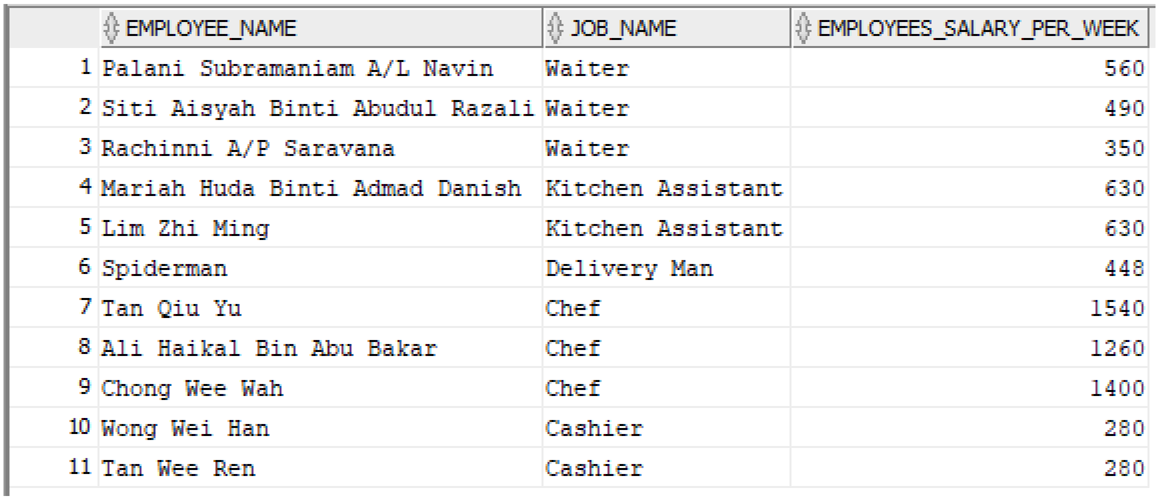
SQL Statement

SELECT EMPLOYEES.Employee_Name, JOBS.Job_Name, EMPLOYEES.Employee_Work_Hours * JOBS.Job_Pay_Rate * 7 **AS** Employees_Salary_Per_Week

FROM EMPLOYEES

INNER JOIN JOBS **ON** EMPLOYEES.Job_ID = JOBS.Job_ID

ORDER BY JOBS.Job_Name DESC;

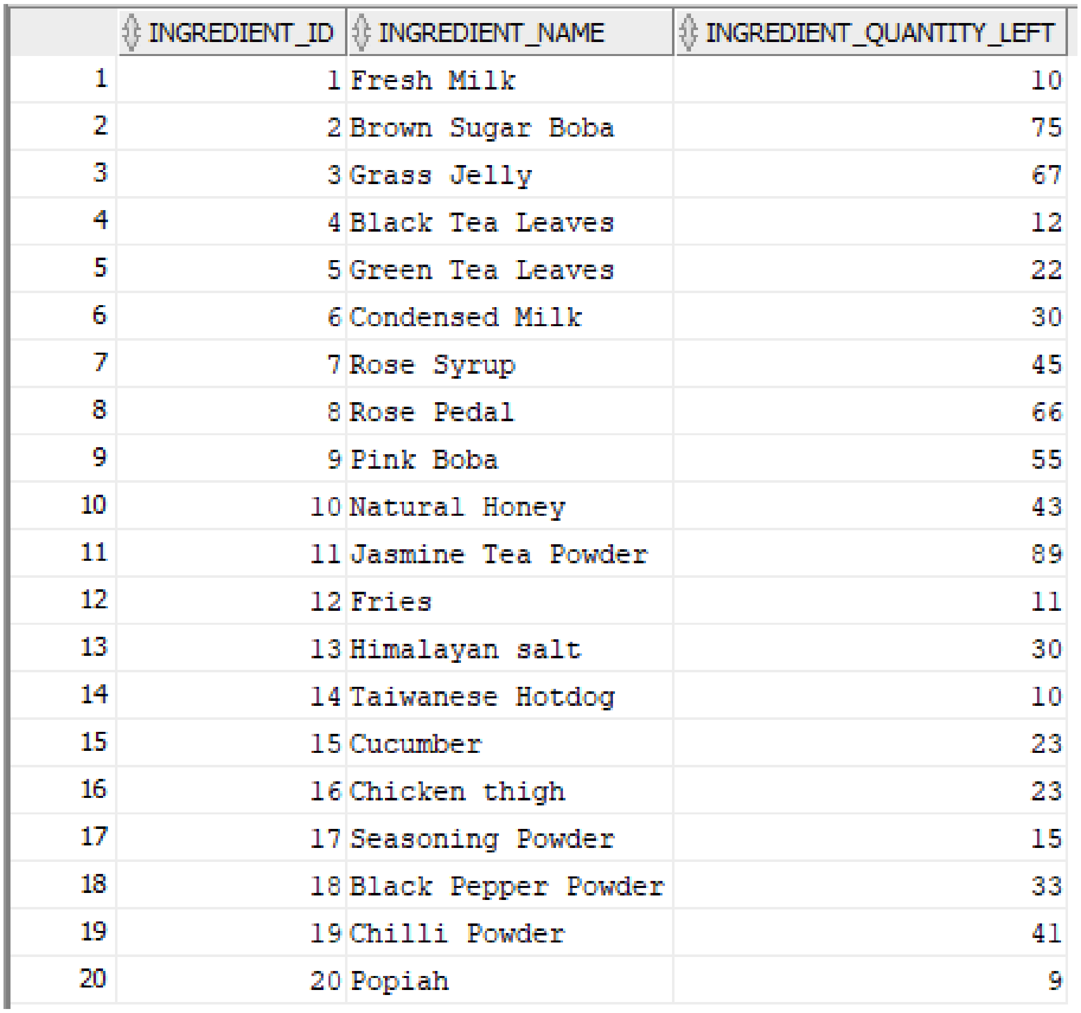


	EMPLOYEE_NAME	JOB_NAME	EMPLOYEES_SALARY_PER_WEEK
1	Palani Subramaniam A/L Navin	Waiter	560
2	Siti Aisyah Binti Abudul Razali	Waiter	490
3	Rachinni A/P Saravana	Waiter	350
4	Mariah Huda Binti Admad Danish	Kitchen Assistant	630
5	Lim Zhi Ming	Kitchen Assistant	630
6	Spiderman	Delivery Man	448
7	Tan Qiu Yu	Chef	1540
8	Ali Haikal Bin Abu Bakar	Chef	1260
9	Chong Wee Wah	Chef	1400
10	Wong Wei Han	Cashier	280
11	Tan Wee Ren	Cashier	280

2.17 SELECT, SUM(), ALIASES, ADD (SQL Arithmetic Operator), INNER JOIN, WHERE, ORDER BY, GROUP BY

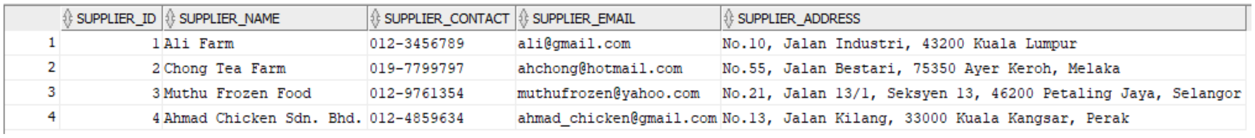
Situation 1: The employer or manager will order the ingredients with the suppliers when the quantity of each ingredient left is less than equal 30. Calculate the total quantity of each ingredient after adding the quantity of each ingredient left with each ingredient supplied by the suppliers.

INGREDIENTS TABLE



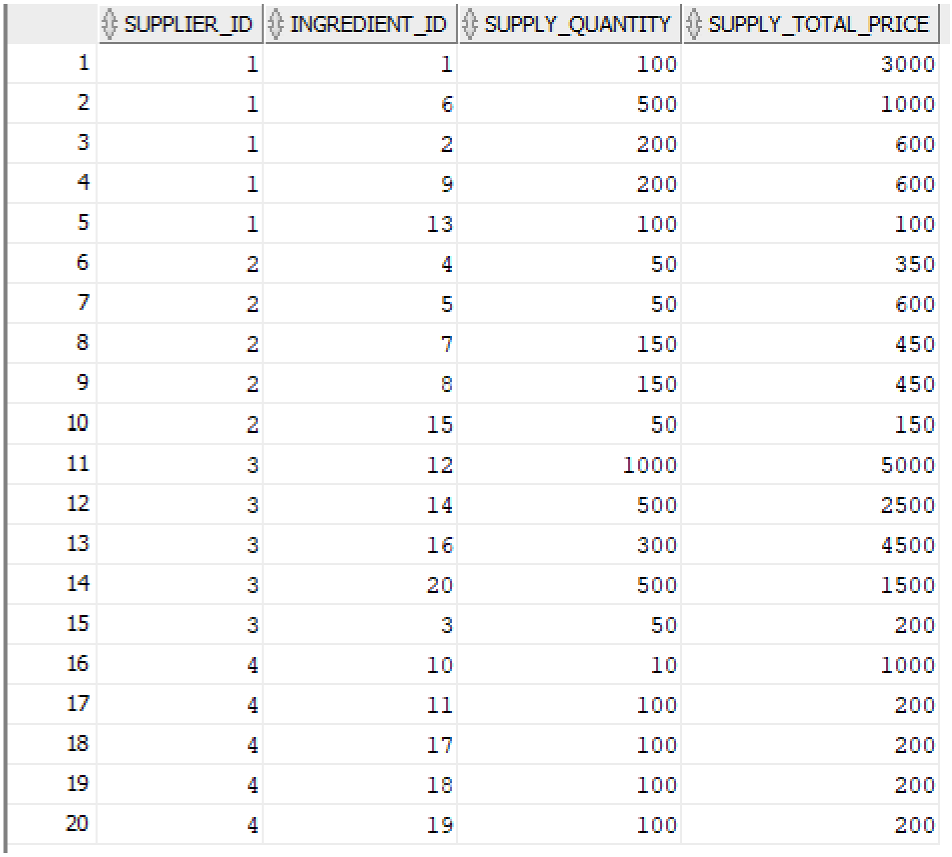
	INGREDIENT_ID	INGREDIENT_NAME	INGREDIENT_QUANTITY_LEFT
1	1	Fresh Milk	10
2	2	Brown Sugar Boba	75
3	3	Grass Jelly	67
4	4	Black Tea Leaves	12
5	5	Green Tea Leaves	22
6	6	Condensed Milk	30
7	7	Rose Syrup	45
8	8	Rose Pedal	66
9	9	Pink Boba	55
10	10	Natural Honey	43
11	11	Jasmine Tea Powder	89
12	12	Fries	11
13	13	Himalayan salt	30
14	14	Taiwanese Hotdog	10
15	15	Cucumber	23
16	16	Chicken thigh	23
17	17	Seasoning Powder	15
18	18	Black Pepper Powder	33
19	19	Chilli Powder	41
20	20	Popiah	9

SUPPLIERS TABLE



	SUPPLIER_ID	SUPPLIER_NAME	SUPPLIER_CONTACT	SUPPLIER_EMAIL	SUPPLIER_ADDRESS
1	1	Ali Farm	012-3456789	ali@gmail.com	No.10, Jalan Industri, 43200 Kuala Lumpur
2	2	Chong Tea Farm	019-7799797	ahchong@hotmail.com	No.55, Jalan Bestari, 75350 Ayer Keroh, Melaka
3	3	Muthu Frozen Food	012-9761354	muthufrozen@yahoo.com	No.21, Jalan 13/1, Seksyen 13, 46200 Petaling Jaya, Selangor
4	4	Ahmad Chicken Sdn. Bhd.	012-4859634	ahmad_chicken@gmail.com	No.13, Jalan Kilang, 33000 Kuala Kangsar, Perak

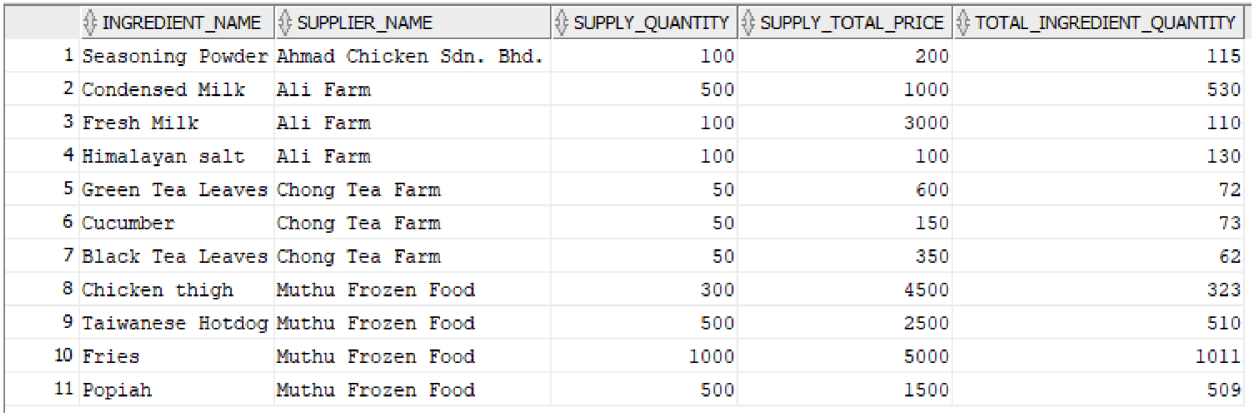
SUPPLY TABLE



	SUPPLIER_ID	INGREDIENT_ID	SUPPLY_QUANTITY	SUPPLY_TOTAL_PRICE
1	1	1	100	3000
2	1	6	500	1000
3	1	2	200	600
4	1	9	200	600
5	1	13	100	100
6	2	4	50	350
7	2	5	50	600
8	2	7	150	450
9	2	8	150	450
10	2	15	50	150
11	3	12	1000	5000
12	3	14	500	2500
13	3	16	300	4500
14	3	20	500	1500
15	3	3	50	200
16	4	10	10	1000
17	4	11	100	200
18	4	17	100	200
19	4	18	100	200
20	4	19	100	200

SQL Statement

```
SELECT INGREDIENTS.Ingredient_Name, SUPPLIERS.Supplier_Name,
SUPPLY.Supply_Quantity, SUPPLY.Supply_Total_Price,
INGREDIENTS.Ingredient_Quantity_Left + SUPPLY.Supply_Quantity AS
Total_Ingredient_Quantity

FROM SUPPLY

INNER JOIN INGREDIENTS ON SUPPLY.Ingredient_ID = INGREDIENTS.Ingredient_ID

INNER JOIN SUPPLIERS ON SUPPLY.Supplier_ID = SUPPLIERS.Supplier_ID

WHERE INGREDIENTS.Ingredient_Quantity_Left <= 30

ORDER BY SUPPLIERS.Supplier_Name;
```

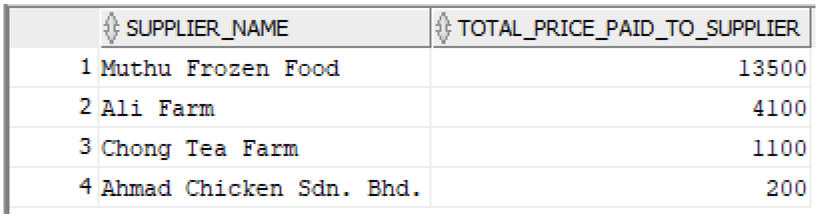


	INGREDIENT_NAME	SUPPLIER_NAME	SUPPLY_QUANTITY	SUPPLY_TOTAL_PRICE	TOTAL_INGREDIENT_QUANTITY
1	Seasoning Powder	Ahmad Chicken Sdn. Bhd.	100	200	115
2	Condensed Milk	Ali Farm	500	1000	530
3	Fresh Milk	Ali Farm	100	3000	110
4	Himalayan salt	Ali Farm	100	100	130
5	Green Tea Leaves	Chong Tea Farm	50	600	72
6	Cucumber	Chong Tea Farm	50	150	73
7	Black Tea Leaves	Chong Tea Farm	50	350	62
8	Chicken thigh	Muthu Frozen Food	300	4500	323
9	Taiwanese Hotdog	Muthu Frozen Food	500	2500	510
10	Fries	Muthu Frozen Food	1000	5000	1011
11	Popiah	Muthu Frozen Food	500	1500	509

Continue Situation 1: After the employer receives the ingredients from the suppliers, the employer needs to pay the suppliers. Calculate the total price that is needed to pay to each supplier.

SQL Statement

```
SELECT Supplier_Name, SUM(Supply_Total_Price) AS Total_Price_Paid_To_Supplier

FROM(

        SELECT INGREDIENTS.Ingredient_Name, SUPPLIERS.Supplier_Name,
        SUPPLY.Supply_Quantity, SUPPLY.Supply_Total_Price,
        INGREDIENTS.Ingredient_Quantity_Left + SUPPLY.Supply_Quantity AS
        Total_Ingredient_Quantity

        FROM SUPPLY

        INNER JOIN INGREDIENTS ON SUPPLY.Ingredient_ID = INGREDIENTS.Ingredient_ID

        INNER JOIN SUPPLIERS ON SUPPLY.Supplier_ID = SUPPLIERS.Supplier_ID

        WHERE INGREDIENTS.Ingredient_Quantity_Left <= 30

        ORDER BY Suppliers.Supplier_Name)

GROUP BY Supplier_Name;
```

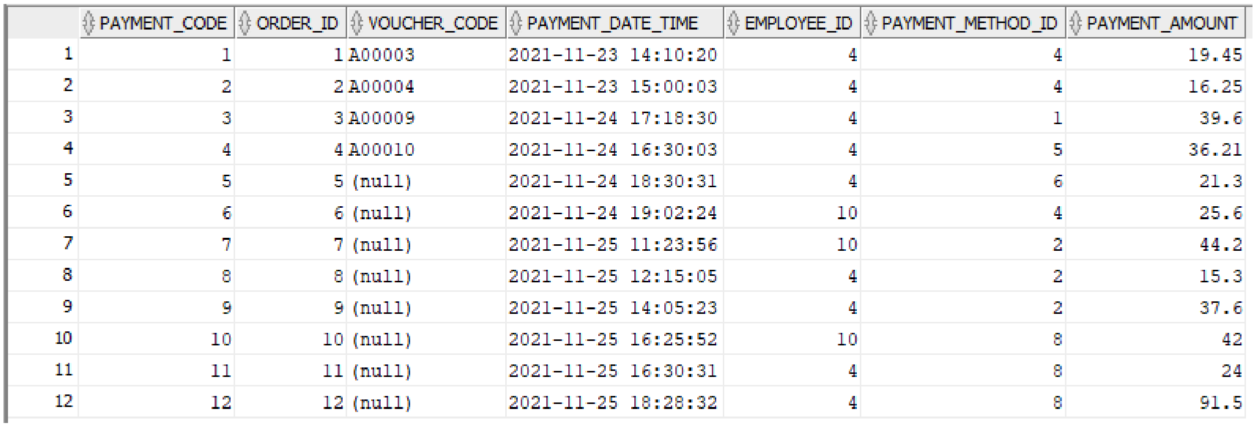


	SUPPLIER_NAME	TOTAL_PRICE_PAID_TO_SUPPLIER
1	Muthu Frozen Food	13500
2	Ali Farm	4100
3	Chong Tea Farm	1100
4	Ahmad Chicken Sdn. Bhd.	200

2.18 SELECT, COUNT(), SUM(), AVG(), ALIASES, LEFT JOIN, GROUP BY, ORDER BY, DESCENDING

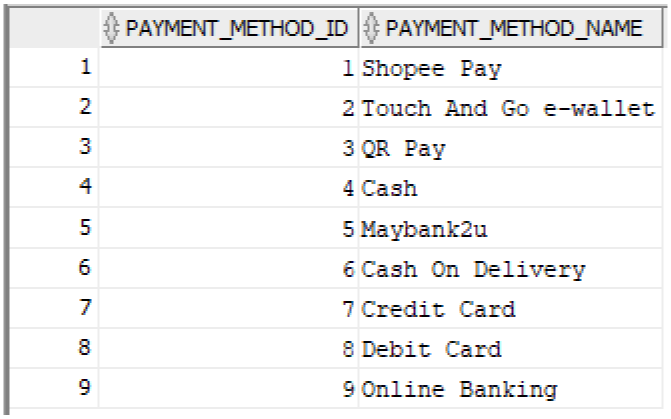
Situation: The employer can calculate the total payments by using each payment method that is made by the customers. Besides, the employer also can calculate the number of payments using each payment method and the average price in each payment by using the payment method.

PAYMENTS TABLE



	PAYMENT_CODE	ORDER_ID	VOUCHER_CODE	PAYMENT_DATE_TIME	EMPLOYEE_ID	PAYMENT_METHOD_ID	PAYMENT_AMOUNT
1	1	1	A00003	2021-11-23 14:10:20	4	4	19.45
2	2	2	A00004	2021-11-23 15:00:03	4	4	16.25
3	3	3	A00009	2021-11-24 17:18:30	4	1	39.6
4	4	4	A00010	2021-11-24 16:30:03	4	5	36.21
5	5	5	(null)	2021-11-24 18:30:31	4	6	21.3
6	6	6	(null)	2021-11-24 19:02:24	10	4	25.6
7	7	7	(null)	2021-11-25 11:23:56	10	2	44.2
8	8	8	(null)	2021-11-25 12:15:05	4	2	15.3
9	9	9	(null)	2021-11-25 14:05:23	4	2	37.6
10	10	10	(null)	2021-11-25 16:25:52	10	8	42
11	11	11	(null)	2021-11-25 16:30:31	4	8	24
12	12	12	(null)	2021-11-25 18:28:32	4	8	91.5

PAYMENT_METHODS TABLE



	PAYMENT_METHOD_ID	PAYMENT_METHOD_NAME
1	1	Shopee Pay
2	2	Touch And Go e-wallet
3	3	QR Pay
4	4	Cash
5	5	Maybank2u
6	6	Cash On Delivery
7	7	Credit Card
8	8	Debit Card
9	9	Online Banking

<u>SQL Statement</u>

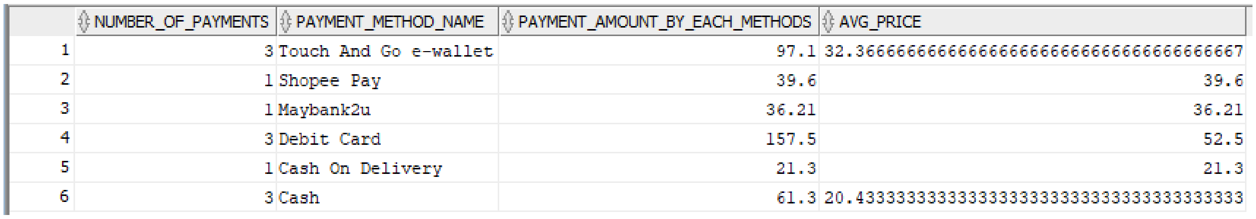
SELECT **COUNT(**PAYMENTS.Payment_Code**)** **AS** Number_Of_Payments, PAYMENT_METHODS.Payment_Method_Name, **SUM(**PAYMENTS.Payment_Amount**)** **AS** Payment_Amount_By_Each_Methods, **AVG(**PAYMENTS.Payment_Amount**)** **AS** AVG_Price

FROM PAYMENTS

LEFT JOIN PAYMENT_METHODS **ON** PAYMENTS.Payment_Method_ID = PAYMENT_METHODS.Payment_Method_ID

GROUP BY Payment_Method_Name

ORDER BY Payment_Method_Name DESC;



	NUMBER_OF_PAYMENTS	PAYMENT_METHOD_NAME	PAYMENT_AMOUNT_BY_EACH_METHODS	AVG_PRICE
1	3	Touch And Go e-wallet	97.1	32.366666666666666666666666666666666667
2	1	Shopee Pay	39.6	39.6
3	1	Maybank2u	36.21	36.21
4	3	Debit Card	157.5	52.5
5	1	Cash On Delivery	21.3	21.3
6	3	Cash	61.3	20.433333333333333333333333333333333333

2.0 Database Testing

No	Date	Test Description	Input	Expected Output	Result	Action
1	27/12/2021	Column testing- Validation of the presence of any unused/ unmapped database tables/ columns	Voucher, Payment, Employee, Job, Payment_Method, Order, Customer, Delivery, Order_Menu, Menu, Ingredient, Supply, Supplier, Menu_Ingredient	All the tables created are mapped to the database tables.	The output is just like the expected outcome.	None
2	27/12/2021	Validation for all the constraints (Primary key and Foreign key) between the database table	Voucher_Code, Payment_Code, Employee_ID, Job_ID, Payment_Method_ID, Order_ID, Customer_ID, Menu_ID, Ingredient_ID, Supplier_ID	All the constraints created are perfectly joined in the database table	The output is just like the expected outcome.	None
3	05/01/2022	Validation of the compatibility of the data type and field lengths.	All the data is inserted into each of the tables with a minimum of one set of data in each table	All the data is inserted into each of the tables with a minimum of 5 data in each table	Some of the data types required modification and some of the field lengths were too huge. (waste space)	Change the data type to the compatible type and fixed the field lengths just suitable to the data.
4	10/01//2022	Checking whether the data queries are logically well organized and the database tables are well implemented	All the data, records, table, constraints and queries inside the database tables	All the data, records, table, constraints and inside the database tables are well implemented.	The output is just like the expected outcome.	None

3.0 Difficulties and Problems

3.1 Problems:

1. The data of business processes and business rules collected are not completed during the data collection process.
2. No idea on how to design a database based on the data collected before.
3. Not familiar with the knowledge of designing the database using SQL statements.
4. Limitation of communication during the discussion because some of our group members practice online learning at home.
5. Time constraints in completing this assignment due to heavy workload from other courses.
6. Errors on the implementable Entity Relationship Diagram.

3.2 Solutions:

1. We solved this problem by conducting several physical and virtual meetings with the manager of Ke Nina Cafe to get more information needed.
2. We have a group discussion with our group members, and also seek opinions and suggestions from other friends and our seniors in order to have a clear understanding of how to design the database.
3. We do the self-learning session on the SQL tutorial provided by the W3schools online learning platform.
4. We have meetings every weekend to make sure all of us understand the assignment task and work in the correct direction.
5. We distribute our tasks to each of our group members and set a due date for the tasks to ensure our progress meets the schedule that we had planned.
6. We do the correction from the previous version of the implementable Entity Relationship Diagram and replace it with the correct version for the database design using SQL statements.

3.3 Future improvement:

1. Learn cloud-based database management to design an online database so that the employer can access it at any time.
2. Approach the company earlier to get business processes and business rules to ensure more features can be added to the database.
3. Improve database to store the information of each branch of the restaurant if the employer opens new branches.
4. Improve database to record the data of all the managers of each branch.
5. Improve this database to help the employer or managers to record the ingredients expiry date so that they will not need to remember or record using spreadsheets like Excel.

3.4 Lesson learnt:

1. Time management is important for all members of the team to complete the assignment on time.
2. Cooperation between group members can help us solve the problems we face more easily.
3. There are many resources we can find from online platforms such as the W3schools platform.

4.0 Peer Work Group Evaluation Form

Course: WIA2001 Semester: 1 Session: 2021/2022
Lecturer: DR. FARIZA HANUM BINTI MD NASARUDDIN
Assignment:**Group Project**

Evaluator (Student's Name): **FRANKIE LIM QI QUAN**
Date: 13 JANUARY 2022

Group Members:

	Matrix Number	Name
1	U2005382/1	GOH CHEE LAM
2	17206915/2	FARISAH ATHIRA BINTI MD ZAMRI
3	U2005396/1	PUA JING YI
4	U2005394/1	KELVIN CHEAH

Directions: In the space below, honestly evaluate the work of other students in your group by answering **yes** or **no** and by using a scale from 1 to 3, **1 being poor, 2 being average, 3 being above average. Please circle your answer.**

		Group Member 1	Group Member 2	Group Member 3	Group Member 4
1	Did this group member complete his/her assigned tasks for the group	Yes	Yes	Yes	Yes
2	How would you rate the quality of this person's work	3	3	3	3
3	How would you rate the timeliness of the completion of the work?	3	3	3	3
4	How would you rate the accuracy of the work	3	3	3	3
5	Overall, how would you rank this group member's performance in the group?	3	3	3	3
6	Would you want to work with this person again? Explain why in the space below.	Yes	Yes	Yes	Yes
		Perfect group members.			

Course: WIA2001 Semester: 1 Session: 2021/2022
Lecturer: DR. FARIZA HANUM BINTI MD NASARUDDIN
Assignment:**Group Project**

Evaluator (Student's Name): **PUA JING YI**
Date: 14 JANUARY 2022

Group Members:

	Matrix Number	Name
1	U2005382/1	GOH CHEE LAM
2	U2005263/1	FRANKIE LIM QI QUAN
3	17206915/2	FARISAH ATHIRA BINTI MD ZAMRI
4	U2005394/1	KELVIN CHEAH

Directions: In the space below, honestly evaluate the work of other students in your group by answering **yes** or **no** and by using a scale from 1 to 3, **1 being poor, 2 being average, 3 being above average. Please circle your answer.**

		Group Member 1	Group Member 2	Group Member 3	Group Member 4
1	Did this group member complete his/her assigned tasks for the group	Yes	Yes	Yes	Yes
2	How would you rate the quality of this person's work	3	3	3	3
3	How would you rate the timeliness of the completion of the work?	3	3	3	3
4	How would you rate the accuracy of the work	3	3	3	3
5	Overall, how would you rank this group member's performance in the group?	3	3	3	3
6	Would you want to work with this person again? Explain why in the space below.	Yes	Yes	Yes	Yes
		Clear division of work and help each other between the group members.			

Course: WIA2001 Semester: 1 Session: 2021/2022
Lecturer: DR. FARIZA HANUM BINTI MD NASARUDDIN
Assignment: **Group Project**

Evaluator (Student's Name): **Farisah Athira Binti Md Zamri**
Date: 13/02/2022

Group Members:

	Matrix Number	Name
1	U2005263/1	FRANKIE LIM QI QUAN
2	U2005382/1	GOH CHEE LAM
3	U2005396/1	PUA JING YI
4	U2005394/1	KELVIN CHEAH

Directions: In the space below, honestly evaluate the work of other students in your group by answering **yes** or **no** and by using a scale from 1 to 3, **1 being poor, 2 being average, 3 being above average. Please circle your answer.**

		Group Member 1	Group Member 2	Group Member 3	Group Member 4
1	Did this group member complete his/her assigned tasks for the group	[✓ Yes] No	[✓ Yes] No	[✓ Yes] No	[✓ Yes] No
2	How would you rate the quality of this person's work	1 2 [✓ 3]	1 2 [✓ 3]	1 2 [✓ 3]	1 2 [✓ 3]
3	How would you rate the timeliness of the completion of the work?	1 2 [✓ 3]	1 2 [✓ 3]	1 2 [✓ 3]	1 2 [✓ 3]
4	How would you rate the accuracy of the work	1 2 [✓ 3]	1 2 [✓ 3]	1 2 [✓ 3]	1 2 [✓ 3]
5	Overall, how would you rank this group member's performance in the group?	1 2 [✓ 3]	1 2 [✓ 3]	1 2 [✓ 3]	1 2 [✓ 3]
6	Would you want to work with this person again? Explain why in the space below.	[✓ Yes] No	[✓ Yes] No	[✓ Yes] No	[✓ Yes] No
		Because all of them are very cooperative and contribute to the group.			

Course: WIA2001 Semester: 1 Session: 2021/2022
Lecturer: DR. FARIZA HANUM BINTI MD NASARUDDIN
Assignment: **Group Project**

Evaluator (Student's Name): **GOH CHEE LAM**
Date: 13 JANUARY 2022

Group Members:

	Matrix Number	Name
1	U2005263/1	FRANKIE LIM QI QUAN
2	17206915/2	FARISAH ATHIRA BINTI MD ZAMRI
3	U2005396/1	PUA JING YI
4	U2005394/1	KELVIN CHEAH

Directions: In the space below, honestly evaluate the work of other students in your group by answering **yes** or **no** and by using a scale from 1 to 3, **1 being poor, 2 being average, 3 being above average. Please circle your answer.**

		Group Member 1	Group Member 2	Group Member 3	Group Member 4
1	Did this group member complete his/her assigned tasks for the group	Yes	Yes	Yes	Yes
2	How would you rate the quality of this person's work	3	3	3	3
3	How would you rate the timeliness of the completion of the work?	3	3	3	3
4	How would you rate the accuracy of the work	3	3	3	3
5	Overall, how would you rank this group member's performance in the group?	3	3	3	3
6	Would you want to work with this person again? Explain why in the space below.	Yes	Yes	Yes	Yes
		I like to work with them because they show high responsibility in doing the project.			

Course: WIA2001 Semester: 1 Session: 2021/2022
Lecturer: DR. FARIZA HANUM BINTI MD NASARUDDIN
Assignment: **Group Project**

Evaluator (Student's Name): **KELVIN CHEAH**
Date: 13 JANUARY 2022

Group Members:

	Matrix Number	Name
1	U2005263/1	FRANKIE LIM QI QUAN
2	17206915/2	FARISAH ATHIRA BINTI MD ZAMRI
3	U2005396/1	PUA JING YI
4	U2005382/1	GOH CHEE LAM

Directions: In the space below, honestly evaluate the work of other students in your group by answering **yes** or **no** and by using a scale from 1 to 3, **1 being poor, 2 being average, 3 being above average. Please circle your answer.**

		Group Member 1	Group Member 2	Group Member 3	Group Member 4
1	Did this group member complete his/her assigned tasks for the group	Yes	Yes	Yes	Yes
2	How would you rate the quality of this person's work	3	3	3	3
3	How would you rate the timeliness of the completion of the work?	3	3	3	3
4	How would you rate the accuracy of the work	3	3	3	3
5	Overall, how would you rank this group member's performance in the group?	3	3	3	3
6	Would you want to work with this person again? Explain why in the space below.	Yes	Yes	Yes	Yes
		I like to work with them because they are cooperative.			