

Student Name: Nikolaos Gkmpenompa
 Student Number: 22213708
 Course: Higher Diploma in Science in Data Analytics - HDSDA_SEP23
 Module: Databases for Analytics
 Lecturer: Dr David Hamill
 CA1: Project: Online Retail Business
 Submission Date: 28-04-2024

Overview and Background

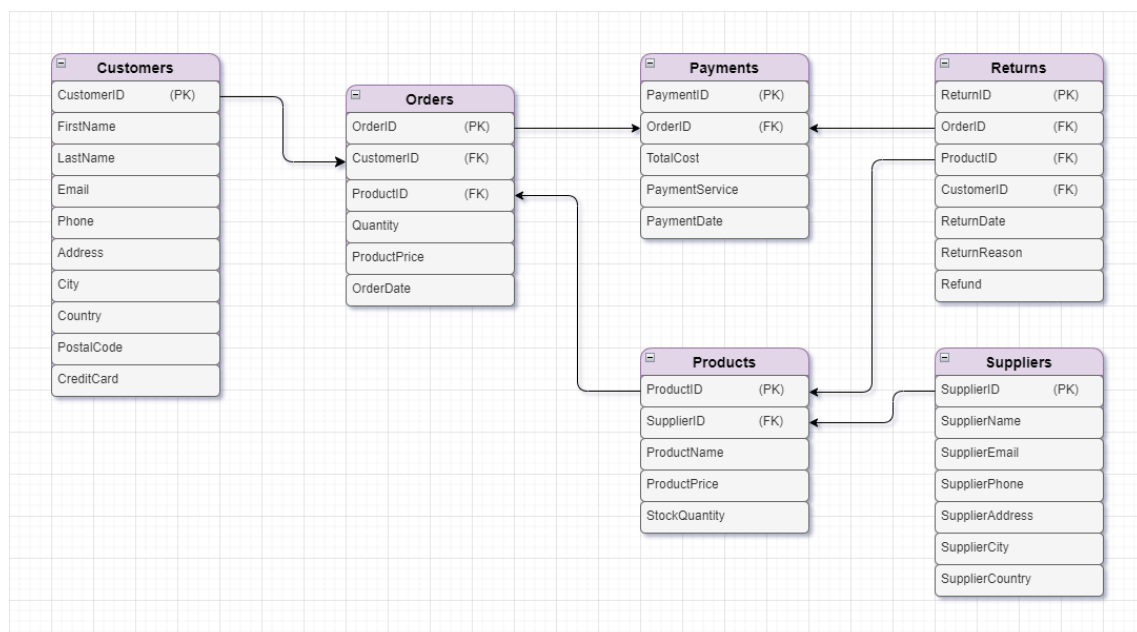
The project is based on an online retail store designed to enable customers purchase products online. In my case, the online business is specialised in combat sports gear and equipment. The idea for this online store comes from my personal passion and involvement in Muay Thai and my respect for Martial Arts philosophy as a whole.

The benefits of an online store are convenience, accessibility, and scalability. Convenience and accessibility because customers can purchase products from any location and outside of store hours. Scalability because as the business grows, an online platform can easily adjust to hold more products, users, and services.

My online store is a small business offering combat sport gear and as any modern retail business, it needs a database system with every stakeholder involved, all attributes, and daily routines. In this project, I will be developing this database.

ERD Diagram

An ERD (Entity-Relationship Diagram) diagram for a retail business is developed based on its structure and relationships between the entities within the business. Following this basic logic, my ERD diagram, as shown below, provides a visual representation of my business entities and how they interact with each other.



My ERD diagram includes the following tables:

- **Customers:** Contains information on the people that purchased products.
Attributes: Customer ID (Primary Key), FirstName, LastName, Email, Phone, Address, City, Country, Postal Code, Credit Card
- **Suppliers:** Contains information on the suppliers of products for my business.
Attributes: Supplier ID (Primary Key), Supplier Name, Supplier Email, Supplier Phone, Supplier Address, Supplier City, Supplier Country
- **Products:** Contains information on products sold by my business and stock quantity.
Attributes: Product ID (Primary Key), Supplier ID (Foreign Key), Product Name, Product Price, Stock Quantity
- **Orders:** Contains information on the orders created by customers.
Attributes: Order ID (Primary Key), Customer ID (Foreign Key), Product ID (Foreign Key), Quantity, Product Price, Order Date
- **Payments:** Contains information on the payments of each order.
Attributes: Payment ID (Primary Key), Order ID (Foreign Key), Total Cost, Payment Service
- **Returns:** Contains information on the returned orders, the products included and the reason.
Attributes: Return ID (Primary Key), Order ID (Foreign Key), Product ID (Foreign Key), Customer ID (Foreign Key), Return Date, Return Reason, Refund

Database Development

To build my database I use MySQL Workbench 8.0. Using the statement **CREATE DATABASE MartialArtsGear;** I created my database. By using the statement **CREATE TABLE TableName (column1 datatype, column2 datatype, ...)** I created the 6 tables that form the basis of my database. In **Customers** table where primary key is **CustomerID**, the customer's credit card is one of the pieces of information included. In **Suppliers** table the primary key is **SupplierID**. In **Products** table the primary key is **ProductID** and foreign key is **SupplierID**. In **Orders** table the primary key is **OrderID** and there are two foreign keys for **CustomerID** and **ProductID**. In **Payments** table the primary key is **PaymentID**, the foreign key is **OrderID** and there is also an attribute **PaymentService** used for the type of payment, like credit card or PayPal. In **Returns** table the primary key is **ReturnID**, there are two foreign keys for **OrderID** and **CustomerID**, the attribute **Refund** is used for the amount (Product Price) to be refunded to customer, and the attribute **ReturnReason** for the reason the order is returned such as: 'Damaged or defective product', 'Wrong product delivered', 'Order arrived too late'.

The data that are later added to all tables, including the Products (STOCK) table, were generated at mocharoo.com, as advised.

All relevant coding to create the database, the tables, insert values, create the Data Mart, and generate the requested Queries are included in the **MartialArtsGear.sql** file, found in the .zip folder.

SQL Queries

In the .sql file, the comment # under every query must be removed for the result to be displayed!

Query 1

(View) - Create a View showing all transactions for a given week in your business.

```
332 #Query 1 (View) - Create a View showing all transactions for a given week in your business.
333 • CREATE VIEW WeekTransactions AS
334 SELECT Orders.OrderID, Customers.CustomerID, Customers.FirstName, Customers.LastName, Products.ProductName, Orders.Quantity,
335 Orders.ProductPrice, Orders.OrderDate, Payments.PaymentID, Payments.TotalCost, Payments.PaymentService, Payments.PaymentDate
336 FROM Orders
337 INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID
338 INNER JOIN Products ON Orders.ProductID = Products.ProductID
339 INNER JOIN Payments ON Orders.OrderID = Payments.OrderID
340 WHERE Payments.PaymentDate >= '2024-04-01'
341 AND Payments.PaymentDate < '2024-04-07';
342
343 • SELECT * FROM WeekTransactions;
```

Result Grid Filter Rows: Export: Wrap Cell Content:												
	OrderID	CustomerID	FirstName	LastName	ProductName	Quantity	ProductPrice	OrderDate	PaymentID	TotalCost	PaymentService	PaymentDate
▶	O07	C04	Marty	Callan	Groin Guard	1	30.00	2024-04-05	07	30.00	creditcard	2024-04-05
	O08	C04	Marty	Callan	Jiu-Jitsu Belt - Brown	1	50.00	2024-04-05	08	50.00	creditcard	2024-04-05
	O11	C06	Paton	Dono	Boxing Headgear	1	50.00	2024-04-03	11	50.00	creditcard	2024-04-03
	O12	C06	Paton	Dono	Boxing Gloves 16oz	1	85.00	2024-04-03	12	85.00	creditcard	2024-04-03
	O19	C10	Ram	Westbury	Boxing Gloves 12oz	1	90.00	2024-04-01	19	90.00	creditcard	2024-04-01
	O20	C10	Ram	Westbury	Thai Pads	1	80.00	2024-04-01	20	80.00	creditcard	2024-04-01
	O41	C21	Byrle	Towndrow	Boxing Headgear	1	50.00	2024-04-02	41	50.00	creditcard	2024-04-02
	O42	C21	Byrle	Towndrow	Boxing Gloves 16oz	1	85.00	2024-04-02	42	85.00	paypal	2024-04-02

Query 2

(Trigger) - Create a Trigger that stores stock levels once a sale takes place.

```
344 #Query 2 (Trigger) - Create a Trigger that stores stock levels once a sale takes place.
345 DELIMITER $$
346 • CREATE TRIGGER StockLevel
347 AFTER INSERT ON Orders
348 FOR EACH ROW
349 BEGIN
350     UPDATE Products
351     SET StockQuantity = StockQuantity - NEW.quantity
352     WHERE ProductID = NEW.ProductID;
353 END $$
354 DELIMITER ;
355
356 • SELECT * FROM Products;
```

Result Grid					
Filter Rows: <input type="text"/>					
Edit: Export/Import: Wrap Cell Content:					
	ProductID	SupplierID	ProductName	ProductPrice	StockQuantity
▶	P01	S01	Boxing Gloves 14oz	80.00	15
	P02	S01	Muay Thai Shorts - Large	35.00	20
	P03	S01	Muay Thai Shorts - Medium	35.00	20
	P04	S01	Muay Thai Shorts - XL	35.00	20
	P05	S01	Knee Pads	25.00	30
	P06	S01	Shin Guards	60.00	25
	P07	S02	Boxing Gloves 12oz	90.00	20

Before placing an order for 3 pieces of the **P04** product the Stock quantity is **20** pieces (above)

After placing an order for 3 pieces of the **P04** product the Stock quantity is **17** pieces (below)





```
358 #Placing an order to check that the trigger actually works
359 INSERT INTO Orders (OrderID, CustomerID, ProductID, Quantity, ProductPrice, OrderDate)
360 VALUES
361 ('071', 'C30', 'P04', 3, 20.00, '2024-04-10');
362
363 • SELECT * FROM Products;
```

Result Grid					
Filter Rows: <input type="text"/>					
Edit: Export/Import: Wrap Cell Content:					
	ProductID	SupplierID	ProductName	ProductPrice	StockQuantity
▶	P01	S01	Boxing Gloves 14oz	80.00	15
	P02	S01	Muay Thai Shorts - Large	35.00	20
	P03	S01	Muay Thai Shorts - Medium	35.00	20
	P04	S01	Muay Thai Shorts - XL	35.00	17
	P05	S01	Knee Pads	25.00	30
	P06	S01	Shin Guards	60.00	25
	P07	S02	Boxing Gloves 12oz	90.00	20
	P08	S02	Mouth Guard	15.00	30

Query 3

(View) - Create a View of stock (by supplier) purchased by you (business).

```
366      #Query 3 (View) - Create a View of stock (by supplier) purchased by you.
367 •    CREATE VIEW MyPurchasedStock AS
368         SELECT Products.ProductID, Products.ProductName, Products.StockQuantity, Suppliers.SupplierName
369         FROM Products
370        JOIN Suppliers ON Products.SupplierID = Suppliers.SupplierID;
371
372 •    SELECT * FROM MyPurchasedStock;
```

Result Grid   Filter Rows: <input type="text"/> Export:  Wrap Cell Content: 				
	ProductID	ProductName	StockQuantity	SupplierName
▶	P01	Boxing Gloves 14oz	15	Fairtex
	P02	Muay Thai Shorts - Large	20	Fairtex
	P03	Muay Thai Shorts - Medium	20	Fairtex
	P04	Muay Thai Shorts - XL	17	Fairtex
	P05	Knee Pads	30	Fairtex
	P06	Shin Guards	25	Fairtex
	P07	Boxing Gloves 12oz	20	Adidas
	P08	Mouth Guard	30	Adidas
	P09	Boxing Hand Wraps	40	Adidas
	P10	Boxing Gloves 16oz	15	Adidas
	P11	Wrist Bands	20	Adidas
	P12	Punching Bag	10	Adidas
	P13	BJJ Gi - White	10	Tatami
	P14	BJJ Gi - Black	10	Tatami
	P15	Jiu-Jitsu Belt - White	20	Tatami
	P16	Grappling Dummy	10	Tatami
	P17	Jiu-Jitsu Belt - Brown	10	Tatami
	P18	Jiu-Jitsu Belt - Blue	20	Tatami
	P19	Sparring Headgear	20	TwinsSpecial
	P20	Boxing Gloves 16oz	25	TwinsSpecial
	P21	Kick Shield	20	TwinsSpecial
	P22	Thai Pads	15	TwinsSpecial
	P23	Boxing Gloves 12oz	15	TwinsSpecial
	P24	Kickboxing Pants - Large	30	TwinsSpecial
	P25	Groin Guard	20	Venum
	P26	MMA Gloves - Red	15	Venum
	P27	MMA Gloves - Black	15	Venum
	P28	Boxing Headgear	20	Venum
	P29	Boxing Punching Bag	10	Venum
	P30	Mouth Guard Case	20	Venum

Query 4

(View) - Create a View of total stock sold to general public (group by supplier).

```
375 #Query 4 (View) - Create a View of total stock sold to general public (group by supplier).
376 #Since all my customers are "public" the script below is applicable.
377 #If I had different types of customers like "public", "business" etc. Before GROUPBY I would add: WHERE Customers.CustomerType = 'Public'
378 • CREATE VIEW TotalStockSold AS
379     SELECT Products.SupplierID, Suppliers.SupplierName, SUM(Orders.Quantity) AS TotalStockSold
380     FROM Orders
381     JOIN Products ON Orders.ProductID = Products.ProductID
382     JOIN Suppliers ON Products.SupplierID = Suppliers.SupplierID
383     GROUP BY Products.SupplierID, Suppliers.SupplierName;
384
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	SupplierID	SupplierName	TotalStockSold
►	S01	Fairtex	18
	S02	Adidas	14
	S03	Tatami	14
	S04	TwinsSpecial	14
	S05	Venum	13

TotalStockSold 8 ~

Query 5

Detail and total all transactions (SALES) for the month-to-date. (A Group By with Roll-Up) – 1 month

```
388 #Query 5 - Detail and total all transactions (SALES) for the month-to-date. (A Group By with Roll-Up) - 1 month
389 DELIMITER $$
390 • CREATE PROCEDURE month_to_date_transactions(current_month INT, current_year INT)
391 BEGIN
392     SELECT
393         COALESCE(OrderDate, 'Total') AS OrderDate,
394         OrderID,
395         SUM(ProductPrice) AS TotalPrice,
396         SUM(Quantity) AS TotalQuantity,
397         CustomerID
398     FROM orders
399     WHERE MONTH(OrderDate) = current_month AND YEAR(OrderDate) = current_year
400     GROUP BY OrderDate, OrderID, CustomerID WITH ROLLUP;
401 END $$
402 DELIMITER ;
403
404 • CALL month_to_date_transactions(MONTH(CURRENT_DATE()), YEAR(CURRENT_DATE()));
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	OrderDate	OrderID	TotalPrice	TotalQuantity	CustomerID
▶	2024-04-01	O19	90.00	1	C10
	2024-04-01	O19	90.00	1	NULL
	2024-04-01	O20	80.00	1	C10
	2024-04-01	O20	80.00	1	NULL
	2024-04-01	NULL	170.00	2	NULL
	2024-04-02	O41	50.00	1	C21
	2024-04-02	O41	50.00	1	NULL
	2024-04-02	O42	85.00	1	C21
	2024-04-02	O42	85.00	1	NULL
	2024-04-02	NULL	135.00	2	NULL
	2024-04-03	O11	50.00	1	C06
	2024-04-03	O11	50.00	1	NULL
	2024-04-03	O12	85.00	1	C06
	2024-04-03	O12	85.00	1	NULL
	2024-04-03	NULL	135.00	2	NULL
	2024-04-05	O07	30.00	1	C04
	2024-04-05	O07	30.00	1	NULL
	2024-04-05	O08	50.00	1	C04

...

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	OrderDate	OrderID	TotalPrice	TotalQuantity	CustomerID
	2024-04-10	NULL	225.00	7	NULL
	2024-04-11	O59	45.00	1	C30
	2024-04-11	O59	45.00	1	NULL
	2024-04-11	O60	100.00	1	C30
	2024-04-11	O60	100.00	1	NULL
	2024-04-11	NULL	145.00	2	NULL
	2024-04-12	O37	30.00	1	C19
	2024-04-12	O37	30.00	1	NULL
	2024-04-12	O38	50.00	1	C19
	2024-04-12	O38	50.00	1	NULL
	2024-04-12	NULL	80.00	2	NULL
	2024-04-13	O69	35.00	1	C05
	2024-04-13	O69	35.00	1	NULL
	2024-04-13	O70	45.00	1	C05
	2024-04-13	O70	45.00	1	NULL
	2024-04-13	NULL	80.00	2	NULL
	Total	NULL	1535.00	29	NULL

Query 6

Detail and total all SALES for the year-to-date. (A Group By with Roll-Up) – All months

```
407 #Query 6 - Detail and total all SALES for the year-to-date. (A Group By with Roll-Up) - All months
408 DELIMITER $$
409 • CREATE PROCEDURE year_to_date_transactions(current_year INT)
410 BEGIN
411     SELECT
412         COALESCE(MONTH(OrderDate), 'Total') AS Month,
413         COALESCE(YEAR(OrderDate), current_year) AS Year,
414         OrderID,
415         SUM(ProductPrice) AS TotalPrice,
416         SUM(Quantity) AS TotalQuantity,
417         CustomerID
418     FROM orders
419     WHERE YEAR(OrderDate) = current_year
420     GROUP BY COALESCE(YEAR(OrderDate), current_year), COALESCE(MONTH(OrderDate), 'Total'), OrderID, CustomerID, OrderDate WITH ROLLUP;
421 END $$
422 DELIMITER ;
423
424 • CALL year_to_date_transactions(YEAR(CURRENT_DATE()));
```

Result Grid

	Month	Year	OrderID	TotalPrice	TotalQuantity	CustomerID
▶	2	2024	O04	60.00	1	C03
	2	2024	O04	60.00	1	C03
	2	2024	O04	60.00	1	NULL
	2	2024	O13	50.00	1	C07
	2	2024	O13	50.00	1	C07
	2	2024	O13	50.00	1	NULL
	2	2024	O14	100.00	1	C07
	2	2024	O14	100.00	1	C07
	2	2024	O14	100.00	1	NULL
	2	2024	O23	5.00	1	C12
	2	2024	O23	5.00	1	C12
	2	2024	O23	5.00	1	NULL
	2	2024	O24	15.00	1	C12
	2	2024	O24	15.00	1	C12
	2	2024	O24	15.00	1	NULL
	2	2024	O25	50.00	1	C13

Result 10

...

	2	2024	O65	35.00	1	C03
	2	2024	O65	35.00	1	C03
	2	2024	O65	35.00	1	NULL
	2	2024	O66	90.00	1	C03
	2	2024	O66	90.00	1	C03
	2	2024	O66	90.00	1	NULL
	2	2024	NULL	1075.00	19	NULL
	3	2024	O03	35.00	1	C02
	3	2024	O03	35.00	1	C02
	3	2024	O03	35.00	1	NULL
	3	2024	O05	35.00	1	C03
	3	2024	O05	35.00	1	C03
	3	2024	O05	35.00	1	NULL
	3	2024	O06	90.00	1	C03
	3	2024	O06	90.00	1	C03
	3	2024	O06	90.00	1	NULL

Result 10

...

	Month	Year	OrderID	TotalPrice	TotalQuantity	CustomerID
	3	2024	O64	60.00	1	NULL
	3	2024	O67	30.00	1	C04
	3	2024	O67	30.00	1	C04
	3	2024	O67	30.00	1	NULL
	3	2024	O68	50.00	1	C04
	3	2024	O68	50.00	1	C04
	3	2024	O68	50.00	1	NULL
	3	2024	NULL	1500.00	25	NULL
	4	2024	O01	80.00	1	C01
	4	2024	O01	80.00	1	C01
	4	2024	O01	80.00	1	NULL
	4	2024	O02	30.00	1	C01
	4	2024	O02	30.00	1	C01
	4	2024	O02	30.00	1	NULL
	4	2024	O07	30.00	1	C04
	4	2024	O07	30.00	1	C04

Result 10

...

	4	2024	O61	80.00	1	C01
	4	2024	O61	80.00	1	NULL
	4	2024	O62	30.00	1	C01
	4	2024	O62	30.00	1	C01
	4	2024	O62	30.00	1	NULL
	4	2024	O69	35.00	1	C05
	4	2024	O69	35.00	1	C05
	4	2024	O69	35.00	1	NULL
	4	2024	O70	45.00	1	C05
	4	2024	O70	45.00	1	C05
	4	2024	O70	45.00	1	NULL
	4	2024	O71	20.00	3	C30
	4	2024	O71	20.00	3	C30
	4	2024	O71	20.00	3	NULL
	4	2024	NULL	1535.00	29	NULL
	NULL	2024	NULL	4110.00	73	NULL

Result 10

Query 7

Display the growth in sales/services (as a percentage) for your business, from the 1st month of opening until now.

```
427 #Query 7 - Display the growth in sales/services (as a percentage) for your business, from the 1st month of opening until now.
428 SELECT
429     CONCAT(FORMAT(((SUM(Orders.TotalAmount) - first_month_sales) / first_month_sales) * 100, 2), "%") AS sales_growth_percentage
430 FROM
431     (SELECT
432         SUM(ProductPrice) AS first_month_sales
433     FROM Orders
434     WHERE YEAR(OrderDate) = 2024 AND MONTH(OrderDate) = 2) AS first_sales,
435     (SELECT
436         YEAR(OrderDate) AS year,
437         MONTH(OrderDate) AS month,
438         SUM(ProductPrice) AS TotalAmount
439     FROM Orders
440     WHERE (YEAR(OrderDate) = 2024 AND MONTH(OrderDate) >= 2) OR (YEAR(OrderDate) = 2024 AND MONTH(OrderDate) <= 4)
441     GROUP BY YEAR(OrderDate), MONTH(OrderDate)) AS Orders
442 GROUP BY first_month_sales;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
sales_growth_percentage			
282.33%			

Query 8

Display all returns to date.

```
445 #Query 8 - Display all returns to date.
446 • SELECT * FROM returns;
```

ReturnID	OrderID	ProductID	CustomerID	ReturnDate	ReturnReason	Refund
R01	O13	P27	C07	2024-02-25	Damaged or defective product	50.00
R02	O14	P29	C07	2024-02-25	Wrong product delivered	100.00
R03	O19	P23	C10	2024-04-15	Order arrived too late	90.00
R04	O20	P22	C10	2024-04-15	Order arrived too late	80.00
R05	O21	P18	C11	2024-03-19	Wrong product delivered	30.00
R06	O24	P09	C12	2024-02-28	Wrong product ordered	15.00
R07	O28	P20	C14	2024-03-05	Order arrived too late	80.00
R08	O32	P08	C16	2024-04-15	Damaged or defective product	30.00
R09	O34	P21	C17	2024-03-30	Wrong product ordered	60.00
R10	O67	P25	C04	2024-04-10	Damaged or defective product	30.00

returns 12 ×

Data Mart (View Results)

The Data Mart containing results only from Views (Questions 1, 3 and 4 ONLY)

```
448 # Creating the Data Mart containing the results ONLY from Views (Questions 1, 3 and 4)
449 • DROP DATABASE IF EXISTS DataMart;
450 • CREATE DATABASE DataMart;
451 • USE DataMart;
452
453 #Query 1 (View) - Create a View showing all transactions for a given week in your business.
454 • CREATE TABLE DataMart.WeekTransactions AS SELECT * FROM MartialArtsGear.WeekTransactions;
455 • select * from WeekTransactions;
456
457 #Query 3 (View) - Create a View of stock (by supplier) purchased by you.
458 • CREATE TABLE DataMart.MyPurchasedStock AS SELECT * FROM MartialArtsGear.MyPurchasedStock;
459 • select * from MyPurchasedStock;
460
461 #Query 4 (View) - Create a View of total stock sold to general public (group by supplier).
462 • CREATE TABLE DataMart.TotalStockSold AS SELECT * FROM MartialArtsGear.TotalStockSold;
463 • select * from TotalStockSold;
```

Result Grid											
Filter Rows: <input type="text"/>				Export: <input type="text"/>		Wrap Cell Content: <input type="text"/>					
OrderID	CustomerID	FirstName	LastName	ProductName	Quantity	ProductPrice	OrderDate	PaymentID	TotalCost	PaymentService	PaymentDate
O07	C04	Marty	Callan	Groin Guard	1	30.00	2024-04-05	07	30.00	creditcard	2024-04-05
O08	C04	Marty	Callan	Jiu-Jitsu Belt - Brown	1	50.00	2024-04-05	08	50.00	creditcard	2024-04-05
O11	C06	Paton	Dono	Boxing Headgear	1	50.00	2024-04-03	11	50.00	creditcard	2024-04-03
O12	C06	Paton	Dono	Boxing Gloves 16oz	1	85.00	2024-04-03	12	85.00	creditcard	2024-04-03
O19	C10	Ram	Westbury	Boxing Gloves 12oz	1	90.00	2024-04-01	19	90.00	creditcard	2024-04-01
O20	C10	Ram	Westbury	Thai Pads	1	80.00	2024-04-01	20	80.00	creditcard	2024-04-01
O41	C21	Byrle	Towndrow	Boxing Headgear	1	50.00	2024-04-02	41	50.00	creditcard	2024-04-02
O42	C21	Byrle	Towndrow	Boxing Gloves 16oz	1	85.00	2024-04-02	42	85.00	paypal	2024-04-02
Result 16		WeekTransactions 17		MyPurchasedStock 18		TotalStockSold 19					

ProductID	ProductName	StockQuantity	SupplierName
P01	Boxing Gloves 14oz	15	Fairtex
P02	Muay Thai Shorts - Large	20	Fairtex
P03	Muay Thai Shorts - Medium	20	Fairtex
P04	Muay Thai Shorts - XL	17	Fairtex
P05	Knee Pads	30	Fairtex
P06	Shin Guards	25	Fairtex
P07	Boxing Gloves 12oz	20	Adidas
P08	Mouth Guard	30	Adidas
P09	Boxing Hand Wraps	40	Adidas
P10	Boxing Gloves 16oz	15	Adidas
P11	Wrist Bands	20	Adidas
P12	Punching Bag	10	Adidas
P13	BJJ Gi - White	10	Tatami
P14	BJJ Gi - Black	10	Tatami
P15	Jiu-Jitsu Belt - White	20	Tatami
P16	Grappling Dummy	10	Tatami
P17	Jiu-Jitsu Belt - Brown	10	Tatami
P18	Jiu-Jitsu Belt - Blue	20	Tatami
P19	Sparring Headgear	20	TwinsSpecial
P20	Boxing Gloves 16oz	25	TwinsSpecial
P21	Kick Shield	20	TwinsSpecial
P22	Thai Pads	15	TwinsSpecial
P23	Boxing Gloves 12oz	15	TwinsSpecial
P24	Kickboxing Pants - Large	30	TwinsSpecial
P25	Groin Guard	20	Venum
P26	MMA Gloves - Red	15	Venum
P27	MMA Gloves - Black	15	Venum
P28	Boxing Headgear	20	Venum
P29	Boxing Punching Bag	10	Venum
P30	Mouth Guard Case	20	Venum

Result 16 WeekTransactions 17 MyPurchasedStock 18 x TotalStockSold 19

SupplierID	SupplierName	TotalStockSold
S01	Fairtex	18
S02	Adidas	14
S03	Tatami	14
S04	TwinsSpecial	14
S05	Venum	13

Result 16 WeekTransactions 17 MyPurchasedStock 18 TotalStockSold 19 x