

Vinicius Verza

MySQL Database Online Watch Shop

1- INTRODUCTION

The following information reports the creation of an online database for an online watch shop as well as views, triggers and stored procedures. The tables were created on MySQL, populated with random relevant information.

2- DATABASE

Below you can observe the structure created for this database. The primary keys are the first one and the relevant foreign keys are within each table although the proper correlations were not set.

Table: customer

Columns:

<u>CustomerID</u>	varchar(10) PK
FirstName	varchar(50)
LastName	varchar(50)
Address	varchar(255)
City	varchar(60)
County	varchar(60)
Country	varchar(60)
Phone	varchar(20)
Email	varchar(50)

Table: stock

Columns:

<u>StockID</u>	varchar(10) PK
ItemBrand	varchar(50)
ItemModel	varchar(50)
QtyStock	int(11)
ItemCost	decimal(6,2)
SupplierID	varchar(10)

Table: basket

Columns:

<u>OrderID</u>	varchar(10) PK
OrderDate	date
PaymentType	varchar(50)
CustomerID	varchar(10)

Table: supplier

Columns:

<u>SupplierID</u>	varchar(10) PK
SupName	varchar(50)
SupEmail	varchar(50)
SupPhone	varchar(20)

Table: orderitems

Columns:

<u>ItemID</u>	varchar(10) PK
QtySold	int(11)
ItemPrice	decimal(6,2)
VAT	decimal(2,2)
OrderID	varchar(10)
StockID	varchar(10)

Table: returns

Columns:

<u>ReturnID</u>	varchar(10) PK
ReturnDate	date
AmountRet	decimal(6,2)
ItemID	varchar(10)
CustomerID	varchar(10)

These are the tables created on a database called 'WatchShop'. Following, all tables were populated with random information. Customer table has 10 entries, Stock has 15, Basket 50, Supplier has 5, OrderItems has 58 and Returns has 10.

The SQL file for its creation and population is found along with this report.

3- VIEWS, TRIGGERS AND STORED PROCEDURES

In this section you can find the input code for every question, following a screenshot with the code output:

1 - [View] Showing all transactions for a given week:

```
CREATE VIEW Transactions_August_Week_2 AS
SELECT Basket.OrderID,
DATE_FORMAT(Basket.OrderDate, '%M %d - %Y') AS Date,
CONCAT((Stock.ItemBrand), ' ', (Stock.ItemModel)) AS Item,
OrderItems.ItemPrice,
Basket.PaymentType,
CONCAT((Customer.FirstName), ' ', (Customer.LastName)) AS Customer
FROM Basket
JOIN OrderItems ON OrderItems.OrderID = Basket.OrderID
JOIN Stock ON Stock.StockID = OrderItems.StockID
JOIN Customer ON Customer.CustomerID = Basket.CustomerID
WHERE OrderDate BETWEEN '2019-08-10' AND '2019-08-17';
```

Output:

	OrderID	Date	Item	ItemPrice	PaymentType	Customer
▶	O32	August 10 - 2019	Casio G-Shock GA2100	86.65	Debit Card	Robert Plant
	O33	August 11 - 2019	Citizen Diver BJ8050-08E	204.13	Credit Card	Freddie Mercury
	O34	August 13 - 2019	Casio Pro-Trek PRW-3100YB	241.33	Debit Card	Robert Plant

2 - [Trigger] Create a Trigger that updates stock levels once a sale takes

```
DELIMITER $$
CREATE TRIGGER StockLevelUpdate
    AFTER INSERT ON OrderItems
    FOR EACH ROW BEGIN
    UPDATE Stock
    SET QtyStock = QtyStock - NEW.QtySold
    WHERE StockID = NEW.StockID;
END$$
DELIMITER ;

# Transaction Test for Trigger Created on Question 2
INSERT INTO OrderItems (ItemID, QtySold, ItemPrice, VAT, OrderID, StockID)
VALUES ("I222", 2, 9.55, 0.23, "01", "ST1");
```

Output:

Stock table before transaction test:

StockID	ItemBrand	ItemModel	QtyStock	ItemCost	SupplierID
ST1	Casio	F-91W	20	7.96	SU1
ST10	Seiko	Prospex SNE499P1	20	398.02	SU3
ST11	Citizen	Skyhawk A-T Red Arrows	20	610.83	SU5
ST12	Citizen	Eco-Drive AT4000-02E	20	298.21	SU5
ST13	Citizen	Diver BJ8050-08E	20	170.11	SU5
ST14	Vostok	Komandirskie 2416B	20	87.94	SU4
ST15	Vostok	Amphibia 120812	20	48.67	SU4
ST2	Seiko	Big Orange SRPC55	20	78.45	SU3
ST3	Casio	G-Shock GA2100	20	72.21	SU1
ST4	Omega	Speedmaster	20	1650.21	SU2
ST5	Vostok	Amphibia Scuba Dude 71...	20	31.01	SU4
ST6	Casio	Edifice EF-527D-1AVEF	20	98.97	SU1
ST7	Casio	Edifice EF-566DB	20	82.59	SU1
ST8	Casio	Pro-Trek PRW-3100YB	20	201.11	SU1
ST9	Seiko	SSB313P1	20	150.26	SU3

Stock table after transaction test:

StockID	ItemBrand	ItemModel	QtyStock	ItemCost	SupplierID
ST1	Casio	F-91W	18	7.96	SU1
ST10	Seiko	Prospex SNE499P1	20	398.02	SU3
ST11	Citizen	Skyhawk A-T Red Arrows	20	610.83	SU5
ST12	Citizen	Eco-Drive AT4000-02E	20	298.21	SU5
ST13	Citizen	Diver BJ8050-08E	20	170.11	SU5
ST14	Vostok	Komandirskie 2416B	20	87.94	SU4
ST15	Vostok	Amphibia 120812	20	48.67	SU4
ST2	Seiko	Big Orange SRPC55	20	78.45	SU3
ST3	Casio	G-Shock GA2100	20	72.21	SU1
ST4	Omega	Speedmaster	20	1650.21	SU2
ST5	Vostok	Amphibia Scuba Dude 71...	20	31.01	SU4
ST6	Casio	Edifice EF-527D-1AVEF	20	98.97	SU1
ST7	Casio	Edifice EF-566DB	20	82.59	SU1
ST8	Casio	Pro-Trek PRW-3100YB	20	201.11	SU1
ST9	Seiko	SSB313P1	20	150.26	SU3

3 - [View] Create a View of all stock, grouped by supplier:

```
CREATE VIEW All_Stock AS
SELECT Supplier.SupName, Stock.StockID, Stock.ItemBrand, Stock.ItemModel, Stock.SupplierID, Stock.QtyStock
FROM Stock
JOIN Supplier
ON Stock.SupplierID=Supplier.SupplierID
ORDER BY Supplier.SupName;
```

Output:

SupName	StockID	ItemBrand	ItemModel	SupplierID	QtyStock
David Hamilton Watch Factory	ST10	Seiko	Prospex SNE499P1	SU3	20
David Hamilton Watch Factory	ST2	Seiko	Big Orange SRPC55	SU3	20
David Hamilton Watch Factory	ST9	Seiko	SSB313P1	SU3	20
DimSum China Watches LTD	ST1	Casio	F-91W	SU1	18
DimSum China Watches LTD	ST3	Casio	G-Shock GA2100	SU1	20
DimSum China Watches LTD	ST6	Casio	Edifice EF-527D-1AVEF	SU1	20
DimSum China Watches LTD	ST7	Casio	Edifice EF-566DB	SU1	20
DimSum China Watches LTD	ST8	Casio	Pro-Trek PRW-3100YB	SU1	20
Its Pork O'Clock! Handmade Watches	ST11	Citizen	Skyhawk A-T Red Arrows	SU5	20
Its Pork O'Clock! Handmade Watches	ST12	Citizen	Eco-Drive AT4000-02E	SU5	20
Its Pork O'Clock! Handmade Watches	ST13	Citizen	Diver BJ8050-08E	SU5	20
NCI Watches Corporation	ST4	Omega	Speedmaster	SU2	20
What Time is It? Corporation	ST14	Vostok	Komandirskie 2416B	SU4	20
What Time is It? Corporation	ST15	Vostok	Amphibia 120812	SU4	20
What Time is It? Corporation	ST5	Vostok	Amphibia Scuba Dude 71...	SU4	20

4 - [Stored Procedure] Detail and total all sales for the year, group these by each month, a Group by with Roll-up:

```
DELIMITER //
CREATE PROCEDURE DetailedSales ()
BEGIN
SELECT
MONTHNAME(Basket.OrderDate) AS Month,
COUNT(Basket.OrderID) AS Items_Sold,
CONCAT('€ ', FORMAT(SUM(OrderItems.ItemPrice), 2)) AS Gross_Income_No_VAT,
CONCAT('€ ', FORMAT(SUM(OrderItems.ItemPrice * 1.23), 2)) AS Gross_Income_VAT,
CONCAT('€ ', FORMAT(SUM(OrderItems.ItemPrice * 0.20), 2)) AS Net_Income
FROM Basket
INNER JOIN OrderItems ON Basket.OrderID = OrderItems.OrderID
GROUP BY MONTH(Basket.OrderDate) WITH ROLLUP;
END//

# Calling the Procedure
CALL DetailedSales
```

Output:

Month	Items_Sold	Gross_Income_No_VAT	Gross_Income_VAT	Net_Income
January	6	€ 2,372.03	€ 2,917.60	€ 474.41
February	8	€ 2,400.06	€ 2,952.07	€ 480.01
March	4	€ 463.78	€ 570.45	€ 92.76
April	5	€ 2,521.13	€ 3,100.99	€ 504.23
May	3	€ 299.55	€ 368.45	€ 59.91
June	6	€ 2,497.66	€ 3,072.12	€ 499.53
July	5	€ 4,174.87	€ 5,135.09	€ 834.97
August	6	€ 2,056.49	€ 2,529.48	€ 411.30
September	4	€ 1,450.93	€ 1,784.64	€ 290.19
October	6	€ 1,311.16	€ 1,612.73	€ 262.23
November	3	€ 680.74	€ 837.31	€ 136.15
December	3	€ 595.37	€ 732.31	€ 119.07
December	59	€ 20,823.77	€ 25,613.24	€ 4,164.75

5 - [View] Display the growth in sales (as a percentage) for your business, from the 1st month of opening until the end of the year:

```
CREATE VIEW Sales_Growth AS
SELECT
MONTHNAME(Basket.OrderDate) AS Months,
COUNT(Basket.OrderID) AS Total_Orders,
CONCAT('€ ', FORMAT(SUM(OrderItems.ItemPrice), 2)) AS Income,
CONCAT('€ ', LAG (SUM(OrderItems.ItemPrice)) OVER (ORDER BY Basket.OrderDate ASC)) AS Previous_Month,
CONCAT(ROUND((SUM(OrderItems.ItemPrice * OrderItems.QtySold) - LAG((SUM(OrderItems.ItemPrice * OrderItems.QtySold)),1)
OVER ()) / (LAG((SUM(OrderItems.ItemPrice * OrderItems.QtySold)),1) OVER()) * 100, 1), ' %') AS Growth
FROM Basket
INNER JOIN OrderItems ON Basket.OrderID = OrderItems.OrderID
GROUP BY MONTH(Basket.OrderDate);
```

Output:

Months	Total_Orders	Income	Previous_Month	Growth
January	6	€ 2,372.03	NULL	NULL
February	8	€ 2,400.06	€ 2372.03	6.6 %
March	4	€ 463.78	€ 2400.06	-74.3 %
April	5	€ 2,521.13	€ 463.78	286.6 %
May	3	€ 299.55	€ 2521.13	-84.7 %
June	6	€ 2,497.66	€ 299.55	546.7 %
July	5	€ 4,174.87	€ 2497.66	67.2 %
August	6	€ 2,056.49	€ 4174.87	-45.9 %
September	4	€ 1,450.93	€ 2056.49	-35.8 %
October	6	€ 1,311.16	€ 1450.93	2.8 %
November	3	€ 680.74	€ 1311.16	-54.4 %
December	3	€ 595.37	€ 680.74	4.9 %

6 - [Stored Procedure] Create a stored procedure that will display all orders by customer and their county:

```
DELIMITER //
CREATE PROCEDURE OrdersByCustomers ()
BEGIN
SELECT CONCAT(Customer.FirstName, ' ', Customer.LastName) AS CustomerName,
Basket.OrderID, Customer.Country, Basket.OrderDate
FROM Basket
JOIN Customer ON Customer.CustomerID = Basket.CustomerID
ORDER BY Customer.FirstName;
END//
```

Output:

CustomerName	OrderID	Country	OrderDate
Arnold Schwarzenegger	O25	USA	2019-06-28
David Gilmour	O20	USA	2019-05-17
David Gilmour	O45	USA	2019-10-24
David Gilmour	O39	USA	2019-09-27
David Gilmour	O28	USA	2019-07-16
David Gilmour	O18	USA	2019-04-23
David Gilmour	O5	USA	2019-02-03
David Gilmour	O47	USA	2019-11-09
Freddie Mercury	O9	Mexico	2019-02-22
Freddie Mercury	O3	Mexico	2019-01-22
Freddie Mercury	O15	Mexico	2019-04-07
Freddie Mercury	O16	Mexico	2019-04-11
Freddie Mercury	O33	Mexico	2019-08-11
Freddie Mercury	O48	Mexico	2019-12-24
Freddie Mercury	O22	Mexico	2019-06-01
George Washington	O6	USA	2019-02-11
George Washington	O27	USA	2019-07-16
George Washington	O12	USA	2019-03-11
George Washington	O29	USA	2019-07-18
George Washington	O31	USA	2019-08-02
George Washington	O43	USA	2019-10-19
James Bond	O14	Ireland	2019-03-26
James Bond	O44	Ireland	2019-10-22
James Bond	O46	Ireland	2019-11-03
Milton Weissbier	O19	Ireland	2019-05-12
Milton Weissbier	O23	Ireland	2019-06-07
Nilson Ale	O35	Ireland	2019-08-19
Nilson Ale	O41	Ireland	2019-10-11
Nilson Ale	O21	Ireland	2019-05-23
Nilson Ale	O2	Ireland	2019-01-19
Nilson Ale	O8	Ireland	2019-02-16

Robert Plant	O10	USA	2019-02-22
Robert Plant	O11	USA	2019-03-10
Robert Plant	O34	USA	2019-08-13
Robert Plant	O42	USA	2019-10-18
Robert Plant	O24	USA	2019-06-26
Robert Plant	O32	USA	2019-08-10
Robert Plant	O50	USA	2019-12-29
Robert Plant	O49	USA	2019-12-25
Whoopi Goldberg	O13	USA	2019-03-12
Whoopi Goldberg	O36	USA	2019-08-30
Whoopi Goldberg	O40	USA	2019-10-01
Whoopi Goldberg	O1	USA	2019-01-06
Whoopi Goldberg	O7	USA	2019-02-11
Wilson Pilsen	O17	Ireland	2019-04-19
Wilson Pilsen	O37	Ireland	2019-09-22
Wilson Pilsen	O38	Ireland	2019-09-24
Wilson Pilsen	O26	Ireland	2019-07-08
Wilson Pilsen	O4	Ireland	2019-01-27
Wilson Pilsen	O30	Ireland	2019-07-26

7 - [Stored Procedure] Create a stored procedure that will display all returns, grouped by month:

```
DELIMITER //
```

```
CREATE PROCEDURE TotalReturns ()
```

```
BEGIN
```

```
SELECT
```

```
MONTHNAME>Returns.ReturnDate) AS Month,
```

```
COUNT>Returns.ReturnID) AS Total_Returns,
```

```
CONCAT('€ ', FORMAT(SUM>Returns.AmountRet), 2)) AS Amount_Returned
```

```
FROM Returns
```

```
GROUP BY MONTH>Returns.ReturnDate) WITH ROLLUP;
```

```
END//
```

```
# Calling the Procedure
```

```
CALL TotalReturns
```

Output:

Month	Total_Returns	Amount_Returned
February	1	€ 241.33
June	1	€ 94.14
July	2	€ 2,713.25
August	2	€ 360.09
September	2	€ 538.16
October	1	€ 118.76
December	1	€ 241.33
December	10	€ 4,307.06

8 - [Stored Procedure] Display a specific customer details and all of their relevant orders to date by passing a parameter (eg: CustomerID):

```
DELIMITER //
CREATE PROCEDURE CustomerInfo (IN ID VARCHAR(5))
BEGIN
SELECT
Customer.CustomerID,
CONCAT(Customer.FirstName, ' ', Customer.LastName) AS Name,
CONCAT(Customer.Address, ', ', Customer.City, ', ', Customer.County, ', ', Customer.Country) AS Address,
Customer.Phone AS Phone,
Customer.Email AS Email,
Basket.OrderID,
Basket.OrderDate AS Date
FROM Customer
LEFT JOIN Basket ON Customer.CustomerID = Basket.CustomerID
WHERE Customer.CustomerID = ID
ORDER BY Basket.OrderDate;
END//

# Calling the Procedure
CALL CustomerInfo ('C1')
```

Output:

CustomerID	Name	Address	Phone	Email	OrderID	Date
C1	Whoopi Goldberg	4th Avenue, Brooklyn, New York, USA	+1 9 91234 5678	whooooopi@gmail.com	O1	2019-01-06
C1	Whoopi Goldberg	4th Avenue, Brooklyn, New York, USA	+1 9 91234 5678	whooooopi@gmail.com	O7	2019-02-11
C1	Whoopi Goldberg	4th Avenue, Brooklyn, New York, USA	+1 9 91234 5678	whooooopi@gmail.com	O13	2019-03-12
C1	Whoopi Goldberg	4th Avenue, Brooklyn, New York, USA	+1 9 91234 5678	whooooopi@gmail.com	O36	2019-08-30
C1	Whoopi Goldberg	4th Avenue, Brooklyn, New York, USA	+1 9 91234 5678	whooooopi@gmail.com	O40	2019-10-01

9 - [Trigger] Create a Trigger that will populate a 'history table' once a customer contact details have been updated:

First, we need to create a table for backup. It was named HistoryCustomer. Code below:

```
CREATE TABLE HistoryCustomer
(
  CustomerID varchar (10),
  FirstName varchar (50),
  LastName varchar (50),
  Address varchar (255),
  City varchar (60),
  County varchar (60),
  Country varchar (60),
  Phone varchar (20),
  Email varchar (50),
  PRIMARY KEY (CustomerID)
);
```

Trigger:

```
DELIMITER //
CREATE TRIGGER History_trigger
AFTER UPDATE ON Customer
FOR EACH ROW
BEGIN
  INSERT INTO HistoryCustomer(CustomerID,FirstName,LastName,Address,City,County,Country,Phone,Email)
  VALUES (OLD.CustomerID,OLD.FirstName,OLD.LastName,OLD.Address,OLD.City,OLD.County,OLD.Country,OLD.Phone,OLD.Email);
END//
DELIMITER ;

# Update Test for Trigger Created on Question 9
UPDATE Customer SET FirstName='Carlos' WHERE CustomerID='C2';
```

Output Customer table before update:

CustomerID	FirstName	LastName	Address	City	County	Country	Phone	Email
C1	Whoopi	Goldberg	4th Avenue	Brooklyn	New York	USA	+1 9 91234 5678	whoooopi@gmail.com
C10	Arnold	Schwarzenegger	2019 Future Street	Austin	Texas	USA	+1 1 6969 6969	astalavista@baby.com
C2	Robert	Plant	87 Singers Avenue	Los Angeles	California	USA	+1 4 1234 9876	blackdog@hotmail.com
C3	David	Gilmour	11 High Hopes Street	Austin	Texas	USA	+1 9 4432 0690	heyou@gmail.com
C4	Freddie	Mercury	666 Avenida de Los Muertos	Ciudad de Mexico	Ciudad de Mexico	Mexico	+52 91221 1111	mamaaa@hotmail.com.mx
C5	James	Bond	Brosnan Square	Navan	Co Meath	Ireland	+353 089 222 3333	bondjamesbond@yahoo.ie
C6	George	Washington	The White House	Washington DC	Washington	USA	+1 2 3456 1234	georgie@whitehouse.com
C7	Wilson	Pilsen	1 Beer Avenue	Wicklow	Leinster	Ireland	+353 81 211 2211	drinkbeer@gmail.com
C8	Nilson	Ale	2 Drinkers Boulevard	Dublin	Leinster	Ireland	+353 89 222 9845	aleisbetter@gmail.com
C9	Milton	Weissbier	3 Holygrail Avenue	Dublin	Leinster	Ireland	+353 81 221 0002	weissthebest@gmail.com

Output Customer table after update:

CustomerID	FirstName	LastName	Address	City	County	Country	Phone	Email
C1	Whoopi	Goldberg	4th Avenue	Brooklyn	New York	USA	+1 9 91234 5678	whoopooopi@gmail.com
C10	Arnold	Schwarzenegger	2019 Future Street	Austin	Texas	USA	+1 1 6969 6969	astalavista@baby.com
C2	Carlos	Plant	87 Singers Avenue	Los Angeles	California	USA	+1 4 1234 9876	blackdog@hotmail.com
C3	David	Gilmour	11 High Hopes Street	Austin	Texas	USA	+1 9 4432 0690	heyoud@gmail.com
C4	Freddie	Mercury	666 Avenida de Los Muertos	Ciudad de Mexico	Ciudad de Mexico	Mexico	+52 91221 1111	mamaaa@hotmail.com.mx
C5	James	Bond	Brosnan Square	Navan	Co Meath	Ireland	+353 089 222 3333	bondjamesbond@yahoo.ie
C6	George	Washington	The White House	Washington DC	Washington	USA	+1 2 3456 1234	georgie@whitehouse.com
C7	Wilson	Pilsen	1 Beer Avenue	Wicklow	Leinster	Ireland	+353 81 211 2211	drinkbeer@gmail.com
C8	Nilson	Ale	2 Drinkers Boulevard	Dublin	Leinster	Ireland	+353 89 222 9845	aleisbetter@gmail.com
C9	Milton	Weissbier	3 Holygrail Avenue	Dublin	Leinster	Ireland	+353 81 221 0002	weissthebest@gmail.com

New entry input by trigger on the HistoryCustomer table for backup:

CustomerID	FirstName	LastName	Address	City	County	Country	Phone	Email
C2	Robert	Plant	87 Singers Avenue	Los Angeles	California	USA	+1 4 1234 9876	blackdog@hotmail.com

10 - [View] Create a View that will display a breakdown of (a) sales (b) profit and (c) returns for each month of the year:

```
CREATE VIEW Store_Breakdown AS
SELECT
MONTHNAME(Basket.OrderDate) AS Months,
#COUNT(Basket.OrderID) AS Total_Orders,
CONCAT('€ ', FORMAT(SUM(OrderItems.ItemPrice), 2)) AS Gross_Income_No_VAT,
CONCAT('€ ', FORMAT(SUM(OrderItems.ItemPrice * 1.23), 2)) AS Gross_Income_VAT,
CONCAT('€ ', FORMAT(SUM(OrderItems.ItemPrice * 0.20), 2)) AS Net_Income,
COUNT>Returns.ReturnID) AS Total_Returns,
CONCAT('€ ', FORMAT(SUM>Returns.AmountRet), 2)) AS Amount_Returned
FROM OrderItems
JOIN Basket ON Basket.OrderID = OrderItems.OrderID
LEFT JOIN>Returns ON OrderItems.ItemID =>Returns.ItemID
GROUP BY MONTH(Basket.OrderDate) WITH ROLLUP;
```

Output:

Months	Gross_Income_No_VAT	Gross_Income_VAT	Net_Income	Total_Returns	Amount_Returned
January	€ 2,372.03	€ 2,917.60	€ 474.41	1	€ 241.33
February	€ 2,400.06	€ 2,952.07	€ 480.01	0	NULL
March	€ 463.78	€ 570.45	€ 92.76	1	€ 241.33
April	€ 2,521.13	€ 3,100.99	€ 504.23	0	NULL
May	€ 299.55	€ 368.45	€ 59.91	0	NULL
June	€ 2,497.66	€ 3,072.12	€ 499.53	1	€ 94.14
July	€ 4,174.87	€ 5,135.09	€ 834.97	1	€ 1,980.25
August	€ 2,056.49	€ 2,529.48	€ 411.30	1	€ 733.00
September	€ 1,450.93	€ 1,784.64	€ 290.19	2	€ 360.09
October	€ 1,311.16	€ 1,612.73	€ 262.23	2	€ 538.16
November	€ 680.74	€ 837.31	€ 136.15	0	NULL
December	€ 595.37	€ 732.31	€ 119.07	1	€ 118.76
December	€ 20,823.77	€ 25,613.24	€ 4,164.75	10	€ 4,307.06

11 – [Database] Add all of the above scripts into one unique SQL file that will run all of the scripts together and output the data into a DataMart:

```
Create Database DataMart;
USE DataMart;
Create Table DataMart.All_Stock_Data SELECT * FROM WatchShop.All_Stock;
Create Table DataMart.Sales_Growth_Data SELECT * FROM WatchShop.Sales_Growth;
Create Table DataMart.Store_Breakdown_Data SELECT * FROM WatchShop.Store_Breakdown;
Create Table DataMart.Transactions_August_Week_2_Data SELECT * FROM WatchShop.Transactions_August_Week_2;

# Testing Tables:
SELECT * FROM datamart.all_stock_data;
SELECT * FROM datamart.sales_growth_data;
SELECT * FROM datamart.store_breakdown_data;
SELECT * FROM datamart.transactions_august_week_2_data;
```

Outputs:

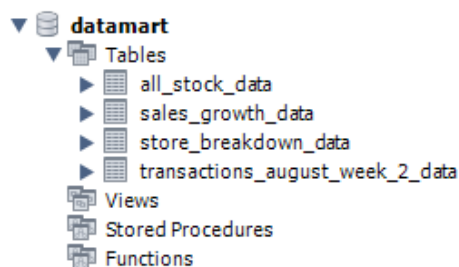
SupName	StockID	ItemBrand	ItemModel	SupplierID	QtyStock
DimSum China Watches LTD	ST1	Casio	F-91W	SU1	18
David Hamilton Watch Factory	ST10	Seiko	Prospex SNE499P1	SU3	20
Its Pork O'Clock! Handmade Watches	ST11	Citizen	Skyhawk A-T Red Arrows	SU5	20
Its Pork O'Clock! Handmade Watches	ST12	Citizen	Eco-Drive AT4000-02E	SU5	20
Its Pork O'Clock! Handmade Watches	ST13	Citizen	Diver BJ8050-08E	SU5	20
What Time is It? Corporation	ST14	Vostok	Komandirskie 2416B	SU4	20
What Time is It? Corporation	ST15	Vostok	Amphibia 120812	SU4	20
David Hamilton Watch Factory	ST2	Seiko	Big Orange SRPC55	SU3	20
DimSum China Watches LTD	ST3	Casio	G-Shock GA2100	SU1	20
NCI Watches Corporation	ST4	Omega	Speedmaster	SU2	20
What Time is It? Corporation	ST5	Vostok	Amphibia Scuba Dude 710059	SU4	20
DimSum China Watches LTD	ST6	Casio	Edifice EF-527D-1AVEF	SU1	20
DimSum China Watches LTD	ST7	Casio	Edifice EF-566DB	SU1	20
DimSum China Watches LTD	ST8	Casio	Pro-Trek PRW-3100YB	SU1	20
David Hamilton Watch Factory	ST9	Seiko	SSB313P1	SU3	20

Months	Total_Orders	Income	Previous_Month	Growth
January	6	€ 2,372.03	NULL	NULL
February	8	€ 2,400.06	€ 2372.03	6.6 %
March	4	€ 463.78	€ 2400.06	-74.3 %
April	5	€ 2,521.13	€ 463.78	286.6 %
May	3	€ 299.55	€ 2521.13	-84.7 %
June	6	€ 2,497.66	€ 299.55	546.7 %
July	5	€ 4,174.87	€ 2497.66	67.2 %
August	6	€ 2,056.49	€ 4174.87	-45.9 %
September	4	€ 1,450.93	€ 2056.49	-35.8 %
October	6	€ 1,311.16	€ 1450.93	2.8 %
November	3	€ 680.74	€ 1311.16	-54.4 %
December	3	€ 595.37	€ 680.74	4.9 %

Months	Gross_Income_No_VAT	Gross_Income_VAT	Net_Income	Total_Returns	Amount_Returned
January	€ 2,372.03	€ 2,917.60	€ 474.41	1	€ 241.33
February	€ 2,400.06	€ 2,952.07	€ 480.01	0	NULL
March	€ 463.78	€ 570.45	€ 92.76	1	€ 241.33
April	€ 2,521.13	€ 3,100.99	€ 504.23	0	NULL
May	€ 299.55	€ 368.45	€ 59.91	0	NULL
June	€ 2,497.66	€ 3,072.12	€ 499.53	1	€ 94.14
July	€ 4,174.87	€ 5,135.09	€ 834.97	1	€ 1,980.25
August	€ 2,056.49	€ 2,529.48	€ 411.30	1	€ 733.00
September	€ 1,450.93	€ 1,784.64	€ 290.19	2	€ 360.09
October	€ 1,311.16	€ 1,612.73	€ 262.23	2	€ 538.16
November	€ 680.74	€ 837.31	€ 136.15	0	NULL
December	€ 595.37	€ 732.31	€ 119.07	1	€ 118.76
December	€ 20,823.77	€ 25,613.24	€ 4,164.75	10	€ 4,307.06

OrderID	Date	Item	ItemPrice	PaymentType	Customer
O32	August 10 - 2019	Casio G-Shock GA2100	86.65	Debit Card	Carlos Plant
O33	August 11 - 2019	Citizen Diver BJ8050-08E	204.13	Credit Card	Freddie Mercury
O34	August 13 - 2019	Casio Pro-Trek PRW-3100YB	241.33	Debit Card	Carlos Plant

Above, an image of the DataMart database with the created tables:



Following, an image of the WatchShop database with all the created tables, views and procedures:

