

## Databases and Business Applications B8IT101

Name: Seán Carroll

Student Number: 20024157

Module Code: B8IT101

### Question 1 - SQL Queries

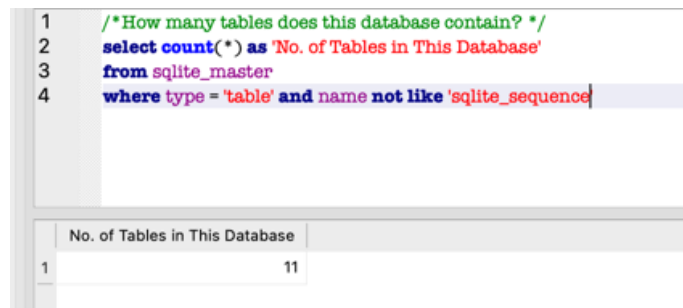
1. *How many tables does this database contain?*

/\*Code for question 1\*/

select count(\*) as 'No. of Tables in This Database'

from sqlite\_master

where type = 'table' and name not like 'sqlite\_sequence'



The screenshot shows a SQL query editor with the following code:

```
1  /*How many tables does this database contain? */
2  select count(*) as 'No. of Tables in This Database'
3  from sqlite_master
4  where type = 'table' and name not like 'sqlite_sequence'
```

Below the query editor, the results are displayed in a table:


No. of Tables in This Database	
1	11

2. *How many invoices are listed in this database?*

/\*Code for question 2\*/

select count(InvoiceId) as 'No. of Invoices in This Database'

from invoices



The screenshot shows a SQL query editor with the following code:

```
1  /*How many invoices are listed in this database?*/
2  select count(InvoiceId) as 'No. of Invoices in This Database'
3  from invoices
```

Below the query editor, the results are displayed in a table:

No. of Invoices in This Database	
1	412

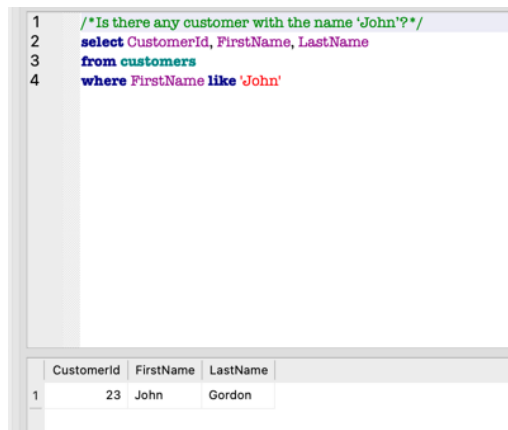
3. *Is there any customer with the name 'John'?*

*/\*Code for question 3\*/*

select CustomerId, FirstName, LastName

from customers

where FirstName like 'John'



```
1  /*Is there any customer with the name 'John'?*/
2  select CustomerId, FirstName, LastName
3  from customers
4  where FirstName like 'John'
```

	CustomerId	FirstName	LastName
1	23	John	Gordon

4. *What is the longest album in the database?*

*/\*Code for question 4\*/*

select albums.Title, sum(tracks.Milliseconds/60000) as 'Length of Album in Minutes'

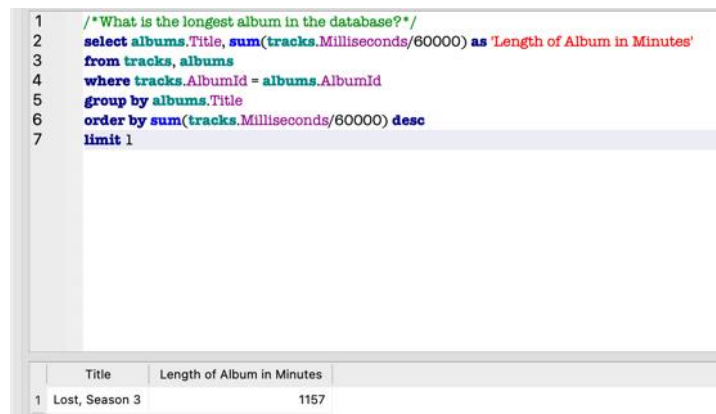
from tracks, albums

where tracks.AlbumId = albums.AlbumId

group by albums.Title

order by sum(tracks.Milliseconds/60000) desc

limit 1



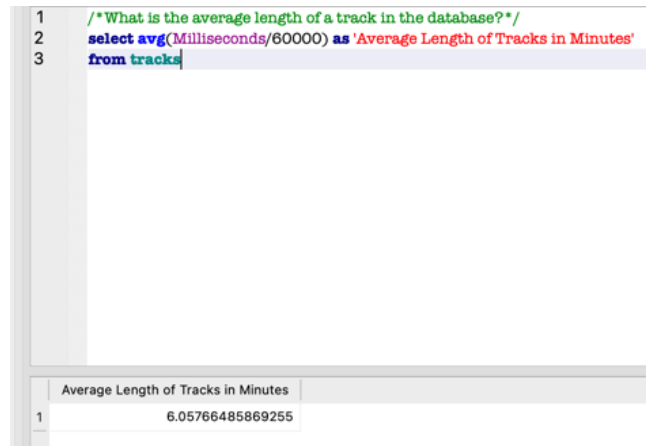
```
1  /*What is the longest album in the database?*/
2  select albums.Title, sum(tracks.Milliseconds/60000) as 'Length of Album in Minutes'
3  from tracks, albums
4  where tracks.AlbumId = albums.AlbumId
5  group by albums.Title
6  order by sum(tracks.Milliseconds/60000) desc
7  limit 1
```

	Title	Length of Album in Minutes
1	Lost, Season 3	1157

5. What is the average length of a track in the database?

/\*Code for question 5\*/

```
select avg(Milliseconds/60000) as 'Average Length of Tracks in Minutes'
from tracks
```



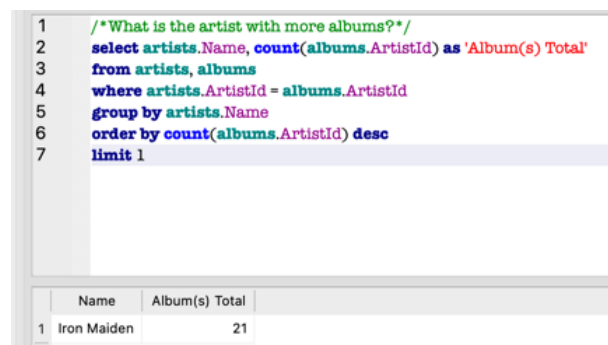
```
1  /*What is the average length of a track in the database?*/
2  select avg(Milliseconds/60000) as 'Average Length of Tracks in Minutes'
3  from tracks
```

Average Length of Tracks in Minutes	
1	6.05766485869255

6. What is the artist with more albums?

/\*Code for question 6\*/

```
select artists.Name, count(albums.ArtistId) as 'Album(s) Total'
from artists, albums
where artists.ArtistId = albums.ArtistId
group by artists.Name
order by count(albums.ArtistId) desc
limit 1
```



```
1  /*What is the artist with more albums?*/
2  select artists.Name, count(albums.ArtistId) as 'Album(s) Total'
3  from artists, albums
4  where artists.ArtistId = albums.ArtistId
5  group by artists.Name
6  order by count(albums.ArtistId) desc
7  limit 1
```

	Name	Album(s) Total
1	Iron Maiden	21

7. *List the top 5 artists with more sales.*

/\*Code for question 7\*/

select artists.Name, sum(invoices.total) as 'Total Sales'

from artists, albums, tracks, invoice\_items, invoices

where artists.ArtistId = albums.ArtistId and albums.AlbumId = tracks.AlbumId and  
tracks.TrackId = invoice\_items.TrackId and invoice\_items.InvoiceId = invoices.InvoiceId

group by artists.Name

order by sum(invoices.total) desc

limit 5

```
1  /*List the top 5 artists with more sales.*/
2  select artists.Name, sum(invoices.total) as 'Total Sales'
3  from artists, albums, tracks, invoice_items, invoices
4  where artists.ArtistId = albums.ArtistId and albums.AlbumId = tracks.AlbumId and tracks.TrackId = invoice_items.TrackId and invoice_items.InvoiceId = invoices.InvoiceId
5  group by artists.Name
6  order by sum(invoices.total) desc
7  limit 5
```

	Name	Total Sales
1	Iron Maiden	1233.54
2	U2	895.5900000000001
3	Lost	833.7
4	Led Zeppelin	620.73
5	Metallica	599.94

8. List the tracks that contain 'you' in their title. List the album and artist of these tracks.

/\*Code for question 8\*/

select tracks.Name as 'Track Title', albums.Title as 'Album Title', artists.Name as 'Artist Name'

from artists, albums, tracks

where artists.ArtistId = albums.ArtistId and albums.AlbumId = tracks.AlbumId and tracks.Name like '%you %'

1	/*List the tracks that contain 'you' in their title. List the album and artist of these tracks.*/		
2	select tracks.Name as 'Track Title', albums.Title as 'Album Title', artists.Name as 'Artist Name'		
3	from artists, albums, tracks		
4	where artists.ArtistId = albums.ArtistId and albums.AlbumId = tracks.AlbumId and tracks.Name like '%you %'		
	Track Title	Album Title	Artist Name
1	You Oughta Know	Jagged Little Pill	Alanis Morissette
2	You Learn	Jagged Little Pill	Alanis Morissette
3	You Oughta Know (Alternate)	Jagged Little Pill	Alanis Morissette
4	Put You Down	Facelift	Alice In Chains
5	What You Are	Audioslave	Audioslave
6	Let Me Love You Baby	The Best Of Buddy Guy - The Millenium Collection	Buddy Guy
7	You Shook Me	BBC Sessions [Disc 1] [Live]	Led Zeppelin
8	I Can't Quit You Baby	BBC Sessions [Disc 1] [Live]	Led Zeppelin
9	I Can't Quit You Baby(2)	BBC Sessions [Disc 1] [Live]	Led Zeppelin
10	You Shook Me(2)	BBC Sessions [Disc 1] [Live]	Led Zeppelin
11	Do You Love Me	Greatest Kiss	Kiss
12	Don't You Cry	Into The Light	David Coverdale
13	Wherever You May Go	Into The Light	David Coverdale
14	You Fool No One	MK III The Final Concerts [Disc 1]	Deep Purple
15	Do You Like The Way	Supernatural	Santana
16	Do You Have Other Loves?	The Best of Ed Motta	Ed Motta
17	You Fool No One (Alternate Version)	The Final Concerts (Disc 2)	Deep Purple
18	Have You Ever Seen The Rain?	Chronicle, Vol. 1	Creedence Clearwater Revival
19	Before You Accuse Me	Chronicle, Vol. 2	Creedence Clearwater Revival
20	You Keep On Moving	Come Taste The Band	Deep Purple
21	You Can't Do it Right (With the One You Love)	Stormbringer	Deep Purple
22	Have You Ever Needed Someone So Bad	Vault: Def Leppard's Greatest Hits	Def Leppard
23	Before You Accuse Me	Unplugged	Eric Clapton
24	Nobody Knows You When You're Down & Out	Unplugged	Eric Clapton
25	I've Got You Under My Skin	My Way: The Best Of Frank Sinatra [Disc 1]	Frank Sinatra
26	Homecoming / The Death Of St. Jimmys / Fast 12th St. /	American Idiot	Green Day

9. What is the average number of items per invoice? List all invoices that have more than the average number of items in them.

/\*Code for question 9\*/

select \*

from invoices

left join (

select InvoiceId, count(InvoiceId) as 'ItemCount'

from invoice\_items

group by InvoiceId

) as 'Count\_InvoiceId' on invoices.InvoiceId = Count\_InvoiceId.InvoiceId

where Count\_InvoiceId.ItemCount > (select avg(ItemCount) from (select InvoiceId, count(\*) as ItemCount from invoice\_items group by InvoiceId))

1	/*What is the average number of items per invoice? List all invoices that have more than the average number of items in them.*/										
2	select *										
3	from invoices										
4	left join (										
5	select InvoiceId, count(InvoiceId) as 'ItemCount'										
6	from invoice_items										
7	group by InvoiceId										
8	) as 'Count_InvoiceId' on invoices.InvoiceId = Count_InvoiceId.InvoiceId										
9	where Count_InvoiceId.ItemCount > (select avg(ItemCount) from (select InvoiceId, count(*) as ItemCount from invoice_items group by InvoiceId))										
10											
	InvoiceId	CustomerId	InvoiceDate	BillingAddress	BillingCity	BillingState	BillingCountry	BillingPostalCode	Total	InvoiceId	ItemCount
1	3	8	2009-01-03 00:00:00	Grétrystraat 63	Brussels	NULL	Belgium	1000	5.94	3	6
2	4	14	2009-01-06 00:00:00	8210 111 ST NW	Edmonton	AB	Canada	T6G 2C7	8.91	4	9
3	5	23	2009-01-11 00:00:00	69 Salem Street	Boston	MA	USA	2113	13.86	5	14
4	10	46	2009-02-03 00:00:00	3 Chatham Street	Dublin	Dublin	Ireland	NULL	5.94	10	6
5	11	52	2009-02-06 00:00:00	202 Hoxton Street	London	NULL	United Kingdom	N1 5LH	8.91	11	9
6	12	2	2009-02-11 00:00:00	Theodor-Heuss-Straße 34	Stuttgart	NULL	Germany	70174	13.86	12	14
7	17	25	2009-03-06 00:00:00	319 N. Frances Street	Madison	WI	USA	53703	5.94	17	6
8	18	31	2009-03-09 00:00:00	194A Chain Lake Drive	Halifax	NS	Canada	B3S 1C5	8.91	18	9
9	19	40	2009-03-14 00:00:00	8, Rue Hanovre	Paris	NULL	France	75002	13.86	19	14
10	24	4	2009-04-06 00:00:00	Ullevålsveien 14	Oslo	NULL	Norway	0171	5.94	24	6
11	25	10	2009-04-09 00:00:00	Rua Dr. Falcão Filho, 155	São Paulo	SP	Brazil	01007-010	8.91	25	9
12	26	19	2009-04-14 00:00:00	1 Infinite Loop	Cupertino	CA	USA	95014	13.86	26	14
13	31	42	2009-05-07 00:00:00	9, Place Louis Barthou	Bordeaux	NULL	France	33000	5.94	31	6
14	32	48	2009-05-10 00:00:00	Lijnbaansgracht 120bg	Amsterdam	VV	Netherlands	1016	8.91	32	9
15	33	57	2009-05-15 00:00:00	Calle Lira, 198	Santiago	NULL	Chile	NULL	13.86	33	14
16	38	21	2009-06-07 00:00:00	801 W 4th Street	Reno	NV	USA	89503	5.94	38	6
17	39	27	2009-06-10 00:00:00	1033 N Park Ave	Tucson	AZ	USA	85719	8.91	39	9
18	40	36	2009-06-15 00:00:00	Tauentzienstraße 8	Berlin	NULL	Germany	10789	13.86	40	14
19	45	59	2009-07-08 00:00:00	3,Raj Bhavan Road	Bangalore	NULL	India	560001	5.94	45	6
20	46	6	2009-07-11 00:00:00	Rilská 3174/6	Prague	NULL	Czech Republic	14300	8.91	46	9
21	47	15	2009-07-16 00:00:00	700 W Pender Street	Vancouver	BC	Canada	V6C 1G8	13.86	47	14
22	52	38	2009-08-08 00:00:00	Barbarossastraße 19	Berlin	NULL	Germany	10779	5.94	52	6

10. How many invoices are there per year? What is the total amount in sales per year?

/\*Code for question 10\*/

```
select strftime('%Y', InvoiceDate) as 'Year', count(*) as 'Total Invoices Per Year', sum(total)  
as 'Total Sales Per Year'
```

```
from invoices
```

```
group by year
```

```
1  /*How many invoices are there per year? What is the total amount in sales per year?*/  
2  select strftime('%Y', InvoiceDate) as 'Year', count(*) as 'Total Invoices Per Year', sum(total) as 'Total Sales Per Year'  
3  from invoices  
4  group by year
```

	Year	Total Invoices Per Year	Total Sales Per Year
1	2009	83	449.46
2	2010	83	481.45
3	2011	83	469.58
4	2012	83	477.53
5	2013	80	450.58

## **Question 2 - Database Design & Implementation**

### **Part I: Conceptual and Logical Design**

1. *Describe/introduce the chosen retail business and its objectives (200-400 words).*

As per Zara's website (2023), Zara is one of the largest international fashion companies and has a significant presence in Ireland. Zara is one of the many retail brands that belongs to Inditex, which is a distribution group that is one of the world's largest. Zara puts the customer at the heart of their business model, which includes design, production, distribution and sales through their extensive retail network.

According to Martin Roll (2021), Zara bases its core values around four simple terms: "beauty, clarity, functionality and sustainability." Zara's primary objective is to provide customers with the latest fashion trends at accessible prices, and it achieves this through a unique and efficient supply chain and production process. This commitment to this objective can be seen in their mission statement where they vouch "to give customers what they want, and to get it to them faster than anyone else" (Wondershare EdrawMind, n.d.). Zara uses a vertical integration model which identifies market preferences and uses a fast fashion strategy to enable both their physical and online stores to feature the most up-to-date and in demand clothing.

In its vision Zara also places an emphasis on contributing to a sustainable development of society and that of the environment with which we interact so that it can align its business practices with the growing consumer demand for environmentally conscious products (Wondershare EdrawMind, n.d.).

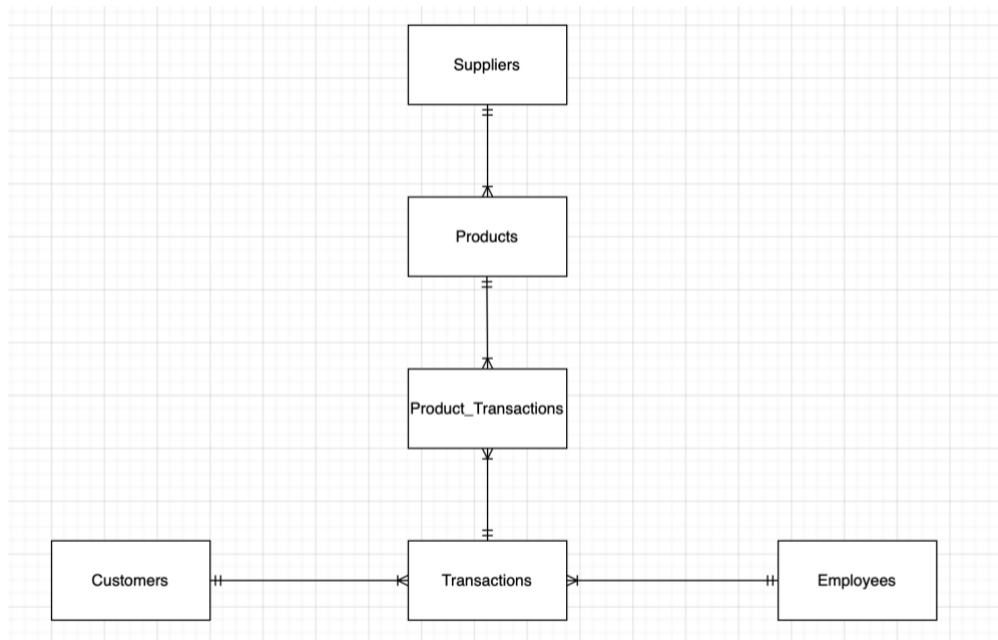


In terms of its presence in Ireland, Zara tailors its objectives to the specific demands and preferences of Irish consumers. This involves a keen understanding of local fashion trends, preferences, and cultural nuances. Additionally, Zara focuses on creating a positive and seamless shopping experience for its Irish customers, both in physical stores and its online retail store.

Overall, Zara's objectives revolve around staying at the forefront of fashion, providing customers with affordable and stylish clothing, and incorporating sustainable practices into its business model. By consistently meeting these objectives, Zara continues to be a prominent retailer in the highly competitive retail industry, offering a compelling mix of fashion-forward products and responsible business practices.

2. Identify the relevant entities of the business and their relationship. Draw a conceptual diagram of your database.

**I drew the following conceptual diagram to help identify the relevant entities of the business and their relationship.**



- The Suppliers table and the Products table share a one to many relationship as each supplier may supply multiple products.
- The Products table and the Products\_Transactions table share a one to many relationship as each product may be associated with multiple product transaction records.
- The Transactions table and the Product\_Transactions table share a one to many relationship as each transaction may have multiple transaction product records.
- The Employees table and the Transactions table share a one to many relationship as each employee may be involved in multiple transactions.
- The Customers table and the Transactions table share a one to many relationship as each customer may make multiple purchases.

3. *Identify the relationships, cardinalities and participation constraints with supporting business rules and assumptions. Identify the entity types, their respective attributes, primary keys and foreign keys. Document your assumptions.*

I designed the following data dictionary to help identify entity types, their respective attributes, primary keys and foreign keys. I will discuss the relationships, cardinalities, and participation constraints beneath each table.

**Products Table**

Attribute	Datatype	Required	Description
product_id	INTEGER	Yes	PK for Products Table.
product_name	VARCHAR(100)	Yes	Product's name, e.g. t-shirt.
price	NUMERIC(10,2)	Yes	Product's <u>price</u> , should be formatted to 2 decimal places.
stock_quantity	INTEGER	Yes	The amount of stock of the product currently available in stock.
supplier_id	INTEGER	Yes	FK for Suppliers Table

- product\_id is the primary key for the Products table.
- supplier\_id is a foreign key for the Suppliers table.

**Suppliers Table**

Attribute	Datatype	Required	Description
supplier_id	INTEGER	Yes	PK for Suppliers Table.
supplier_name	VARCHAR(100)	Yes	Name of the supplier, e.g. Alibaba Group
contact_person	VARCHAR(100)	Yes	Name of the contact person for the supplier. Should be a full name, e.g. Mary Smith
contact_email	VARCHAR(100)	Yes	Contact person's email <u>address</u> , does not have to be a company address.
contact_phone	VARCHAR(11)	Yes	Contact person's phone number. Phone number should be an Irish number, e.g. 087-6934267

- supplier\_id is the primary key for the Suppliers table.

**Product\_Transactions Table**

Attribute	Datatype	Required	Description
product_transactions_id	INTEGER	Yes	PK for Product_Transactions Table.
transaction_id	INTEGER	Yes	FK for Transactions Table.
product_id	INTEGER	Yes	FK for Products Table.
quantity	INTEGER	Yes	The quantity of products purchased by the customer upon completion of the transaction.
unit_price	NUMERIC(10,2)	Yes	The price of the product, should be formatted to 2 decimal places.

- The Product\_Transactions table is a junction table and represents the many-to-many relationship between transactions and products.
- product\_transactions\_id is the primary key for the Product\_Transactions table.
- transaction\_id is a foreign key for the Transactions table.
- product\_id is a foreign key for the Products table.

**Transactions Table**

Attribute	Datatype	Required	Description
transaction_id	INTEGER	Yes	PK for Transactions Table.
customer_id	INTEGER	Yes	FK for Customers Table.
employee_id	INTEGER	Yes	FK for Employees Table.
transaction_date	DATETIME	Yes	The Date the transaction took place in the year 2023, should be in the format YYYY/MM/DD, e.g. 2023/07/30
total_amount	NUMERIC(10,2)	Yes	The quantity of a product purchased multiplied by the price of the product and the total amount paid by the customer upon completion of the transaction. Should be formatted to 2 decimal places.

- transaction\_id is the primary key for the Transactions table.
- customer\_id is a foreign key for the Customers table.
- employee\_id is a foreign key for the Employees Table

**Employees Table**

Attribute	Datatype	Required	Description
employee_id	INTEGER	Yes	PK for Employees Table.
first_name	VARCHAR(100)	Yes	Employee Name, this is a first name like John.
last_name	VARCHAR(100)	Yes	Employee Name, this is a last name like Davids.
position	VARCHAR(100)	Yes	Employee's job title within the company, e.g. General Manager
address	VARCHAR(200)	Yes	This is a full address, should include Street
email	VARCHAR(100)	Yes	Employee's email address, should be a Google account, i.e. @Google.ie
phone_number	VARCHAR(11)	Yes	Phone number should be an Irish number, e.g. 087-6934267

- employee\_id is the primary key for the Employees table.

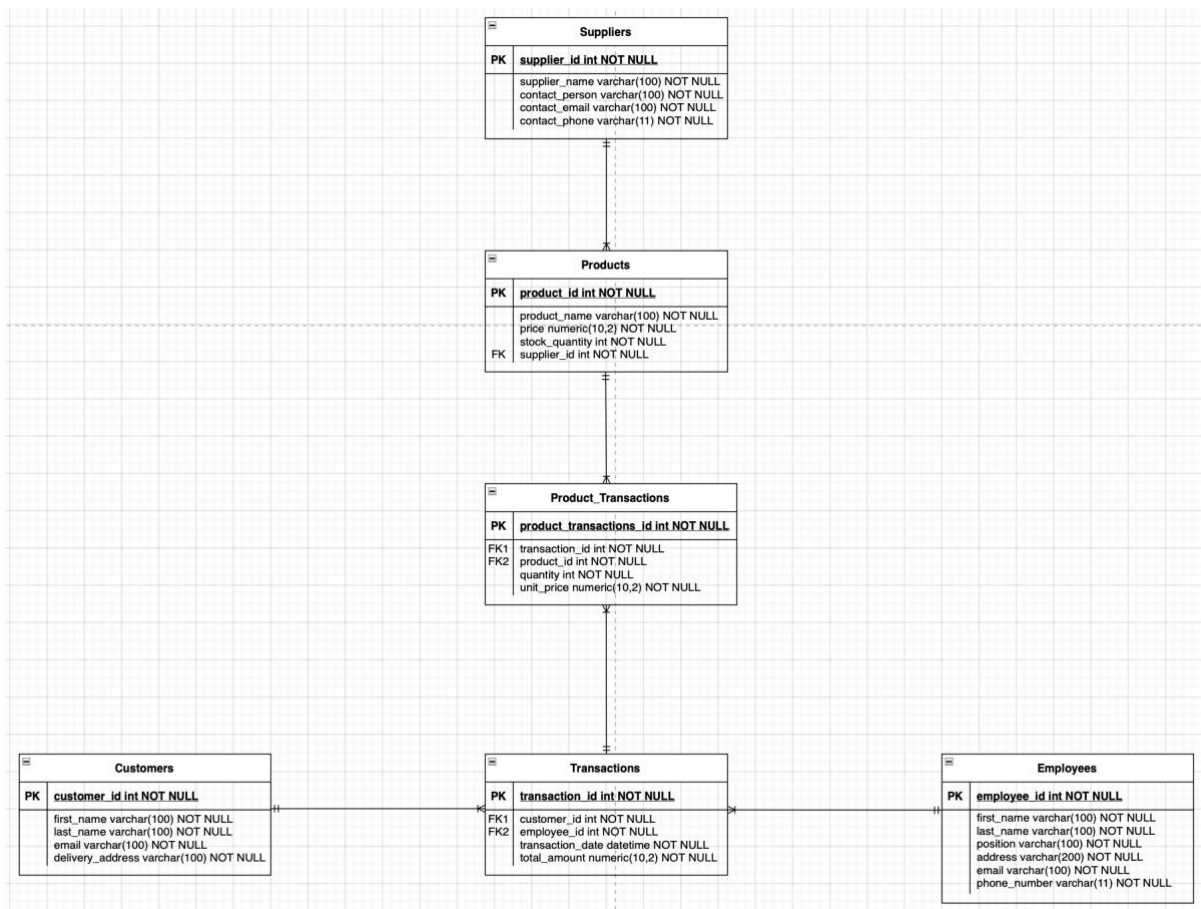
**Customers Table**

Attribute	Datatype	Required	Description
customer_id	INTEGER	Yes	PK for Customers Table.
first_name	VARCHAR(100)	Yes	Customer name, this is a first name like Ciara.
last_name	VARCHAR(100)	Yes	Customer name, this is a last name like Brodigan.
email	VARCHAR(100)	Yes	Customer's email address, should be a personal email address.
delivery_address	VARCHAR(200)	Yes	This is a full address, should include house number if possible and street address.

- customer\_id is the primary key for the Customers table.

4. Draw an ER diagram for the system depicting the entities, relationships, cardinalities, participations using your preferred ERD software. Make sure you include the data types and that tables are in a 3rd normal form.

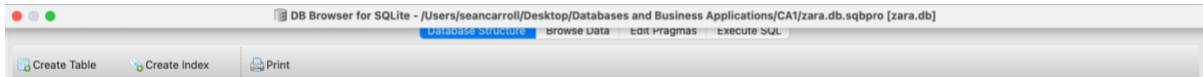
I drew the following ER diagram for the system depicting the entities, relationships, cardinalities, and participations.



## Part II: Physical Design

1. *Create the corresponding database using DDL.*

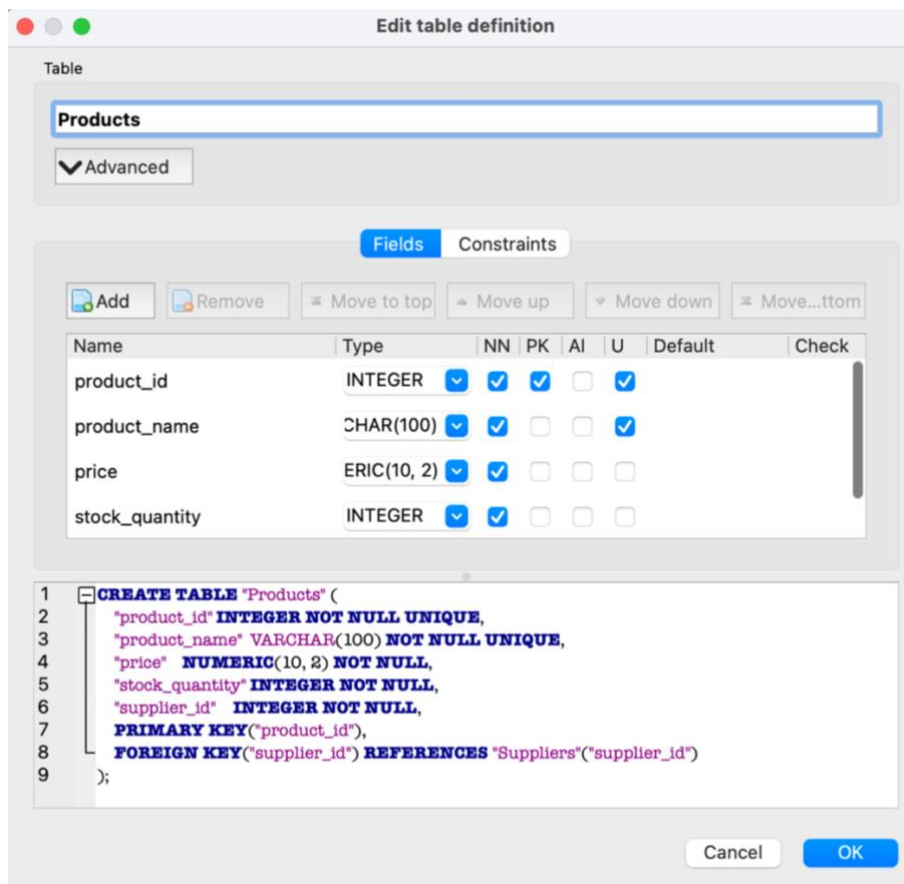
I named my database zara.db as this is the name of my chosen online retail store.



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

Code to create 'Products' Table.

```
CREATE TABLE "Products" (  
    "product_id" INTEGER NOT NULL UNIQUE,  
    "product_name" VARCHAR(100) NOT NULL UNIQUE,  
    "price" NUMERIC(10, 2) NOT NULL,  
    "stock_quantity" INTEGER NOT NULL,  
    "supplier_id" INTEGER NOT NULL,  
    PRIMARY KEY("product_id"),  
    FOREIGN KEY("supplier_id") REFERENCES "Suppliers"("supplier_id")  
);
```



## Code to create 'Suppliers' Table.

```
CREATE TABLE "Suppliers" (  
    "supplier_id" INTEGER NOT NULL UNIQUE,  
    "supplier_name" VARCHAR(100) NOT NULL,  
    "contact_person" VARCHAR(100) NOT NULL,  
    "contact_email" VARCHAR(100) NOT NULL,  
    "contact_phone" VARCHAR(11) NOT NULL,  
    PRIMARY KEY("supplier_id")  
);
```

Table: Suppliers

Advanced

Fields Constraints

Add Remove Move to top Move up Move down Move...tton

Name	Type	NN	PK	AI	U	Default	Check
supplier_id	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
supplier_name	CHAR(100)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
contact_person	CHAR(100)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
contact_email	CHAR(100)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

```
1 CREATE TABLE "Suppliers" (  
2     "supplier_id" INTEGER NOT NULL UNIQUE,  
3     "supplier_name" VARCHAR(100) NOT NULL,  
4     "contact_person" VARCHAR(100) NOT NULL,  
5     "contact_email" VARCHAR(100) NOT NULL,  
6     "contact_phone" VARCHAR(11) NOT NULL,  
7     PRIMARY KEY("supplier_id")  
8 );
```

Cancel OK



## Code to create 'Product\_Transactions' Table.

```
CREATE TABLE "Product_Transactions" (  
    "product_transaction_id"    INTEGER NOT NULL UNIQUE,  
    "transaction_id"           INTEGER NOT NULL,  
    "product_id"               INTEGER NOT NULL,  
    "quantity"                 INTEGER NOT NULL,  
    "unit_price"               NUMERIC(10, 2) NOT NULL,  
    PRIMARY KEY("product_transaction_id"),  
    FOREIGN KEY("product_id") REFERENCES "Products"("product_id"),  
    FOREIGN KEY("transaction_id") REFERENCES  
    "Transactions"("transaction_id")  
);
```

The screenshot shows the 'Edit table definition' dialog box for the 'Product\_Transactions' table. The table name is 'Product\_Transactions'. The 'Advanced' section is expanded, showing the 'Fields' tab. The fields are listed in a table with columns for Name, Type, NN, PK, AI, U, Default, and Check.

Name	Type	NN	PK	AI	U	Default	Check
product_transaction_id	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
transaction_id	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
product_id	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
quantity	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

The SQL code is displayed in a text area below the fields table:

```
1 CREATE TABLE "Product_Transactions" (  
2     "product_transaction_id"    INTEGER NOT NULL UNIQUE,  
3     "transaction_id"           INTEGER NOT NULL,  
4     "product_id"               INTEGER NOT NULL,  
5     "quantity"                 INTEGER NOT NULL,  
6     "unit_price"               NUMERIC(10, 2) NOT NULL,  
7     PRIMARY KEY("product_transaction_id"),  
8     FOREIGN KEY("product_id") REFERENCES "Products"("product_id"),  
9     FOREIGN KEY("transaction_id") REFERENCES "Transactions"("transaction_id")  
10 );
```

At the bottom right, there are 'Cancel' and 'OK' buttons.

## Code to create 'Transactions' Table.

```
CREATE TABLE "Transactions" (  
    "transaction_id"      INTEGER NOT NULL UNIQUE,  
    "customer_id" INTEGER NOT NULL,  
    "employee_id" INTEGER NOT NULL,  
    "transaction_date"    DATETIME NOT NULL,  
    "total_amount"        NUMERIC NOT NULL,  
    PRIMARY KEY("transaction_id"),  
    FOREIGN KEY("employee_id") REFERENCES  
"Employees"("employee_id") ON DELETE NO ACTION ON UPDATE NO  
ACTION,  
    FOREIGN KEY("customer_id") REFERENCES  
"Customers"("customer_id") ON DELETE NO ACTION ON UPDATE NO  
ACTION  
);
```

Table

Transactions

Advanced

Fields Constraints

Add Remove Move to top Move up Move down Move...ttom

Name	Type	NN	PK	AI	U	Default	Check
transaction_id	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
customer_id	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
employee_id	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
transaction_date	DATETIME	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

```
1 CREATE TABLE "Transactions" (  
2     "transaction_id" INTEGER NOT NULL UNIQUE,  
3     "customer_id" INTEGER NOT NULL,  
4     "employee_id" INTEGER NOT NULL,  
5     "transaction_date" DATETIME NOT NULL,  
6     "total_amount" NUMERIC NOT NULL,  
7     PRIMARY KEY("transaction_id"),  
8     FOREIGN KEY("customer_id") REFERENCES "Customers"("customer_id") ON DELETE NO A  
9     FOREIGN KEY("employee_id") REFERENCES "Employees"("employee_id") ON DELETE NO A  
10 );
```

Cancel OK

## Code to create 'Customers' Table.

```
CREATE TABLE "Customers" (  
    "customer_id" INTEGER NOT NULL UNIQUE,  
    "first_name" VARCHAR(100) NOT NULL,  
    "last_name" VARCHAR(100) NOT NULL,  
    "email" VARCHAR(100) NOT NULL,  
    "delivery_address" VARCHAR(200) NOT NULL,  
    PRIMARY KEY("customer_id")  
);
```

Table: **Customers**

Advanced

Fields Constraints

Add Remove Move to top Move up Move down Move...ttop

Name	Type	NN	PK	AI	U	Default	Check
customer_id	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
first_name	CHAR(100)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
last_name	CHAR(100)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
email	CHAR(100)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

```
1 CREATE TABLE "Customers" (  
2     "customer_id" INTEGER NOT NULL UNIQUE,  
3     "first_name" VARCHAR(100) NOT NULL,  
4     "last_name" VARCHAR(100) NOT NULL,  
5     "email" VARCHAR(100) NOT NULL,  
6     "delivery_address" VARCHAR(200) NOT NULL,  
7     PRIMARY KEY("customer_id")  
8 );
```

Cancel OK

## Code to create 'Employees' Table.

```
CREATE TABLE "Employees" (  
    "employee_id" INTEGER NOT NULL UNIQUE,  
    "first_name" VARCHAR(100) NOT NULL,  
    "last_name" VARCHAR(100) NOT NULL,  
    "position" VARCHAR(100) NOT NULL,  
    "address" VARCHAR(200) NOT NULL,  
    "email" VARCHAR(100) NOT NULL,  
    "phone_number" VARCHAR(11) NOT NULL,  
    PRIMARY KEY("employee_id")  
);
```

Table: **Employees**

Advanced

Fields Constraints

Add Remove Move to top Move up Move down Move to bottom

Name	Type	NN	PK	AI	U	Default	Check
employee_id	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
first_name	CHAR(100)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
last_name	CHAR(100)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
position	CHAR(100)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

```
1 CREATE TABLE "Employees" (  
2     "employee_id" INTEGER NOT NULL UNIQUE,  
3     "first_name" VARCHAR(100) NOT NULL,  
4     "last_name" VARCHAR(100) NOT NULL,  
5     "position" VARCHAR(100) NOT NULL,  
6     "address" VARCHAR(200) NOT NULL,  
7     "email" VARCHAR(100) NOT NULL,  
8     "phone_number" VARCHAR(11) NOT NULL,  
9     PRIMARY KEY("employee_id")  
10 );
```

Cancel OK

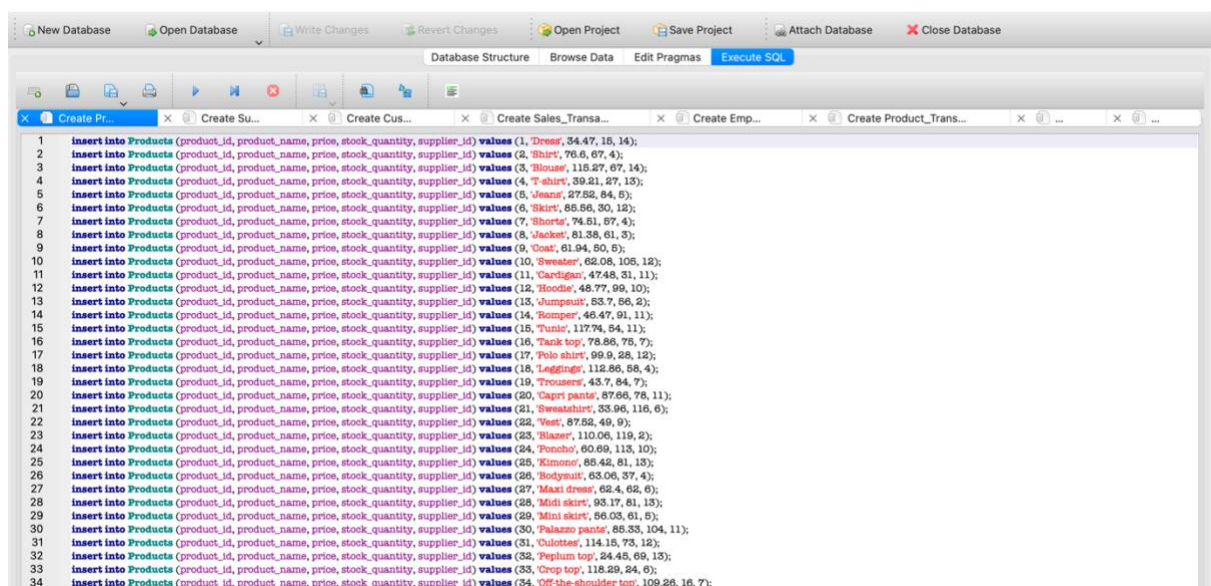
Each table generated.

Name	Type	Schema
Customers		
customer_id	INTEGER	CREATE TABLE "Customers" ( "customer_id" INTEGER NOT NULL UNIQUE, "first_name" VARCHAR(100) NOT NULL, "last_name" VARCHAR(100) NOT NULL, "email" VARCHAR(100) NOT NULL, "delivery_address" VARCHAR(200) NOT NULL, "phone_number" VARCHAR(11) NOT NULL, "primary_key" VARCHAR(100) NOT NULL, "CONSTRAINT" VARCHAR(100) NOT NULL );
first_name	VARCHAR(100)	"first_name" VARCHAR(100) NOT NULL
last_name	VARCHAR(100)	"last_name" VARCHAR(100) NOT NULL
email	VARCHAR(100)	"email" VARCHAR(100) NOT NULL
delivery_address	VARCHAR(200)	"delivery_address" VARCHAR(200) NOT NULL
phone_number	VARCHAR(11)	"phone_number" VARCHAR(11) NOT NULL
primary_key	VARCHAR(100)	"primary_key" VARCHAR(100) NOT NULL
CONSTRAINT	VARCHAR(100)	"CONSTRAINT" VARCHAR(100) NOT NULL
Employees		
employee_id	INTEGER	CREATE TABLE "Employees" ( "employee_id" INTEGER NOT NULL UNIQUE, "first_name" VARCHAR(100) NOT NULL, "last_name" VARCHAR(100) NOT NULL, "position" VARCHAR(100) NOT NULL, "address" VARCHAR(200) NOT NULL, "email" VARCHAR(100) NOT NULL, "phone_number" VARCHAR(11) NOT NULL, "primary_key" VARCHAR(100) NOT NULL, "CONSTRAINT" VARCHAR(100) NOT NULL );
first_name	VARCHAR(100)	"first_name" VARCHAR(100) NOT NULL
last_name	VARCHAR(100)	"last_name" VARCHAR(100) NOT NULL
position	VARCHAR(100)	"position" VARCHAR(100) NOT NULL
address	VARCHAR(200)	"address" VARCHAR(200) NOT NULL
email	VARCHAR(100)	"email" VARCHAR(100) NOT NULL
phone_number	VARCHAR(11)	"phone_number" VARCHAR(11) NOT NULL
primary_key	VARCHAR(100)	"primary_key" VARCHAR(100) NOT NULL
CONSTRAINT	VARCHAR(100)	"CONSTRAINT" VARCHAR(100) NOT NULL
Product_Transactions		
product_transaction_id	INTEGER	CREATE TABLE "Product_Transactions" ( "product_transaction_id" INTEGER NOT NULL UNIQUE, "transaction_id" INTEGER NOT NULL, "product_id" INTEGER NOT NULL, "quantity" INTEGER NOT NULL, "unit_price" NUMERIC(10, 2) NOT NULL, "CONSTRAINT" VARCHAR(100) NOT NULL );
transaction_id	INTEGER	"transaction_id" INTEGER NOT NULL
product_id	INTEGER	"product_id" INTEGER NOT NULL
quantity	INTEGER	"quantity" INTEGER NOT NULL
unit_price	NUMERIC(10, 2)	"unit_price" NUMERIC(10, 2) NOT NULL
CONSTRAINT	VARCHAR(100)	"CONSTRAINT" VARCHAR(100) NOT NULL
Products		
product_id	INTEGER	CREATE TABLE "Products" ( "product_id" INTEGER NOT NULL UNIQUE, "product_name" VARCHAR(100) NOT NULL, "price" NUMERIC(10, 2) NOT NULL, "stock_quantity" INTEGER NOT NULL, "supplier_id" INTEGER NOT NULL, "CONSTRAINT" VARCHAR(100) NOT NULL );
product_name	VARCHAR(100)	"product_name" VARCHAR(100) NOT NULL
price	NUMERIC(10, 2)	"price" NUMERIC(10, 2) NOT NULL
stock_quantity	INTEGER	"stock_quantity" INTEGER NOT NULL
supplier_id	INTEGER	"supplier_id" INTEGER NOT NULL
CONSTRAINT	VARCHAR(100)	"CONSTRAINT" VARCHAR(100) NOT NULL
Suppliers		
supplier_id	INTEGER	CREATE TABLE "Suppliers" ( "supplier_id" INTEGER NOT NULL UNIQUE, "supplier_name" VARCHAR(100) NOT NULL, "contact_person" VARCHAR(100) NOT NULL, "contact_email" VARCHAR(100) NOT NULL, "contact_phone" VARCHAR(11) NOT NULL, "CONSTRAINT" VARCHAR(100) NOT NULL );
supplier_name	VARCHAR(100)	"supplier_name" VARCHAR(100) NOT NULL
contact_person	VARCHAR(100)	"contact_person" VARCHAR(100) NOT NULL
contact_email	VARCHAR(100)	"contact_email" VARCHAR(100) NOT NULL
contact_phone	VARCHAR(11)	"contact_phone" VARCHAR(11) NOT NULL
CONSTRAINT	VARCHAR(100)	"CONSTRAINT" VARCHAR(100) NOT NULL
Transactions		
transaction_id	INTEGER	CREATE TABLE "Transactions" ( "transaction_id" INTEGER NOT NULL UNIQUE, "customer_id" INTEGER NOT NULL, "employee_id" INTEGER NOT NULL, "transaction_date" DATETIME NOT NULL, "total_amount" NUMERIC(10, 2) NOT NULL, "CONSTRAINT" VARCHAR(100) NOT NULL );
customer_id	INTEGER	"customer_id" INTEGER NOT NULL

3. *Populate your database with DML insert statements. You may use data generators such as Mockaroo. Make sure that you include a sample one-year transaction (01/01/2023 - 31/12/2023) on each table. In your final document, there is no need to copy and paste all insert into statements, only 2 to 10 lines would be sufficient.*

### Code to insert values into 'Products' table.

```
insert into Products (product_id, product_name, price, stock_quantity, supplier_id) values (1, 'Dress', 34.47, 15, 14);
insert into Products (product_id, product_name, price, stock_quantity, supplier_id) values (2, 'Shirt', 76.6, 67, 4);
insert into Products (product_id, product_name, price, stock_quantity, supplier_id) values (3, 'Blouse', 115.27, 67, 14);
insert into Products (product_id, product_name, price, stock_quantity, supplier_id) values (4, 'T-shirt', 39.21, 27, 13);
insert into Products (product_id, product_name, price, stock_quantity, supplier_id) values (5, 'Jeans', 27.52, 84, 5);
insert into Products (product_id, product_name, price, stock_quantity, supplier_id) values (6, 'Skirt', 85.56, 30, 12);
insert into Products (product_id, product_name, price, stock_quantity, supplier_id) values (7, 'Shorts', 74.51, 57, 4);
insert into Products (product_id, product_name, price, stock_quantity, supplier_id) values (8, 'Jacket', 81.38, 61, 3);
insert into Products (product_id, product_name, price, stock_quantity, supplier_id) values (9, 'Coat', 61.94, 50, 5);
insert into Products (product_id, product_name, price, stock_quantity, supplier_id) values (10, 'Sweater', 62.08, 105, 12);
```



The screenshot shows a database management tool interface with a menu bar (New Database, Open Database, Write Changes, Revert Changes, Open Project, Save Project, Attach Database, Close Database) and a toolbar. Below the toolbar is a tabbed interface with tabs for 'Create Pr...', 'Create Su...', 'Create Cus...', 'Create Sales\_Transa...', 'Create Emp...', 'Create Product\_Trans...', and others. The 'Create Pr...' tab is active, displaying a list of 34 insert statements for the 'Products' table. The statements are numbered 1 through 34 and follow the same format as the code provided in the previous block, with product names like 'Dress', 'Shirt', 'Blouse', 'T-shirt', 'Jeans', 'Skirt', 'Shorts', 'Jacket', 'Coat', 'Sweater', etc.



## Code to insert values into 'Suppliers' table.

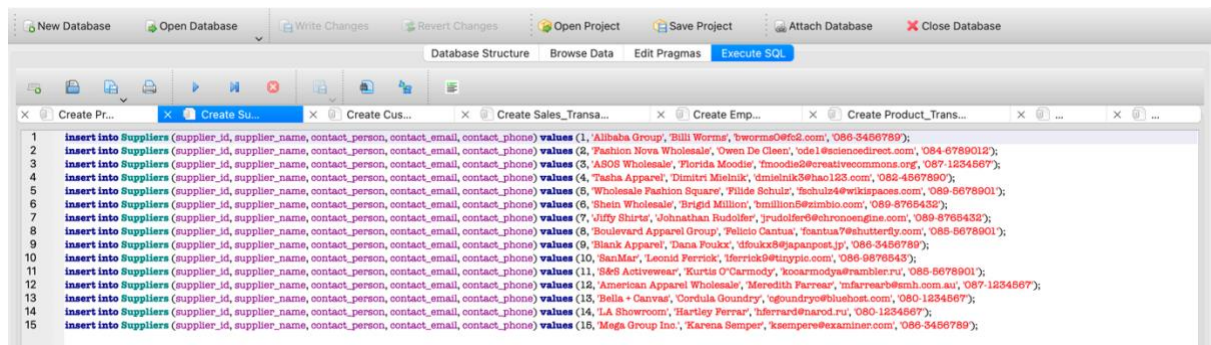
insert into Suppliers (supplier\_id, supplier\_name, contact\_person, contact\_email, contact\_phone) values (1, 'Alibaba Group', 'Billi Worms', 'bworms0@fc2.com', '086-3456789');

insert into Suppliers (supplier\_id, supplier\_name, contact\_person, contact\_email, contact\_phone) values (2, 'Fashion Nova Wholesale', 'Owen De Cleen', 'ode1@sciencedirect.com', '084-6789012');

insert into Suppliers (supplier\_id, supplier\_name, contact\_person, contact\_email, contact\_phone) values (3, 'ASOS Wholesale', 'Florida Moodie', 'fmoodie2@creativecommons.org', '087-1234567');

insert into Suppliers (supplier\_id, supplier\_name, contact\_person, contact\_email, contact\_phone) values (4, 'Tasha Apparel', 'Dimitri Mielnik', 'dmielnik3@hao123.com', '082-4567890');

insert into Suppliers (supplier\_id, supplier\_name, contact\_person, contact\_email, contact\_phone) values (5, 'Wholesale Fashion Square', 'Filide Schulz', 'fschulz4@wikispaces.com', '089-5678901');



```
1 insert into Suppliers (supplier_id, supplier_name, contact_person, contact_email, contact_phone) values (1, 'Alibaba Group', 'Billi Worms', 'bworms0@fc2.com', '086-3456789');
2 insert into Suppliers (supplier_id, supplier_name, contact_person, contact_email, contact_phone) values (2, 'Fashion Nova Wholesale', 'Owen De Cleen', 'ode1@sciencedirect.com', '084-6789012');
3 insert into Suppliers (supplier_id, supplier_name, contact_person, contact_email, contact_phone) values (3, 'ASOS Wholesale', 'Florida Moodie', 'fmoodie2@creativecommons.org', '087-1234567');
4 insert into Suppliers (supplier_id, supplier_name, contact_person, contact_email, contact_phone) values (4, 'Tasha Apparel', 'Dimitri Mielnik', 'dmielnik3@hao123.com', '082-4567890');
5 insert into Suppliers (supplier_id, supplier_name, contact_person, contact_email, contact_phone) values (5, 'Wholesale Fashion Square', 'Filide Schulz', 'fschulz4@wikispaces.com', '089-5678901');
6 insert into Suppliers (supplier_id, supplier_name, contact_person, contact_email, contact_phone) values (6, 'Stain Wholesale', 'Bridgit Millon', 'bmillon@zimbio.com', '089-8765432');
7 insert into Suppliers (supplier_id, supplier_name, contact_person, contact_email, contact_phone) values (7, 'Jiffy Shirts', 'Johnathan Rudolfer', 'jrudolfer@chrononline.com', '089-8765432');
8 insert into Suppliers (supplier_id, supplier_name, contact_person, contact_email, contact_phone) values (8, 'Boulevard Apparel Group', 'Felicio Cantua', 'fcantua7@shutterstock.com', '088-5678901');
9 insert into Suppliers (supplier_id, supplier_name, contact_person, contact_email, contact_phone) values (9, 'Blank Apparel', 'Dana Foukx', 'dfoukx8@japanpost.jp', '086-3456789');
10 insert into Suppliers (supplier_id, supplier_name, contact_person, contact_email, contact_phone) values (10, 'SanMar', 'Leonid Ferrick', 'lferrick9@tinypic.com', '086-9876543');
11 insert into Suppliers (supplier_id, supplier_name, contact_person, contact_email, contact_phone) values (11, 'S&S Activewear', 'Kurtis O'Carmony', 'kocarmody@rambler.ru', '085-6678901');
12 insert into Suppliers (supplier_id, supplier_name, contact_person, contact_email, contact_phone) values (12, 'American Apparel Wholesale', 'Meredith Parver', 'mparver@bush.com.au', '087-1234567');
13 insert into Suppliers (supplier_id, supplier_name, contact_person, contact_email, contact_phone) values (13, 'Bella + Canvas', 'Cordula Goudry', 'cgoudry@bluehost.com', '080-1234567');
14 insert into Suppliers (supplier_id, supplier_name, contact_person, contact_email, contact_phone) values (14, 'LA Showroom', 'Hartley Ferrar', 'hferrard@narod.ru', '080-1234567');
15 insert into Suppliers (supplier_id, supplier_name, contact_person, contact_email, contact_phone) values (15, 'Maga Group Inc', 'Karina Sempere', 'ksempere@examiner.com', '086-3456789');
```

## Code to insert values into 'Product\_Transactions' table.

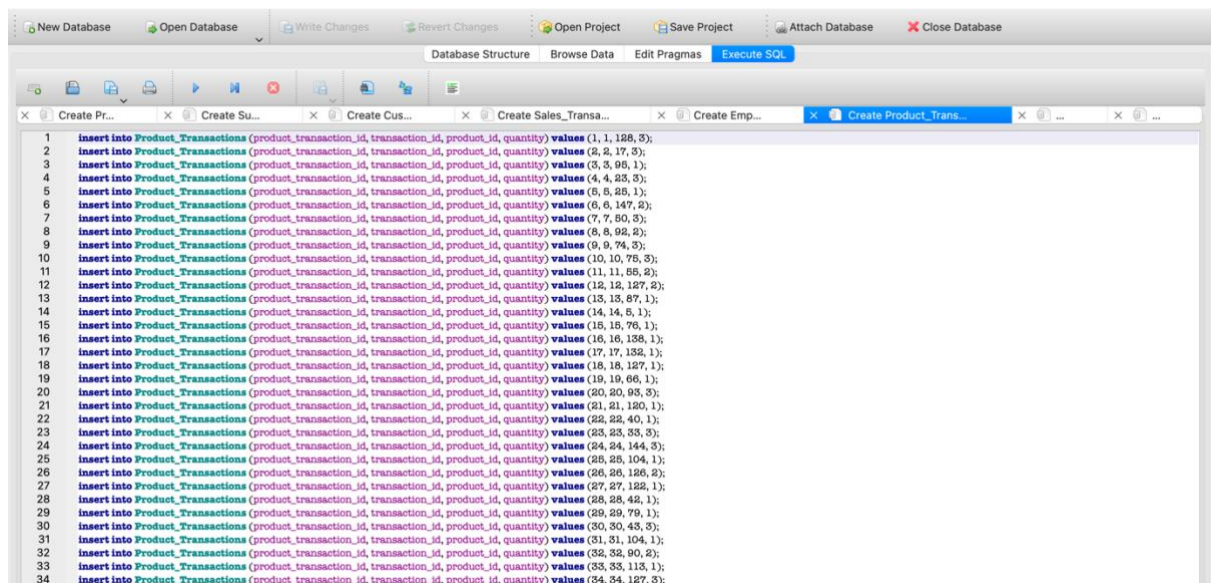
insert into Product\_Transactions (product\_transaction\_id, transaction\_id, product\_id, quantity) values (1, 1, 128, 3);

insert into Product\_Transactions (product\_transaction\_id, transaction\_id, product\_id, quantity) values (2, 2, 17, 3);

insert into Product\_Transactions (product\_transaction\_id, transaction\_id, product\_id, quantity) values (3, 3, 95, 1);

insert into Product\_Transactions (product\_transaction\_id, transaction\_id, product\_id, quantity) values (4, 4, 23, 3);

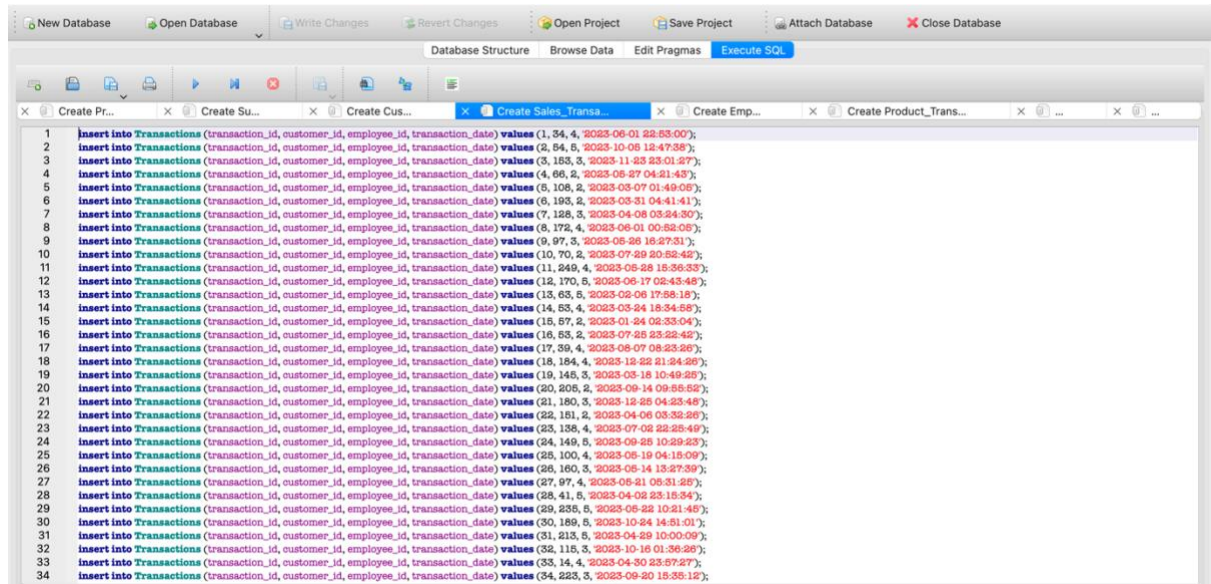
insert into Product\_Transactions (product\_transaction\_id, transaction\_id, product\_id, quantity) values (5, 5, 25, 1);



```
1 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (1, 1, 128, 3);
2 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (2, 2, 17, 3);
3 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (3, 3, 95, 1);
4 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (4, 4, 23, 3);
5 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (5, 5, 25, 1);
6 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (6, 6, 28, 1);
7 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (6, 6, 147, 2);
8 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (7, 7, 80, 3);
9 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (8, 8, 92, 2);
10 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (9, 9, 74, 5);
11 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (10, 10, 75, 3);
12 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (11, 11, 65, 2);
13 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (12, 12, 127, 2);
14 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (13, 13, 87, 1);
15 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (14, 14, 5, 1);
16 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (15, 15, 76, 1);
17 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (16, 16, 138, 1);
18 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (17, 17, 132, 1);
19 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (18, 18, 127, 1);
20 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (19, 19, 66, 1);
21 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (20, 20, 93, 3);
22 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (21, 21, 120, 1);
23 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (22, 22, 40, 1);
24 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (23, 23, 33, 3);
25 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (24, 24, 144, 3);
26 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (25, 25, 104, 1);
27 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (26, 26, 126, 2);
28 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (27, 27, 122, 1);
29 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (28, 28, 42, 1);
30 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (29, 29, 79, 1);
31 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (30, 30, 43, 5);
32 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (31, 31, 104, 1);
33 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (32, 32, 90, 2);
34 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (33, 33, 113, 1);
35 insert into Product_Transactions (product_transaction_id, transaction_id, product_id, quantity) values (34, 34, 127, 3);
```

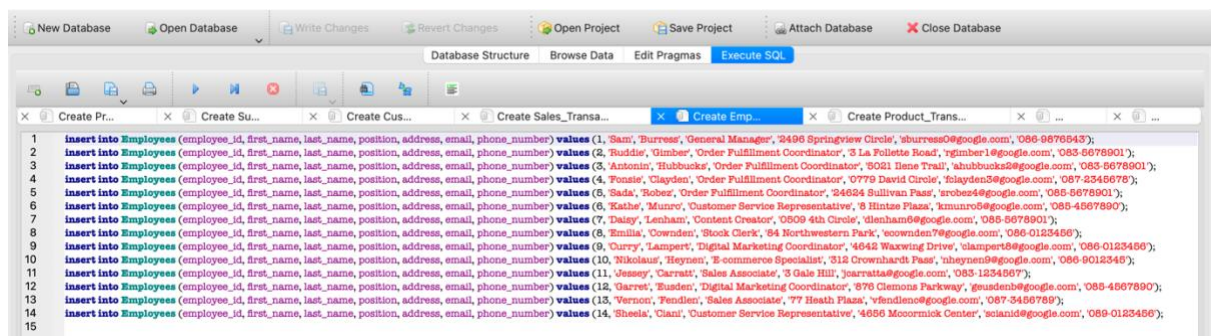
## Code to insert values into 'Transactions' table.

```
insert into Transactions (transaction_id, customer_id, employee_id, transaction_date) values (1, 34, 4, '2023-06-01 22:53:00');
insert into Transactions (transaction_id, customer_id, employee_id, transaction_date) values (2, 54, 5, '2023-10-05 12:47:38');
insert into Transactions (transaction_id, customer_id, employee_id, transaction_date) values (3, 153, 3, '2023-11-23 23:01:27');
insert into Transactions (transaction_id, customer_id, employee_id, transaction_date) values (4, 66, 2, '2023-05-27 04:21:43');
insert into Transactions (transaction_id, customer_id, employee_id, transaction_date) values (5, 108, 2, '2023-03-07 01:49:05');
```



## Code to insert values into 'Employees' table.

```
insert into Employees (employee_id, first_name, last_name, position, address, email, phone_number) values (1, 'Sam',
'Burress', 'General Manager', '2496 Springview Circle', 'sburress0@google.com', '086-9876543');
insert into Employees (employee_id, first_name, last_name, position, address, email, phone_number) values (2, 'Ruddie',
'Gimber', 'Order Fulfillment Coordinator', '3 La Follette Road', 'rgimber1@google.com', '083-5678901');
insert into Employees (employee_id, first_name, last_name, position, address, email, phone_number) values (3, 'Antonin',
'Hubbucks', 'Order Fulfillment Coordinator', '5021 Ilene Trail', 'ahubbucks2@google.com', '083-5678901');
insert into Employees (employee_id, first_name, last_name, position, address, email, phone_number) values (4, 'Fonsie',
'Clayden', 'Order Fulfillment Coordinator', '0779 David Circle', 'fclayden3@google.com', '087-2345678');
insert into Employees (employee_id, first_name, last_name, position, address, email, phone_number) values (5, 'Sada',
'Robez', 'Order Fulfillment Coordinator', '24624 Sullivan Pass', 'srobez4@google.com', '085-5678901');
```





## Code to insert values into 'Customers' table.

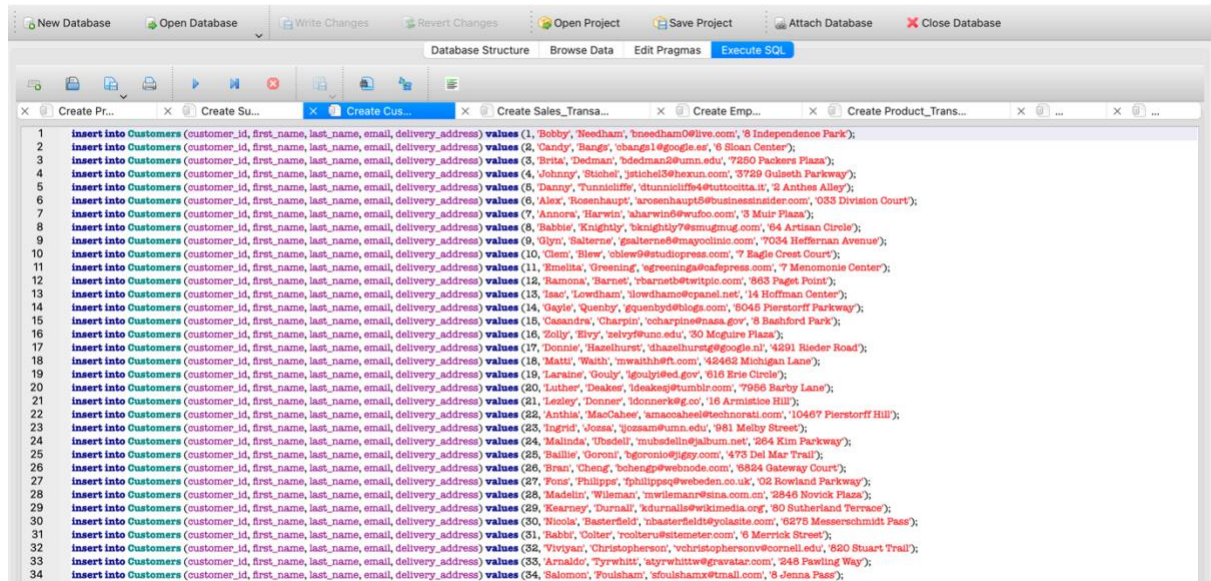
```
insert into Customers (customer_id, first_name, last_name, email, delivery_address) values (1, 'Bobby', 'Needham', 'bneedham0@live.com', '8 Independence Park');
```

```
insert into Customers (customer_id, first_name, last_name, email, delivery_address) values (2, 'Candy', 'Bangs', 'cbangs1@google.es', '6 Sloan Center');
```

```
insert into Customers (customer_id, first_name, last_name, email, delivery_address) values (3, 'Brita', 'Dedman', 'bdedman2@umn.edu', '7250 Packers Plaza');
```

```
insert into Customers (customer_id, first_name, last_name, email, delivery_address) values (4, 'Johnny', 'Stichel', 'jstichel3@hexun.com', '3729 Gulseth Parkway');
```

```
insert into Customers (customer_id, first_name, last_name, email, delivery_address) values (5, 'Danny', 'Tunncliffe', 'dtunncliffe4@tuttocitta.it', '2 Anthes Alley');
```



The screenshot shows a database management interface with a toolbar at the top containing buttons like 'New Database', 'Open Database', 'Write Changes', 'Revert Changes', 'Open Project', 'Save Project', 'Attach Database', and 'Close Database'. Below the toolbar is a menu bar with 'Database Structure', 'Browse Data', 'Edit Pragma', and 'Execute SQL'. The main window displays a list of 34 INSERT statements for the Customers table, each with a unique customer ID, first name, last name, email, and delivery address. The statements are numbered 1 through 34 and are all of the form: `insert into Customers (customer_id, first_name, last_name, email, delivery_address) values (ID, 'First Name', 'Last Name', 'Email', 'Address');`

**Code to find 'unit\_price' with results to be copied and pasted into 'unit\_price' column in 'Product\_Transactions' table. This avoids different values between 'price' column in 'Products' table and 'unit\_price' column in the 'Product\_Transactions' table.**

```
select products.*, Product_Transactions.product_transaction_id,  
Product_Transactions.transaction_id, Product_Transactions.product_id,  
Product_Transactions.quantity, Products.price as 'unit_price'  
from products, Product_Transactions  
where products.product_id = Product_Transactions.product_id  
order by Product_Transactions.product_transaction_id asc
```



Code to find 'unit\_price' with results to be copied and pasted into

```

1 /*Find unit_price*/
2 select products.*, Product_Transactions.product_transaction_id, Product_Transactions.transaction_id, Product_Transactions.product_id, Product_Transactions.quantity, Products.price as 'unit_price'
3 from products, Product_Transactions
4 where products.product_id = Product_Transactions.product_id
5 order by Product_Transactions.product_transaction_id asc

```

	product_id	product_name	price	stock_quantity	supplier_id	product_transaction_id	transaction_id	product_id	quantity	unit_price
1	128	Lehenga	78.59	54	8	1	1	128	3	78.59
2	17	Polo shirt	99.9	28	12	2	2	17	3	99.9
3	95	Belt	60.97	32	7	3	3	95	1	60.97
4	23	Blazer	110.06	119	2	4	4	23	3	110.06
5	25	Kimono	85.42	81	13	5	5	25	1	85.42
6	147	Mantilla	60.22	100	1	6	6	147	2	60.22
7	50	Trench coat	94.95	36	5	7	7	50	3	94.95
8	92	Beanie	52.96	109	6	8	8	92	2	52.96
9	74	Sandals	70.23	85	12	9	9	74	3	70.23
10	75	Espadrilles	32.5	26	11	10	10	75	3	32.5
11	55	Bomber jacket	100.41	114	3	11	11	55	2	100.41
12	127	Cheongsam	49.01	80	2	12	12	127	2	49.01
13	87	Satchel	92.37	73	5	13	13	87	1	92.37
14	5	Jeans	27.52	84	5	14	14	5	1	27.52
15	76	Mules	58.51	20	7	15	15	76	1	58.51
16	138	Pashmina shawl	85.72	43	5	16	16	138	1	85.72
17	132	Harem pants	42.75	29	11	17	17	132	1	42.75
18	127	Cheongsam	49.01	80	2	18	18	127	1	49.01
19	66	Skinny jeans	24.35	31	1	19	19	66	1	24.35
20	93	Beret	109.77	10	7	20	20	93	3	109.77
21	120	Swing coat	101.7	26	15	21	21	120	1	101.7
22	40	Shift dress	94.6	33	5	22	22	40	1	94.6
23	33	Crop top	118.29	24	6	23	23	33	3	118.29
24	144	Galters	81.06	84	2	24	24	144	3	81.06
25	104	Sarong	32.45	62	10	25	25	104	1	32.45

Code to find 'total\_amount' with results to be copied and pasted into

'total\_amount' column in 'Transactions' table. This is the multiplication of the 'quantity' value and the 'unit\_price' value in the 'Product\_Transactions' table.

```

select product_transactions.*, Transactions.transaction_id, Transactions.customer_id,
Transactions.employee_id, Transactions.transaction_date, quantity * unit_price as
'total_amount'
from Product_Transactions, Transactions
where Product_Transactions.transaction_id = Transactions.transaction_id
order by Transactions.transaction_id asc

```

Code to find 'total\_amount' with results to be copied and pasted into

```

1 /* Find total_amount */
2 select product_transactions.*, Transactions.transaction_id, Transactions.customer_id, Transactions.employee_id, Transactions.transaction_date, quantity * unit_price as 'total_amount'
3 from Product_Transactions, Transactions
4 where Product_Transactions.transaction_id = Transactions.transaction_id
5 order by Transactions.transaction_id asc

```

	product_transaction_id	transaction_id	product_id	quantity	unit_price	transaction_id	customer_id	employee_id	transaction_date	total_amount
1	1	1	128	3	78.59	1	34	4	2023-06-01 22:53:00	235.77
2	2	2	17	3	99.9	2	54	5	2023-10-05 12:47:38	299.7
3	3	3	95	1	60.97	3	153	3	2023-11-23 23:01:27	60.97
4	4	4	23	3	110.06	4	66	2	2023-05-27 04:21:43	330.18
5	5	5	25	1	85.42	5	108	2	2023-03-07 01:49:05	85.42
6	6	6	147	2	60.22	6	193	2	2023-03-31 04:41:41	120.44
7	7	7	50	3	94.95	7	128	3	2023-04-08 03:24:30	284.85
8	8	8	92	2	52.96	8	172	4	2023-06-01 00:52:05	105.92
9	9	9	74	3	70.23	9	97	3	2023-05-26 16:27:31	210.69
10	10	10	75	3	32.5	10	70	2	2023-07-29 20:52:42	97.5
11	11	11	55	2	100.41	11	249	4	2023-05-28 15:36:33	200.82
12	12	12	127	2	49.01	12	170	5	2023-06-17 02:43:48	98.02
13	13	13	87	1	92.37	13	63	5	2023-02-06 17:58:18	92.37
14	14	14	5	1	27.52	14	53	4	2023-03-24 18:34:58	27.52
15	15	15	76	1	58.51	15	57	2	2023-01-24 02:33:04	58.51
16	16	16	138	1	85.72	16	53	2	2023-07-25 23:22:42	85.72
17	17	17	132	1	42.75	17	39	4	2023-08-07 06:23:26	42.75
18	18	18	127	1	49.01	18	184	4	2023-12-22 21:24:26	49.01
19	19	19	66	1	24.35	19	145	3	2023-03-18 10:49:25	24.35
20	20	20	93	3	109.77	20	205	2	2023-09-14 09:55:52	329.31

### Part III: Write SQL Statements to answer the following queries

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

/\*Code for question 1\*/

```
select *
from Products
where Price > 100
order by price asc
```

```
1  /*Show all the details of the products that have a price greater than 100.*/
2  select *
3  from Products
4  where Price > 100
5  order by price asc
```

	product_id	product_name	price	stock_quantity	supplier_id
1	55	Bomber jacket	100.41	114	3
2	120	Swing coat	101.7	26	15
3	135	Sherwani	103.15	63	15
4	45	Joggers	103.84	57	6
5	137	Chemisette	105.5	96	6
6	61	Ruffle blouse	105.74	69	5
7	62	Peter Pan collar dress	107.53	10	8
8	36	Tie-neck blouse	107.69	13	1
9	141	Caftan	108.98	43	14
10	133	Kilt	109.18	35	5
11	34	Off-the-shoulder top	109.26	16	7
12	93	Beret	109.77	10	7
13	125	Sarafan dress	109.97	115	10
14	23	Blazer	110.06	119	2
15	122	Paperbag waist pants	110.51	111	15
16	101	Onesie	111.24	86	7
17	140	Sari	111.39	108	11
18	18	Leggings	112.86	58	4
19	41	Sheath dress	113.53	60	10
20	31	Culottes	114.15	73	12
21	3	Blouse	115.27	67	14

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

/\*Code for question 2\*/

```
select products.product_id, Products.product_name, Products.price, Products.stock_quantity,
Suppliers.*
from Products, Suppliers
where Products.supplier_id = Suppliers.supplier_id
```

```
1  /*Show all the products along with the supplier detail who supplied the products.*/
2  select products.product_id, Products.product_name, Products.price, Products.stock_quantity, Suppliers.*
3  from Products, Suppliers
4  where Products.supplier_id = Suppliers.supplier_id
```

	product_id	product_name	price	stock_quantity	supplier_id	supplier_name	contact_person	contact_email	contact_phone
1	1	Dress	34.47	15	14	LA Showroom	Hartley Ferrar	hferrard@snad.ru	080-1234567
2	2	Shirt	76.6	67	4	Tasha Apparel	Dimitri Mielnik	dmieink3@hao123.com	082-4567890
3	3	Blouse	115.27	67	14	LA Showroom	Hartley Ferrar	hferrard@snad.ru	080-1234567
4	4	T-shirt	39.21	27	13	Bella + Canvas	Cordula Goundry	cgoundryc@bluehost.com	080-1234567
5	5	Jeans	27.52	84	5	Wholesale Fashion Square	Filide Schulz	fschulz4@wikispaces.com	089-5678901
6	6	Skirt	85.56	30	12	American Apparel Wholesale	Meredith Farnear	mfarnearb@snh.com.au	087-1234567
7	7	Shorts	74.51	57	4	Tasha Apparel	Dimitri Mielnik	dmieink3@hao123.com	082-4567890
8	8	Jacket	81.38	61	3	ASOS Wholesale	Florida Moodie	fmoodie2@creativecommons.org	087-1234567
9	9	Coat	61.94	50	5	Wholesale Fashion Square	Filide Schulz	fschulz4@wikispaces.com	089-5678901
10	10	Sweater	62.08	105	12	American Apparel Wholesale	Meredith Farnear	mfarnearb@snh.com.au	087-1234567
11	11	Cardigan	47.48	31	11	S&S Activewear	Kurtis O'Carmony	kocarmodya@rambler.ru	085-5678901
12	12	Hoodie	48.77	99	10	SanMar	Leonid Ferrick	lferrick9@tinypic.com	086-9876543
13	13	Jumpsuit	53.7	56	2	Fashion Nova Wholesale	Owen De Cleen	ode1@sciencedirect.com	084-6789012
14	14	Rompers	46.47	91	11	S&S Activewear	Kurtis O'Carmony	kocarmodya@rambler.ru	085-5678901
15	15	Tunic	117.74	54	11	S&S Activewear	Kurtis O'Carmony	kocarmodya@rambler.ru	085-5678901
16	16	Tank top	78.86	75	7	Jiffy Shirts	Johnathan Rudolfer	jrudolfer@chronosengine.com	089-8765432
17	17	Polo shirt	99.9	28	12	American Apparel Wholesale	Meredith Farnear	mfarnearb@snh.com.au	087-1234567
18	18	Leggings	112.86	58	4	Tasha Apparel	Dimitri Mielnik	dmieink3@hao123.com	082-4567890
19	19	Trousers	43.7	84	7	Jiffy Shirts	Johnathan Rudolfer	jrudolfer@chronosengine.com	089-8765432
20	20	Capri pants	87.66	78	11	S&S Activewear	Kurtis O'Carmony	kocarmodya@rambler.ru	085-5678901
21	21	Sweatshirt	33.96	116	6	Shein Wholesale	Brigid Millon	bmillon5@zeibio.com	089-8765432
22	22	Vest	87.52	49	9	Blank Apparel	Dana Foulx	dfoulx8@apartpost.jp	086-3456789
23	23	Blazer	110.06	119	2	Fashion Nova Wholesale	Owen De Cleen	ode1@sciencedirect.com	084-6789012
24	24	Poncho	60.69	113	10	SanMar	Leonid Ferrick	lferrick9@tinypic.com	086-9876543
25	25	Kimono	85.42	81	13	Bella + Canvas	Cordula Goundry	cgoundryc@bluehost.com	080-1234567

3. Show all the products bought by the top 5 customers (the customers that spent the most in the store).

/\*Code for question 3\*/

```
select Customers.customer_id, Customers.first_name, Customers.last_name,
Products.product_name,
Product_Transactions.quantity, Product_Transactions.unit_price,
Transactions.transaction_date, Transactions.total_amount
from Products, Product_Transactions, Transactions, Customers
where Product_Transactions .product_id = Products.product_id and
Transactions.transaction_id = Product_Transactions.transaction_id
and Customers.customer_id = Transactions.customer_id
order by Transactions.total_amount desc
limit 5
```

```
1  /*Show all the products bought by the top 5 customers (the customers that spent the most in the store).*/
2  select Customers.customer_id, Customers.first_name, Customers.last_name, Products.product_name,
3  Product_Transactions.quantity, Product_Transactions.unit_price, Transactions.transaction_date, Transactions.total_amount
4  from Products, Product_Transactions, Transactions, Customers
5  where Product_Transactions .product_id = Products.product_id and Transactions.transaction_id = Product_Transactions.transaction_id
6  and Customers.customer_id = Transactions.customer_id
7  order by Transactions.total_amount desc
8  limit 5
```

	customer_id	first_name	last_name	product_name	quantity	unit_price	transaction_date	total_amount
1	59	Blane	Ivashnikov	Snood	3	119.93	2023-06-20 17:20:54	359.79
2	138	Nathaniel	Colter	Crop top	3	118.29	2023-07-02 22:25:49	354.87
3	130	Conny	Meikle	Crop top	3	118.29	2023-08-11 18:16:42	354.87
4	33	Arnaldo	Tyrwhitt	Crop top	3	118.29	2023-06-12 06:43:32	354.87
5	10	Clem	Blew	Clutch	3	117.93	2023-05-11 15:39:42	353.79

4. Show the information of the top 3 suppliers with more products.

/\*Code for question 4\*/

```
select Suppliers.*, count(Products.product_id) as 'Total Products Supplied'
from Suppliers, Products
where Products.supplier_id = Suppliers.supplier_id
group by Suppliers.supplier_id
order by count(Products.product_id) desc
limit 3
```

```
1  /*Show the information of the top 3 suppliers with more products. */
2  select Suppliers.*, count(Products.product_id) as 'Total Products Supplied'
3  from Suppliers, Products
4  where Products.supplier_id = Suppliers.supplier_id
5  group by Suppliers.supplier_id
6  order by count(Products.product_id) desc
7  limit 3
```

	supplier_id	supplier_name	contact_person	contact_email	contact_phone	Total Products Supplied
1	6	Shein Wholesale	Brigid Million	bmillion5@zimbio.com	089-8765432	17
2	10	SanMar	Leonid Ferrick	lferrick9@tinypic.com	086-9876543	13
3	5	Wholesale Fashion Square	Filide Schulz	fschulz4@wikispaces.com	089-5678901	13

5. *What is the product with the highest value of sales?*

/\*Code for question 5\*/

```
select Products.*, sum(Product_Transactions.quantity) as 'No. of Products Sold',  
sum(Product_Transactions.quantity * Product_Transactions.unit_price) as 'Sales Value'  
from Products, Product_Transactions  
where Product_Transactions.product_id = Products.product_id  
group by Products.product_id  
order by sum(Product_Transactions.quantity * Product_Transactions.unit_price) desc  
limit 1
```

```
1  /*What is the product with the highest value of sales? */
2  select Products.*, sum(Product_Transactions.quantity) as 'No. of Products Sold', sum(Product_Transactions.quantity * Product_Transactions.unit_price) as 'Sales Value'
3  from Products, Product_Transactions
4  where Product_Transactions.product_id = Products.product_id
5  group by Products.product_id
6  order by sum(Product_Transactions.quantity * Product_Transactions.unit_price) desc
7  limit 1
```

	product_id	product_name	price	stock_quantity	supplier_id	No. of Products Sold	Sales Value
1	126	Zoot suit	88.89	113	9	25	2222.25

6. *Sum the total sales by month.*

/\*Code for question 6\*/

```
select strftime('%m', transaction_date) as 'Month', sum (total_amount) as 'Total Sales'  
from Transactions  
group by strftime('%m', transaction_date)  
order by strftime('%m', transaction_date) asc
```

```
1  /*Sum the total sales by month.*/
2  select strftime('%m', transaction_date) as 'Month', sum (total_amount) as 'Total Sales'
3  from Transactions
4  group by strftime('%m', transaction_date)
5  order by strftime('%m', transaction_date) asc
6
```

	Month	Total Sales
1	01	10078.93
2	02	9316.49
3	03	11067.01
4	04	12187.21
5	05	10165.78
6	06	11270.72
7	07	13083.08
8	08	11940.54
9	09	14367.14
10	10	12124.96
11	11	11853.61
12	12	9401.46

7. Create a view that shows the total number of items a customer bought from the business in October 2023 along with the total price.

/\*Code for question 7\*/

```
select strftime('%m', transaction_date) as 'Month', count(transaction_id) as 'No. of Transactions', sum(Transactions.total_amount) as 'Total Price'
from Transactions
where strftime('%m', transaction_date) = '10'
group by strftime('%m', transaction_date)
```

```
1  /*Create a view that shows the total number of items a customer bought from the business in October 2023 along with the total price. */
2  select strftime('%m', transaction_date) as 'Month', count(transaction_id) as 'No. of Transactions', sum(Transactions.total_amount) as 'Total Price'
3  from Transactions
4  where strftime('%m', transaction_date) = '10'
5  group by strftime('%m', transaction_date)
```

	Month	No. of Transactions	Total Price
1	10	87	12124.96

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

/\*Code for question 8\*/

```
delete from Customers
where customer_id not in (select distinct customer_id from Transactions);
```

```
1  /*Delete all customers who never bought a product from the business. */
2  delete from Customers
3  where customer_id not in (select distinct customer_id from Transactions);
```

9. List all the customers whose name starts with B.

/\*Code for question 9\*/

```
select *
from Customers
where first_name like 'b%'
order by customer_id
```

```
1  /*List all the customers whose name starts with B.*/
2  select *
3  from Customers
4  where first_name like 'b%'
5  order by customer_id
```

	customer_id	first_name	last_name	email	delivery_address
1	1	Bobby	Needham	bneedham0@live.com	8 Independence Park
2	3	Brita	Dedman	bdedman2@umn.edu	7250 Packers Plaza
3	8	Babbie	Knightly	bknightly7@smugmug.com	64 Artisan Circle
4	25	Baillie	Goroni	bgoronio@jigsy.com	473 Del Mar Trail
5	26	Bran	Cheng	bchengp@webnode.com	6824 Gateway Court
6	59	Blane	Ivashnikov	bivashnikov1m@wikipedia.org	8928 American Point
7	70	Berkeley	Fibbitts	bfibbitts1x@tripadvisor.com	420 Manufacturers Lane
8	77	Bertrand	Harrowsmith	bharrowsmith24@yahoo.co.jp	9 Sycamore Circle
9	119	Brig	Sanbroke	bsanbroke3a@mozilla.com	807 Loomis Place
10	139	Berty	Colvine	bcolvine3u@ebay.com	712 Tennyson Plaza
11	144	Bessy	Heersma	bheersma3z@rambler.ru	98 Forest Dale Center
12	180	Bartlet	Greenstreet	bgreenstreet42@ask.com	7434 Westend Hill
13	181	Barbaraanne	Duddin	bduddin50@ed.gov	71143 Cambridge Avenue
14	183	Berke	Denyukhin	bdenyukhin52@ft.com	26 Delaware Lane
15	193	Balduin	Brockest	brockest5c@sohu.com	66 Commercial Pass
16	212	Bryce	Naris	bnaris5v@sphinn.com	3 Village Road
17	213	Baldwin	Warburton	bwarburton5w@whitehouse.gov	5 Paget Way
18	216	Brandtr	Unthank	bunthank5z@etsy.com	3904 Tomscot Hill
19	222	Bryanty	Picot	bpicot65@tuttocitta.it	492 Hazelcrest Point

10. What supplier sold more products?

/\*Code for question 10\*/

```
select Suppliers.*, count(Products.supplier_id) as 'Total Products Sold'
from Suppliers, Products
where Products.supplier_id = Suppliers.supplier_id
group by Products.supplier_id
order by count(Products.supplier_id) desc
limit 1
```

```
1  /*What supplier sold more products?*/
2  select Suppliers.*, count(Products.supplier_id) as 'Total Products Sold'
3  from Suppliers, Products
4  where Products.supplier_id = Suppliers.supplier_id
5  group by Products.supplier_id
6  order by count(Products.supplier_id) desc
7  limit 1
```

	supplier_id	supplier_name	contact_person	contact_email	contact_phone	Total Products Sold
1	6	Shein Wholesale	Brigid Million	bmillion5@zimbio.com	089-8765432	17

## References

- Roll, M. (2021). *The Secret of Zara's Success: a Culture of Customer Co-creation* / Martin Roll. [online] Martin Roll. Available at:  
<https://martinroll.com/resources/articles/strategy/the-secret-of-zaras-success-a-culture-of-customer-co-creation/>. [Accessed 14 Nov. 2023].
- Wondershare EdrawMind. (n.d.). *Zara Mission and Vision Statement Analysis*. [online] Edrawsoft. Available at: <https://www.edrawmind.com/article/zara-mission-and-vision-statement-analysis.html>. [Accessed 14 Nov. 2023].
- Zara. (2023). *Company + Info* / ZARA Ireland. [online] Zara.com. Available at:  
<https://www.zara.com/ie/en/z-company-corp1391.html?v1=2313140>. [Accessed 14 Nov. 2023].