NATIONAL COLLEGE OF IRELAND

HIGHER DIPLOMA IN SCIENCE IN COMPUTING

(HDCSDEV_INT)

Introduction to Databases

Fiama Marques de Souza Viana – 20121181

DUBLIN - IRELAND

2020

Online Shoes Sale Retail

E-commerce sales have greatly expanded over the past 5 years, there is no doubt that this new sales method offers many benefits to its customers such as the ease of buying anything anywhere besides the quick price comparison between several stories in a short time. In addition, online shopping is expected to increase even more in the coming years as they become more secure and convenient for customers.

Like many retail sectors, the Online Shoe Sales Retail has considerably flourished, and it is one of the categories in which customers are most likely to purchase online. People would feel more comfortable buying shoes online instead of clothes, due to the uniform size of the shoes. Although online shoe retail is responsible for most of the sales industry growth, there are many online shoe sellers that make it more difficult for a new company to stand out in front of so many competitors.

Nowadays there are many challenges to set off a new business in this field, the most important of which is to win Customer Loyalty. What will guarantee the endurance of a company is how satisfied the customer is not only with the products but mainly with the customer service. Also, customers expect a great user experience, which means that the navigability of the system should be seamless.

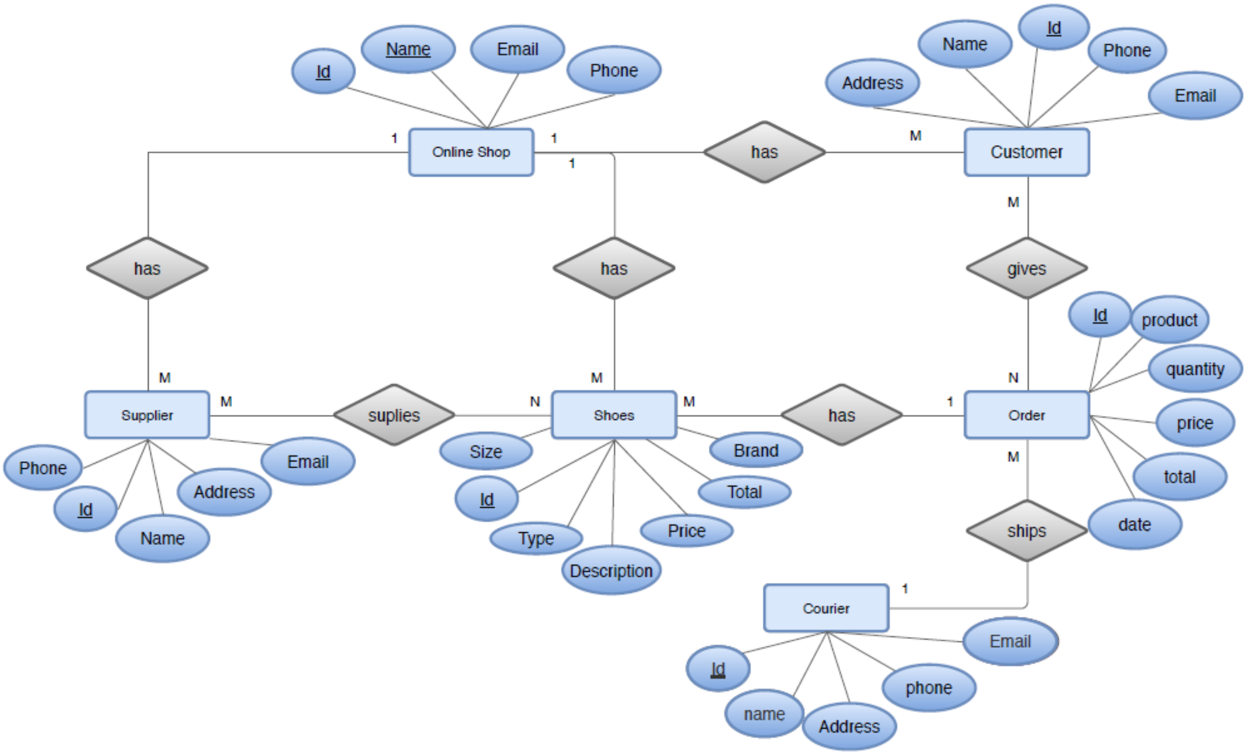
There are many other challenges that a retailer will face such as delivery, shipping, and return. Buyers want to try on their new shoes as soon as possible and they hate to face difficulties when need to return a shoe. Moreover, the website should provide all details about its products, fees, and additional charges. Also, the brand and about us, and cybersecurity must be considered, as people will feel more confident to buy and become a loyal customer.

We are living in a new era and people no longer want to be in a never-ending aisle to buy new shoes, they want the best products and services of excellence. In order to become a leader in online shoe sales retail, it is expected to overcome all these challenges and offer the best possible experience.

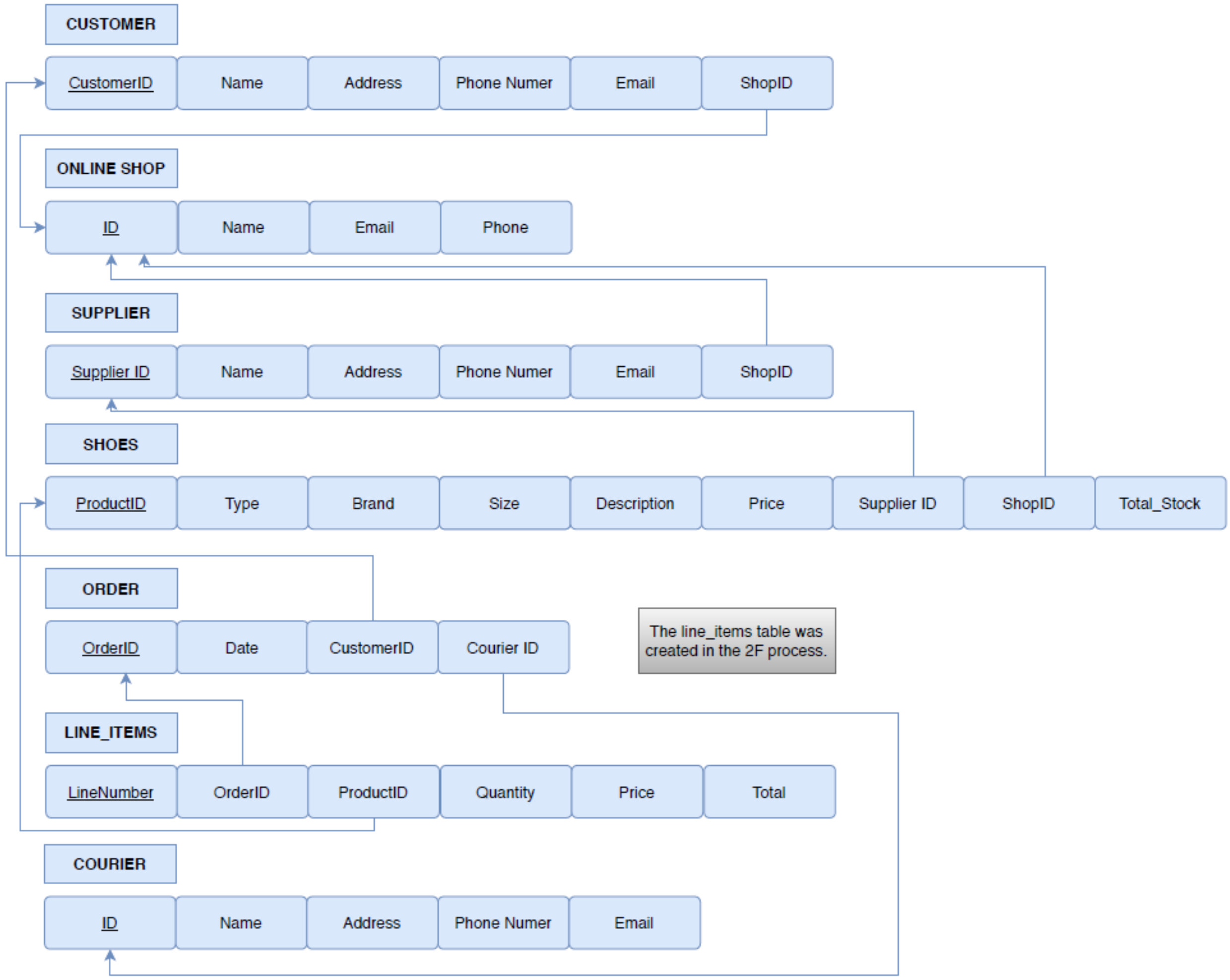
Entities	Attributes
Customer type: strong	CustomerID (PK) Name Address Phone Email
Shoe Online Shop type: strong	ShopID(PK) Name Email Phone
Supplier type: strong	SupplierID (PK) Name Address Phone Email
Shoes type: strong	ProductID (PK) Type Brand Size Description Price Total Stock
Order type: strong	OrderId (PK) Date Product Quantity Total Price
Courier type: strong	CourierID (PK) Name Phone Address Email

Relationships	Cardinality	Constraints
Online Shop has shoes	One-To-Many	Partial
Online Shop has suppliers	One-To-Many	Partial
Online Shop has customers	One-To-Many	Partial
Supplier supplies shoes	Many-To-Many	Partial
Customers gives orders	Many-To-Many	Partial
Orders has products	One-To-Many	Partial
Courier ships orders	One-To-Many	Total

ER Diagram


Id
Name
Email
Phone
Online Shop
1
1
1
has
M
Customer
Address
Name
Id
Phone
Email
M
has
has
gives
M
M
M
Supplier
suplies
N
Shoes
M
has
1
Order
N
Id
product
quantity
price
total
date
M
ships
Phone
Id
Name
Address
Email
Size
Id
Type
Description
Price
Total
Brand
1
Courier
Email
phone
Address
name
Id

Normalization – Third Normal Form



CUSTOMER

CustomerID	Name	Address	Phone Numer	Email	ShopID

ONLINE SHOP

ID	Name	Email	Phone

SUPPLIER

Supplier ID	Name	Address	Phone Numer	Email	ShopID

SHOES

ProductID	Type	Brand	Size	Description	Price	Supplier ID	ShopID	Total_Stock

ORDER

OrderID	Date	CustomerID	Courier ID

The line_items table was created in the 2F process.

LINE_ITEMS

LineNumber	OrderID	ProductID	Quantity	Price	Total

COURIER

ID	Name	Address	Phone Numer	Email

Part 2: Logical and physical design

1. Create the corresponding database using DDL.

```
CREATE DATABASE ONLINE_SHOES_SALE;
USE ONLINE_SHOES_SALE;
```

2. Create all the necessary tables identified above using DDL.

```
CREATE TABLE ONLINESHOP(
    shopId INTEGER NOT NULL UNIQUE,
    shopName VARCHAR(20),
    email VARCHAR(20),
    phone VARCHAR(15),
    PRIMARY KEY(shopId)
);
ALTER TABLE ONLINESHOP MODIFY email VARCHAR(255);
ALTER TABLE ONLINESHOP MODIFY phone VARCHAR(20);

CREATE TABLE CUSTOMER(
    customer_id INTEGER NOT NULL UNIQUE,
    customer_name VARCHAR(30),
    address VARCHAR(60),
    phone VARCHAR(15),
    email VARCHAR(20),
    shopId INTEGER,
    PRIMARY KEY(customer_id),
    FOREIGN KEY (shopId)
          REFERENCES ONLINESHOP(shopId)
);

CREATE TABLE SUPPLIER(
    supplier_id INTEGER NOT NULL UNIQUE,
    supplier_name VARCHAR(30),
    address VARCHAR(60),
    phone VARCHAR(15),
    email VARCHAR(20),
    shopId INTEGER,
    PRIMARY KEY(supplier_id),
    FOREIGN KEY(shopId)
          REFERENCES ONLINESHOP(shopID)
);
ALTER TABLE SUPPLIER MODIFY email VARCHAR(255);
```

```
CREATE TABLE SHOES(
   product_id INTEGER NOT NULL UNIQUE,
   shoes_type VARCHAR(10),
   brand VARCHAR(10),
   size INTEGER,
   shoes_description VARCHAR(255),
   price DECIMAL,
   supplier_id INTEGER,
   shopId INTEGER,
   total_stock INTEGER,
   PRIMARY KEY(product_id),
   FOREIGN KEY(supplier_id)
        REFERENCES SUPPLIER(supplier_id),
   FOREIGN KEY(shopId)
        REFERENCES ONLINESHOP(shopID)
);

CREATE TABLE COURIER(
   courier_id INTEGER NOT NULL UNIQUE,
   courier_name VARCHAR(20),
   address VARCHAR(60),
   phone VARCHAR(15),
   email VARCHAR(20),
   PRIMARY KEY(courier_id)
);

CREATE TABLE SHOES_ORDER(
   order_id INTEGER NOT NULL UNIQUE,
   order_date date,
   customer_id INTEGER,
   courier_id INTEGER,
   PRIMARY KEY(order_id),
   FOREIGN KEY(customer_id) REFERENCES CUSTOMER(customer_id),
   FOREIGN KEY(courier_id) REFERENCES COURIER(courier_id)
);
CREATE TABLE LINE_ITEMS(
   line_number INTEGER NOT NULL,
   order_id INTEGER NOT NULL,
   product_id INTEGER,
   unit_price DECIMAL,
   quantity INTEGER,
   total DECIMAL,
   PRIMARY KEY(line_number),
   FOREIGN KEY(order_id) REFERENCES SHOES_ORDER(order_id),
   FOREIGN KEY(product_id) REFERENCES SHOES(product_id)
);
```

3. Populate at least three of your tables with some data using DML (insert into statement)

INSERT INTO ONLINESHOP(shopId,shopName,email,phone) VALUES(1,"THEBESTSHOES","thebestshoes@gmaiL","+353 083-222-3333");

INSERT INTO COURIER(courier_id,courier_name,address,phone,email) VALUES(1425,"UPS", "080 Warner Pass", "803-217-8593", "skienl9@illinois.edu");

INSERT INTO COURIER(courier_id,courier_name,address,phone,email) VALUES(1457,"AN POST", "O'connel Street", "(01) 705 7600", "care@anpostmobile.ie");

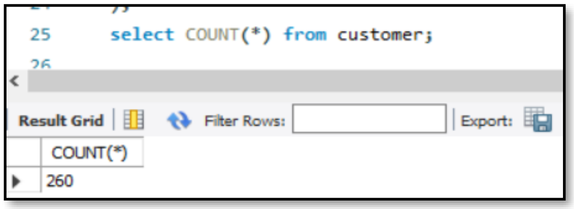
INSERT INTO COURIER(courier_id,courier_name,address,phone,email) VALUES(2105,"Fedex", "9 Elgar Road", "165-162-5948", "btidball2@oracle.com");

INSERT INTO SUPPLIER(supplier_id,supplier_name,address,phone,email,shopId) VALUES(1348,"NIKE","Westend Retail Park Unit 6, Blanchardstown, Dublin", "(01) 811 1140","nike@nike.com.ie",1);

INSERT INTO SUPPLIER(supplier_id,supplier_name,address,phone,email,shopId) VALUES(1349,"MYSHOES","North Retail Park Unit 8, Blanchardstown, Dublin", "(01) 111 444","myshoes@myshoes.com.ie",1);

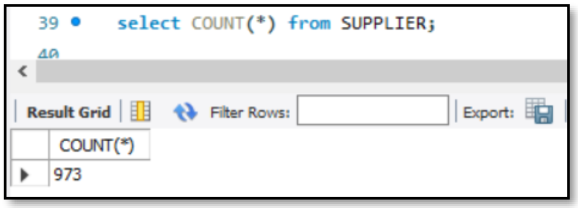
INSERT INTO SUPPLIER(supplier_id,supplier_name,address,phone,email,shopId) VALUES(1350,"THEBESTSHOES","City West, Dublin", "(01) 324 1248","thebest@thebest.com.ie",1);

4. Populate your database with a large data set representing a one-year transaction (01/01/2020 - 31/12/2020) on each table. (Use online data generators such as Mockaroo or generate data to generate synthetic data.)



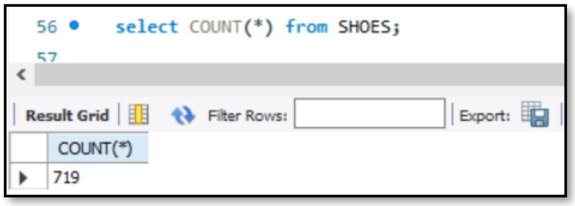
```
select COUNT(*) from customer;
```

```
COUNT(*)
260
```



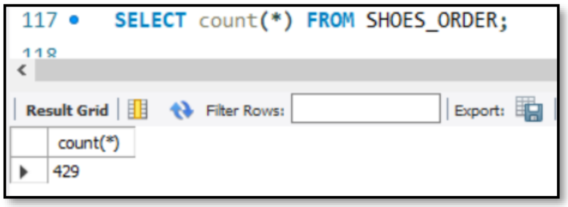
```
select COUNT(*) from SUPPLIER;
```

```
COUNT(*)
973
```



```
select COUNT(*) from SHOES;
```

```
COUNT(*)
719
```



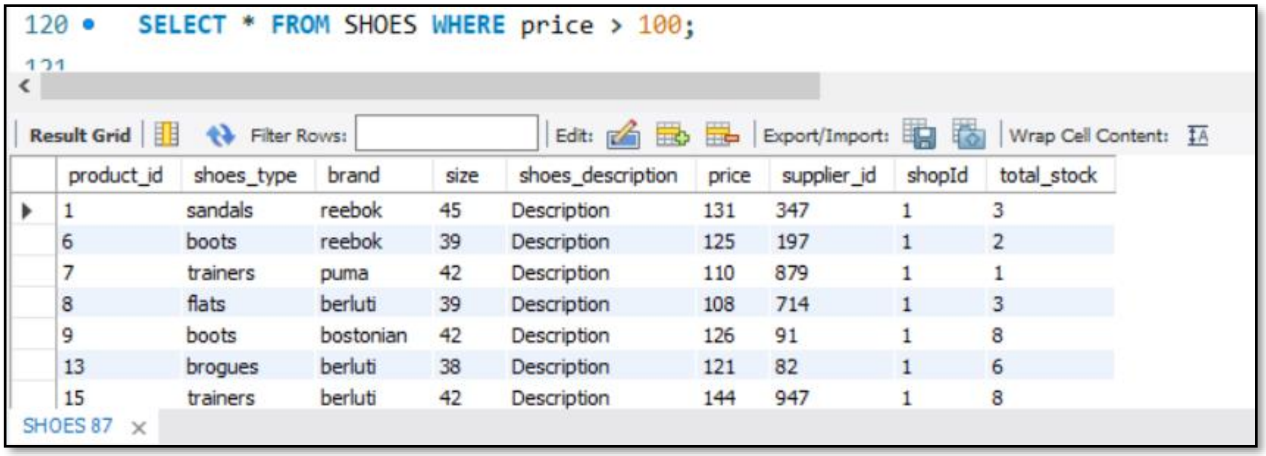
```
SELECT count(*) FROM SHOES_ORDER;
```

```
count(*)
429
```

Part 3: Write SQL Statements to answer the following queries - 40% (5 marks each)

1. Show all the details of the products that have a price greater than 100.

SELECT * FROM SHOES WHERE price > 100;

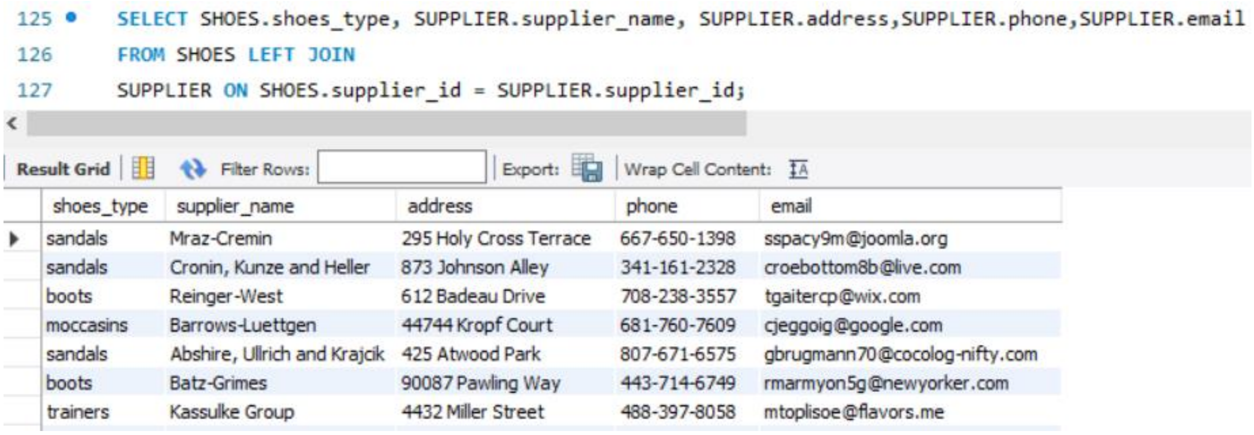


```
SELECT * FROM SHOES WHERE price > 100;
```

	product_id	shoes_type	brand	size	shoes_description	price	supplier_id	shopId	total_stock
▶	1	sandals	reebok	45	Description	131	347	1	3
	6	boots	reebok	39	Description	125	197	1	2
	7	trainers	puma	42	Description	110	879	1	1
	8	flats	berluti	39	Description	108	714	1	3
	9	boots	bostonian	42	Description	126	91	1	8
	13	brogues	berluti	38	Description	121	82	1	6
	15	trainers	berluti	42	Description	144	947	1	8

2. Show all the products along with the supplier detail who supplied the products.

SELECT SHOES.shoes_type, SUPPLIER.supplier_name,
SUPPLIER.address,SUPPLIER.phone,SUPPLIER.email
FROM SHOES LEFT JOIN
SUPPLIER ON SHOES.supplier_id = SUPPLIER.supplier_id;

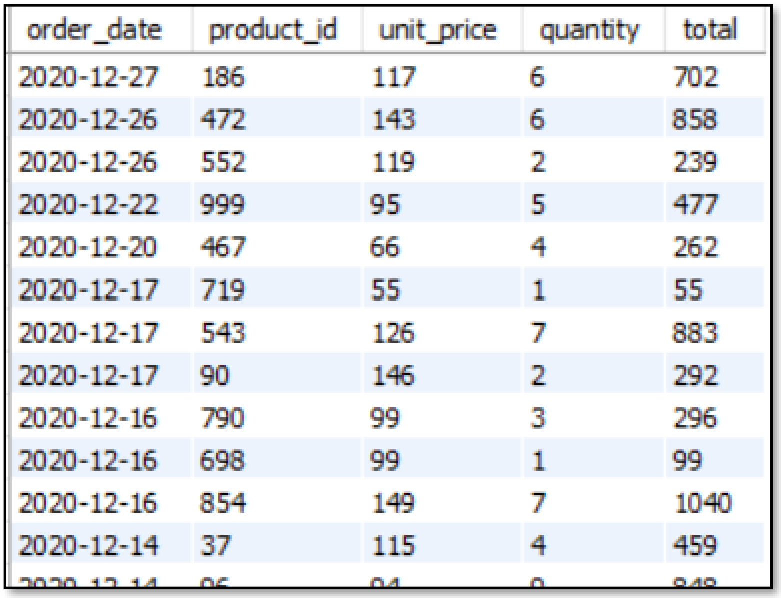


```
SELECT SHOES.shoes_type, SUPPLIER.supplier_name, SUPPLIER.address,SUPPLIER.phone,SUPPLIER.email
FROM SHOES LEFT JOIN
SUPPLIER ON SHOES.supplier_id = SUPPLIER.supplier_id;
```

	shoes_type	supplier_name	address	phone	email
▶	sandals	Mraz-Cremin	295 Holy Cross Terrace	667-650-1398	sspacy9m@joomla.org
	sandals	Cronin, Kunze and Heller	873 Johnson Alley	341-161-2328	croebottom8b@live.com
	boots	Reinger-West	612 Badeau Drive	708-238-3557	tgaitercp@wix.com
	moccasins	Barrows-Luettgen	44744 Kropf Court	681-760-7609	cjeggoig@google.com
	sandals	Abshire, Ullrich and Krajcik	425 Atwood Park	807-671-6575	gbrugmann70@cocolog-nifty.com
	boots	Batz-Grimes	90087 Pawling Way	443-714-6749	rmarmyon5g@newyorker.com
	trainers	Kassulke Group	4432 Miller Street	488-397-8058	mtoplisoe@flavors.me

3.	Create a stored procedure that takes the start and end dates of the sales and display all the sales transactions between the start and the end dates.

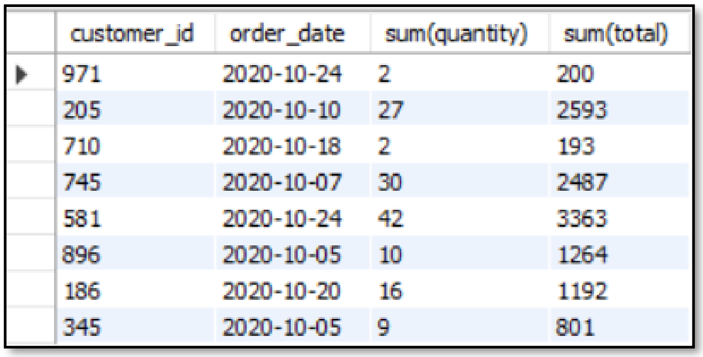
```
DELIMITER //

CREATE PROCEDURE GetSales()
BEGIN
   SELECT shoes_order.order_date, product_id,unit_price,quantity,total
   FROM line_items
   LEFT JOIN shoes_order
   ON shoes_order.order_id = line_items.order_id
   WHERE order_date >= '2020-01-01'
   AND order_date <= '2020-12-31' ORDER BY order_date DESC;
END //

DELIMITER ;

CALL GetSales();
```



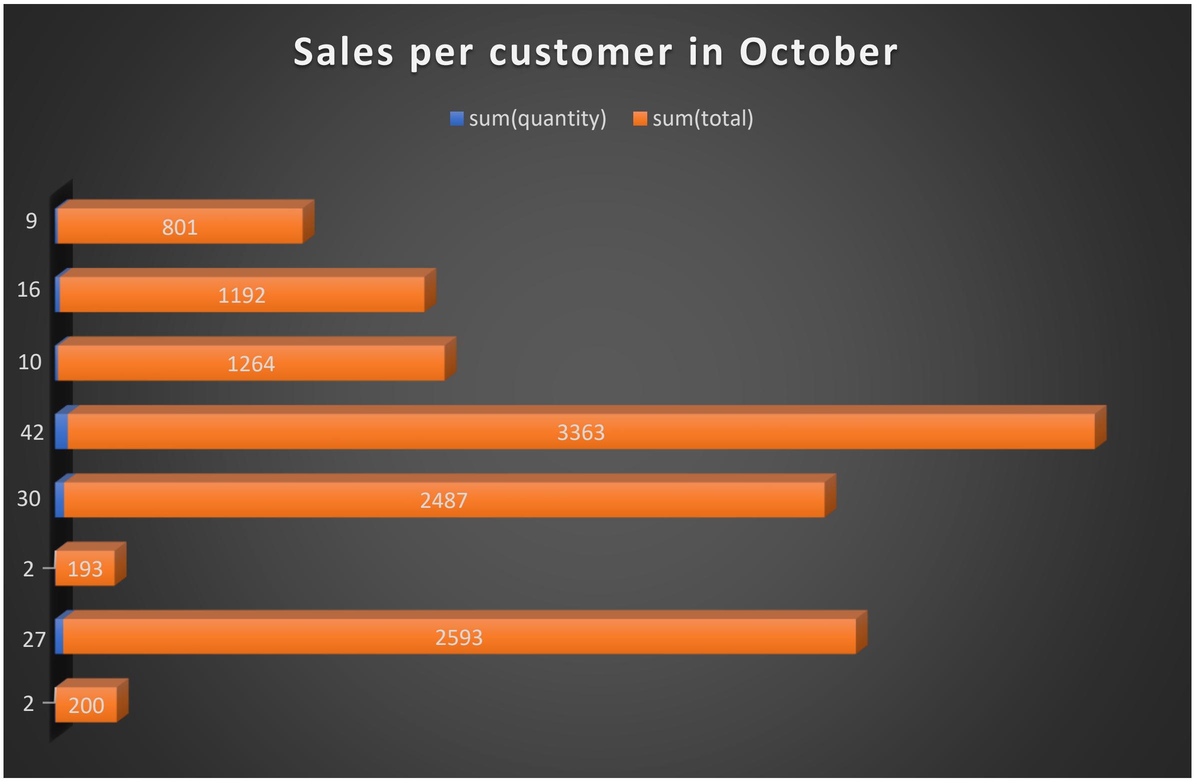
order_date	product_id	unit_price	quantity	total
2020-12-27	186	117	6	702
2020-12-26	472	143	6	858
2020-12-26	552	119	2	239
2020-12-22	999	95	5	477
2020-12-20	467	66	4	262
2020-12-17	719	55	1	55
2020-12-17	543	126	7	883
2020-12-17	90	146	2	292
2020-12-16	790	99	3	296
2020-12-16	698	99	1	99
2020-12-16	854	149	7	1040
2020-12-14	37	115	4	459
2020-12-14	96	94	9	848

4.	Create a view that shows the total number of items a customer buys from the business in October 2020 along with the total price (use group by)

```
CREATE VIEW sales_per_customer_october AS
   select shoes_order.customer_id, shoes_order.order_date, sum(quantity), sum(total)
   from line_items
   left join shoes_order on line_items.order_id = shoes_order.order_id
   where order_date >= '2020-10-01' and order_date <= '2020-10-30'
   group by quantity;
SELECT * FROM sales_per_customer_october;
```



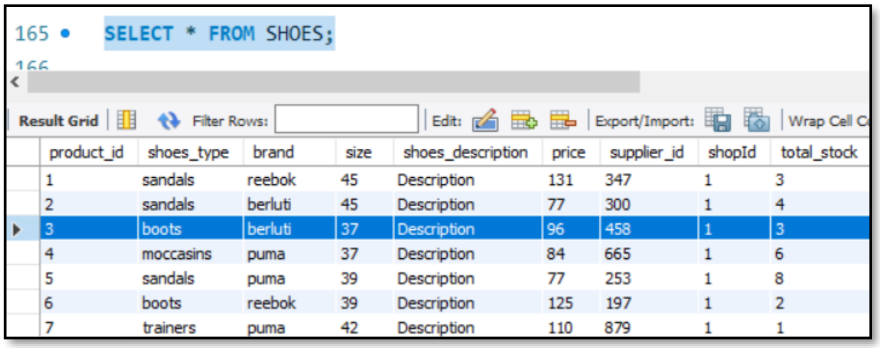
	customer_id	order_date	sum(quantity)	sum(total)
▶	971	2020-10-24	2	200
	205	2020-10-10	27	2593
	710	2020-10-18	2	193
	745	2020-10-07	30	2487
	581	2020-10-24	42	3363
	896	2020-10-05	10	1264
	186	2020-10-20	16	1192
	345	2020-10-05	9	801



Graphic created by using Excel as a reporting tool

5. Create a trigger that adjusts the stock level every time a product is sold.

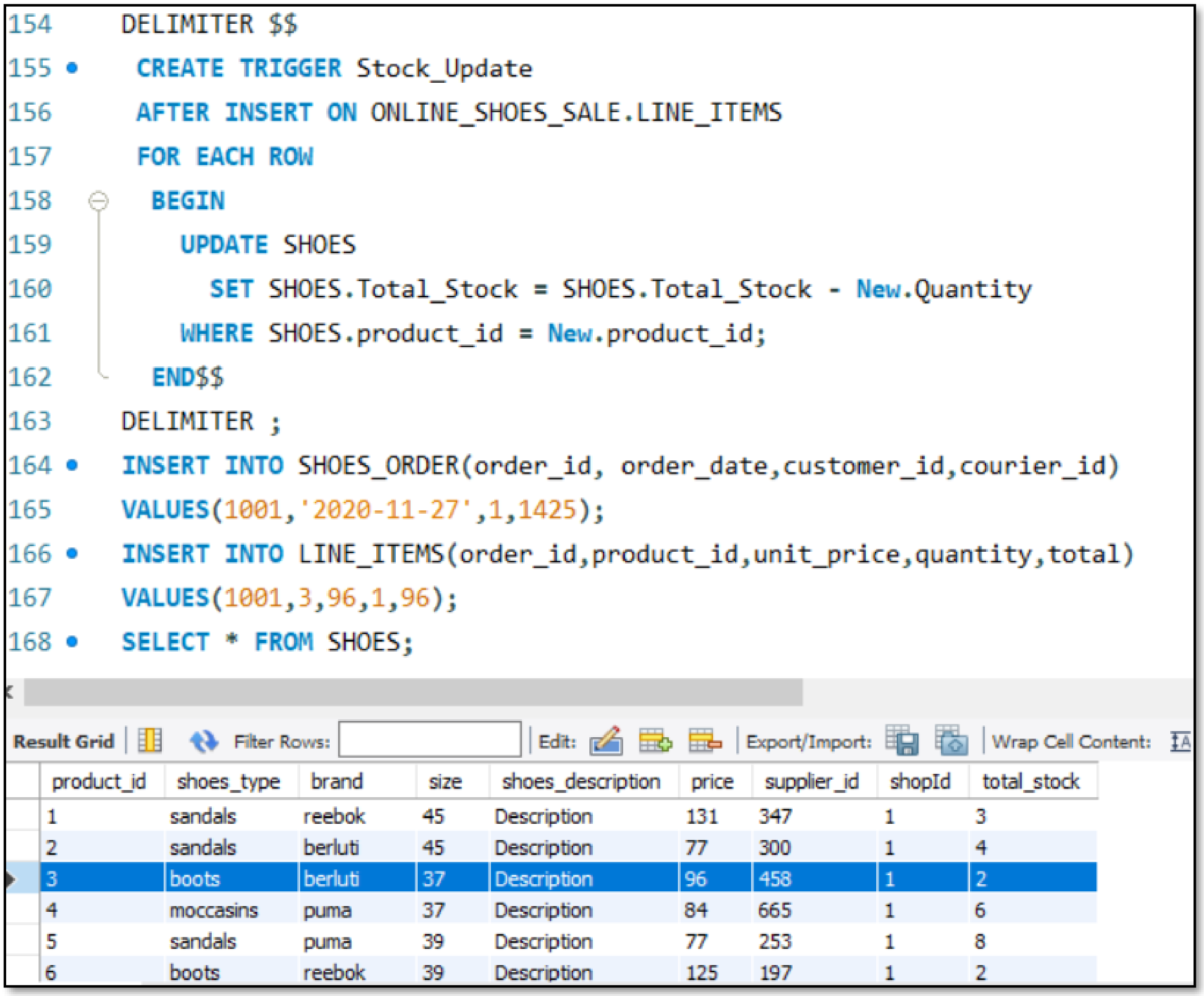
```
DELIMITER $$
CREATE TRIGGER Stock_Update
AFTER INSERT ON ONLINE_SHOES_SALE.LINE_ITEMS
FOR EACH ROW
BEGIN
        UPDATE SHOES
        SET SHOES.Total_Stock = SHOES.Total_Stock - New.Quantity
        WHERE SHOES.product_id = New.product_id;
END$$
DELIMITER ;
```



```
SELECT * FROM SHOES;
```

product_id	shoes_type	brand	size	shoes_description	price	supplier_id	shopId	total_stock
1	sandals	reebok	45	Description	131	347	1	3
2	sandals	berluti	45	Description	77	300	1	4
3	boots	berluti	37	Description	96	458	1	3
4	moccasins	puma	37	Description	84	665	1	6
5	sandals	puma	39	Description	77	253	1	8
6	boots	reebok	39	Description	125	197	1	2
7	trainers	puma	42	Description	110	879	1	1

There were 3 items of boots Berluti.



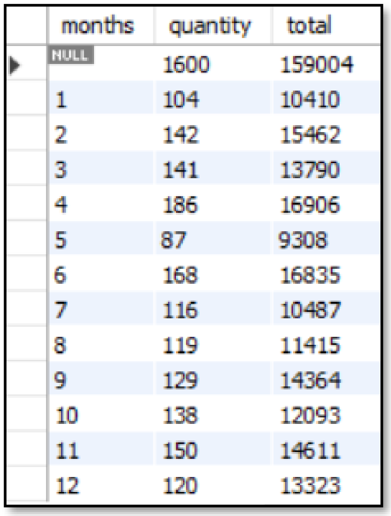
```
DELIMITER $$
 CREATE TRIGGER Stock_Update
 AFTER INSERT ON ONLINE_SHOES_SALE.LINE_ITEMS
 FOR EACH ROW
  BEGIN
    UPDATE SHOES
      SET SHOES.Total_Stock = SHOES.Total_Stock - New.Quantity
    WHERE SHOES.product_id = New.product_id;
  END$$
DELIMITER ;
INSERT INTO SHOES_ORDER(order_id, order_date,customer_id,courier_id)
VALUES(1001,'2020-11-27',1,1425);
INSERT INTO LINE_ITEMS(order_id,product_id,unit_price,quantity,total)
VALUES(1001,3,96,1,96);
SELECT * FROM SHOES;
```

product_id	shoes_type	brand	size	shoes_description	price	supplier_id	shopId	total_stock
1	sandals	reebok	45	Description	131	347	1	3
2	sandals	berluti	45	Description	77	300	1	4
3	boots	berluti	37	Description	96	458	1	2
4	moccasins	puma	37	Description	84	665	1	6
5	sandals	puma	39	Description	77	253	1	8
6	boots	reebok	39	Description	125	197	1	2

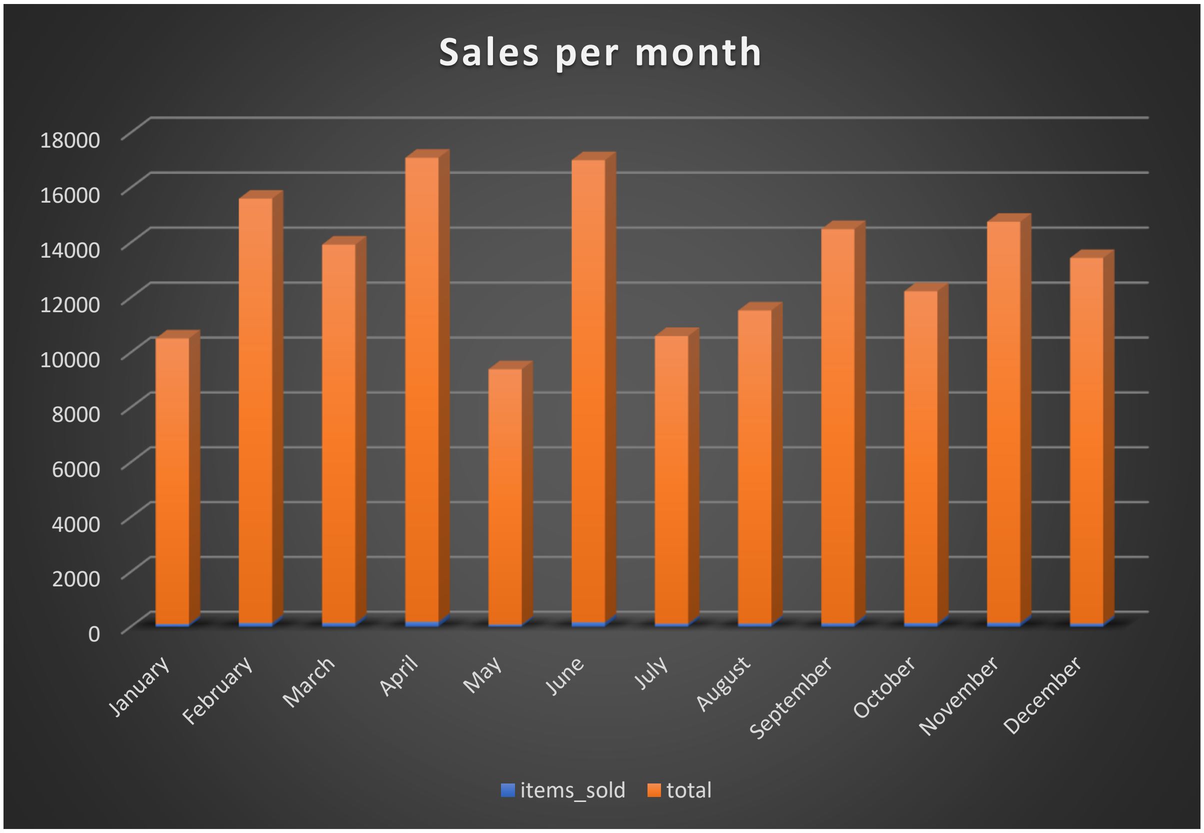
After the trigger applied and a sale was made there were 2 boots remaining in the stock.

6. Create a report of the annual sales (2020) of the business showing the total number of products sold and the total price sold every month (use A group by with roll-up).

```
CREATE VIEW sales_per_month AS
        SELECT MONTH(shoes_order.order_date) as
        months,sum(quantity) as quantity,sum(total) as
        total
        FROM LINE_ITEMS
        INNER JOIN SHOES_ORDER
        ON shoes_order.order_id = line_items.order_id
        GROUP BY MONTH(order_date) with rollup
        ORDER BY MONTH(order_date) ASC;

SELECT * FROM sales_per_month;
```



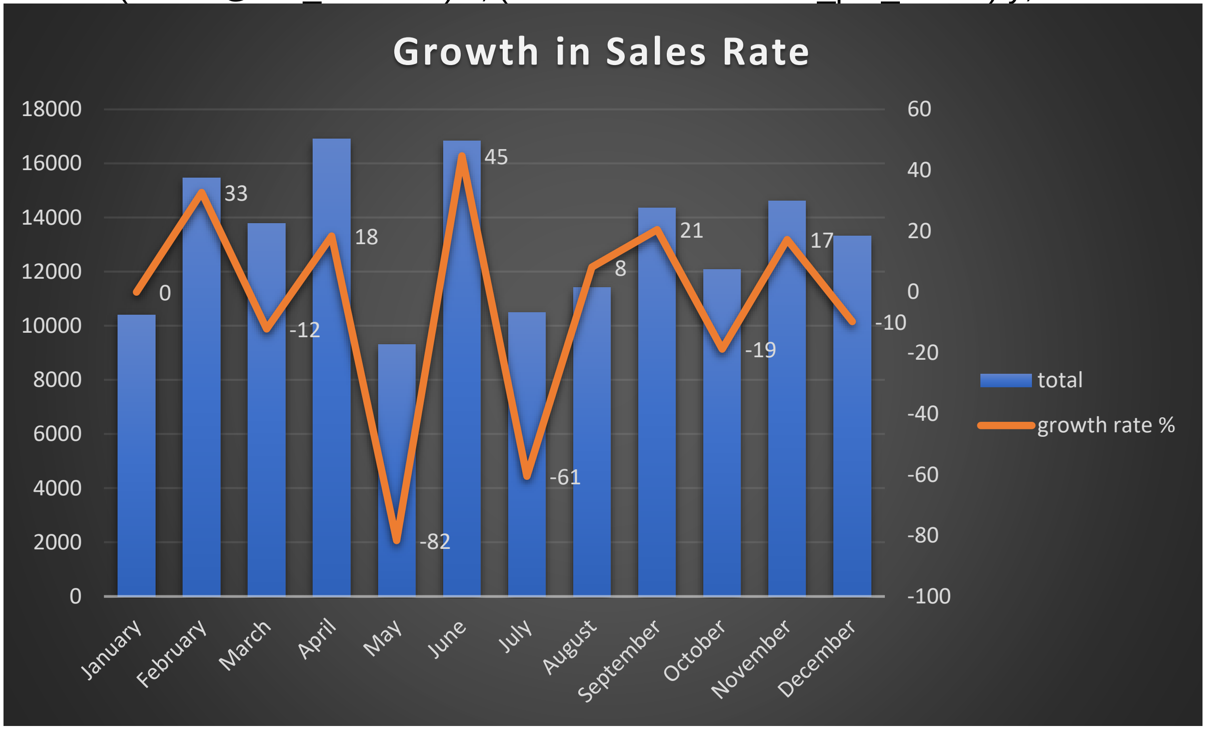
months	quantity	total
NULL	1600	159004
1	104	10410
2	142	15462
3	141	13790
4	186	16906
5	87	9308
6	168	16835
7	116	10487
8	119	11415
9	129	14364
10	138	12093
11	150	14611
12	120	13323



Graphic created by using Excel as a reporting tool

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

```
SELECT total,
if (@last_total = 0, 0, ((total - @last_total) / total) * 100) "growth rate %", @last_total := total
from
        (select @last_total := 0) x, (select total from sales_per_month) y;
```



Graphic created by using Excel as a reporting tool

8. Delete all customers who never buy a product from the business.

```
SET SQL_SAFE_UPDATES = 1;

DELETE FROM customer
WHERE NOT EXISTS (
   SELECT *
   FROM shoes_order
   WHERE customer_id = customer.customer_id
)
```

REFERENCES

www.ibisworld.com. (n.d.). E-Commerce vs. Brick and Mortar: Shoe Industry | IBISWorld Industry Insider. [online] Available at: https://www.ibisworld.com/industry-insider/analyst-insights/e-commerce-vs-brick-and-mortar-shoe-industry/ [Accessed 20 Nov. 2020].

Connect, A. (2018). 7 Challenges in online shoe retail. [online] Arema Connect. Available at: https://aremaconnect.com/2018/10/challenges-online-shoe-retail/ [Accessed 20 Nov. 2020].