Introduction to Databases CA
April 2023

Miguel Angel Vinas

Student Number: x22116133

INDEX

PART 1: Conceptual Design

1. Describe/introduce the chosen retail business and its objectives.

2. Identify the relevant entities of the business with their respective attributes, entity types and primary keys. Include all the core entities of the business.

3. Identify the relationships, cardinalities and participation constraints with supporting business rules and assumptions.

4. Draw an ER diagram for the system depicting the entities, relationships, cardinalities, participations using your preferred ERD notation.

5. Convert the entities in the conceptual design into a logical table. Use the following template to identify the tables. Make sure that the tables are in a 3rd normal form.

6. ER diagram with the system depicting the entities, relationships, cardinalities, participations after the 3rd Normal Form.

PART 2: Logical and physical design

1. Create the corresponding database using DDL.

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

3. Populate your tables with some data using DML (insert into statement).

4. Populate your database with DML insert statements OR using a large data set representing a sample one-year transaction (01/01/2023 - 31/12/2023) on each table, using online data generators such as Mockaroo or generate data to generate synthetic data.

PART 3: Write SQL Statements to answer the following queries

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

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

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.

4. Create a view that shows the total number of items a customer buys from the business in October 2023 along with the total price (use group by).
(SELECT * name of the view probably)

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

6. Create a query to view the annual sales over the last 12 months of the business showing the total number of products sold and the total price sold every month (use A group by with roll-up).

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

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

1. Describe/introduce the chosen retail business and its objectives.

My retail business is a family-owned bookshop called "Book Trek Shop" that is located in Lucan, County Dublin.

Even though it has different categories of products, such as books to read, books to study, office supplies and magazines we specialize in Science Fiction.

Within each category there are sub-categories.

An example of a category would be books to read and a sub-category would be Science Fiction or Period

We do not have any online presence and our inventory is done through an Excel file.

We have applied for the Trading Online Voucher Scheme from our Local Enterprise Office (Trading Online Voucher Scheme - Local Enterprise Office) because we want customers to have the choice of buying from their homes. Because of that, we need to build a database where all our information is stored in a secure way.

After doing our research we have identified that we need 7 tables and within each table there will be several attributes.

Because we want to differentiate the physical store from the digital store we are going to have one of the attributes as "storeID" and another one as "storeName" within the STORE entity, that will give us the ability to add new physical stores in the future when we expand.

We would like to have the less possible friction when looking up things in the database and as we will want to see how many purchases our customers make we will have an attribute called "customerPurchases" set to an int within our CUSTOMER entity.

This will give us the ability to filter by customers who purchase the most (or the less) in our database to send accurate marketing campaigns.

2. Identify the relevant entities of the business with their respective attributes, entity types and primary keys. Include all the core entities of the business.

An entity in a relational database is a real world object or concept that can be identified by a set of attributes or properties and that can be represented in the database.

Entities are typically represented in a database as a table, with each row in the table representing a specific instance of the entity.

Examples of entities can be a university course or a job or a customer.

Within the customer entity table we could have columns like customerID, name, email, etc, which would represent the attributes of the customer entity.

Entities in relational databases are related to each other through relationships, which define how data in one entity relates to data in another entity.

The entities for the "Book Trek" database are as follow:

1. Customer

- Attributes: customerID, customerName, customerSurname, customerAddress1, customerAddress2, customerEircode, customerPhoneNumber, customerEmail, customerAge, customerPurchases, customerBirthday

2. Product

- Attributes: productID, productName, productPrice, productSupplier, productStock, productCategory, productSubcategory1, productOnSale

By creating a productSubcategory1 we will be able to add more SubCategories in the future if needed.

3. Supplier

- Attributes: supplierID, supplierName, supplierAddress1, supplierAddress2, supplierPostcode, supplierPhoneNumber, supplierEmail.

4. Order

- Attributes: orderID, orderCustomerID, orderDate, orderDescription, orderTotalPrice

An order in the case of our business is a customer's request for purchasing something from us.

5. Store

- Attributes: storeID, storeName, storeStock, storeAddress1, storeAddress2, storeEircode, storeEmail, storePhoneNumber.

6. Sales

- Attributes: saleID, saleCustomerID, orderID, storeID

I have entered orderID and storeID within the SALE entity because it will help displaying the growth in sales by store. Although I think that I might not need it in the end.

A sale in the case of our business is the actual transaction / payment of the order that the customer has requested.

3. Identify the relationships, cardinalities and participation constraints with supporting business rules and assumptions.
All these relationships have been made before the entities being in 3rd Normal Form.

1. Store – Order: One store can have multiple orders, however, one order cannot be placed in different stores. Therefore, the relationship between the Store and the Order entities is one to many. The cardinality notation is 1:N.

2. Store – Sales: One store can have multiple sales. One sale can be placed only in one store at the time. The relationship between the Store and the Sales entities is one to many. The cardinality notation is 1:N.

3. Store – Supplier: One store can have multiple suppliers. The suppliers can supply products to 2 stores ("Main store" and "Digital Store"). The cardinality notation in this case is 1:N for the Store – Supplier and 1..2 for the Supplier – Store. In this way we can keep the stock of each store separated.

4. Store – Product: One store can have multiple products. One product can only be in one store at the time. The relationship between the Store and the Sales entities is one to many. The cardinality notation is 1:N.

5. Store – Customer: One store can have multiple customers. However, one customer can only make a purchase from one store at the time. The relationship between the Store and the Customer entities is one to many. The cardinality notation is 1:N.

6. Order – Product: One order can include multiple products. One product can be in multiple orders too. The relationship between the Order and the Product entities is many to many. The cardinality notation is N:N.

7. Order – Sales: Multiple orders can be included in just one sale. The relationship between the Order and the Sales entities is many to one. The cardinality notation is N:1.

8. Customer – Order: One customer can create multiple orders. However, one order can only be made by one customer. The relationship between the Customer and the Order entities is one to many. The cardinality notation is 1:N.
Do we have to think about "Many customers" can create "multiple orders" instead? If this is the case, the relationship is totally different!.

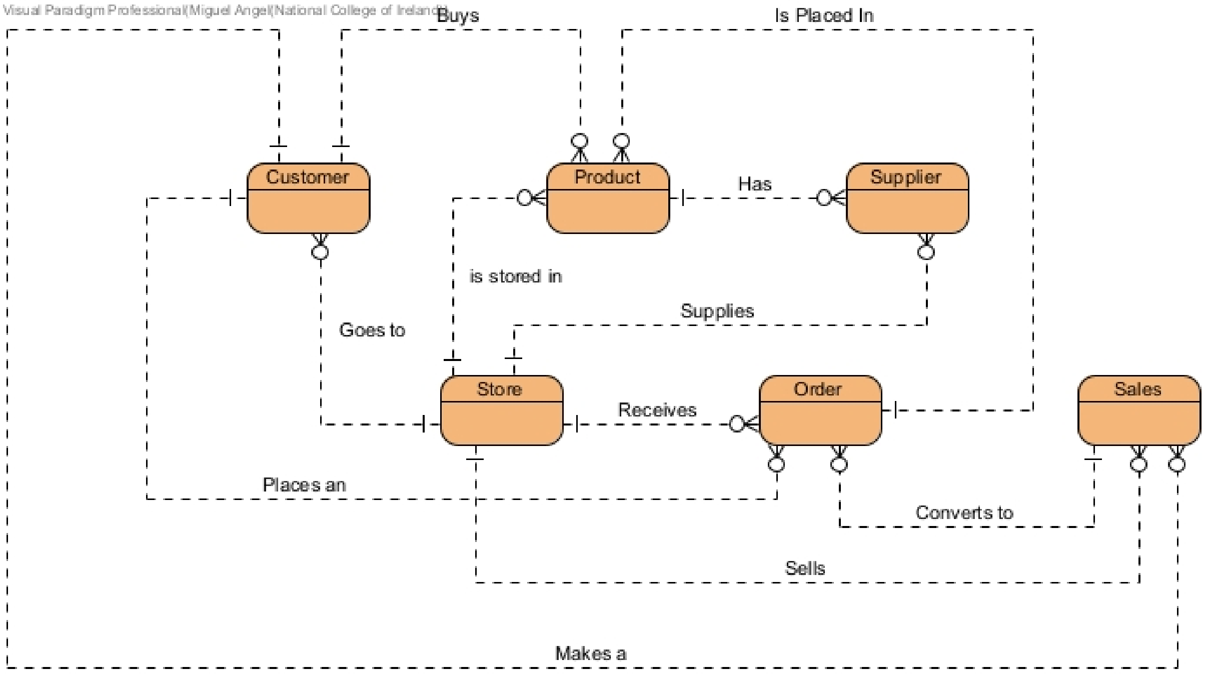
9. Customer – Sales: One customer can make many sales. One sale can be made out of many orders. Consequently one sale can only be made by one customer. The relationship between the Customer and the Sales entities is one to one. The cardinality notation is 1:N. We might need to introduce a foreign key to reflect that one sale can be composed of many orders.

10. Customer – Product: One customer can purchase many products. One product can be purchased by many customers. The relationship between the Customer and the Product entities is many to many. The cardinality notation is N:N.

11. Product – Supplier: One product can be supplied by one supplier. One supplier can supply many products. The relationship between the Product and the Supplier entities is one to one. The relationship between the Supplier and the Product entities is one to many. The cardinality notation in Product – Supplier is 1:1, the cardinality notation in Supplier – Product is 1:N.

4. Draw an ER diagram for the system depicting the entities, relationships, cardinalities, participations using your preferred ERD notation.
This ER diagram is before the entities being in the 3rd Normal Form.

There is another one in question 6 after the entities in the 3rd Normal Form.



5. Convert the entities in the conceptual design into a logical table. Use the following template to identify the tables. Make sure that the tables are in a 3rd normal form.

*a. **CUSTOMER** (customerID (PK), firstName, surname, email, age, birthday, purchases, addressID (FK), phoneNumber)*
for the phoneNumber attribute I have chosen an integer with a value of 20 because phone numbers can be very lengthy in some places around the world.

For the birthday attribute I have chosen a data type of date with the value to be year-month-day

*b. **ADDRESS** (addressID (PK), address1, address2, eircode)*

*c. **PRODUCT** (productID (PK), supplierID (FK), name, price, stock, onSale, categoryID (FK))*
for the price attribute I have chosen a decimal type because prices are not whole numbers and in my research I found that by giving a type decimal the calculations would be more exact and we will be able to calculate our sales accurately as opposed as using a double / float type. I chose DECIMAL (10, 2) because that means that the total digit will be 10 characters and two digits will appear after the decimal point.

*d. **CATEGORY** (categoryID (PK), name, subCategory1ID (FK))*

*e. **SUBCATEGORY** (subCategory1ID (PK), subCategory1Name)*

*f. **SUPPLIER** (supplierID (PK), name, addressID (FK), email, phoneNumber)*

for the phoneNumber attribute I have chosen an integer with a value of 20 because phone numbers can be very lengthy in some places around the world.

*g. **ORDERDETAILS** (orderdetailsId (PK), saleId (FK), itemDescription, Quantity, SalePrice, productId (FK), productOrderId (FK))*

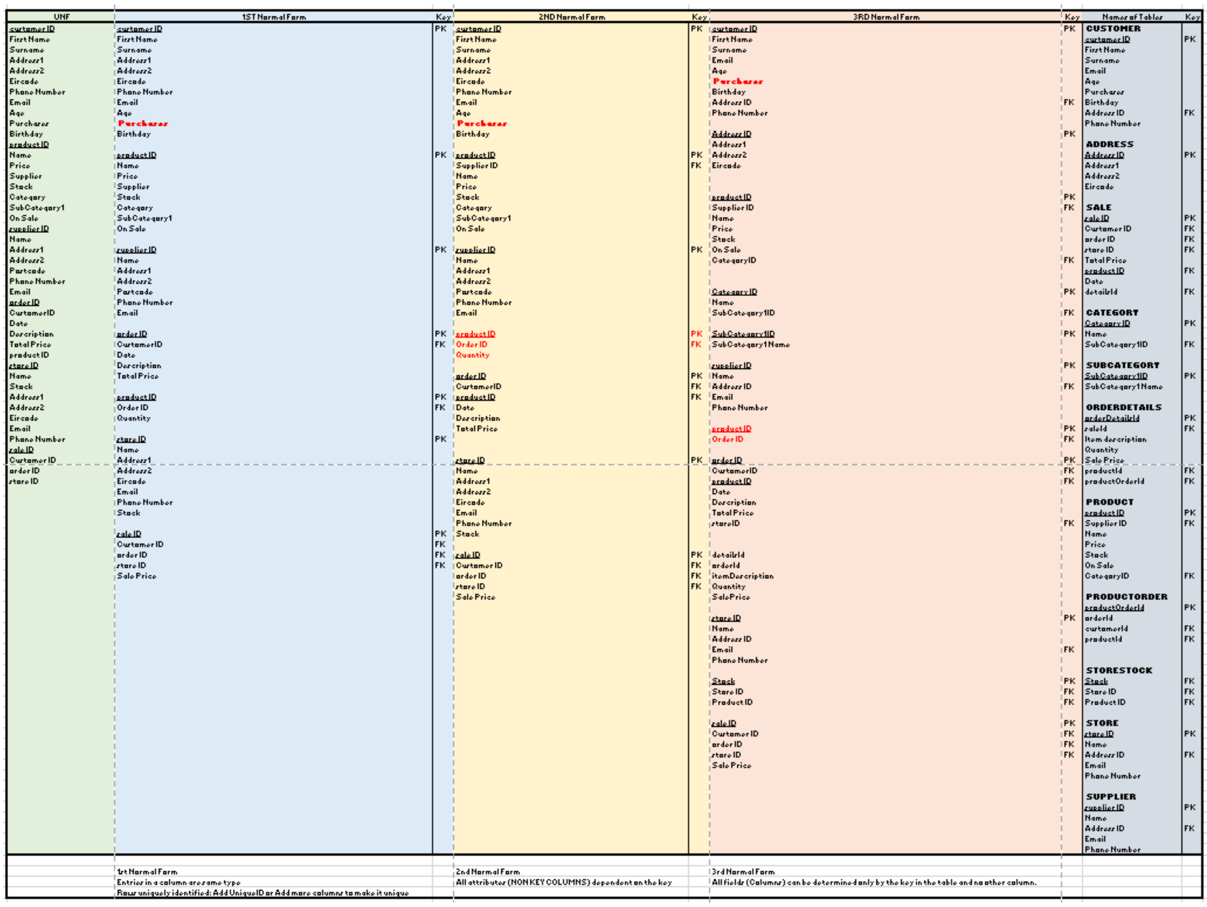
The entity cannot be named ORDER so I chose ORDERCREATED *i. **STORE** (storeID (PK), storename, addressID (FK), email, phoneNumber)*

for the phoneNumber attribute I have chosen an integer with a value of 20 because phone numbers can be very lengthy in some places around the world.

*j. **STORESTOCK** (stock, storeID (FK), productID (FK))*
I do not have a Primary Key here..

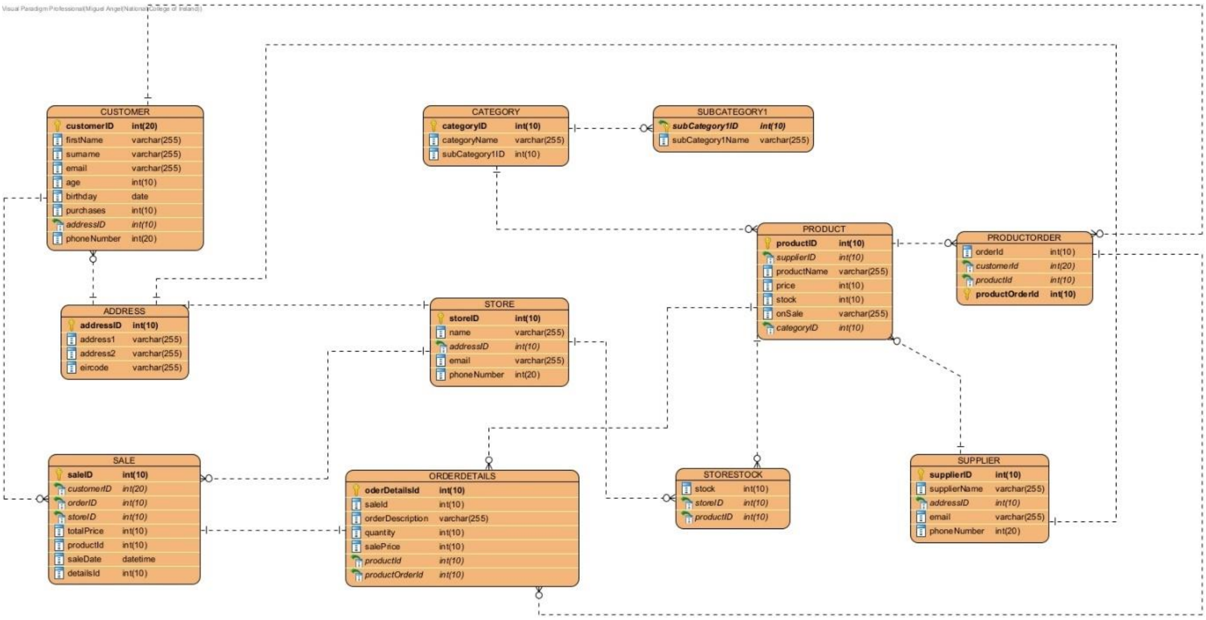
*k. **SALE** (saleID (PK), customerID (FK), orderID (FK), storeID (FK), totalPrice, productId (FK), date, detailsId(FK))*

*l. **PRODUCTORDER** (productOrderId (PK), orderId, customerId (FK), productId (FK))*



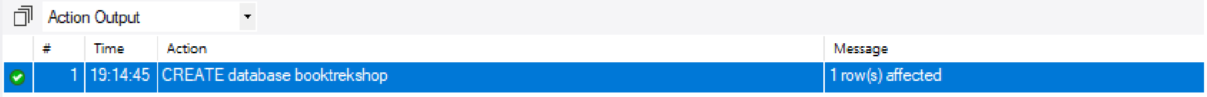
UNF	1ST Normal Form	Key	2ND Normal Form	Key	3RD Normal Form	Key	Names of Tables	Key
customerID	customerID	PK	customerID	PK	customerID	PK	**CUSTOMER**	
First Name	First Name		First Name		First Name		customerID	PK
Surname	Surname		Surname		Surname		First Name	
Address1	Address1		Address1		Email		Surname	
Address2	Address2		Address2		Age		Email	
Eircode	Eircode		Eircode		Purchases		Age	
Phone Number	Phone Number		Phone Number		Birthday		Purchases	
Email	Email		Email		Address ID	FK	Birthday	
Age	Age		Age		Phone Number		Address ID	FK
Purchases	Purchases		Purchases				Phone Number	
Birthday	Birthday		Birthday		AddressID	PK		
productID					Address1		**ADDRESS**	
Name	productID	PK	productID	PK	Address2		AddressID	PK
Price	Name		Supplier ID	FK	Eircode		Address1	
Supplier	Price		Name				Address2	
Stock	Supplier		Price				Eircode	
Category	Stock		Stock		productID	PK		
SubCategory1	Category		Category		Supplier ID	FK	**SALE**	
On Sale	SubCategory1		SubCategory1		Name		saleID	PK
supplierID	On Sale		On Sale		Price		Customer ID	FK
Name					Stock		order ID	FK
Address1	supplierID	PK	supplierID	PK	On Sale		store ID	FK
Address2	Name		Name		CategoryID	FK	Total Price	
Postcode	Address1		Address1				productID	FK
Phone Number	Address2		Address2				Date	
Email	Postcode		Postcode		CategoryID	PK	detailsId	FK
orderID	Phone Number		Phone Number		Name			
CustomerID	Email		Email		SubCategory1ID	FK	**CATEGORT**	
Date							CategoryID	PK
Description	orderID	PK	productID	PK	SubCategory1ID	PK	Name	
Total Price	CustomerID	FK	Order ID	FK	SubCategory1Name		SubCategory1ID	FK
product ID	Date		Quantity					
storeID	Description				supplierID	PK	**SUBCATEGORT**	
Name	Total Price		orderID	PK	Name		SubCategory1ID	PK
Stock			CustomerID	FK	Address ID	FK	SubCategory1Name	
Address1	productID	PK	productID	FK	Email			
Address2	Order ID	FK	Date		Phone Number		**ORDERDETAILS**	
Eircode	Quantity		Description				orderDetailsId	PK
Email			Total Price		productID	PK	saleId	FK
Phone Number	storeID	PK			Order ID	FK	Item description	
saleID	Name						Quantity	
Customer ID	Address1		storeID	PK	orderID	PK	Sale Price	
order ID	Address2		Name		CustomerID	FK	productId	FK
store ID	Eircode		Address1		productID	FK	productOrderId	FK
	Email		Address2		Date			
	Phone Number		Eircode		Description		**PRODUCT**	
	Stock		Email		Total Price		productID	PK
			Phone Number		storeID	FK	Supplier ID	FK
	saleID	PK	Stock				Name	
	Customer ID	FK					Price	
	order ID	FK	saleID	PK	detailsId		Stock	
	store ID	FK	Customer ID	FK	orderId		On Sale	
	Sale Price		order ID	FK	itemDescription		CategoryID	FK
			store ID	FK	Quantity			
			Sale Price		SalePrice		**PRODUCTORDER**	
							productOrderId	PK
					storeID	PK	orderId	
					Name		customerId	FK
					Address ID		productId	FK
					Email	FK		
					Phone Number			
							STORESTOCK	
					Stock	PK	Stock	FK
					Store ID	FK	Store ID	FK
					Product ID	FK	Product ID	FK
					saleID	PK	**STORE**	
					Customer ID	FK	storeID	PK
					order ID	FK	Name	
					store ID	FK	Address ID	FK
					Sale Price		Email	
							Phone Number	
							SUPPLIER	
							supplierID	PK
							Name	
							Address ID	FK
							Email	
							Phone Number	
	1st Normal Form		2nd Normal Form		3rd Normal Form			
	Entries in a column are same type		All attributes (NON KEY COLUMNS) dependent on the key		All fields (Columns) can be determined only by the key in the table and no other column.			
	Rows uniquely identified: Add UniqueID or Add more columns to make it unique							

6. ER diagram with the system depicting the entities, relationships, cardinalities, participations after the 3rd Normal Form.



PART 2: Logical and physical design

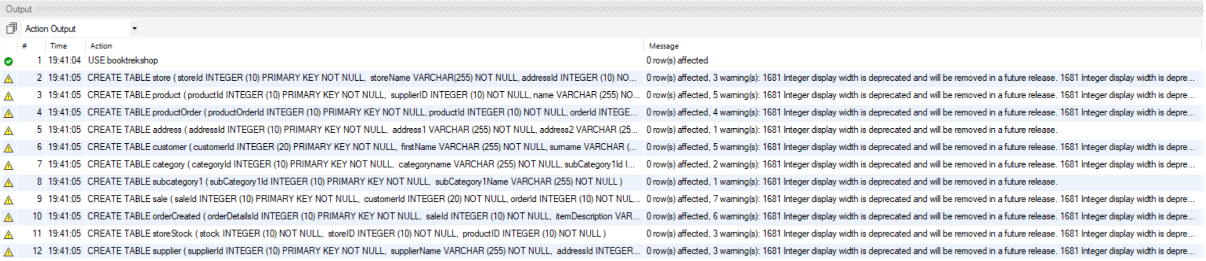
1. Create the corresponding database using DDL.



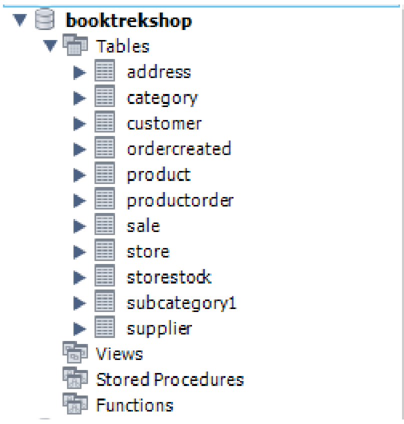
Action Output

	#	Time	Action	Message
✓	1	19:14:45	CREATE database booktrekshop	1 row(s) affected

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

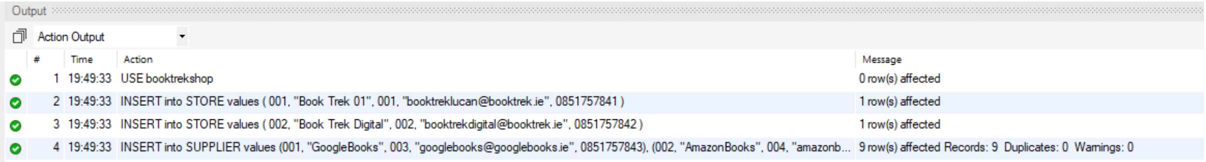


```
Output
Action Output
    #  Time      Action                                                                                                                   Message
✓   1  19:41:04  USE booktrekshop                                                                                                         0 row(s) affected
⚠   2  19:41:05  CREATE TABLE store ( storeId INTEGER (10) PRIMARY KEY NOT NULL,  storeName VARCHAR(255) NOT NULL, addressId INTEGER (10) NO...  0 row(s) affected, 3 warning(s): 1681 Integer display width is deprecated and will be removed in a future release. 1681 Integer display width is depre...
⚠   3  19:41:05  CREATE TABLE product ( productId INTEGER (10) PRIMARY KEY NOT NULL,  supplierID INTEGER (10) NOT NULL, name VARCHAR (255) NO...  0 row(s) affected, 5 warning(s): 1681 Integer display width is deprecated and will be removed in a future release. 1681 Integer display width is depre...
⚠   4  19:41:05  CREATE TABLE productOrder ( productOrderId INTEGER (10) PRIMARY KEY NOT NULL, productId INTEGER (10) NOT NULL, orderId INTEGE...  0 row(s) affected, 4 warning(s): 1681 Integer display width is deprecated and will be removed in a future release. 1681 Integer display width is depre...
⚠   5  19:41:05  CREATE TABLE address ( addressId INTEGER (10) PRIMARY KEY NOT NULL,  address1 VARCHAR (255) NOT NULL, address2 VARCHAR (25...  0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future release.
⚠   6  19:41:05  CREATE TABLE customer ( customerId INTEGER (20) PRIMARY KEY NOT NULL,  firstName VARCHAR (255) NOT NULL, surname VARCHAR (...  0 row(s) affected, 5 warning(s): 1681 Integer display width is deprecated and will be removed in a future release. 1681 Integer display width is depre...
⚠   7  19:41:05  CREATE TABLE category ( categoryId INTEGER (10) PRIMARY KEY NOT NULL,  categoryname VARCHAR (255) NOT NULL, subCategory1Id I...  0 row(s) affected, 2 warning(s): 1681 Integer display width is deprecated and will be removed in a future release. 1681 Integer display width is depre...
⚠   8  19:41:05  CREATE TABLE subcategory1 ( subCategory1Id INTEGER (10) PRIMARY KEY NOT NULL,  subCategory1Name VARCHAR (255) NOT NULL )      0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future release.
⚠   9  19:41:05  CREATE TABLE sale ( saleId INTEGER (10) PRIMARY KEY NOT NULL,  customerId INTEGER (20) NOT NULL, orderId INTEGER (10) NOT NUL...  0 row(s) affected, 7 warning(s): 1681 Integer display width is deprecated and will be removed in a future release. 1681 Integer display width is depre...
⚠  10  19:41:05  CREATE TABLE orderCreated ( orderDetailsId INTEGER (10) PRIMARY KEY NOT NULL,  saleId INTEGER (10) NOT NULL,  itemDescription VAR...  0 row(s) affected, 6 warning(s): 1681 Integer display width is deprecated and will be removed in a future release. 1681 Integer display width is depre...
⚠  11  19:41:05  CREATE TABLE storeStock ( stock INTEGER (10) NOT NULL,  storeID INTEGER (10) NOT NULL,  productID INTEGER (10) NOT NULL )      0 row(s) affected, 3 warning(s): 1681 Integer display width is deprecated and will be removed in a future release. 1681 Integer display width is depre...
⚠  12  19:41:05  CREATE TABLE supplier ( supplierId INTEGER (10) PRIMARY KEY NOT NULL,  supplierName VARCHAR (255) NOT NULL,  addressId INTEGER...  0 row(s) affected, 3 warning(s): 1681 Integer display width is deprecated and will be removed in a future release. 1681 Integer display width is depre...
```



```
▼ booktrekshop
  ▼ Tables
    ▶ address
    ▶ category
    ▶ customer
    ▶ ordercreated
    ▶ product
    ▶ productorder
    ▶ sale
    ▶ store
    ▶ storestock
    ▶ subcategory1
    ▶ supplier
    Views
    Stored Procedures
    Functions
```

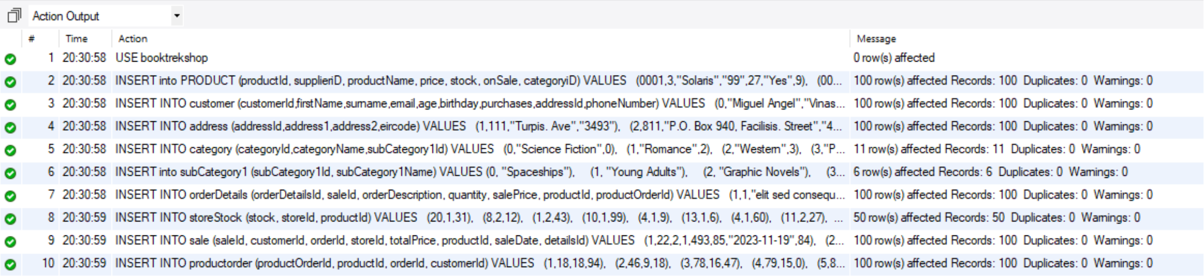
3. Populate your tables with some data using DML (insert into statement).



```
Output
Action Output
 #  Time      Action                                                                                                                  Message
 1  19:49:33  USE booktrekshop                                                                                                        0 row(s) affected
 2  19:49:33  INSERT into STORE values ( 001, "Book Trek 01", 001, "booktreklucan@booktrek.ie", 0851757841 )                          1 row(s) affected
 3  19:49:33  INSERT into STORE values ( 002, "Book Trek Digital", 002, "booktrekdigital@booktrek.ie", 0851757842 )                   1 row(s) affected
 4  19:49:33  INSERT into SUPPLIER values (001, "GoogleBooks", 003, "googlebooks@googlebooks.ie", 0851757843), (002, "AmazonBooks", 004, "amazonb...  9 row(s) affected Records: 9  Duplicates: 0  Warnings: 0
```

4. Populate your database with DML insert statements OR using a large data set representing a sample one-year transaction (01/01/2023 - 31/12/2023) on each table, using online data generators such as Mockaroo or generate data to generate synthetic data.

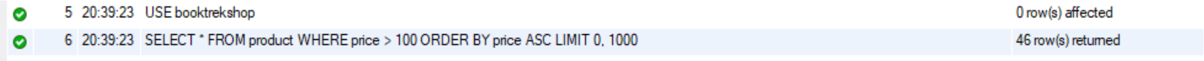
Used generatedata.com partially



```
Action Output
 #  Time      Action                                                                                                                                                   Message
 1  20:30:58  USE booktrekshop                                                                                                                                         0 row(s) affected
 2  20:30:58  INSERT into PRODUCT (productId, supplieriD, productName, price, stock, onSale, categoryiD) VALUES  (0001,3,"Solaris","99",27,"Yes",9),  (00...         100 row(s) affected Records: 100  Duplicates: 0  Warnings: 0
 3  20:30:58  INSERT INTO customer (customerId,firstName,surname,email,age,birthday,purchases,addressId,phoneNumber) VALUES  (0,"Miguel Angel","Vinas...            100 row(s) affected Records: 100  Duplicates: 0  Warnings: 0
 4  20:30:58  INSERT INTO address (addressId,address1,address2,eircode) VALUES  (1,111,"Turpis. Ave","3493"),  (2,811,"P.O. Box 940, Facilisis. Street","4...         100 row(s) affected Records: 100  Duplicates: 0  Warnings: 0
 5  20:30:58  INSERT INTO category (categoryId,categoryName,subCategory1Id) VALUES  (0,"Science Fiction",0),  (1,"Romance",2),  (2,"Western",3),  (3,"P...          11 row(s) affected Records: 11  Duplicates: 0  Warnings: 0
 6  20:30:58  INSERT into subCategory1 (subCategory1Id, subCategory1Name) VALUES (0, "Spaceships"),   (1, "Young Adults"),   (2, "Graphic Novels"),   (3...          6 row(s) affected Records: 6  Duplicates: 0  Warnings: 0
 7  20:30:58  INSERT INTO orderDetails (orderDetailsId, saleId, orderDescription, quantity, salePrice, productId, productOrderId) VALUES  (1,1,"elit sed consequ...   100 row(s) affected Records: 100  Duplicates: 0  Warnings: 0
 8  20:30:59  INSERT INTO storeStock (stock, storeId, productId) VALUES  (20,1,31),  (8,2,12),  (1,2,43),  (10,1,99),  (4,1,9),  (13,1,6),  (4,1,60),  (11,2,27),  ...   50 row(s) affected Records: 50  Duplicates: 0  Warnings: 0
 9  20:30:59  INSERT INTO sale (saleId, customerId, orderId, storeId, totalPrice, productId, saleDate, detailsId) VALUES  (1,22,2,1,493,85,"2023-11-19",84),  (2...  100 row(s) affected Records: 100  Duplicates: 0  Warnings: 0
10  20:30:59  INSERT INTO productorder (productOrderId, productId, orderId, customerId) VALUES  (1,18,18,94),  (2,46,9,18),  (3,78,16,47),  (4,79,15,0),  (5,8...      100 row(s) affected Records: 100  Duplicates: 0  Warnings: 0
```

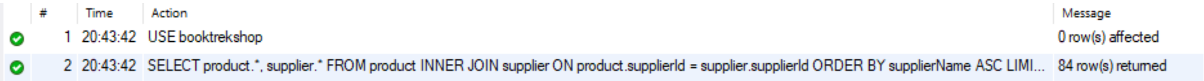
PART 3: Write SQL Statements to answer the following queries

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



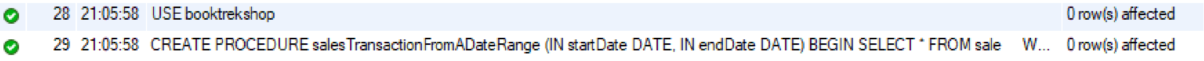
```
5  20:39:23  USE booktrekshop                                                    0 row(s) affected
6  20:39:23  SELECT * FROM product WHERE price > 100 ORDER BY price ASC LIMIT 0, 1000   46 row(s) returned
```

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

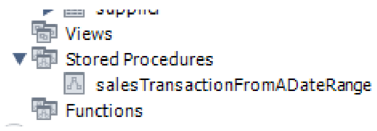


```
#  Time      Action                                                                                                  Message
1  20:43:42  USE booktrekshop                                                                                        0 row(s) affected
2  20:43:42  SELECT product.*, supplier.* FROM product INNER JOIN supplier ON product.supplierId = supplier.supplierId ORDER BY supplierName ASC LIMI...  84 row(s) returned
```

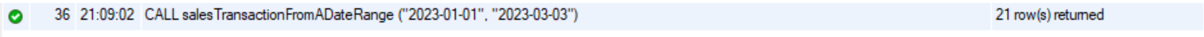
3. Create a stored procedure that takes the start and end dates of the sales and display all the sales transactions between the start and the end dates.



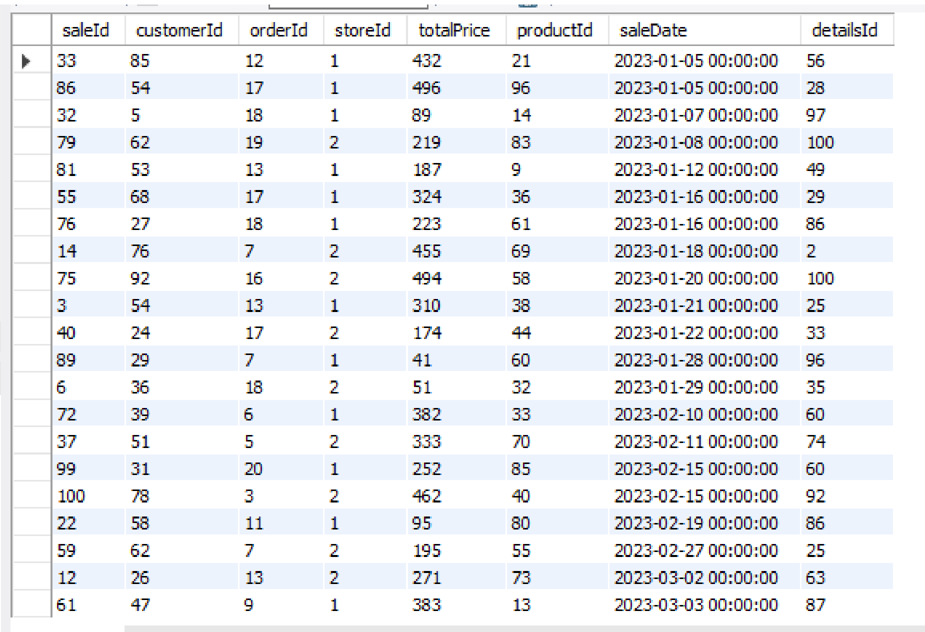
```
28  21:05:58  USE booktrekshop                                                                                        0 row(s) affected
29  21:05:58  CREATE PROCEDURE salesTransactionFromADateRange (IN startDate DATE, IN endDate DATE) BEGIN SELECT * FROM sale    W...  0 row(s) affected
```



```
Views
Stored Procedures
    salesTransactionFromADateRange
Functions
```

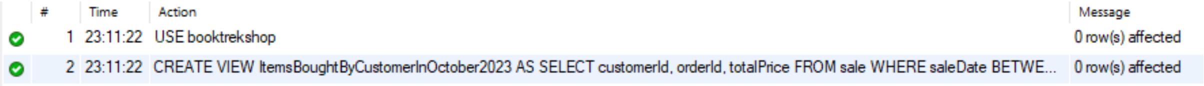


```
36  21:09:02  CALL salesTransactionFromADateRange ("2023-01-01", "2023-03-03")       21 row(s) returned
```

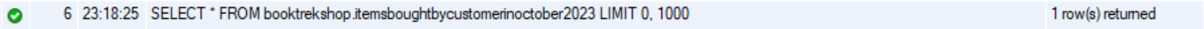


saleId	customerId	orderId	storeId	totalPrice	productId	saleDate	detailsId
33	85	12	1	432	21	2023-01-05 00:00:00	56
86	54	17	1	496	96	2023-01-05 00:00:00	28
32	5	18	1	89	14	2023-01-07 00:00:00	97
79	62	19	2	219	83	2023-01-08 00:00:00	100
81	53	13	1	187	9	2023-01-12 00:00:00	49
55	68	17	1	324	36	2023-01-16 00:00:00	29
76	27	18	1	223	61	2023-01-16 00:00:00	86
14	76	7	2	455	69	2023-01-18 00:00:00	2
75	92	16	2	494	58	2023-01-20 00:00:00	100
3	54	13	1	310	38	2023-01-21 00:00:00	25
40	24	17	2	174	44	2023-01-22 00:00:00	33
89	29	7	1	41	60	2023-01-28 00:00:00	96
6	36	18	2	51	32	2023-01-29 00:00:00	35
72	39	6	1	382	33	2023-02-10 00:00:00	60
37	51	5	2	333	70	2023-02-11 00:00:00	74
99	31	20	1	252	85	2023-02-15 00:00:00	60
100	78	3	2	462	40	2023-02-15 00:00:00	92
22	58	11	1	95	80	2023-02-19 00:00:00	86
59	62	7	2	195	55	2023-02-27 00:00:00	25
12	26	13	2	271	73	2023-03-02 00:00:00	63
61	47	9	1	383	13	2023-03-03 00:00:00	87

4. Create a view that shows the total number of items a customer buys from the business in October 2023 along with the total price (use group by)



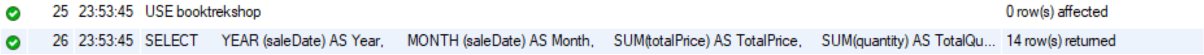
```
#  Time      Action                                                                                                                          Message
1  23:11:22  USE booktrekshop                                                                                                                0 row(s) affected
2  23:11:22  CREATE VIEW ItemsBoughtByCustomerInOctober2023 AS SELECT customerId, orderId, totalPrice FROM sale WHERE saleDate BETWE...     0 row(s) affected
```



```
6  23:18:25  SELECT * FROM booktrekshop.itemsboughtbycustomerinoctober2023 LIMIT 0, 1000     1 row(s) returned
```

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

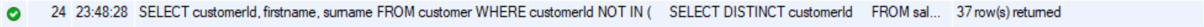
6. Create a query to view the annual sales over the last 12 months of the business showing the total number of products sold and the total price sold every month (use A group by with roll-up).



```
25  23:53:45  USE booktrekshop                                                                                     0 row(s) affected
26  23:53:45  SELECT    YEAR (saleDate) AS Year,    MONTH (saleDate) AS Month,    SUM(totalPrice) AS TotalPrice,    SUM(quantity) AS TotalQu...    14 row(s) returned
```

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

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



```
24  23:48:28  SELECT customerId, firstname, surname FROM customer WHERE customerId NOT IN (    SELECT DISTINCT customerId    FROM sal...    37 row(s) returned
```