



TIS1101 Database Fundamentals

Assignment 2

Title: FoodPanda Food Delivery

Prepared by: Group 8

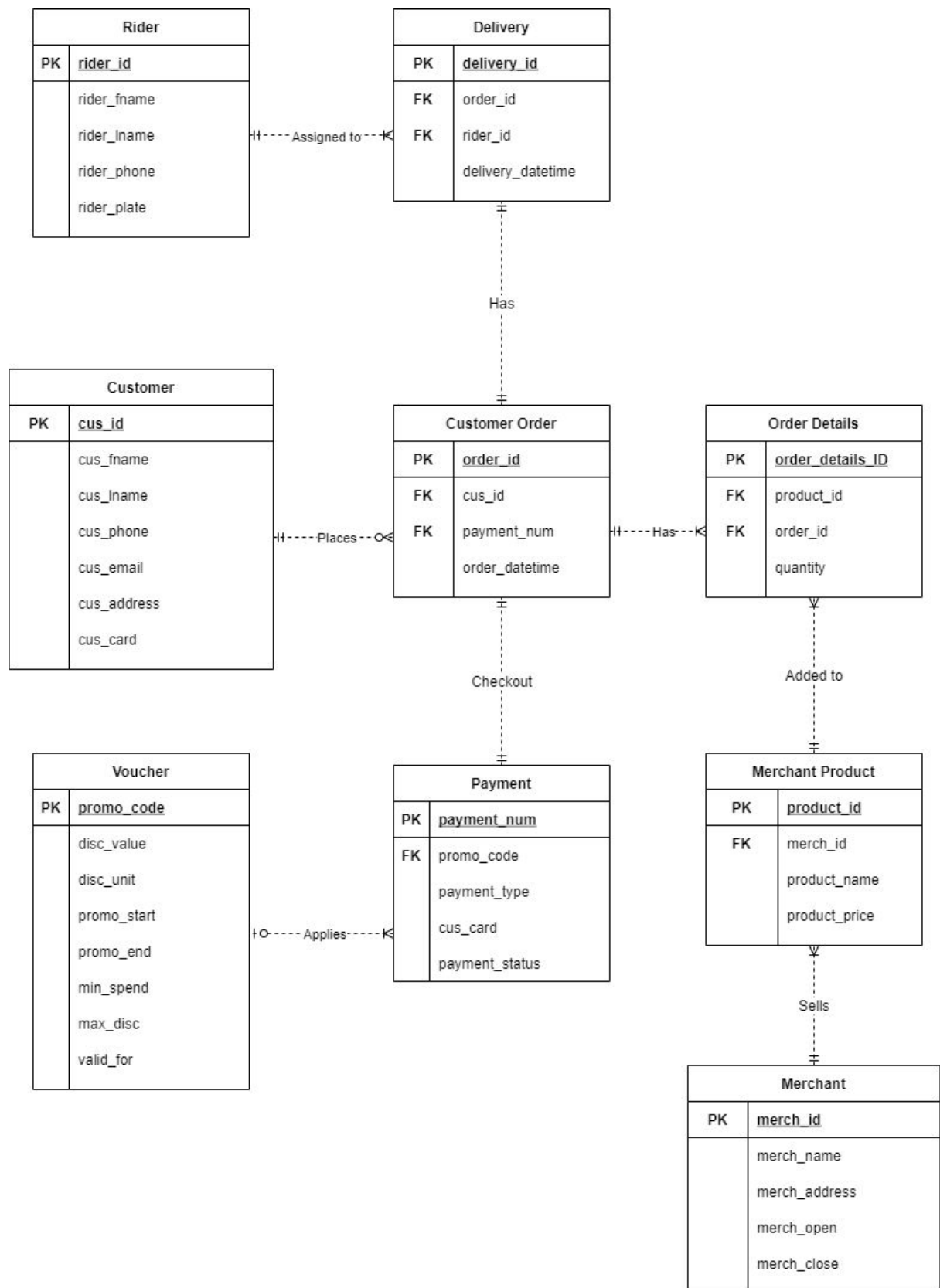
ID	NAME	EMAIL
1181103501	Lim Wei Jie (Leader)	1181103501@student.mmu.edu.my
1181103353	Tee Ming Yuan	1181103353@student.mmu.edu.my
1181103230	Loo Chen Zhi	1181103230@student.mmu.edu.my
1181103362	Chang See Jie	1181103362@student.mmu.edu.my

Table of Contents

Table of Contents	1
Entity-Relationship Diagram (ER-D)	3
Data Dictionary	4
Creation of Tables	8
3.1 Table Name: customer	8
3.2 Table Name: rider	8
3.3 Table Name: merchant	8
3.4 Table Name: voucher	8
3.5 Table Name: payment	9
3.6 Table Name: merchant_product	9
3.7 Table Name: Customer Order	9
3.8 Table Name: Order Details	9
3.9 Table Name: Delivery	10
Data Insertion	11
4.1 Table Name: customer	11
4.2 Table Name: rider	11
4.3 Table Name: merchant	12
4.4 Table Name: voucher	12
4.5 Table Name: payment	13
4.6 Table Name: merchant_product	13
4.7 Table Name: customer_order	14
4.8 Table Name: order_details	15
4.9 Table Name: Delivery	15
Data Manipulation with SQL	16
5.1 AGGREGATE FUNCTION (COUNT, MAX, MIN, AVG, SUM)	16
5.1.1 Get the ID of active customers and number of purchase (COUNT)	16
5.1.2 Get the product with highest price among all merchants (MAX)	16
5.1.3 Get the voucher with the smallest minimum spend (MIN)	16
5.1.4 Get the average delivery time (AVG)	17
5.1.5 Get the total number of items ordered for each order (SUM)	17
5.2 GROUP BY AND HAVING CLAUSES	18
5.2.1 Get the amount of product purchased that have prices lower than 10	18
5.3 TRIGGERS	19

5.3.1 Create a trigger to apply delivery fee on order after March 2021	19
5.3.2 Merchants are not allowed to extend closing time but allowed to shorten closing time	21
5.4 STORED PROCEDURE	23
5.4.1 Users can select merchants to increase product price by 3% or 5% according to original price.	23
5.5 VIEW	25
5.5.1 Create a view for products having price larger than 10	25
5.5.2 Create a view for merchants having opening time at or after 11am	25
5.6 USE OF SUBQUERIES/NESTED QUERIES	26
5.6.1 Get the subtotal for each order	26
5.6.2 Get the rider who have delivered equal or more than 2 orders	26
5.7 QUERIES NOT COVERED IN LECTURE	27
5.7.1 Top 5 lowest product price among all merchants (Using ROWNUM)	27
5.7.2 Generate VIP number with auto increment by 1 for customers who have more than 1 order history (Using CREATE SEQUENCE)	27
5.7.3 Get the lowest price product for each merchant (Using RANK)	28
5.7.4 Get February 2021 order history (Using EXTRACT)	29
Contribution	30

1. Entity-Relationship Diagram (ER-D)



2. Data Dictionary

Table Name	Field Name	Description	Data Type	PK/ FK	FK Reference Table
customer	cus_id	Customer account ID	VARCHAR(5)	PK	
	cus_fname	Customer first name	VARCHAR(15)		
	cus_lname	Customer last name	VARCHAR(15)		
	cus_phone	Customer phone number	NUMBER(11)		
	cus_email	Customer email address	VARCHAR(30)		
	cus_address	Customer address	VARCHAR(80)		
	cus_card	Customer credit or debit card number	NUMBER(16)		
rider	rider_id	Rider account ID	VARCHAR(5)	PK	
	rider_fname	Rider first name	VARCHAR(30)		
	rider_lname	Rider Last name	VARCHAR(30)		
	rider_phone	Rider phone number	NUMBER(11)		
	rider_plate	Rider vehicle plate number	VARCHAR(7)		
merchant	merch_id	Merchant ID	VARCHAR(5)	PK	
	merch_name	Merchant name	VARCHAR(40)		

	merch_addresses	Merchant address	VARCHAR(100)		
	merch_open	Merchant opening time	NUMBER(4)		
	merch_close	Merchant closing time	NUMBER(4)		
voucher	promo_code	Promotional codes	VARCHAR(20)	PK	
	disc_value	Discount value	NUMBER(2)		
	disc_unit	Discount unit (Percentage/Flat currency amount)	VARCHAR(10)		
	promo_start	Promotion starts date	DATE		
	promo_end	Promotion ends date	DATE		
	min_spend	Minimum spend to qualify for the promotion	NUMBER(2)		
	max_disc	Maximum discount amount	NUMBER(2)		
	valid_for	Promotion terms and conditions	VARCHAR(30)		
payment	payment_num	Payment number	VARCHAR(5)	PK	
	promo_code	Promotional code	VARCHAR(20)	FK	voucher
	payment_type	Payment type (Cash/Online banking/Credit or Debit card)	VARCHAR(20)		

	cus_card	Customer credit or debit card number	NUMBER(16)		
	payment_status	Payment status (Unpaid/Failed/Expired/Paid/Refunding/Refunded)	VARCHAR(10)		
merchant_product	product_id	Product ID	VARCHAR(7)	PK	
	merch_id	Merchant ID	VARCHAR(5)	FK	merchant
	product_name	Product name	VARCHAR(30)		
	product_price	Product price	NUMBER(5,2)		
customer_order	order_id	Order ID	VARCHAR(5)	PK	
	cus_id	Customer account ID	VARCHAR(5)	FK	customer
	payment_num	Payment number	VARCHAR(5)	FK	payment
	order_datetime	Order date time	TIMESTAMP		
order_details	order_details_id	Order details ID	VARCHAR(7)	PK	
	product_id	Product ID	VARCHAR(7)	FK	product
	order_id	Order ID	VARCHAR(5)	FK	customer_order
	quantity	Product quantity	NUMBER(2)		

delivery	delivery_id	Delivery ID	VARCHAR(5)	PK	
	order_id	Order ID	VARCHAR(5)	FK	customer_order
	rider_id	Rider account ID	VARCHAR(5)	FK	rider
	deliery_dateti me	Delivery date time	TIMESTAMP		

3. Creation of Tables

3.1 Table Name: customer

```
CREATE TABLE customer (  
  cus_id    VARCHAR(5) NOT NULL PRIMARY KEY,  
  cus_fname VARCHAR(15),  
  cus_lname VARCHAR(15),  
  cus_phone NUMBER(11),  
  cus_email VARCHAR(30),  
  cus_address VARCHAR(80),  
  cus_card  NUMBER(16)  
)
```

3.2 Table Name: rider

```
CREATE TABLE rider(  
  rider_id  VARCHAR(5) NOT NULL PRIMARY KEY,  
  rider_fname VARCHAR (30),  
  rider_lname VARCHAR (30),  
  rider_phone NUMBER (11),  
  rider_plate VARCHAR (7)  
)
```

3.3 Table Name: merchant

```
CREATE TABLE merchant (  
  merch_id    VARCHAR(5) NOT NULL PRIMARY KEY,  
  merch_name  VARCHAR(40),  
  merch_address VARCHAR(100),  
  merch_open  NUMBER(4),  
  merch_close NUMBER(4)  
)
```

3.4 Table Name: voucher

```
CREATE TABLE voucher (  
  promo_code VARCHAR(20) NOT NULL PRIMARY KEY,  
  disc_value  NUMBER(2),  
  disc_unit   VARCHAR(10),  
  promo_start DATE,  
  promo_end   DATE,  
  min_spend   NUMBER(2),  
  max_disc    NUMBER(2),  
  valid_for   VARCHAR(30)  
)
```

3.5 Table Name: payment

```
CREATE TABLE payment (  
    payment_num VARCHAR(5) NOT NULL PRIMARY KEY,  
    promo_code VARCHAR(20),  
    payment_type VARCHAR(20),  
    cus_card NUMBER(16),  
    payment_status VARCHAR(10),  
    FOREIGN KEY (promo_code) REFERENCES voucher  
)
```

3.6 Table Name: merchant_product

```
CREATE TABLE merchant_product (  
    product_id VARCHAR(7) NOT NULL PRIMARY KEY,  
    merch_id VARCHAR(5),  
    product_name VARCHAR(30),  
    product_price NUMBER(5,2),  
    FOREIGN KEY (merch_id) REFERENCES merchant(merch_id)  
)
```

3.7 Table Name: Customer Order

```
CREATE TABLE customer_order (  
    order_id VARCHAR(5) NOT NULL PRIMARY KEY,  
    cus_id VARCHAR(5) NOT NULL,  
    payment_num VARCHAR(5) NOT NULL,  
    order_datetime TIMESTAMP,  
    FOREIGN KEY (cus_id) REFERENCES customer(cus_id),  
    FOREIGN KEY (payment_num) REFERENCES payment(payment_num)  
)
```

3.8 Table Name: Order Details

```
CREATE TABLE order_details (  
    order_details_ID VARCHAR(7) NOT NULL PRIMARY KEY,  
    product_id VARCHAR(7),  
    order_id VARCHAR(5),  
    quantity NUMBER(2),  
    FOREIGN KEY (order_id) REFERENCES customer_order(order_id),  
    FOREIGN KEY (product_id) REFERENCES merchant_product(product_id)  
)
```

3.9 Table Name: Delivery

```
CREATE TABLE delivery (  
  delivery_id    VARCHAR(5) NOT NULL PRIMARY KEY,  
  order_id      VARCHAR(5),  
  rider_id      VARCHAR(5),  
  delivery_datetime  TIMESTAMP,  
  FOREIGN KEY (order_id) REFERENCES customer_order(order_id),  
  FOREIGN KEY (rider_id) REFERENCES rider(rider_id)  
)
```

4. Data Insertion

4.1 Table Name: customer

`INSERT INTO customer VALUES ('C0001', 'Lee', 'Zi Sheng', 60133721917, 'zisheng@gmail.com', '74, Jalan Mahmood, Bandar Kota Bharu, 15200 Kota Bharu, Kelantan', 5479188719084505);`

`INSERT INTO customer VALUES ('C0002', 'Mahsuri', 'Nawar', 60188252238, 'mahsuri@hotmail.com', '10, Jalan Yeo Guan Hup, 41000 Klang, Selangor', NULL);`

`INSERT INTO customer VALUES ('C0003', 'Tan', 'Jessica', 60179316719, 'jessica@hotmail.com', '1083, Jln Merpati, 98000 Miri, Sarawak', 5546635239309761);`

`INSERT INTO customer VALUES ('C0004', 'Wong', 'Jun Jie', 60140733537, 'junjie@gmail.com', '19, Jln Pjs 11/7, Taman Bandar Sunway, 46150 Petaling Jaya, Selangor', 4532972414825033);`

`INSERT INTO customer VALUES ('C0005', 'Kong', 'Xin Yen', 60129745385, 'xyen@gmail.com', '5, Jalan Sungai Tiram 2, 11900 Bayan Lepas, Penang', NULL);`

`INSERT INTO customer VALUES ('C0006', 'Mark', 'Lee', 60163137981, 'mark99@hotmail.com', '241, Jalan PJS 6/5C, PJS 6, 46150 Petaling Jaya, Selangor', 4916431891509681);`

`INSERT INTO customer VALUES ('C0007', 'Maya', 'Izzati', 60178462874, 'mayamaya@hotmail.com', '2, Jln Lada Hitam, Taman Makmur, 85000 Kluang, Johor', 5244738200881183);`

4.2 Table Name: rider

`INSERT INTO rider VALUES ('R0001', 'Haashid', 'Tareef', 60121455591, 'JDL1177');`

`INSERT INTO rider VALUES ('R0002', 'Abdur Rahman', 'Sadi', 60141455530, 'JVS3268');`

`INSERT INTO rider VALUES ('R0003', 'Eddy Siong', 'Pao Ziang', 60125455542, 'WQF360');`

`INSERT INTO rider VALUES ('R0004', 'Muhamad', 'Yaqzan Azni', 60135865577, 'PDC8228');`

`INSERT INTO rider VALUES ('R0005', 'Muhamad Sharizal Bashah', 'Safrizal Firdaus', 60198649871, 'LS8250');`

`INSERT INTO rider VALUES ('R0006', 'Murugayan Munusamy', 'Pakiam', 60193730121, 'LMF7699');`

`INSERT INTO rider VALUES ('R0007', 'Kwong', 'Pheng Foo', 60169867489, 'KDW55');`

`INSERT INTO rider VALUES ('R0008', 'Sukhjit', 'Theseira', 60167647466, 'AJW3399');`

```
INSERT INTO rider VALUES ('R0009', 'Ng', 'Khor Hung', 60142357786,
'VAT9850');
INSERT INTO rider VALUES ('R0010', 'Mutahir', 'Chandran', 60123311969,
'NBG6460');
```

4.3 Table Name: merchant

```
INSERT INTO merchant VALUES ('M0001', 'Nasi Lalap Afieyah', 'Lorong
Bahagia, Taman Chi Leong, Pelabuhan Klang, Selangor', 1000, 2000);
INSERT INTO merchant VALUES ('M0002', 'HODI', '49, Jalan Ria Jaya 1,
Taman Ria Jaya, 32000 Sitiawan, Negeri Perak', 1115, 2045);
INSERT INTO merchant VALUES ('M0003', 'N Food Stall @ 138 Cafe', 'Lot
279, No.19, Jalan Nahkota Gampar, Beautiful Jade Centre, 98000 Miri,
Sarawak', 1730, 2130);
INSERT INTO merchant VALUES ('M0004', 'Lung Meng Fried Radish Cake',
'NEW LUNG MENG ENTERPRISE CO. LOT.2911 BLK5 MCLD MIRI', 0800,
1130);
INSERT INTO merchant VALUES ('M0005', 'WADDY THAI FOOD', 'NO 2D,
JALAN MEDAN BAYAN, RELAU, 11900 BAYAN LEPAS, PULAU PINANG',
1130, 2130);
INSERT INTO merchant VALUES ('M0006', 'Zus Coffee - Sunway Pinnacle',
'Unit G-02, Ground Floor, The Pinnacle Annexe, Persiaran Lagoon, Sunway
City, 47500, Selangor', 0900, 1730);
INSERT INTO merchant VALUES ('M0007', 'KU ANIS CAKE AND BAKERY',
'LOT 1823 KAMPUNG WAKAF MEK ZAINAB, DAERAH PANJI MUKIM TG
CHAT, 15300 KOTA BHARU KELANTAN.', 1100, 2100);
INSERT INTO merchant VALUES ('M0008', 'Warung Hanna Cyberjaya', 'G16
Paragon @ Pangaea, Persiaran Bestari 1, Cyber 11', 0815, 2345);
INSERT INTO merchant VALUES ('M0009', 'Isaac Toast Heuk Hwa Dang
Taylors', 'G-16, Syopz Mall, Jalan Taylor', 0815, 1745);
INSERT INTO merchant VALUES ('M0010', 'Feng Cha Klang', '36 T/ Bawah
Lorong Batu Nilam 3A, Bandar Bukit Tinggi', 1100, 1900);
```

4.4 Table Name: voucher

```
INSERT INTO voucher VALUES ('CUBAPANDA', 10, 'Currency',
'16-FEB-2021', '23-MAR-2021', 20, 10, 'New Customers');
INSERT INTO voucher VALUES ('HUATSUPCNY', 9, 'Currency',
'21-JAN-2021', '14-MAR-2021', 39, 9, 'Existing Customers');
INSERT INTO voucher VALUES ('TASTEOFJOY', 36, 'Percentage',
'25-FEB-2021', '06-MAR-2021', 18, 5, 'Existing Customers');
INSERT INTO voucher VALUES ('ONLYONPANDA', 7, 'Currency',
'18-FEB-2021', '22-MAR-2021', 30, 7, 'Existing Customers');
INSERT INTO voucher VALUES ('LUCKYEATS', 8, 'Currency', '14-JAN-2021',
'19-MAR-2021', 26, 8, 'Existing Customers');
```

4.5 Table Name: payment

```
INSERT INTO payment VALUES ('P0001', NULL, 'Cash', NULL, 'Paid');
INSERT INTO payment VALUES ('P0002', 'HUATSUPCNY', 'Online Banking',
NULL, 'Paid');
INSERT INTO payment VALUES ('P0003', NULL, 'Credit or Debit Card',
4532972414825033, 'Paid');
INSERT INTO payment VALUES ('P0004', NULL, 'Cash', NULL, 'Paid');
INSERT INTO payment VALUES ('P0005', 'ONLYONPANDA', 'Online
Banking', NULL, 'Paid');
INSERT INTO payment VALUES ('P0006', NULL, 'Online Banking', NULL,
'Paid');
INSERT INTO payment VALUES ('P0007', 'ONLYONPANDA', 'Online
Banking', NULL, 'Paid');
INSERT INTO payment VALUES ('P0008', NULL, 'Credit or Debit Card',
4532972414825033, 'Paid');
INSERT INTO payment VALUES ('P0009', NULL, 'Credit or Debit Card',
5479188719084505, 'Paid');
INSERT INTO payment VALUES ('P0010', NULL, 'Online Banking', NULL,
'Paid');
```

4.6 Table Name: merchant_product

```
INSERT INTO merchant_product VALUES ('M000101', 'M0001', 'Nasi Lalap
Set', 12.50);
INSERT INTO merchant_product VALUES ('M000102', 'M0001', 'Ayam
Penyet Set', 12.50);
INSERT INTO merchant_product VALUES ('M000103', 'M0001', 'Sup Ayam',
8.00);
INSERT INTO merchant_product VALUES ('M000104', 'M0001', 'Teh Ais',
2.00);
INSERT INTO merchant_product VALUES ('M000105', 'M0001', 'ABC Tabur
Milo', 6.50);
INSERT INTO merchant_product VALUES ('M000201', 'M0002', '3Q Milk
Tea', 5.90);
INSERT INTO merchant_product VALUES ('M000202', 'M0002', 'Black Tea
with Cheese Foam', 8.90);
INSERT INTO merchant_product VALUES ('M000301', 'M0003', 'Sour and
Spicy Soup', 11.00);
INSERT INTO merchant_product VALUES ('M000302', 'M0003', 'Ginger
Onion Chicken', 12.00);
INSERT INTO merchant_product VALUES ('M000303', 'M0003', 'Nestum
Prawn', 23.00);
INSERT INTO merchant_product VALUES ('M000401', 'M0004', 'Fried Radish
Cake', 7.00);
```

```

INSERT INTO merchant_product VALUES ('M000501', 'M0005', 'Thai Laksa',
12.90);
INSERT INTO merchant_product VALUES ('M000502', 'M0005', 'Phat Phet
Chicken Rice Set', 13.90);
INSERT INTO merchant_product VALUES ('M000601', 'M0006', 'Cafe Latte',
8.90);
INSERT INTO merchant_product VALUES ('M000701', 'M0007', 'Chocolate
Moist Cake', 12.00);
INSERT INTO merchant_product VALUES ('M000702', 'M0007', 'Red Velvet
Cake', 15.00);
INSERT INTO merchant_product VALUES ('M000801', 'M0008', 'Nasi Lemak
Ayam Crispy', 7.50);
INSERT INTO merchant_product VALUES ('M000901', 'M0009', 'Turkey Slice
Cheese Special', 9.00);
INSERT INTO merchant_product VALUES ('M001001', 'M0010', 'Black Sugar
Mousse', 9.90);
INSERT INTO merchant_product VALUES ('M001002', 'M0010', 'Mango
Yogurt', 14.90);

```

4.7 Table Name: customer_order

```

INSERT INTO customer_order VALUES ('O0001', 'C0005', 'P0001',
'12-JAN-2021 15:34');
INSERT INTO customer_order VALUES ('O0002', 'C0003', 'P0002',
'26-JAN-2021 20:03');
INSERT INTO customer_order VALUES ('O0003', 'C0004', 'P0003',
'01-FEB-2021 09:15');
INSERT INTO customer_order VALUES ('O0004', 'C0006', 'P0004',
'03-FEB-2021 10:26');
INSERT INTO customer_order VALUES ('O0005', 'C0002', 'P0005',
'10-FEB-2021 12:45');
INSERT INTO customer_order VALUES ('O0006', 'C0003', 'P0006',
'21-FEB-2021 08:43');
INSERT INTO customer_order VALUES ('O0007', 'C0002', 'P0007',
'15-MAR-2021 19:23');
INSERT INTO customer_order VALUES ('O0008', 'C0004', 'P0008',
'04-APR-2021 10:03');
INSERT INTO customer_order VALUES ('O0009', 'C0001', 'P0009',
'05-APR-2021 16:52');
INSERT INTO customer_order VALUES ('O0010', 'C0002', 'P0010',
'29-APR-2021 17:24');

```

4.8 Table Name: order_details

```
INSERT INTO order_details VALUES ('O000101', 'M000501', 'O0001', 1);
INSERT INTO order_details VALUES ('O000102', 'M000502', 'O0001', 1);
INSERT INTO order_details VALUES ('O000201', 'M000301', 'O0002', 1);
INSERT INTO order_details VALUES ('O000202', 'M000303', 'O0002', 1);
INSERT INTO order_details VALUES ('O000203', 'M000302', 'O0002', 1);
INSERT INTO order_details VALUES ('O000301', 'M000601', 'O0003', 2);
INSERT INTO order_details VALUES ('O000401', 'M000901', 'O0004', 2);
INSERT INTO order_details VALUES ('O000501', 'M000104', 'O0005', 2);
INSERT INTO order_details VALUES ('O000502', 'M000101', 'O0005', 2);
INSERT INTO order_details VALUES ('O000503', 'M000105', 'O0005', 1);
INSERT INTO order_details VALUES ('O000601', 'M000401', 'O0006', 3);
INSERT INTO order_details VALUES ('O000701', 'M000102', 'O0007', 1);
INSERT INTO order_details VALUES ('O000702', 'M000103', 'O0007', 1);
INSERT INTO order_details VALUES ('O000703', 'M000101', 'O0007', 1);
INSERT INTO order_details VALUES ('O000704', 'M000105', 'O0007', 2);
INSERT INTO order_details VALUES ('O000801', 'M000601', 'O0008', 2);
INSERT INTO order_details VALUES ('O000901', 'M000702', 'O0009', 2);
INSERT INTO order_details VALUES ('O000902', 'M000701', 'O0009', 2);
INSERT INTO order_details VALUES ('O001001', 'M001001', 'O0010', 1);
INSERT INTO order_details VALUES ('O001002', 'M001002', 'O0010', 2);
```

4.9 Table Name: Delivery

```
INSERT INTO delivery VALUES ('D0001', 'O0001', 'R0003', '12-JAN-2021
15:57');
INSERT INTO delivery VALUES ('D0002', 'O0002', 'R0002', '26-JAN-2021
20:21');
INSERT INTO delivery VALUES ('D0003', 'O0003', 'R0004', '01-FEB-2021
09:33');
INSERT INTO delivery VALUES ('D0004', 'O0004', 'R0010', '03-FEB-2021
10:55');
INSERT INTO delivery VALUES ('D0005', 'O0005', 'R0008', '10-FEB-2021
13:05');
INSERT INTO delivery VALUES ('D0006', 'O0006', 'R0005', '21-FEB-2021
08:58');
INSERT INTO delivery VALUES ('D0007', 'O0007', 'R0008', '15-MAR-2021
19:48');
INSERT INTO delivery VALUES ('D0008', 'O0008', 'R0004', '04-APR-2021
10:23');
INSERT INTO delivery VALUES ('D0009', 'O0009', 'R0001', '05-APR-2021
17:10');
INSERT INTO delivery VALUES ('D0010', 'O0010', 'R0007', '29-APR-2021
17:52');
```


5. Data Manipulation with SQL

5.1 AGGREGATE FUNCTION (COUNT, MAX, MIN, AVG, SUM)

5.1.1 Get the ID of active customers and number of purchase (COUNT)

```
SELECT cus_id, COUNT(cus_id) AS number_of_purchase
FROM customer_order
GROUP BY cus_id
HAVING COUNT(*) >= 1
ORDER BY cus_id;
```

CUS_ID	NUMBER_OF_PURCHASE
C0001	1
C0002	3
C0003	2
C0004	2
C0005	1
C0006	1

5.1.2 Get the product with highest price among all merchants (MAX)

```
SELECT merch_name, product_name, product_price
FROM merchant_product, merchant
WHERE merchant_product.merch_id = merchant.merch_id
      AND product_price = (SELECT MAX(product_price)
                          FROM merchant_product);
```

MERCH_NAME	PRODUCT_NAME	PRODUCT_PRICE
N Food Stall @ 138 Cafe Nestum Prawn		23

5.1.3 Get the voucher with the smallest minimum spend (MIN)

```
SELECT promo_code, min_spend
FROM voucher
WHERE min_spend = (SELECT MIN(min_spend)
                  FROM voucher);
```

PROMO_CODE	MIN_SPEND
TASTE OF JOY	18

5.1.4 Get the average delivery time (AVG)

```
SELECT AVG(EXTRACT(MINUTE FROM(delivery_datetime -  
order_datetime))) average_delivery_time_in_minute  
FROM customer_order, delivery  
WHERE customer_order.order_id = delivery.order_id;
```

AVERAGE_DELIVERY_TIME_IN_MINUTE
21.4

5.1.5 Get the total number of items ordered for each order (SUM)

```
SELECT order_id, SUM(quantity) AS number_of_items  
FROM order_details  
GROUP BY order_id  
ORDER BY order_id;
```

ORDER_ID	NUMBER_OF_ITEMS
O0001	2
O0002	3
O0003	2
O0004	2
O0005	5
O0006	3
O0007	5
O0008	2
O0009	4
O0010	3

5.2 GROUP BY AND HAVING CLAUSES

5.2.1 Get the amount of product purchased that have prices lower than 10

```
SELECT merchant_product.product_name, COUNT(order_details.order_id),  
merchant_product.product_id, merchant_product.product_price  
FROM (order_details INNER JOIN merchant_product ON  
order_details.product_id = merchant_product.product_id)  
GROUP BY product_name, merchant_product.product_name,  
merchant_product.product_id, merchant_product.product_price  
HAVING product_price < 10  
ORDER BY merchant_product.product_name ASC;
```

PRODUCT_NAME	COUNT(ORDER_DETAILS.ORDER_ID)	PRODUCT_ID	PRODUCT_PRICE
¹ ABC Tabur Milo	2	M000105	6.5
² Black Sugar Mousse	1	M001001	9.9
³ Cafe Latte	2	M000601	8.9
⁴ Fried Radish Cake	1	M000401	7
⁵ Sup Ayam	1	M000103	8
⁶ Teh Ais	1	M000104	2
⁷ Turkey Slice Cheese Special	1	M000901	9

5.3 TRIGGERS

5.3.1 Create a trigger to apply delivery fee on order after March 2021

```
ALTER TABLE payment ADD delivery_fee NUMBER(1);
```

```
CREATE TRIGGER trg_delivery_fee  
AFTER INSERT ON customer_order  
BEGIN
```

```
    UPDATE payment  
    SET delivery_fee = 0;
```

```
    FOR datetime IN (SELECT payment_num, order_datetime  
                     FROM customer_order  
                     WHERE order_datetime > '01-MAR-2021')
```

```
    LOOP
```

```
        UPDATE payment  
        SET delivery_fee = 3  
        WHERE payment_num = datetime.payment_num;
```

```
    END LOOP;
```

```
END;
```

```
INSERT INTO payment VALUES ('P0011', NULL, 'Cash', NULL, 'Paid',  
NULL);
```

```
INSERT INTO customer_order VALUES ('O0011', 'C0002', 'P0011', sysdate);
```

```
SELECT payment.payment_num, TO_CHAR(order_datetime,  
'DD/MM/YYYY') order_datetime, delivery_fee  
FROM payment, customer_order  
WHERE payment.payment_num = customer_order.payment_num;
```

Before trigger:

PAYMENT_NUM	ORDER_DATETIME	DELIVERY_FEE
P0001	12/01/2021	(null)
P0002	26/01/2021	(null)
P0003	01/02/2021	(null)
P0004	03/02/2021	(null)
P0005	10/02/2021	(null)
P0006	21/02/2021	(null)
P0007	15/03/2021	(null)
P0008	04/04/2021	(null)
P0009	05/04/2021	(null)
P0010	29/04/2021	(null)

After trigger:

PAYMENT_NUM	ORDER_DATETIME	DELIVERY_FEE
P0001	12/01/2021	0
P0002	26/01/2021	0
P0003	01/02/2021	0
P0004	03/02/2021	0
P0005	10/02/2021	0
P0006	21/02/2021	0
P0007	15/03/2021	3
P0008	04/04/2021	3
P0009	05/04/2021	3
P0010	29/04/2021	3
P0011	09/02/2021	0

5.3.2 Merchants are not allowed to extend closing time but allowed to shorten closing time

```
CREATE VIEW virtual_merchant AS
SELECT merch_id, merch_name, merch_open, merch_close
FROM merchant;
```

```
CREATE TRIGGER trg_closing_time
INSTEAD OF UPDATE ON virtual_merchant
FOR EACH ROW
BEGIN
    IF :NEW.merch_close > :OLD.merch_close THEN
        UPDATE merchant
        SET merch_close = :OLD.merch_close
        WHERE merch_id = :NEW.merch_id;
    ELSE
        UPDATE merchant
        SET merch_close = :NEW.merch_close
        WHERE merch_id = :NEW.merch_id;
    END IF;
END;
```

```
UPDATE virtual_merchant
SET merch_close = 2200
WHERE merch_id = 'M0001';
```

```
UPDATE virtual_merchant
SET merch_close = 2000
WHERE merch_id = 'M0003';
```

```
SELECT merch_id, merch_name, merch_open, merch_close
FROM merchant;
```

Before trigger:

MERCH_ID	MERCH_NAME	MERCH_OPEN	MERCH_CLOSE
M0001	Nasi Lalap Afieyah	1000	2000
M0002	HODI	1115	2045
M0003	N Food Stall @ 138 Cafe	1730	2130
M0004	Lung Meng Fried Radish Cake	800	1130
M0005	WADDY THAI FOOD	1130	2130
M0006	Zus Coffee - Sunway Pinnacle	900	1730
M0007	KU ANIS CAKE AND BAKERY	1100	2100
M0008	Warung Hanna Cyberjaya	815	2345
M0009	Isaac Toast Heuk Hwa Dang Taylors	815	1745
M0010	Feng Cha Klang	1100	1900

After trigger (change occurs on Row 3):

MERCH_ID	MERCH_NAME	MERCH_OPEN	MERCH_CLOSE
M0001	Nasi Lalap Afieyah	1000	2000
M0002	HODI	1115	2045
M0003	N Food Stall @ 138 Cafe	1730	2000
M0004	Lung Meng Fried Radish Cake	800	1130
M0005	WADDY THAI FOOD	1130	2130
M0006	Zus Coffee - Sunway Pinnacle	900	1730
M0007	KU ANIS CAKE AND BAKERY	1100	2100
M0008	Warung Hanna Cyberjaya	815	2345
M0009	Isaac Toast Heuk Hwa Dang Taylors	815	1745
M0010	Feng Cha Klang	1100	1900

5.4 STORED PROCEDURE

5.4.1 Users can select merchants to increase product price by 3% or 5% according to original price.

```
CREATE PROCEDURE increase_price (merchant_in IN VARCHAR)
AS
BEGIN
    UPDATE merchant_product
    SET product_price = CASE
        WHEN (product_price <= 10 AND merch_id = merchant_in)
            THEN ROUND((product_price*0.05) + product_price, 1)
        WHEN (product_price > 10 AND merch_id = merchant_in)
            THEN ROUND((product_price*0.03) + product_price, 1)
        WHEN (merch_id != merchant_in)
            THEN product_price
    END;
END;

EXEC increase_price('M0001')

SELECT * FROM merchant_product
```

Before executing procedure:

PRODUCT_ID	MERCH_ID	PRODUCT_NAME	PRODUCT_PRICE
M000101	M0001	Nasi Lalap Set	12.5
M000102	M0001	Ayam Penyet Set	12.5
M000103	M0001	Sup Ayam	8
M000104	M0001	Teh Ais	2
M000105	M0001	ABC Tabur Milo	6.5
M000201	M0002	3Q Milk Tea	5.9
M000202	M0002	Black Tea with Cheese Foam	8.9
M000301	M0003	Sour and Spicy Soup	11
M000302	M0003	Ginger Onion Chicken	12
M000303	M0003	Nestum Prawn	23
M000401	M0004	Fried Radish Cake	7
M000501	M0005	Thai Laksa	12.9
M000502	M0005	Phat Phet Chicken Rice Set	13.9
M000601	M0006	Cafe Latte	8.9
M000701	M0007	Chocolate Moist Cake	12
M000702	M0007	Red Velvet Cake	15
M000801	M0008	Nasi Lemak Ayam Crispy	7.5
M000901	M0009	Turkey Slice Cheese Special	9
M001001	M0010	Black Sugar Mousse	9.9
M001002	M0010	Mango Yogurt	14.9

After executing procedure (changes occur on Row 1-5):

PRODUCT_ID	MERCH_ID	PRODUCT_NAME	PRODUCT_PRICE
M000101	M0001	Nasi Lalap Set	12.9
M000102	M0001	Ayam Penyet Set	12.9
M000103	M0001	Sup Ayam	8.4
M000104	M0001	Teh Ais	2.1
M000105	M0001	ABC Tabur Milo	6.8
M000201	M0002	3Q Milk Tea	5.9
M000202	M0002	Black Tea with Cheese Foam	8.9
M000301	M0003	Sour and Spicy Soup	11
M000302	M0003	Ginger Onion Chicken	12
M000303	M0003	Nestum Prawn	23
M000401	M0004	Fried Radish Cake	7
M000501	M0005	Thai Laksa	12.9
M000502	M0005	Phat Phet Chicken Rice Set	13.9
M000601	M0006	Cafe Latte	8.9
M000701	M0007	Chocolate Moist Cake	12
M000702	M0007	Red Velvet Cake	15
M000801	M0008	Nasi Lemak Ayam Crispy	7.5
M000901	M0009	Turkey Slice Cheese Special	9
M001001	M0010	Black Sugar Mousse	9.9
M001002	M0010	Mango Yogurt	14.9

5.5 VIEW

5.5.1 Create a view for products having price larger than 10

```
CREATE VIEW product_price_10 AS
SELECT product_id, merch_id, product_name, product_price
FROM merchant_product
WHERE product_price > 10

SELECT * FROM product_price_10;
```

	PRODUCT_ID	MERCH_ID	PRODUCT_NAME	PRODUCT_PRICE
1	M000101	M0001	Nasi Lalap Set	12.5
2	M000102	M0001	Ayam Penyet Set	12.5
3	M000301	M0003	Sour and Spicy Soup	11
4	M000302	M0003	Ginger Onion Chicken	12
5	M000303	M0003	Nestum Prawn	23
6	M000501	M0005	Thai Laksa	12.9
7	M000502	M0005	Phat Phet Chicken Rice Set	13.9
8	M000701	M0007	Chocolate Moist Cake	12
9	M000702	M0007	Red Velvet Cake	15
10	M001002	M0010	Mango Yogurt	14.9

5.5.2 Create a view for merchants having opening time at or after 11am

```
CREATE VIEW merchant_open_after_11am AS
SELECT merch_id, merch_name, merch_open, merch_close
FROM merchant
WHERE merch_open > '1059'

SELECT * FROM merchant_open_after_11am;
```

	MERCH_ID	MERCH_NAME	MERCH_OPEN	MERCH_CLOSE
1	M0002	HODI	1115	2045
2	M0003	N Food Stall @ 138 Cafe	1730	2130
3	M0005	WADDY THAI FOOD	1130	2130
4	M0007	KU ANIS CAKE AND BAKERY	1100	2100
5	M0010	Feng Cha Klang	1100	1900

5.6 USE OF SUBQUERIES/NESTED QUERIES

5.6.1 Get the subtotal for each order

```
SELECT order_id, SUM(quantity * (SELECT product_price
                                FROM merchant_product
                                WHERE merchant_product.product_id =
                                order_details.product_id)) AS subtotal
FROM order_details
GROUP BY order_id
ORDER BY order_id;
```

ORDER_ID	SUBTOTAL
O0001	26.8
O0002	46
O0003	17.8
O0004	18
O0005	35.5
O0006	21
O0007	46
O0008	17.8
O0009	54
O0010	39.7

5.6.2 Get the rider who have delivered equal or more than 2 orders

```
SELECT rider_id, rider_fname, rider_lname
FROM rider
WHERE rider_id IN (SELECT rider_id
                  FROM delivery
                  GROUP BY rider_id
                  HAVING COUNT(rider_id) >= 2);
```

RIDER_ID	RIDER_FNAME	RIDER_LNAME
R0004	Muhamad	Yaqzan Azni
R0008	Sukhjit	Theseira

5.7 QUERIES NOT COVERED IN LECTURE

5.7.1 Top 5 lowest product price among all merchants (Using ROWNUM)

```
SELECT ROWNUM AS rank, product_id, merch_id, product_name,  
product_price  
FROM (SELECT product_price, product_id, merch_id, product_name  
      FROM merchant_product  
      ORDER BY product_price)  
WHERE ROWNUM <= 5
```

	RANK	PRODUCT_ID	MERCH_ID	PRODUCT_NAME	PRODUCT_PRICE
1	1	M000104	M0001	Teh Ais	2
2	2	M000201	M0002	3Q Milk Tea	5.9
3	3	M000105	M0001	ABC Tabur Milo	6.5
4	4	M000401	M0004	Fried Radish Cake	7
5	5	M000801	M0008	Nasi Lemak Ayam Crispy	7.5

5.7.2 Generate VIP number with auto increment by 1 for customers who have more than 1 order history (Using CREATE SEQUENCE)

```
ALTER TABLE customer ADD vip NUMBER(2);
```

```
CREATE SEQUENCE auto_increment  
MINVALUE 1  
START WITH 1  
INCREMENT BY 1  
CACHE 20;
```

```
UPDATE customer  
SET vip = auto_increment.nextval  
WHERE cus_id IN (SELECT cus_id  
                 FROM customer_order  
                 GROUP BY cus_id  
                 HAVING COUNT(cus_id) > 1) AND  
vip IS NULL;
```

```
SELECT cus_id, cus_fname, cus_lname, vip  
FROM customer;
```

⚡ CUS_ID	⚡ CUS_FNAME	⚡ CUS_LNAME	⚡ VIP
C0001	Lee	Zi Sheng	(null)
C0002	Mahsuri	Nawar	1
C0003	Tan	Jessica	2
C0004	Wong	Jun Jie	3
C0005	Kong	Xin Yen	(null)
C0006	Mark	Lee	(null)
C0007	Maya	Izzati	(null)

5.7.3 Get the lowest price product for each merchant (Using RANK)

```

SELECT product_id, merch_id, product_name, product_price
FROM (SELECT product_id, merch_id, product_name, product_price,
      RANK() OVER (PARTITION BY merch_id
                   ORDER BY product_price) AS rank
      FROM merchant_product)
WHERE RANK = 1;

```

⚡	⚡ PRODUCT_ID	⚡ MERCH_ID	⚡ PRODUCT_NAME	⚡ PRODUCT_PRICE
1	M000104	M0001	Teh Ais	2
2	M000201	M0002	3Q Milk Tea	5.9
3	M000301	M0003	Sour and Spicy Soup	11
4	M000401	M0004	Fried Radish Cake	7
5	M000501	M0005	Thai Laksa	12.9
6	M000601	M0006	Cafe Latte	8.9
7	M000701	M0007	Chocolate Moist Cake	12
8	M000801	M0008	Nasi Lemak Ayam Crispy	7.5
9	M000901	M0009	Turkey Slice Cheese Special	9
10	M001001	M0010	Black Sugar Mousse	9.9

5.7.4 Get February 2021 order history (Using EXTRACT)

```
SELECT order_id, TO_CHAR(order_datetime, 'DD/MM/YYYY HH24:MI')  
order_datetime  
FROM customer_order  
WHERE (EXTRACT (MONTH FROM order_datetime)) = 2 AND  
      (EXTRACT (YEAR FROM order_datetime)) = 2021;
```

ORDER_ID	ORDER_DATETIME
00003	01/02/2021 09:15
00004	03/02/2021 10:26
00005	10/02/2021 12:45
00006	21/02/2021 08:43

6. Contribution

- | | |
|------------------|-------------------|
| 1. Lim Wei Jie | 1181103501 : 100% |
| 2. Tee Ming Yuan | 1181103353 : 100% |
| 3. Loo Chen Zhi | 1181103230 : 100% |
| 4. Chang See Jie | 1181103362 : 100% |