



**UTM**  
UNIVERSITI TEKNOLOGI MALAYSIA

**SCHOOL OF COMPUTING**  
**SESSION 2023/2024, SEMESTER 1**  
**SECV2113**  
**DATABASE**  
**SECTION 10**

**PHASE 3: DATABASE LOGICAL DESIGN & SQL**  
**GROUP: DATA NINJAS**

**LECTURER'S NAME: DR ROZILAWATI**

<b>NO.</b>	<b>NAME</b>	<b>MATRICS NO</b>
<b>1.</b>	<b>SHUHD NASHWAN SAEED SHARAF</b>	<b>A22EC4014</b>
<b>2.</b>	<b>NURUL IZZAH BINTI RASHID</b>	<b>A22EC0256</b>
<b>3.</b>	<b>LYDIA AZRA BINTI MUHAMMAD NOR</b>	<b>B23CS0042</b>

## TABLE OF CONTENT

NO.	TITLE	PAGE NUMBER
1.0	Introduction	2
2.0	Overview of Project	3-4
3.0	Database Conceptual Design	4-5
	3.1 Updated Business Rule	4
	3.2 Conceptual ERD	5
4.0	Database Logical Design	6
	4.1 Logical ERD	7
	4.2 Updated Data Dictionary	8-10
	4.3 Normalization	11-17
5.0	Relational DB Schemas (after normalization)	18-19
6.0	SQL Statements (DDL & DML)	20-34
7.0	Summary	35

## **1.0 Introduction**

A lot of UTM (Universiti Teknologi Malaysia) students engage in small businesses on campus, primarily for financial gain, project assignments or to occupy their leisure time. However, they encounter a significant challenge in terms of lacking a platform for promoting their businesses. What they already do is promote their businesses on WhatsApp or Telegram groups, requiring them to join multiple groups for broader outreach, which, in turn, consumes a considerable amount of time, effort, and mobile space.

To overcome these challenges, we are going to build a platform for small business owners around UTM. All small business owners can use this platform solely to promote their businesses. This platform will encompass a wide range of features and functionalities, including e-commerce capabilities, marketing and customer engagement, and analytics and insights. People around the UTM will be able to participate in the platform to get to know what businesses are there and buy their products. This will make it easier for those students who do not have any transportation to go outside the university to buy the items needed.

This project is a collaborative effort that involves a dedicated team of developers, system analysts, and small business owners. We are committed to delivering a solution that is not only technologically advanced but also responsive to the real-world challenges faced by small business owners.

Through this platform, we are all about inspiring creativity and giving a helping hand to small business owners in UTM. We're dedicated to bringing a user-friendly platform that perfectly fits the needs of small business owners in the area.

## 2.0 Overview of Project

A significant number of Universiti Teknologi Malaysia (UTM) students are actively involved in small businesses on campus, driven by financial motives, academic projects, or leisure pursuits. However, these budding entrepreneurs face a considerable challenge in promoting their businesses effectively. The prevalent method involves utilizing messaging platforms like WhatsApp or Telegram groups, necessitating participation in multiple groups for wider outreach. This approach proves time-consuming, demanding substantial effort and consuming valuable mobile space.

### Project Objective:

To address these challenges, the UTM Small Business project aims to develop a dedicated platform for small business owners within the UTM community. This platform will serve as a centralized hub, offering an array of features to streamline business promotion, enhance e-commerce capabilities, facilitate marketing and customer engagement, and provide valuable analytics and insights.

### Key Features:

**E-commerce Capabilities:** The platform will enable business owners to showcase and sell their products or services online, providing a convenient channel for transactions.

**Marketing and Customer Engagement:** Robust marketing tools will be incorporated, allowing businesses to effectively reach their target audience. Customer engagement features such as notifications and updates will enhance interaction.

**Analytics and Insights:** Business owners can leverage data analytics to gain insights into their performance, customer behavior, and market trends, aiding in strategic decision-making.

**User Participation:** The UTM community, including students and faculty, can actively participate in the platform, discovering local businesses, exploring products, and making purchases. This promotes a thriving local economy and eliminates the need for transportation beyond the university.

#### Mission and Values:

The UTM Business Connect project is driven by a mission to inspire creativity and provide crucial support to small business owners at UTM. The focus is on delivering a user-friendly platform tailored to the specific needs of the local entrepreneurial community. By doing so, the project aims to foster economic growth within the university while simplifying the business promotion process for students.

In conclusion, UTM Business Connect is poised to make a significant impact by creating a thriving ecosystem for small businesses within the UTM community. The platform's comprehensive features and collaborative approach underscore its commitment to empowering student entrepreneurs and promoting a vibrant local business environment.

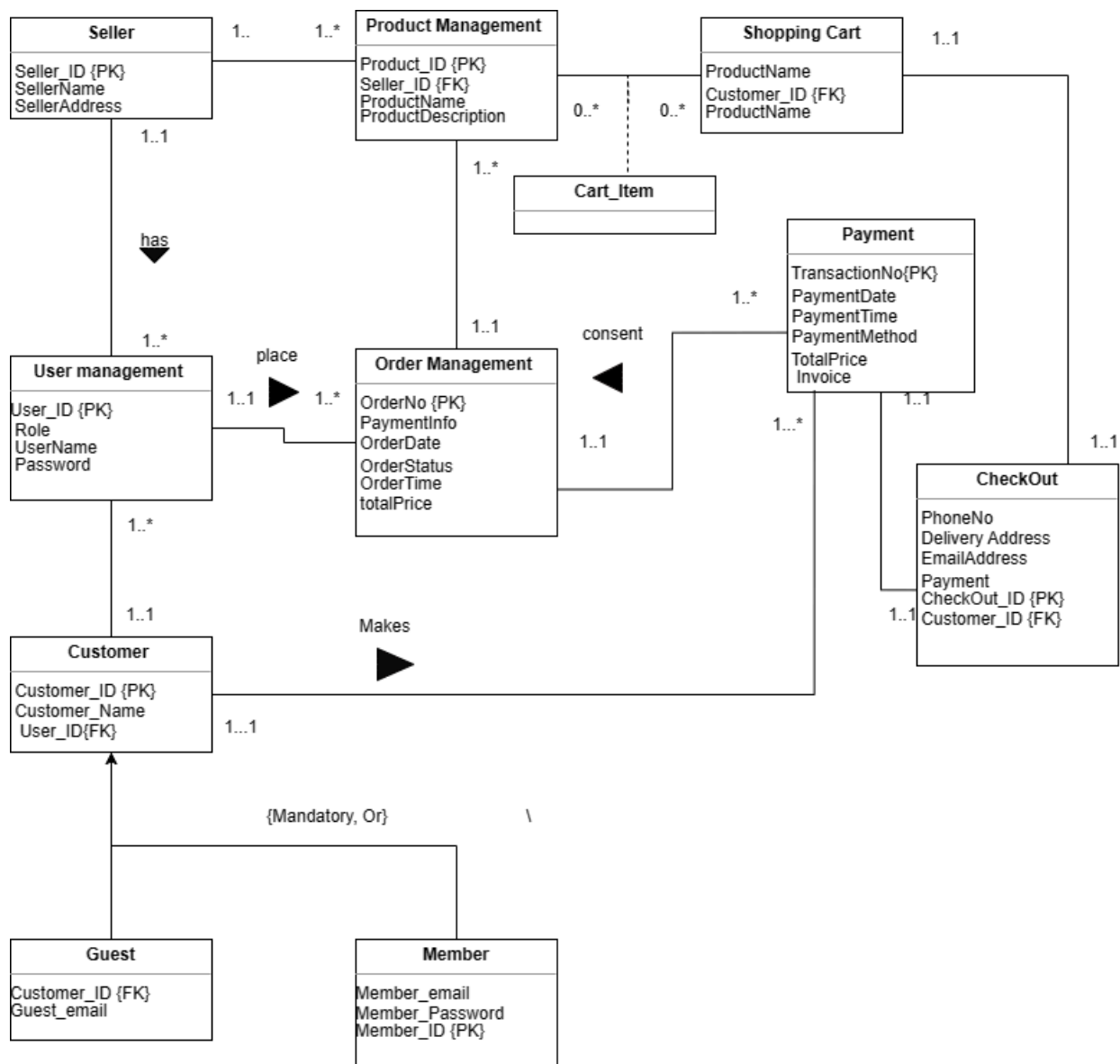
### **3.0 Database Conceptual Design**

#### **3.1 Updated Business Rule**

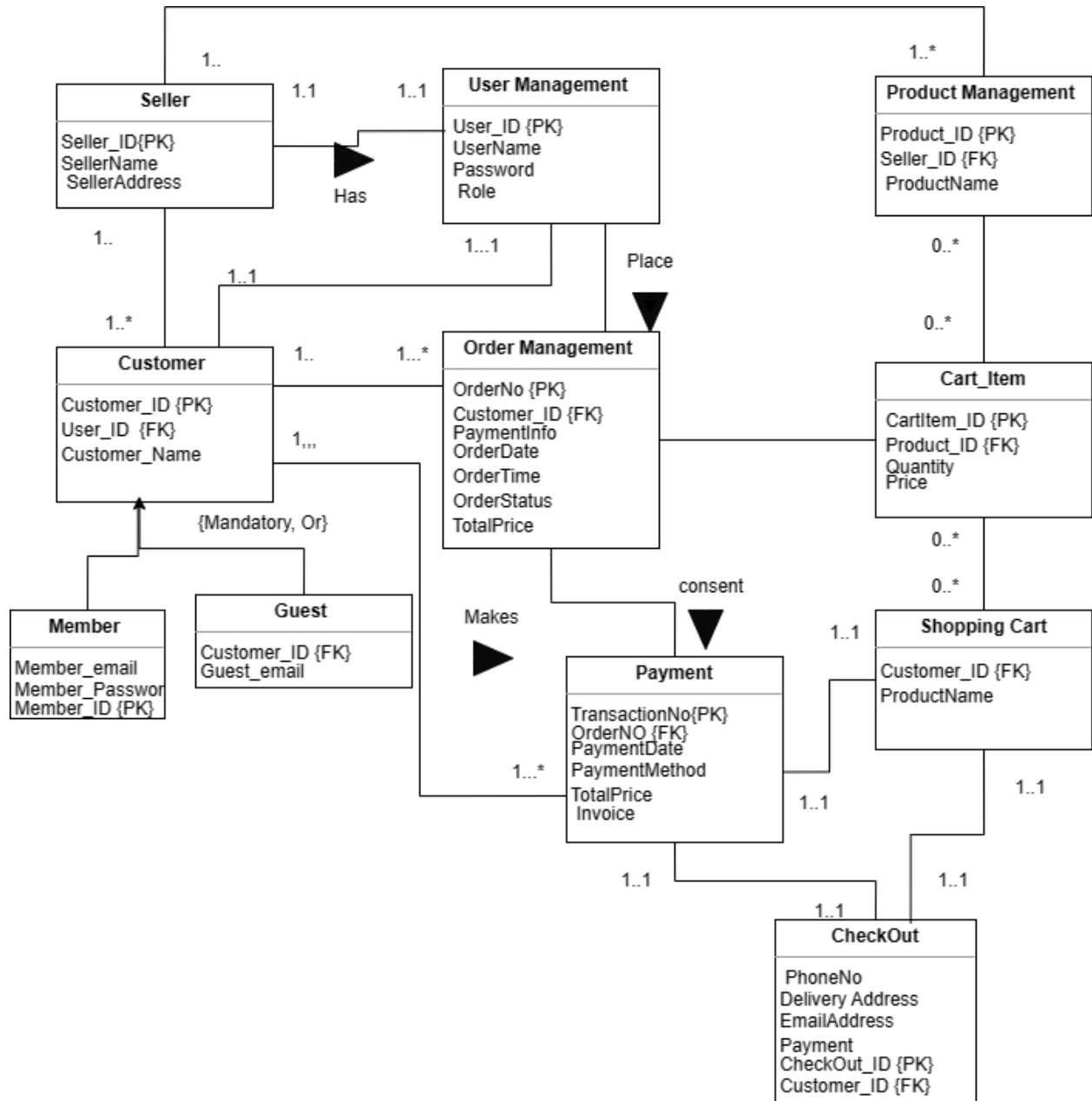
UTM Small Business system:

1. The owner of a small business registers and fills out their profile.
2. A small business owner updates their listing with new goods or services.
3. Users peruse offerings and put goods in their shopping carts.
4. When placing an order, customers supply information about payment and address.
5. Order alerts are sent to small business owners, who then get orders ready for delivery.
6. Owners of small businesses deliver orders and give clients access to tracking details.
7. Small business owners can get orders from their customers and receive feedback from them.

### 3.2 Conceptual ERD



## 4.0 Database Logical Design



#### 4.1 Logical ERD

1. Strong Entity Type

Customer (Customer\_ID,user\_ID,Customer\_Name)

PK: Customer\_ID

2. User Management (User\_ID,UserName>Password,Role)

PK: User\_ID

3. Seller (Seller\_ID,sellerName,SphoneNo,sellerAddress)

PK: Seller\_ID



## 4.2 Updated Data Dictionary

Entity Name	Attributes	Description	Data Type & Length	Nulls	Multivalued
User management	User_ID {PK}	Uniquely identifies the user ID	VARCHAR(40)	NO	NO
	UserName	Name of the user.	VARCHAR(40)	NO	NO
	Password	Password of the user.	VARCHAR(15)	NO	NO
	Role	Role of user	VARCHAR(15)	NO	NO
Seller	Seller_ID	Seller ID	VARCHAR(40)	NO	NO
	SellerName	Seller name.	VARCHAR(40)	NO	NO
	SellerAddress	Seller's address.	VARCHAR(40)	NO	NO
Product Management	Product_ID	Holds product names.	VARCHAR(40)	NO	NO
	ProductName	Name of the product	VARCHAR(40)	NO	NO
	Price	Price of the product	VARCHAR(10)	NO	NO
Order Management	OrderNo {PK}	Uniquely identifies the order number.	VARCHAR(15)	NO	NO
	PaymentInfo	Detail of the payment.	VARCHAR(15)	NO	NO
	OrderDate	Date of the order.	Date	NO	NO
	OrderStatus	Status of the order.	VARCHAR(15)	NO	NO
	OrderTime	Time of the order	Time	NO	NO

	totalPrice	Total price of the ordered item.	VARCHAR(10)	NO	NO
Shopping Cart	Customer_ID	Customer ID	INTEGER	NO	NO
	ProductName	Name of the product.	VARCHAR(40)	NO	YES
Payment	TransactionNO {PK}	Uniquely identifies the transaction number.	INTEGER	NO	NO
	PaymentDate	Date of the payment.	Date	YES	NO
	PaymentTime	Time of the payment	Time	NO	NO
	PaymentMethod	Method of payment.	VARCHAR(10)	NO	NO
	Invoice	written description for the payment	VARCHAR(10)	NO	NO
	TotalPrice	Total price of the ordered item.	INTEGER(6)	NO	NO
Check Out	Checkout_ID	Checkout ID	VARCHAR(40)	NO	NO
	Customer_ID	Customer ID	VARCHAR(40)	NO	NO
	Payment	Status of the payment.	VARCHAR(15)	NO	YES
	EmailAddress	Email of customer.	VARCHAR(40)	NO	NO
	DeliveryAddress	Address of the user.	VARCHAR(40)	NO	YES
	PhoneNO	Phone number.	INTEGER	NO	NO
Customer	Customer_ID	An identification	INTEGER(10)	NO	NO

		for the customer			
	User_ID	User ID	VARCHAR(40)	NO	NO
	Customer_Name	Personal call name for the customer	VARCHAR(40)	NO	NO
Guest	Customer_ID	Customer ID	VARCHAR(40)	NO	NO
	Guest_Email	An active email address for the guest	VARCHAR(15)	NO	NO
Member	Member_email	An active member email address	VARCHAR(15)	NO	NO
	Member_Password	A password for the member account	VARCHAR(15)	NO	NO
	Member_ID {PK}	A unique Identification for the member	INTEGER(10)	NO	NO

### 4.3 Normalization

1. User Management (User\_ID, UserName, Password, Role)

fd1 User\_ID, UserName, Password, Role

#### 1NF

'User\_ID' is the primary key. The User Management table is already in 1NF because each attribute (User\_ID, UserName, Password, Role) all attributes are atomic (no multivalued attributes) and no repeating groups within a table.

User_ID	UserName	Password	Role
US0001	lydiaazra	password1	customer
US0002	izzah	password2	customer
US0003	shahd	adminpass1	seller
US0004	mashicat	password3	customer
US0005	Admin2	adminpass2	seller
US0006	Alana123	password4	customer
US0007	CharlieBrown	password5	customer
US0008	Admin3	adminpass3	seller
US0009	DavidMiller	password6	customer
US0010	Farhana Azmi	password7	customer
US0011	Admin4	adminpass4	seller
US0012	Hani	password8	customer
US0013	GraceTaylor	password9	customer
US0014	Ahmad	adminpass5	seller
US0015	Amirul	password10	customer

#### 2NF

There should be no partial dependencies on the primary key. User\_ID is the primary key and all other attributes are fully dependent on it, the table is in 2NF.

### 3NF

There should be no transitive dependencies. The User Management table is already in 3NF because there are no transitive dependencies; all non-prime attributes are directly dependent on the primary key (customerID).

## 2. Seller (Seller\_ID, SellerName, SellerAddress)

### 1NF

'Seller\_ID' is the primary key. The Seller table is already in 1NF because each attribute (Seller\_ID, SellerName, SellerAddress) all attributes are atomic (no multivalued attributes) and no repeating groups within a table.

Seller_ID	SellerName	SellerAddress
SL0001	Pena	MA5 KTDI UTM
SL0002	Admin2	Taman Universiti, Johor
SL0003	Kek Sedap	Jalan Ampang, Kuala Lumpur
SL0004	Muhammad	Kolej Tun Razak, UTM, Johor
SL0005	Atikah BookStore	Blok U4 Kolej Perdana
SL0006	Kedai Perhiasan Mutiara	Blok L01, Kolej Tun Hussein Onn
SL0007	Bukuku	Jalan Pahang, Kuala Lumpur
SL0008	Kedai Kita	S10, Kolej Tuanku Canselor, UTM
SL0009	Warung Makan Sedap	Kolej 13, Kolej Datin Seri Endon
SL0010	Pusat Komputer Canggih	Kolej 10, UTM Skudai

### 2NF

There should be no partial dependencies on the primary key. Since Seller\_ID is the primary key and all other attributes are fully dependent on it, the table is in 2NF.

### 3NF

There should be no transitive dependencies. The Seller table is already in 3NF because there are no transitive dependencies; all non-prime attributes are directly dependent on the primary key (Seller\_ID).

### 3. Product Management (Product\_ID, ProductName, Seller\_ID)

#### 1NF

'Product\_ID' is the primary key. The Product Management table is already in 1NF because each attribute (Product\_ID, ProductName, Seller\_ID) all attributes are atomic (no multivalued attributes) and no repeating groups within a table.

Product_ID	ProductName	Seller_ID
PR0001	Notebook Set	SL0001
PR0002	Campus T-Shirt	SL0002
PR0003	Healthy Snack Pack	SL0003
PR0004	Laptop Backpack	SL0004
PR0005	Scientific Calculator	SL0005
PR0006	Desk Organizer	SL0006
PR0007	Coffee Mug Set	SL0007
PR0008	Textbook Bundle	SL0008
PR0009	USB Flash Drive	SL0009
PR0010	Student Cookware Set	SL00010

#### 2NF

There should be no partial dependencies on the primary key. Since Product\_ID is the primary key and all other attributes are fully dependent on it, the table is in 2NF.

#### 3NF

There should be no transitive dependencies. The Product Management is already in 3NF because there are no transitive dependencies; all non-prime attributes are directly dependent on the primary key (Product\_ID)

#### 4. Order Management (OrderNo,PaymentInfo,OrderDate,OrderStatus,totalPrice)

##### 1NF

‘OrderNo’ is the primary key. The Order Management table is already in 1NF because each attribute (OrderNo,PaymentInfo,OrderDate,OrderStatus,totalPrice) all attributes are atomic (no multivalued attributes) and no repeating groups within a table.

OrderNo	PaymentInfo	OrderDate	OrderStatus	totalPrice	Product_ID
ORD0001	Credit Card - 1234	01/22/2024	Received	150	PR0001
ORD0002	PayPal- user@example.com	01/23/2024	Delivered	75.5	PR0002
ORD0003	Debit Card -5678	01/24/2024	Delivered	200	PR0003
ORD0004	Bank Transfer - ABC Bank	01/25/2024	Received	120.75	PR0004
ORD0005	Credit Card - 5678	01/26/2024	Delivered	90	PR0005
ORD0006	PayPal - user2@example.com	01/27/2024	Delivered	180.25	PR0006
ORD0007	Debit Card - 1234	01/28/2024	Delivered	130.5	PR0007
ORD0008	Bank Transfer - XYZ Bank	01/29/2024	Received	95	PR0008
ORD0009	Credit Card - 9876	01/30/2024	Received	210	PR0009
ORD0010	PayPal- user3@example.com	01/31/2024	Delivered	110.75	PR0010

##### 2NF

There should be no partial dependencies on the primary key. Since OrderNo is the primary key and all other attributes are fully dependent on it, the table is in 2NF.

##### 3NF

There should be no transitive dependencies. The Order Management is already in 3NF because there are no transitive dependencies; all non-prime attributes are directly dependent on the primary key (OrderNo)

## 5. Payment

(TransactionNO,PaymentDate,PaymentTime,PaymentMethod,Invoice,TotalPrice)

### 1NF

‘TransactionNO’ is the primary key. The Payment table is already in 1NF because each attribute(TransactionNO,PaymentDate,PaymentTime,PaymentMethod,Invoice,TotalPrice ) all attributes are atomic (no multivalued attributes) and no repeating groups within a table.

TransactionNO	PaymentDate	PaymentTime	PaymentMethod	Invoice	TotalPrice
TR0005	01/26/2024	16:00	Credit Card	INV0005	90
TR0006	01/27/2024	13:10	TnG	INV0006	180.25
TR0008	01/29/2024	08:30	Bank Transfer	INV0008	95
TR0009	01/30/2024	15:20	Credit Card	INV0009	210
TR0003	01/24/2024	11:20	Debit Card	INV0003	200
TR0004	01/25/2024	09:15	Bank Transfer	INV0004	120.75
TR0010	01/31/2024	11:55	TnG	INV0010	110.75
TR0007	01/28/2024	10:45	Debit Card	INV0007	130.5
TR0001	01/22/2024	12:30	Credit Card	INV0001	150
TR0002	01/23/2024	14:45	TnG	INV0002	75.5

### 2NF

There should be no partial dependencies on the primary key. Since TransactionNO is the primary key and all other attributes are fully dependent on it, the table is in 2NF.

### 3NF

There should be no transitive dependencies. The payment table is already in 3NF because there are no transitive dependencies; all non-prime attributes are directly dependent on the primary key (TransactionNO).



6. CheckOut  
(Checkout\_ID, Customer\_ID, Payment, EmailAddress, DeliveryAddress, PhoneNO)

### 1NF

'Checkout\_ID' is the primary key. The Payment table is already in 1NF because each attribute (TransactionNO, PaymentDate, PaymentTime, PaymentMethod, Invoice, TotalPrice) all attributes are atomic (no multivalued attributes) and no repeating groups within a table.

Checkout_ID	Customer_ID	Payment	EmailAddress	DeliveryAddress	PhoneNO	OrderNo
CO0005	CUST0005	TR0005	admin2@example.com	Universiti Teknologi Malaysia, Skudai	60112233445	ORD0005
CO0008	CUST0008	TR0008	aliindra@example.com	Universiti Teknologi Malaysia, Skudai	60199887766	ORD0008
CO0009	CUST0009	TR0009	karina@example.com	Universiti Teknologi Malaysia, Skudai	60112233445	ORD0009
CO0007	CUST0007	TR0007	mahmud189@example.com	Universiti Teknologi Malaysia, Skudai	60112233445	ORD0007
CO0001	CUST0001	TR0001	lydiaazra@example.com	Universiti Teknologi Malaysia, Skudai	60123456789	ORD0001
CO0004	CUST0004	TR0004	maria.johnson@example.com	Universiti Teknologi Malaysia, Skudai	60199887766	ORD0004
CO0006	CUST0006	TR0006	ahmadfisal@example.com	Universiti Teknologi Malaysia, Skudai	60199887766	ORD0006
CO0002	CUST0002	TR0002	izzah@example.com	Universiti Teknologi Malaysia, Skudai	60129876543	ORD0002

CO0003	CUST00 03	TR0003	admin1@example.com	Universiti Teknologi Malaysia, Skudai	601122334 45	ORD0003
--------	--------------	--------	--------------------	---	-----------------	---------

### **2NF**

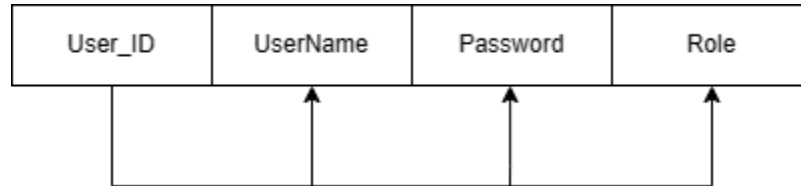
There should be no partial dependencies on the primary key. Since Checkout\_ID is the primary key and all other attributes are fully dependent on it, the table is in 2NF.

### **3NF**

There should be no transitive dependencies. The CheckOut table is already in 3NF because there are no transitive dependencies; all non-prime attributes are directly dependent on the primary key (Checkout\_ID).

## 5.0 Relational DB Schemas

### 1. User Management



PK: User\_ID

FD1: User\_ID  $\longrightarrow$  UserName, Password, Role

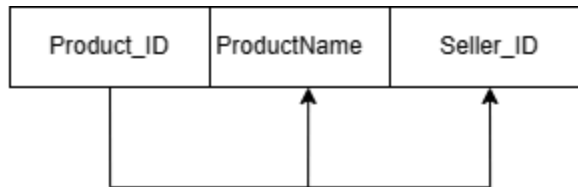
### 2. Seller



PK: Seller\_ID

FD1: Seller\_ID  $\longrightarrow$  SellerName, SellerAddress

### 3. Product Management

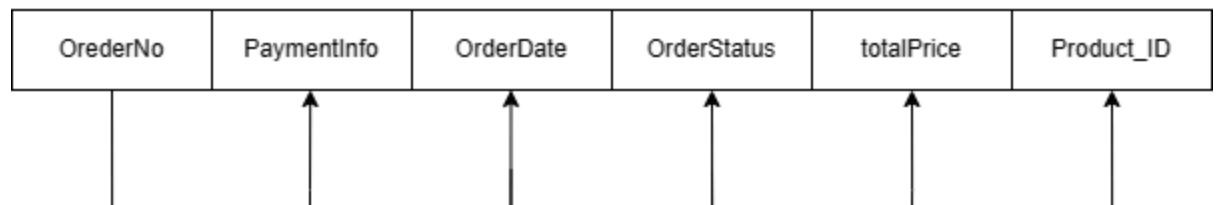


PK: Product\_ID

FK: Seller\_ID

FD1: Product\_ID  $\longrightarrow$  ProductName, Seller\_ID

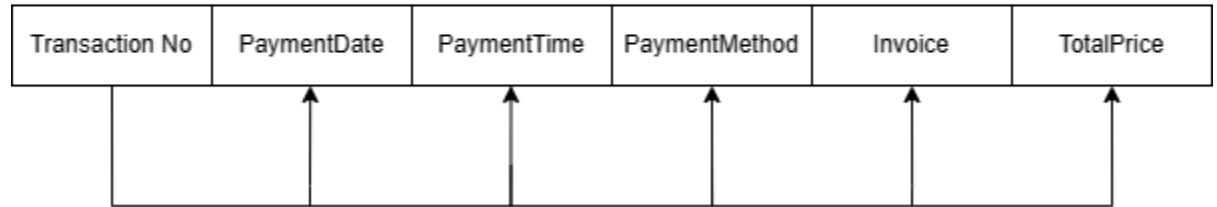
### 4. Order Management



PK: OrderNO

FD1: OrderNO  $\longrightarrow$  PaymentInfo, OrderData, OrderStatus, totalPrice, Product\_ID

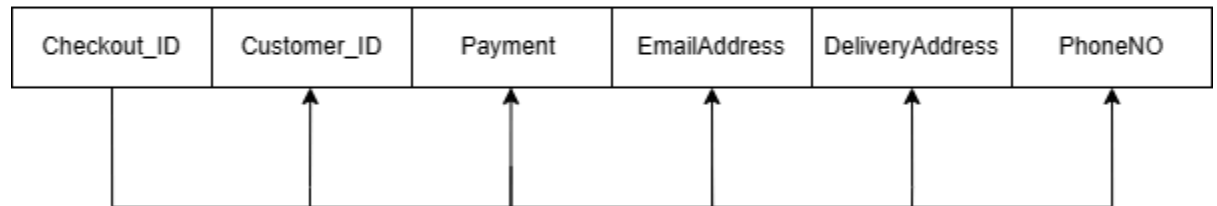
## 5. Payment



PK: TransactionNO

FD1: TransactionNO  $\longrightarrow$  PaymentDate,PaymentTime,PaymentMethod,TotalPrice,Invoice

## 6. Checkout



PK:CheckOut\_ID

FK:Customer\_ID

FD1: CheckOut\_ID  $\longrightarrow$  Customer\_ID,Payment,EmailAddress,DeliveryAddress,PhoneNO

### Final of normalized Relation

1. USER MANAGEMNT(User\_ID,UserName, Password, Role)
2. SELLER(Seller\_ID,SellerName, SellerAddress)
3. PRODUCT MANAGEMNT(Product\_ID,ProductName, Seller\_ID)
4. ORDER MANAGEMENT(ORDER\_NO,PaymentInfo,OrderData,OrderStatus,totalPrice, Product\_ID)
5. PAYMENT(TransactionNO,PaymentDate,PaymentTime,PaymentMethod,TotalPrice ,Invoice)
6. CHECKOUT(Checkout\_ID,Customer\_ID,Payment,EmailAddress,DeliveryAddress ,PhoneNO)

## 6.0 SQL Statement

### 6.1 Create Database and Table

```
CREATE TABLE UserManagement (  
    User_ID VARCHAR2(20) PRIMARY KEY,  
    UserName VARCHAR2(255) NOT NULL,  
    Password VARCHAR2(255) NOT NULL,  
    Role VARCHAR2(50) NOT NULL  
);  
  
CREATE TABLE Seller (  
    Seller_ID VARCHAR2(20) PRIMARY KEY,  
    SellerName VARCHAR2(255) NOT NULL,  
    SellerAddress VARCHAR2(255) NOT NULL  
);  
  
CREATE TABLE Customer (  
    Customer_ID VARCHAR2(20) PRIMARY KEY,  
    User_ID VARCHAR2(20),  
    Customer_Name VARCHAR2(255) NOT NULL,  
    FOREIGN KEY (User_ID) REFERENCES UserManagement(User_ID)  
);  
  
CREATE TABLE ProductManagement (  
    Product_ID VARCHAR2(20) PRIMARY KEY,  
    ProductName VARCHAR2(255) NOT NULL,  
    Seller_ID VARCHAR2(20),  
    FOREIGN KEY (Seller_ID) REFERENCES Seller(Seller_ID)  
);  
  
CREATE TABLE OrderManagement (  
    OrderNo VARCHAR2(20) PRIMARY KEY,  
    PaymentInfo VARCHAR2(255) NOT NULL,  
    OrderDate DATE NOT NULL,  
    OrderStatus VARCHAR2(50) NOT NULL,  
    TotalPrice DECIMAL(10, 2) NOT NULL,  
    Product_ID VARCHAR(20),  
    FOREIGN KEY (Product_ID) REFERENCES ProductManagement(Product_ID)  
);
```

```

CREATE TABLE Payment (
    TransactionNO VARCHAR2(50) PRIMARY KEY,
    PaymentDate DATE NOT NULL,
    PaymentTime TIME NOT NULL,
    PaymentMethod VARCHAR2(50) NOT NULL,
    Invoice VARCHAR2(255) NOT NULL,
    TotalPrice DECIMAL(10, 2) NOT NULL
);

CREATE TABLE Checkout (
    Checkout_ID VARCHAR2(20) PRIMARY KEY,
    Customer_ID VARCHAR2(20),
    Payment INT,
    EmailAddress VARCHAR2(255) NOT NULL,
    DeliveryAddress VARCHAR2(255) NOT NULL,
    PhoneNO VARCHAR2(20) NOT NULL,
    OrderNo VARCHAR2(20),
    FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID),
    FOREIGN KEY (Payment) REFERENCES Payment(TransactionNO),
    FOREIGN KEY (OrderNo) REFERENCES OrderManagement(OrderNo)
);

```

## 6.2 Insert value User Management

```

INSERT INTO UserManagement (User_ID, UserName, Password, Role)
VALUES
    ('US0001', 'lydiaazra', 'password1', 'customer');

```

```

INSERT INTO UserManagement (User_ID, UserName, Password, Role)
VALUES
    ('US0002', 'izzah', 'password2', 'customer');

```

```

INSERT INTO UserManagement (User_ID, UserName, Password, Role)
VALUES
    ('US0003', 'shahd', 'adminpass1', 'seller');

```

```

INSERT INTO UserManagement (User_ID, UserName, Password, Role)
VALUES
    ('US0004', 'mashicat', 'password3', 'customer');

```

```
INSERT INTO UserManagement (User_ID, UserName, Password, Role)
VALUES
```

```
('US0005', 'Admin2', 'adminpass2', 'seller');
```

```
INSERT INTO UserManagement (User_ID, UserName, Password, Role)
VALUES
```

```
('US0006', 'Alana123', 'password4', 'customer');
```

```
INSERT INTO UserManagement (User_ID, UserName, Password, Role)
VALUES
```

```
('US0007', 'CharlieBrown', 'password5', 'customer');
```

```
INSERT INTO UserManagement (User_ID, UserName, Password, Role)
VALUES
```

```
('US0008', 'Admin3', 'adminpass3', 'seller');
```

```
INSERT INTO UserManagement (User_ID, UserName, Password, Role)
VALUES
```

```
('US0009', 'DavidMiller', 'password6', 'customer');
```

```
INSERT INTO UserManagement (User_ID, UserName, Password, Role)
VALUES
```

```
('US0010', 'Farhana Azmi', 'password7', 'customer');
```

```
INSERT INTO UserManagement (User_ID, UserName, Password, Role)
VALUES
```

```
('US0011', 'Admin4', 'adminpass4', 'seller');
```

```
INSERT INTO UserManagement (User_ID, UserName, Password, Role)
VALUES
```

```
('US0012', 'Hani', 'password8', 'customer');
```

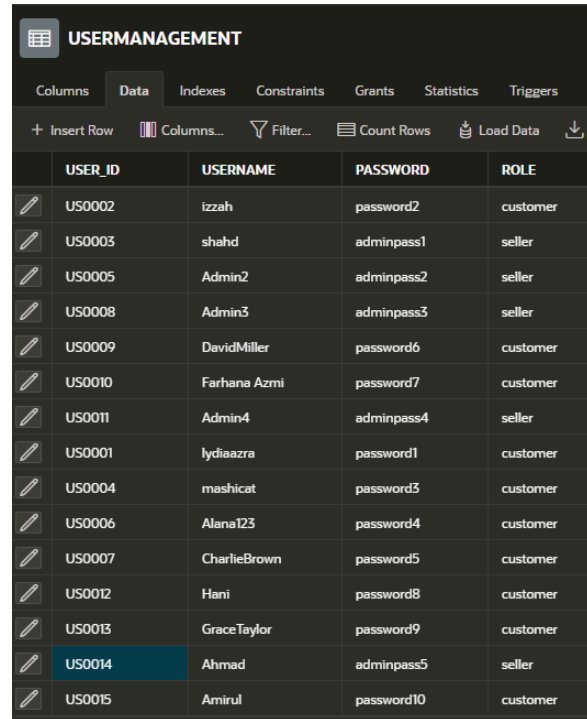
```
INSERT INTO UserManagement (User_ID, UserName, Password, Role)
VALUES
```

```
('US0013', 'GraceTaylor', 'password9', 'customer');
```

```
INSERT INTO UserManagement (User_ID, UserName, Password, Role)
VALUES
```

```
('US0014', 'Ahmad', 'adminpass5', 'seller');
```

```
INSERT INTO UserManagement (User_ID, UserName, Password, Role)
VALUES
('US0015', 'Amirul', 'password10', 'customer');
```



	USER_ID	USERNAME	PASSWORD	ROLE
	US0002	izzah	password2	customer
	US0003	shahd	adminpass1	seller
	US0005	Admin2	adminpass2	seller
	US0008	Admin3	adminpass3	seller
	US0009	DavidMiller	password6	customer
	US0010	Farhana Azmi	password7	customer
	US0011	Admin4	adminpass4	seller
	US0001	lydiaazra	password1	customer
	US0004	meshicat	password3	customer
	US0006	Alana123	password4	customer
	US0007	CharlieBrown	password5	customer
	US0012	Hani	password8	customer
	US0013	GraceTaylor	password9	customer
	US0014	Ahmed	adminpass5	seller
	US0015	Amirul	password10	customer

### 6.3 Insert value Seller

```
INSERT INTO Seller (Seller_ID, SellerName, SellerAddress)
VALUES
('SL0001', 'Pena', 'MA5 KTDI UTM');
```

```
INSERT INTO Seller (Seller_ID, SellerName, SellerAddress)
VALUES
('SL0002', 'Admin2', 'Taman Universiti, Johor');
```

```
INSERT INTO Seller (Seller_ID, SellerName, SellerAddress)
VALUES
('SL0003', 'Kek Sedap', 'Jalan Ampang, Kuala Lumpur');
```

```
INSERT INTO Seller (Seller_ID, SellerName, SellerAddress)
VALUES
('SL0004', 'Muhammad', 'Kolej Tun Razak, UTM, Johor');
```



```
INSERT INTO Seller (Seller_ID, SellerName, SellerAddress)
VALUES
```

```
('SL0005', 'Atikah BookStore', 'Blok U4 Kolej Perdana');
```

```
INSERT INTO Seller (Seller_ID, SellerName, SellerAddress)
VALUES
```

```
('SL0006', 'Kedai Perhiasan Mutiara', 'Blok L01, Kolej Tun Hussein Onn');
```

```
INSERT INTO Seller (Seller_ID, SellerName, SellerAddress)
VALUES
```

```
('SL0007', 'Bukuku', 'Jalan Pahang, Kuala Lumpur');
```

```
INSERT INTO Seller (Seller_ID, SellerName, SellerAddress)
VALUES
```

```
('SL0008', 'Kedai Kita', 'S10, Kolej Tuanku Canselor, UTM');
```

```
INSERT INTO Seller (Seller_ID, SellerName, SellerAddress)
VALUES
```

```
('SL0009', 'Warung Makan Sedap', 'Kolej 13, Kolej Datin Seri Endon');
```

```
INSERT INTO Seller (Seller_ID, SellerName, SellerAddress)
VALUES
```

```
('SL0010', 'Pusat Komputer Canggih', 'Kolej 10, UTM Skudai');
```

SELLER

Columns

Data

Indexes

Constraints

Grants

Statistics

Triggers

Dependencies

+ Insert Row

Columns...

Filter...

Count Rows

Load Data

Download

	SELLER_ID	SELLERNAME	SELLERADDRESS
<div></div>	SL0001	Pena	MA5 KTDI UTM
<div></div>	SL0003	Kek Sedap	Jalan Ampang, Kuala Lumpur
<div></div>	SL0002	Admin2	Taman Universiti, Johor
<div></div>	SL0006	Kedai Perhiasan Mutiara	Blok L01, Kolej Tun Hussein Onn
<div></div>	SL0004	Muhammad	Kolej Tun Razak, UTM, Johor
<div></div>	SL0007	Bukuku	Jalan Pahang, Kuala Lumpur
<div></div>	SL0008	Kedai Kita	S10, Kolej Tuanku Canselor, UTM
<div></div>	SL0005	Atikah BookStore	Blok U4 Kolej Perdana
<div></div>	SL0009	Warung Makan Sedap	Kolej 13, Kolej Datin Seri Endon
<div></div>	SL0010	Pusat Komputer Canggih	Kolej 10, UTM Skudai

#### 6.4 Insert value Customer

```
INSERT INTO Customer (Customer_ID, User_ID, Customer_Name)
VALUES
('CUST0001', 'US0001', 'lydiaazra');
```

```
INSERT INTO Customer (Customer_ID, User_ID, Customer_Name)
VALUES
('CUST0002', 'US0002', 'izzah');
```

```
INSERT INTO Customer (Customer_ID, User_ID, Customer_Name)
VALUES
('CUST0003', 'US0004', 'mashicat');
```

```
INSERT INTO Customer (Customer_ID, User_ID, Customer_Name)
VALUES
('CUST0004', 'US0006', 'Alana123');
```

```
INSERT INTO Customer (Customer_ID, User_ID, Customer_Name)
VALUES
('CUST0005', 'US0009', 'DavidMiller');
```










```
INSERT INTO Customer (Customer_ID, User_ID, Customer_Name)
VALUES
('CUST0006', 'US0010', 'Farhana Azmi');
```

```
INSERT INTO Customer (Customer_ID, User_ID, Customer_Name)
VALUES
('CUST0007', 'US0012', 'Khairul Amri');
```

```
INSERT INTO Customer (Customer_ID, User_ID, Customer_Name)
VALUES
('CUST0008', 'US0013', 'Norliza Yusof');
```

```
INSERT INTO Customer (Customer_ID, User_ID, Customer_Name)
VALUES
('CUST0009', 'US0015', 'Zainab Mustafa');
```

```
INSERT INTO Customer (Customer_ID, User_ID, Customer_Name)
VALUES
('CUST0010', 'US0017', 'Ismail Ibrahim');
```

CUSTOMER			
Columns Data Indexes Constraints Grants Stat			
+ Insert Row Columns... Filter... Count Rows			
	CUSTOMER_ID	USER_ID	CUSTOMER_NAME
	CUST0005	US0009	DavidMiller
	CUST0001	US0001	lydiaazra
	CUST0002	US0002	izzah
	CUST0003	US0004	mashicat
	CUST0004	US0006	Alana123
	CUST0006	US0010	Farhana Azmi
	CUST0007	US0012	Khairul Amri
	CUST0008	US0013	Norliza Yusof
	CUST0009	US0015	Zainab Mustafa

## 6.5 Insert value Product Management

```
INSERT INTO ProductManagement (Product_ID, ProductName, Seller_ID)
VALUES
('PR0001', 'Notebook Set', 'SL0001');
```

```
INSERT INTO ProductManagement (Product_ID, ProductName, Seller_ID)
VALUES
('PR0002', 'Campus T-Shirt', 'SL0002');
```

```
INSERT INTO ProductManagement (Product_ID, ProductName, Seller_ID)
VALUES
('PR0003', 'Healthy Snack Pack', 'SL0003');
```

```
INSERT INTO ProductManagement (Product_ID, ProductName, Seller_ID)
VALUES
('PR0004', 'Laptop Backpack', 'SL0004');
```

```
INSERT INTO ProductManagement (Product_ID, ProductName, Seller_ID)
VALUES
('PR0005', 'Scientific Calculator', 'SL0005');
```

```
INSERT INTO ProductManagement (Product_ID, ProductName, Seller_ID)
VALUES
('PR0006', 'Desk Organizer', 'SL0006');
```











```
INSERT INTO ProductManagement (Product_ID, ProductName, Seller_ID)
VALUES
('PR0007', 'Coffee Mug Set', 'SL0007');
```

```
INSERT INTO ProductManagement (Product_ID, ProductName, Seller_ID)
VALUES
('PR0008', 'Textbook Bundle', 'SL0008');
```

```
INSERT INTO ProductManagement (Product_ID, ProductName, Seller_ID)
VALUES
('PR0009', 'USB Flash Drive', 'SL0009');
```

```
INSERT INTO ProductManagement (Product_ID, ProductName, Seller_ID)
VALUES
```

('PR0010', 'Student Cookware Set', 'SL0010');

PRODUCTMANAGEMENT			
Columns Data Indexes Constraints Grants Statistics			
+ Insert Row Columns... Filter... Count Rows			
	PRODUCT_ID	PRODUCTNAME	SELLER_ID
	PR0001	Notebook Set	SL0001
	PR0002	Campus T-Shirt	SL0002
	PR0005	Scientific Calculator	SL0005
	PR0003	Healthy Snack Pack	SL0003
	PR0004	Laptop Backpack	SL0004
	PR0006	Desk Organizer	SL0006
	PR0007	Coffee Mug Set	SL0007
	PR0008	Textbook Bundle	SL0008
	PR0009	USB Flash Drive	SL0009
	PR0010	Student Cookware Set	SL0010

## 6.6 Insert value Order Management

```
INSERT INTO OrderManagement (OrderNo, PaymentInfo, OrderDate, OrderStatus, TotalPrice, Product_ID)
```

```
VALUES
```

```
  ('ORD0001', 'Credit Card - 1234', TO_DATE('2024-01-22', 'YYYY-MM-DD'), 'Received', 150.00, 'PR0001');
```

```
INSERT INTO OrderManagement (OrderNo, PaymentInfo, OrderDate, OrderStatus, TotalPrice, Product_ID)
```

```
VALUES
```

```
  ('ORD0002', 'PayPal - user@example.com', TO_DATE('2024-01-23', 'YYYY-MM-DD'), 'Delivered', 75.50, 'PR0002');
```

```
INSERT INTO OrderManagement (OrderNo, PaymentInfo, OrderDate, OrderStatus, TotalPrice, Product_ID)
```

```
VALUES
```

```
('ORD0003', 'Debit Card - 5678', TO_DATE('2024-01-24', 'YYYY-MM-DD'),  
'Delivered', 200.00, 'PR0003');
```

```
INSERT INTO OrderManagement (OrderNo, PaymentInfo, OrderDate, OrderStatus,  
TotalPrice, Product_ID)  
VALUES
```

```
('ORD0004', 'Bank Transfer - ABC Bank', TO_DATE('2024-01-25', 'YYYY-MM-DD'),  
'Received', 120.75, 'PR0004');
```

```
INSERT INTO OrderManagement (OrderNo, PaymentInfo, OrderDate, OrderStatus,  
TotalPrice, Product_ID)  
VALUES
```

```
('ORD0005', 'Credit Card - 5678', TO_DATE('2024-01-26', 'YYYY-MM-DD') ,  
'Delivered', 90.00, 'PR0005');
```

```
INSERT INTO OrderManagement (OrderNo, PaymentInfo, OrderDate, OrderStatus,  
TotalPrice, Product_ID)  
VALUES
```

```
('ORD0006', 'PayPal - user2@example.com', TO_DATE('2024-01-27',  
'YYYY-MM-DD') , 'Received', 180.25, 'PR0006');
```

```
INSERT INTO OrderManagement (OrderNo, PaymentInfo, OrderDate, OrderStatus,  
TotalPrice, Product_ID)  
VALUES
```

```
('ORD0007', 'Debit Card - 1234', TO_DATE('2024-01-28', 'YYYY-MM-DD') ,  
'Delivered', 130.50, 'PR0007');
```

```
INSERT INTO OrderManagement (OrderNo, PaymentInfo, OrderDate, OrderStatus,  
TotalPrice, Product_ID)  
VALUES
```

```
('ORD0008', 'Bank Transfer - XYZ Bank', TO_DATE('2024-01-29', 'YYYY-MM-DD'),  
'Received', 95.00, 'PR0008');
```

```
INSERT INTO OrderManagement (OrderNo, PaymentInfo, OrderDate, OrderStatus,  
TotalPrice, Product_ID)  
VALUES
```

```
('ORD0009', 'Credit Card - 9876', TO_DATE('2024-01-30', 'YYYY-MM-DD'),  
'Received', 210.00, 'PR0009');
```

```
INSERT INTO OrderManagement (OrderNo, PaymentInfo, OrderDate, OrderStatus,
TotalPrice, Product_ID)
VALUES
('ORD0010', 'PayPal - user3@example.com', TO_DATE('2024-01-31',
'YYYY-MM-DD') , 'Delivered', 110.75, 'PR0010');
```

ORDERMANAGEMENT

Columns

Data

Indexes

Constraints

Grants

Statistics

Triggers

Dependencies

DDL

Sample Queries

+ Insert Row

Columns...

Filter...

Count Rows

Load Data

Download

Refresh

	ORDERNO	PAYMENTINFO	ORDERDATE	ORDERSTATUS	TOTALPRICE	PRODUCT_ID
<div></div>	ORD0001	Credit Card - 1234	01/22/2024	Received	150	PR0001
<div></div>	ORD0003	Debit Card - 5678	01/24/2024	Delivered	200	PR0003
<div></div>	ORD0004	Bank Transfer - ABC Bank	01/25/2024	Received	120.75	PR0004
<div></div>	ORD0005	Credit Card - 5678	01/26/2024	Delivered	90	PR0005
<div></div>	ORD0007	Debit Card - 1234	01/28/2024	Delivered	130.5	PR0007
<div></div>	ORD0009	Credit Card - 9876	01/30/2024	Received	210	PR0009
<div></div>	ORD0006	PayPal - user2@example.com	01/27/2024	Received	180.25	PR0006
<div></div>	ORD0008	Bank Transfer - XYZ Bank	01/29/2024	Received	95	PR0008
<div></div>	ORD0002	PayPal - user@example.com	01/23/2024	Delivered	75.5	PR0002
<div></div>	ORD0010	PayPal - user3@example.com	01/31/2024	Delivered	110.75	PR0010

1 cells selected

## 6.7 Insert value Payment

```
INSERT INTO Payment (TransactionNO, PaymentDate, PaymentTime, PaymentMethod,
Invoice, TotalPrice)
VALUES
('TR0001', TO_DATE('2024-01-22', 'YYYY-MM-DD') , '12:30', 'Credit Card',
'INV0001', 150.00);
```

```
INSERT INTO Payment (TransactionNO, PaymentDate, PaymentTime, PaymentMethod,
Invoice, TotalPrice)
VALUES
('TR0002', TO_DATE('2024-01-23', 'YYYY-MM-DD') , '14:45', 'TnG', 'INV0002',
75.50);
```

```
INSERT INTO Payment (TransactionNO, PaymentDate, PaymentTime, PaymentMethod,
Invoice, TotalPrice)
VALUES
```

```
('TR0003', TO_DATE('2024-01-24', 'YYYY-MM-DD') , '11:20', 'Debit Card',  
'INV0003', 200.00);
```

```
INSERT INTO Payment (TransactionNO, PaymentDate, PaymentTime, PaymentMethod,  
Invoice, TotalPrice)  
VALUES
```

```
('TR0004', TO_DATE('2024-01-25', 'YYYY-MM-DD') , '09:15', 'Bank Transfer',  
'INV0004', 120.75);
```

```
INSERT INTO Payment (TransactionNO, PaymentDate, PaymentTime, PaymentMethod,  
Invoice, TotalPrice)  
VALUES
```

```
('TR0005', TO_DATE('2024-01-26', 'YYYY-MM-DD') , '16:00', 'Credit Card',  
'INV0005', 90.00);
```

```
INSERT INTO Payment (TransactionNO, PaymentDate, PaymentTime, PaymentMethod,  
Invoice, TotalPrice)  
VALUES
```

```
('TR0006', TO_DATE('2024-01-27', 'YYYY-MM-DD') , '13:10', 'TnG', 'INV0006',  
180.25);
```

```
INSERT INTO Payment (TransactionNO, PaymentDate, PaymentTime, PaymentMethod,  
Invoice, TotalPrice)  
VALUES
```

```
('TR0007', TO_DATE('2024-01-28', 'YYYY-MM-DD') , '10:45', 'Debit Card',  
'INV0007', 130.50);
```

```
INSERT INTO Payment (TransactionNO, PaymentDate, PaymentTime, PaymentMethod,  
Invoice, TotalPrice)  
VALUES
```

```
('TR0008', TO_DATE('2024-01-29', 'YYYY-MM-DD') , '08:30', 'Bank Transfer',  
'INV0008', 95.00);
```

```
INSERT INTO Payment (TransactionNO, PaymentDate, PaymentTime, PaymentMethod,  
Invoice, TotalPrice)  
VALUES
```

```
('TR0009', TO_DATE('2024-01-30', 'YYYY-MM-DD') , '15:20', 'Credit Card',  
'INV0009', 210.00);
```



```

INSERT INTO Payment (TransactionNO, PaymentDate, PaymentTime, PaymentMethod,
Invoice, TotalPrice)
VALUES
('TR0010', TO_DATE('2024-01-31', 'YYYY-MM-DD') , '11:55', 'TnG', 'INV0010',
110.75);

```

PAYMENT						
Columns Data Indexes Constraints Grants Statistics Triggers Dependencies DDL Sample Queries						
+ Insert Row Columns... Filter... Count Rows Load Data Download Refresh						
	TRANSACTIONNO	PAYMENTDATE	PAYMENTTIME	PAYMENTMETHOD	INVOICE	TOTALPRICE
	TR0005	01/26/2024	16:00	Credit Card	INV0005	90
	TR0006	01/27/2024	13:10	TnG	INV0006	180.25
	TR0008	01/29/2024	08:30	Bank Transfer	INV0008	95
	TR0009	01/30/2024	15:20	Credit Card	INV0009	210
	TR0003	01/24/2024	11:20	Debit Card	INV0003	200
	TR0004	01/25/2024	09:15	Bank Transfer	INV0004	120.75
	TR0010	01/31/2024	11:55	TnG	INV0010	110.75
	TR0007	01/28/2024	10:45	Debit Card	INV0007	130.5
	TR0001	01/22/2024	12:30	Credit Card	INV0001	150
	TR0002	01/23/2024	14:45	TnG	INV0002	75.5

## 6.8 Insert value Checkout

```

INSERT INTO Checkout (Checkout_ID, Customer_ID, Payment, EmailAddress,
DeliveryAddress, PhoneNO, OrderNo)
VALUES
('CO0001', 'CUST0001', 'TR0001', 'lydiaazra@example.com', 'Universiti Teknologi
Malaysia, Skudai, Johor', '60123456789', 'ORD0001');

```

```

INSERT INTO Checkout (Checkout_ID, Customer_ID, Payment, EmailAddress,
DeliveryAddress, PhoneNO, OrderNo)
VALUES
('CO0002', 'CUST0002', 'TR0002', 'izzah@example.com', 'Universiti Teknologi
Malaysia, Skudai, Johor', '60129876543', 'ORD0002');

```

```

INSERT INTO Checkout (Checkout_ID, Customer_ID, Payment, EmailAddress,
DeliveryAddress, PhoneNO, OrderNo)
VALUES
('CO0003', 'CUST0003', 'TR0003', 'admin1@example.com', 'Universiti Teknologi
Malaysia, Skudai, Johor', '60112233445', 'ORD0003');

```

```

INSERT INTO Checkout (Checkout_ID, Customer_ID, Payment, EmailAddress,
DeliveryAddress, PhoneNO, OrderNo)
VALUES

```

('CO0004', 'CUST0004', 'TR0004', 'maria.johnson@example.com', 'Universiti Teknologi Malaysia, Skudai, Johor', '60199887766', 'ORD0004');

INSERT INTO Checkout (Checkout\_ID, Customer\_ID, Payment, EmailAddress, DeliveryAddress, PhoneNO, OrderNo)

VALUES

('CO0005', 'CUST0005', 'TR0005', 'admin2@example.com', 'Universiti Teknologi Malaysia, Skudai, Johor', '60112233445', 'ORD0005');

INSERT INTO Checkout (Checkout\_ID, Customer\_ID, Payment, EmailAddress, DeliveryAddress, PhoneNO, OrderNo)

VALUES

('CO0006', 'CUST0006', 'TR0006', 'ahmadfisal@example.com', 'Universiti Teknologi Malaysia, Skudai, Johor', '60199887766', 'ORD0006');

INSERT INTO Checkout (Checkout\_ID, Customer\_ID, Payment, EmailAddress, DeliveryAddress, PhoneNO, OrderNo)

VALUES

('CO0007', 'CUST0007', 'TR0007', 'mahmud189@example.com', 'Universiti Teknologi Malaysia, Skudai, Johor', '60112233445', 'ORD0007');

INSERT INTO Checkout (Checkout\_ID, Customer\_ID, Payment, EmailAddress, DeliveryAddress, PhoneNO, OrderNo)

VALUES

('CO0008', 'CUST0008', 'TR0008', 'aliindra@example.com', 'Universiti Teknologi Malaysia, Skudai, Johor', '60199887766', 'ORD0008');

INSERT INTO Checkout (Checkout\_ID, Customer\_ID, Payment, EmailAddress, DeliveryAddress, PhoneNO, OrderNo)

VALUES

('CO0009', 'CUST0009', 'TR0009', 'karina@example.com', 'Universiti Teknologi Malaysia, Skudai, Johor', '60112233445', 'ORD0009');

INSERT INTO Checkout (Checkout\_ID, Customer\_ID, Payment, EmailAddress, DeliveryAddress, PhoneNO, OrderNo)

VALUES

('CO0010', 'CUST0010', 'TR0010', 'winky@example.com', 'Universiti Teknologi Malaysia, Skudai, Johor', '60199887766', 'ORD0010');

CHECKOUT							
Columns Data Indexes Constraints Grants Statistics Triggers Dependencies DDL Sample Queries							
+ Insert Row Columns... Filter... Count Rows Load Data Download Refresh							
	CHECKOUT_ID	CUSTOMER_ID	PAYMENT	EMAILADDRESS	DELIVERYADDRESS	PHONENO	ORDERNO
	CO0005	CUST0005	TR0005	admin2@example.com	Universiti Teknologi Malaysia, Skudai, ...	6012233445	ORD0005
	CO0008	CUST0008	TR0008	aliindra@example.com	Universiti Teknologi Malaysia, Skudai, ...	60199887766	ORD0008
	CO0009	CUST0009	TR0009	karina@example.com	Universiti Teknologi Malaysia, Skudai, ...	6012233445	ORD0009
	CO0007	CUST0007	TR0007	mahmud189@example.com	Universiti Teknologi Malaysia, Skudai, ...	6012233445	ORD0007
	CO0001	CUST0001	TR0001	lydiaazra@example.com	Universiti Teknologi Malaysia, Skudai, ...	60123456789	ORD0001
	CO0004	CUST0004	TR0004	maria.johnson@example.com	Universiti Teknologi Malaysia, Skudai, ...	60199887766	ORD0004
	CO0006	CUST0006	TR0006	ahmadfaisal@example.com	Universiti Teknologi Malaysia, Skudai, ...	60199887766	ORD0006
	CO0002	CUST0002	TR0002	izzah@example.com	Universiti Teknologi Malaysia, Skudai, ...	60129876543	ORD0002
	CO0003	CUST0003	TR0003	admin1@example.com	Universiti Teknologi Malaysia, Skudai, ...	6012233445	ORD0003

## 6.9 Drop table Checkout

DROP TABLE Checkout;

## 7.0 Delete Customer from table Customer

DELETE FROM Customer  
WHERE Customer\_ID = 'CUST0003';

## 7.1 Update Seller table

UPDATE Seller  
SET SellerAddress = 'M25, KTDI, UTM';  
WHERE Seller\_ID = 'CUST0005';

## 7.0 Summary

The UTM Small Business Platform is a collaborative effort at Universiti Teknologi Malaysia (UTM) aimed at addressing the issues that small company entrepreneurs experience on campus. The platform's goal is to revolutionize small business operations by offering a centralized, user-friendly solution with e-commerce capabilities, marketing tools, and cashless payment choices. The platform aims to create a long-term, financially viable solution in addition to reducing present arduous operations. Through the active participation of small business owners and the development of new revenue sources, the program intends to build a marketplace that benefits both vendors and the UTM community. In general, the UTM Small Business Platform is intended to improve visibility, streamline operations, and develop a dynamic ecosystem of small businesses within the university.

In this phase, learning SQL statements in a database subject involves mastering the essential components of Structured Query Language (SQL) for managing and querying relational databases. This encompasses grasping concepts such as Data Definition Language (DDL) for creating and modifying database schema, Data Manipulation Language (DML) for inserting, updating, and deleting data, and Querying Data for retrieving and filtering information using SELECT statements. We got to implement all the knowledge in this project which will help us have a deeper understanding of database subjects.