Introduction:

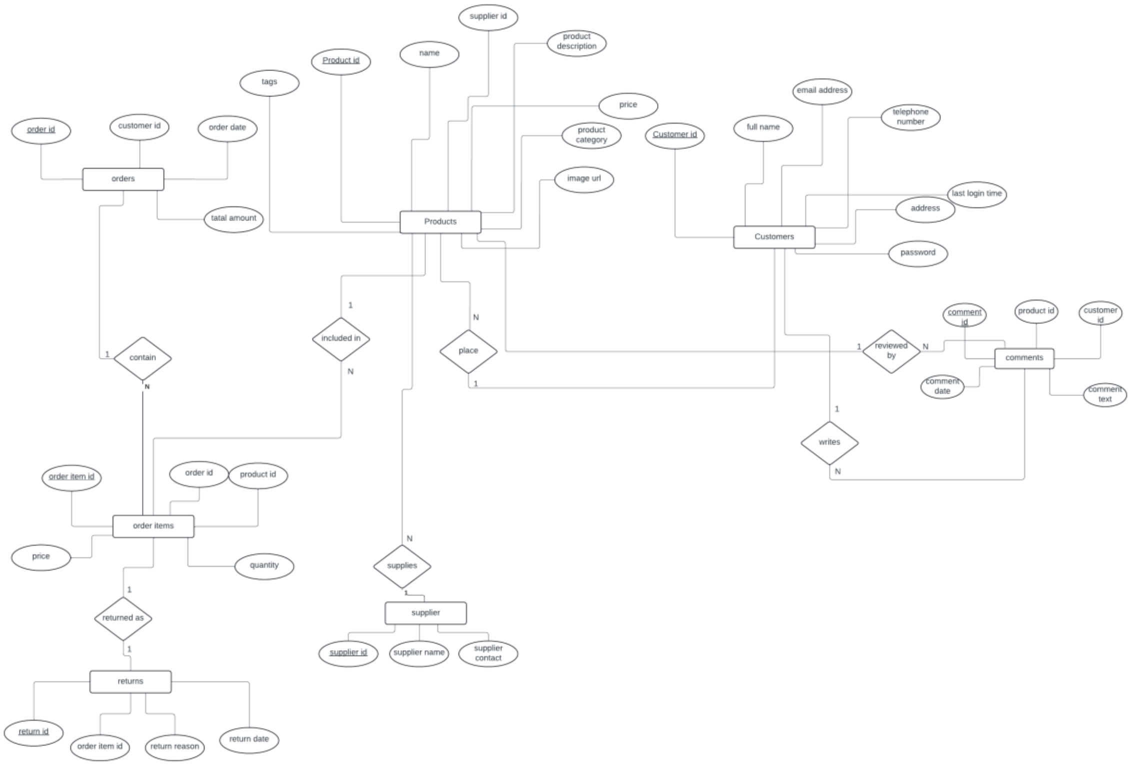
This report is about the development of a data base schema for ABC company, an online shop that sells items all over Europe. The goal is to store and manage customer information, product information, orders, returns and comments.

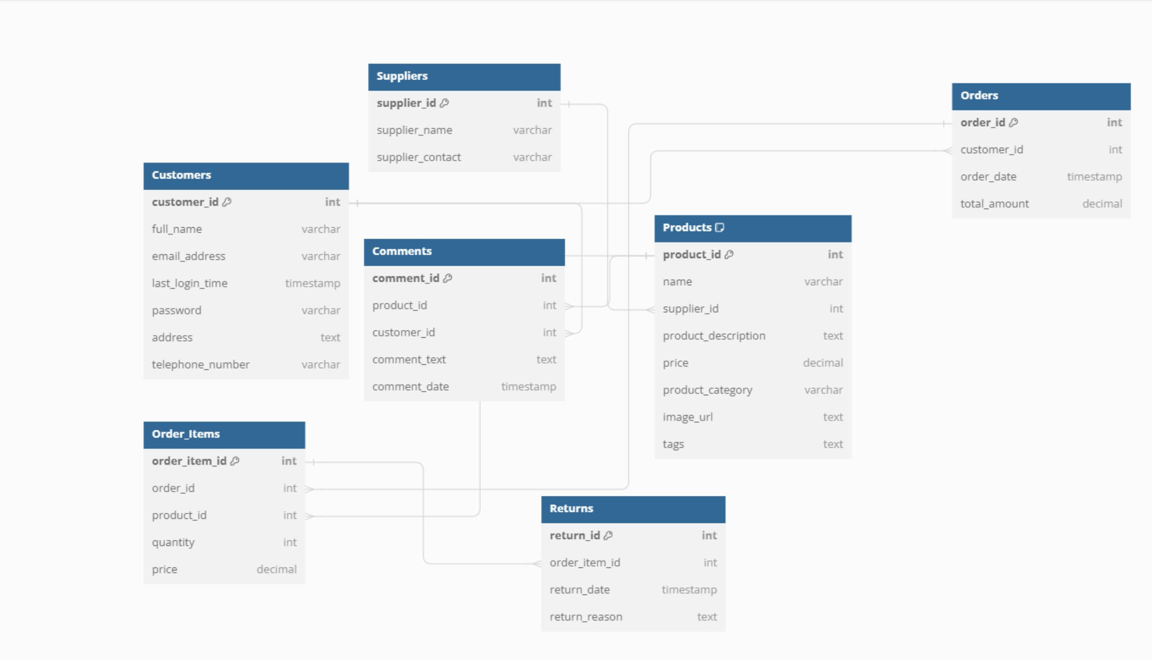
The objectives of the project are:

- Data base schema (ER Diagram)
- Implementing the data base with Maria db.
- Add sample data
- Write sql queries relevant to the database and business problem

Data base design:

ER Diagram:





Suppliers	
supplier_id	int
supplier_name	varchar
supplier_contact	varchar

Orders	
order_id	int
customer_id	int
order_date	timestamp
total_amount	decimal

Customers	
customer_id	int
full_name	varchar
email_address	varchar
last_login_time	timestamp
password	varchar
address	text
telephone_number	varchar

Comments	
comment_id	int
product_id	int
customer_id	int
comment_text	text
comment_date	timestamp

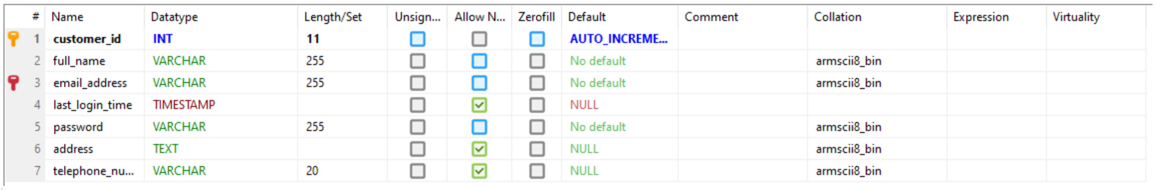
Products	
product_id	int
name	varchar
supplier_id	int
product_description	text
price	decimal
product_category	varchar
image_url	text
tags	text

Order_Items	
order_item_id	int
order_id	int
product_id	int
quantity	int
price	decimal

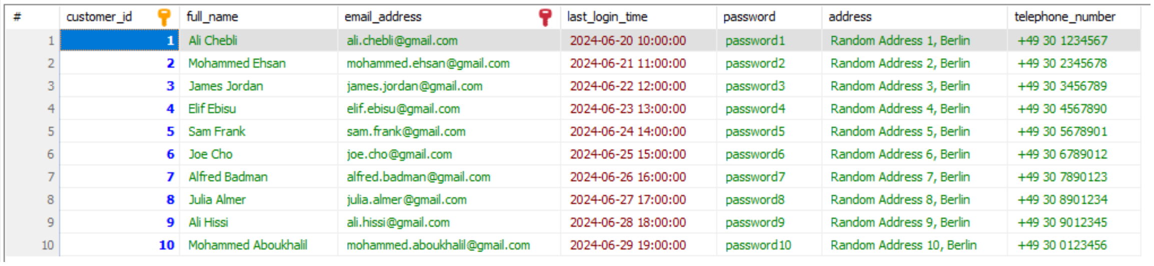
Returns	
return_id	int
order_item_id	int
return_date	timestamp
return_reason	text

Tables:

- Customers: contains customer info:

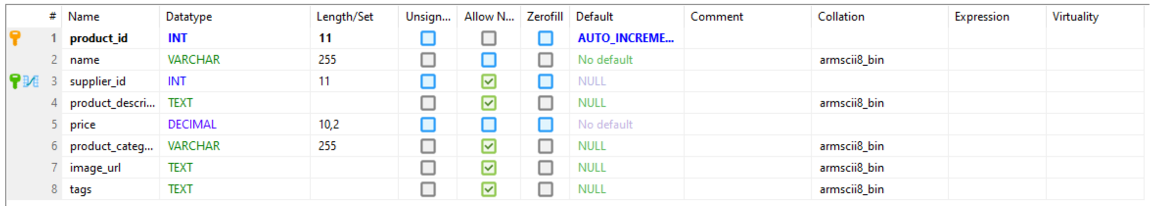


#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	**customer_id**	INT	11	□	□	□	AUTO_INCREME...				
2	full_name	VARCHAR	255	□	□	□	No default		armscii8_bin		
3	email_address	VARCHAR	255	□	□	□	No default		armscii8_bin		
4	last_login_time	TIMESTAMP		□	☑	□	NULL				
5	password	VARCHAR	255	□	□	□	No default		armscii8_bin		
6	address	TEXT		□	☑	□	NULL		armscii8_bin		
7	telephone_nu...	VARCHAR	20	□	☑	□	NULL		armscii8_bin		

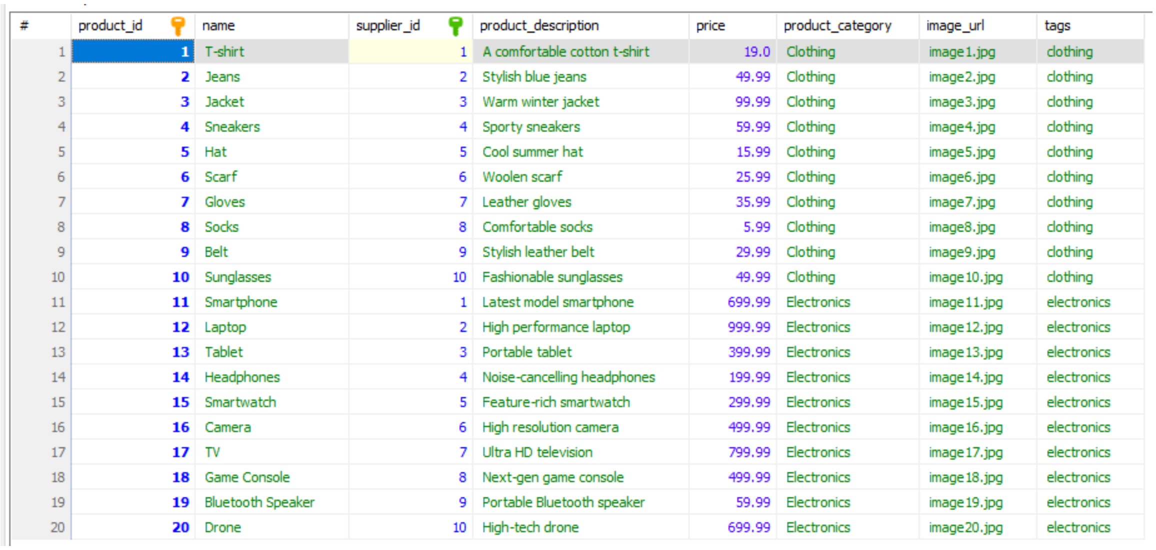


#	customer_id	full_name	email_address	last_login_time	password	address	telephone_number
1	1	Ali Chebli	ali.chebli@gmail.com	2024-06-20 10:00:00	password1	Random Address 1, Berlin	+49 30 1234567
2	2	Mohammed Ehsan	mohammed.ehsan@gmail.com	2024-06-21 11:00:00	password2	Random Address 2, Berlin	+49 30 2345678
3	3	James Jordan	james.jordan@gmail.com	2024-06-22 12:00:00	password3	Random Address 3, Berlin	+49 30 3456789
4	4	Elif Ebisu	elif.ebisu@gmail.com	2024-06-23 13:00:00	password4	Random Address 4, Berlin	+49 30 4567890
5	5	Sam Frank	sam.frank@gmail.com	2024-06-24 14:00:00	password5	Random Address 5, Berlin	+49 30 5678901
6	6	Joe Cho	joe.cho@gmail.com	2024-06-25 15:00:00	password6	Random Address 6, Berlin	+49 30 6789012
7	7	Alfred Badman	alfred.badman@gmail.com	2024-06-26 16:00:00	password7	Random Address 7, Berlin	+49 30 7890123
8	8	Julia Almer	julia.almer@gmail.com	2024-06-27 17:00:00	password8	Random Address 8, Berlin	+49 30 8901234
9	9	Ali Hissi	ali.hissi@gmail.com	2024-06-28 18:00:00	password9	Random Address 9, Berlin	+49 30 9012345
10	10	Mohammed Aboukhalil	mohammed.aboukhalil@gmail.com	2024-06-29 19:00:00	password10	Random Address 10, Berlin	+49 30 0123456

- Products: Contains product information:

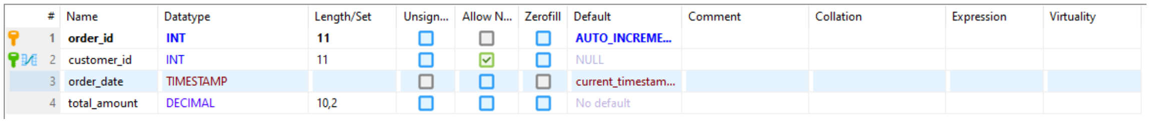


#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	**product_id**	INT	11	□	□	□	AUTO_INCREME...				
2	name	VARCHAR	255	□	□	□	No default		armscii8_bin		
3	supplier_id	INT	11	□	☑	□	NULL				
4	product_descri...	TEXT		□	☑	□	NULL		armscii8_bin		
5	price	DECIMAL	10,2	□	□	□	No default				
6	product_categ...	VARCHAR	255	□	☑	□	NULL		armscii8_bin		
7	image_url	TEXT		□	☑	□	NULL		armscii8_bin		
8	tags	TEXT		□	☑	□	NULL		armscii8_bin		

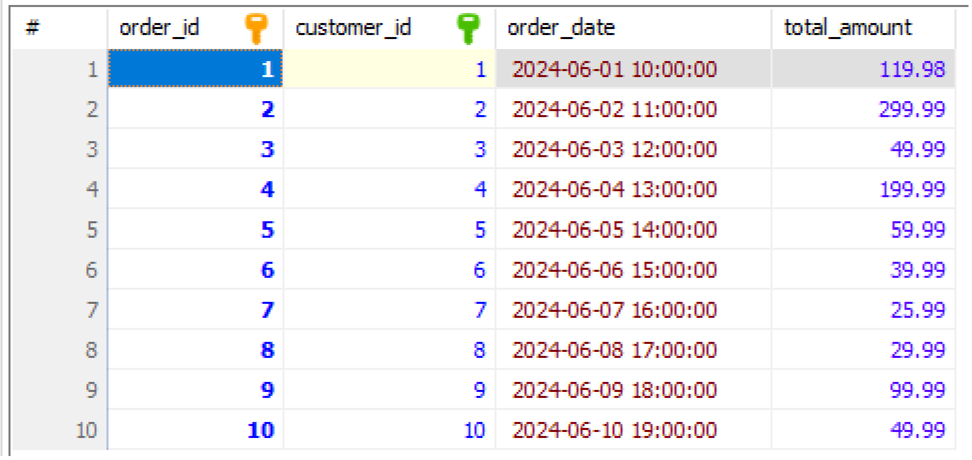


#	product_id	name	supplier_id	product_description	price	product_category	image_url	tags
1	1	T-shirt	1	A comfortable cotton t-shirt	19.0	Clothing	image1.jpg	clothing
2	2	Jeans	2	Stylish blue jeans	49.99	Clothing	image2.jpg	clothing
3	3	Jacket	3	Warm winter jacket	99.99	Clothing	image3.jpg	clothing
4	4	Sneakers	4	Sporty sneakers	59.99	Clothing	image4.jpg	clothing
5	5	Hat	5	Cool summer hat	15.99	Clothing	image5.jpg	clothing
6	6	Scarf	6	Woolen scarf	25.99	Clothing	image6.jpg	clothing
7	7	Gloves	7	Leather gloves	35.99	Clothing	image7.jpg	clothing
8	8	Socks	8	Comfortable socks	5.99	Clothing	image8.jpg	clothing
9	9	Belt	9	Stylish leather belt	29.99	Clothing	image9.jpg	clothing
10	10	Sunglasses	10	Fashionable sunglasses	49.99	Clothing	image10.jpg	clothing
11	11	Smartphone	1	Latest model smartphone	699.99	Electronics	image11.jpg	electronics
12	12	Laptop	2	High performance laptop	999.99	Electronics	image12.jpg	electronics
13	13	Tablet	3	Portable tablet	399.99	Electronics	image13.jpg	electronics
14	14	Headphones	4	Noise-cancelling headphones	199.99	Electronics	image14.jpg	electronics
15	15	Smartwatch	5	Feature-rich smartwatch	299.99	Electronics	image15.jpg	electronics
16	16	Camera	6	High resolution camera	499.99	Electronics	image16.jpg	electronics
17	17	TV	7	Ultra HD television	799.99	Electronics	image17.jpg	electronics
18	18	Game Console	8	Next-gen game console	499.99	Electronics	image18.jpg	electronics
19	19	Bluetooth Speaker	9	Portable Bluetooth speaker	59.99	Electronics	image19.jpg	electronics
20	20	Drone	10	High-tech drone	699.99	Electronics	image20.jpg	electronics

- Orders table :Has order details

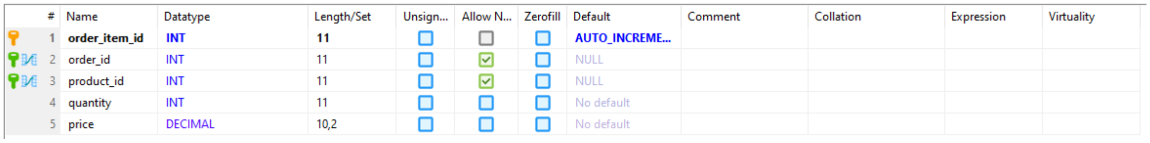


#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	order_id	INT	11	☐	☐	☐	AUTO_INCREME...				
2	customer_id	INT	11	☐	☑	☐	NULL				
3	order_date	TIMESTAMP		☐	☐	☐	current_timestam...				
4	total_amount	DECIMAL	10,2	☐	☐	☐	No default				

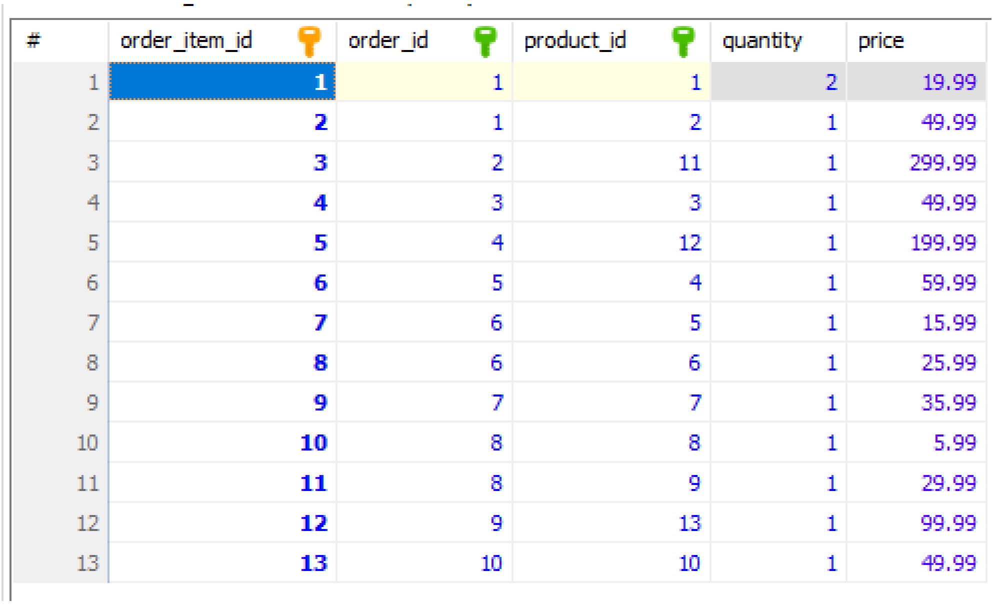


#	order_id	customer_id	order_date	total_amount
1	1	1	2024-06-01 10:00:00	119.98
2	2	2	2024-06-02 11:00:00	299.99
3	3	3	2024-06-03 12:00:00	49.99
4	4	4	2024-06-04 13:00:00	199.99
5	5	5	2024-06-05 14:00:00	59.99
6	6	6	2024-06-06 15:00:00	39.99
7	7	7	2024-06-07 16:00:00	25.99
8	8	8	2024-06-08 17:00:00	29.99
9	9	9	2024-06-09 18:00:00	99.99
10	10	10	2024-06-10 19:00:00	49.99

- Order items : contains items that are in each order :

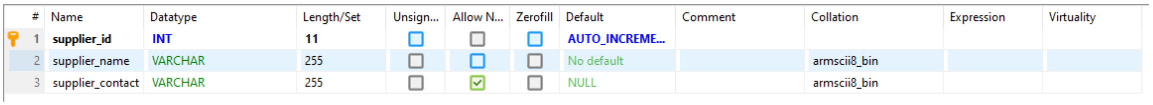


#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	order_item_id	INT	11	☐	☐	☐	AUTO_INCREME...				
2	order_id	INT	11	☐	☑	☐	NULL				
3	product_id	INT	11	☐	☑	☐	NULL				
4	quantity	INT	11	☐	☐	☐	No default				
5	price	DECIMAL	10,2	☐	☐	☐	No default				

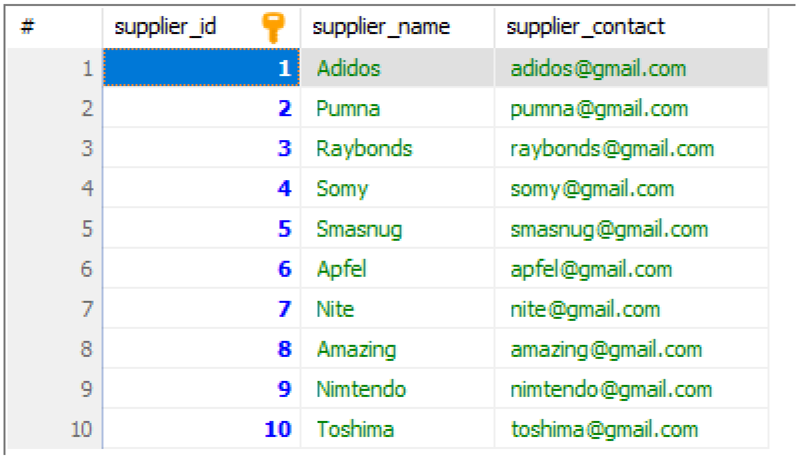


#	order_item_id	order_id	product_id	quantity	price
1	1	1	1	2	19.99
2	2	1	2	1	49.99
3	3	2	11	1	299.99
4	4	3	3	1	49.99
5	5	4	12	1	199.99
6	6	5	4	1	59.99
7	7	6	5	1	15.99
8	8	6	6	1	25.99
9	9	7	7	1	35.99
10	10	8	8	1	5.99
11	11	8	9	1	29.99
12	12	9	13	1	99.99
13	13	10	10	1	49.99

- Suppliers : contains supplier details:

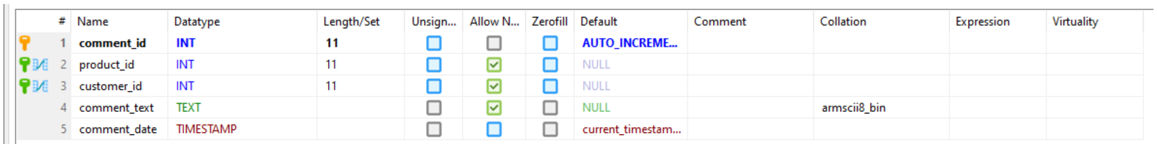


#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	supplier_id	INT	11	□	□	□	AUTO_INCREME...				
2	supplier_name	VARCHAR	255	□	□	□	No default		armscii8_bin		
3	supplier_contact	VARCHAR	255	□	☑	□	NULL		armscii8_bin		

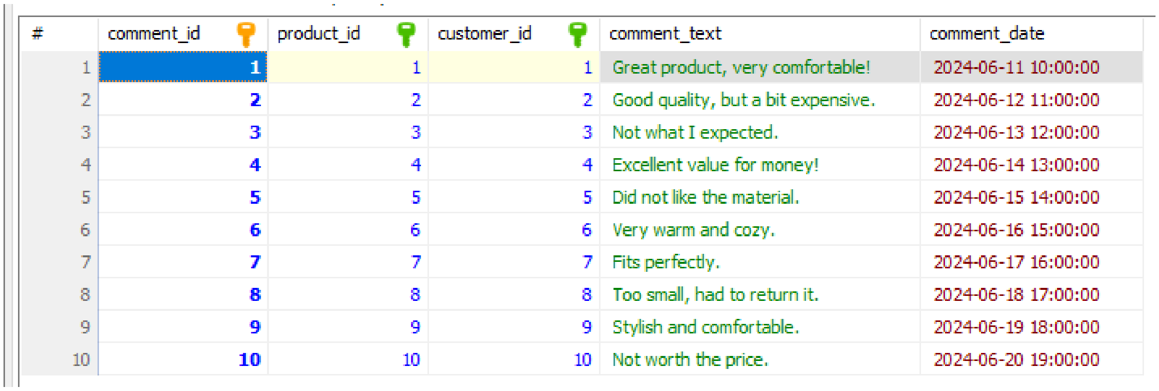


#	supplier_id	supplier_name	supplier_contact
1	1	Adidos	adidos@gmail.com
2	2	Pumna	pumna@gmail.com
3	3	Raybonds	raybonds@gmail.com
4	4	Somy	somy@gmail.com
5	5	Smasnug	smasnug@gmail.com
6	6	Apfel	apfel@gmail.com
7	7	Nite	nite@gmail.com
8	8	Amazing	amazing@gmail.com
9	9	Nimtendo	nimtendo@gmail.com
10	10	Toshima	toshima@gmail.com

- Comments: stores comment data

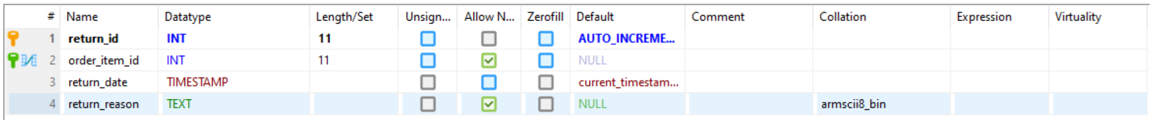


#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	comment_id	INT	11	□	□	□	AUTO_INCREME...				
2	product_id	INT	11	□	☑	□	NULL				
3	customer_id	INT	11	□	☑	□	NULL				
4	comment_text	TEXT		□	☑	□	NULL		armscii8_bin		
5	comment_date	TIMESTAMP		□	□	□	current_timestam...				

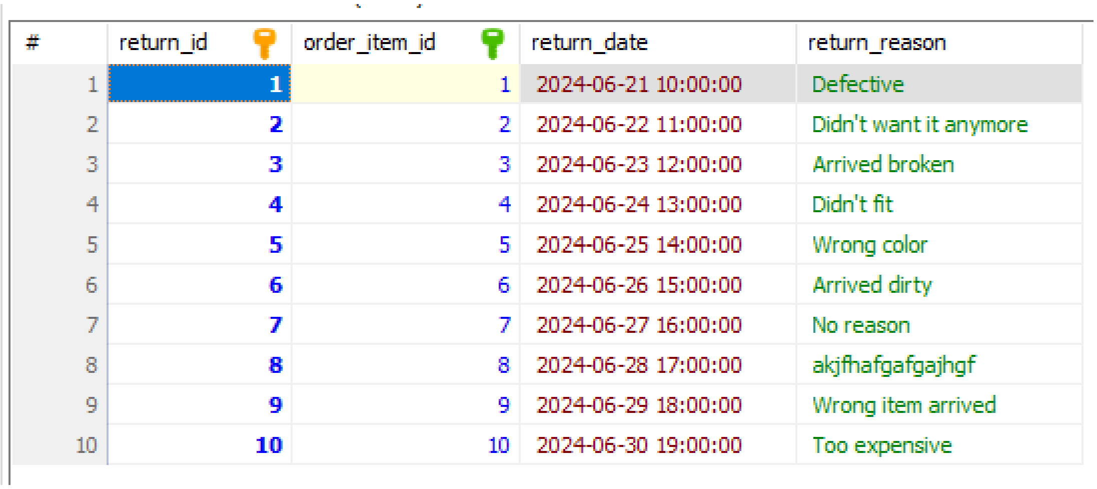


#	comment_id	product_id	customer_id	comment_text	comment_date
1	1	1	1	Great product, very comfortable!	2024-06-11 10:00:00
2	2	2	2	Good quality, but a bit expensive.	2024-06-12 11:00:00
3	3	3	3	Not what I expected.	2024-06-13 12:00:00
4	4	4	4	Excellent value for money!	2024-06-14 13:00:00
5	5	5	5	Did not like the material.	2024-06-15 14:00:00
6	6	6	6	Very warm and cozy.	2024-06-16 15:00:00
7	7	7	7	Fits perfectly.	2024-06-17 16:00:00
8	8	8	8	Too small, had to return it.	2024-06-18 17:00:00
9	9	9	9	Stylish and comfortable.	2024-06-19 18:00:00
10	10	10	10	Not worth the price.	2024-06-20 19:00:00

- Returns table stores data about the returned items :



#	Name	Datatype	Length/Set	Unsign...	Allow N...	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	return_id	INT	11	□	□	□	AUTO_INCREME...				
2	order_item_id	INT	11	□	☑	□	NULL				
3	return_date	TIMESTAMP		□	□	□	current_timestam...				
4	return_reason	TEXT		□	☑	□	NULL		armscii8_bin		

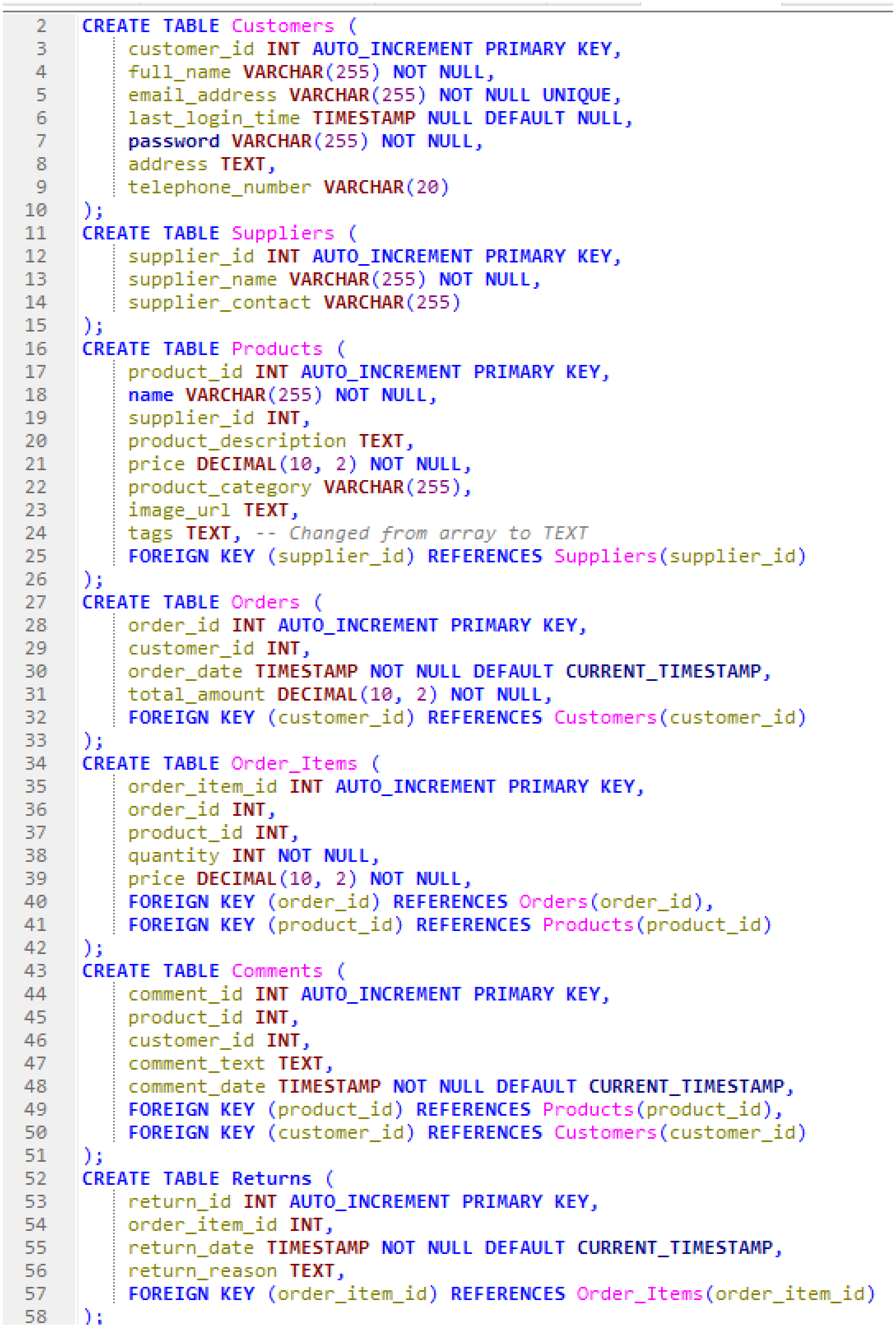


#	return_id	order_item_id	return_date	return_reason
1	1	1	2024-06-21 10:00:00	Defective
2	2	2	2024-06-22 11:00:00	Didn't want it anymore
3	3	3	2024-06-23 12:00:00	Arrived broken
4	4	4	2024-06-24 13:00:00	Didn't fit
5	5	5	2024-06-25 14:00:00	Wrong color
6	6	6	2024-06-26 15:00:00	Arrived dirty
7	7	7	2024-06-27 16:00:00	No reason
8	8	8	2024-06-28 17:00:00	akjfhafgafgajhgf
9	9	9	2024-06-29 18:00:00	Wrong item arrived
10	10	10	2024-06-30 19:00:00	Too expensive

The relationships in this data base are:
Customer – (1) one to many(N)-> Order
Order – (1) one to many(N)-> Order Item
Product – (1) one to many(N)-> Order Item
Supplier – (1) one to many(N)-> Product
Product – (1) one to many(N)-> Comment
Customer – (1) one to many(N)-> Comment
Order Item – (1) one to one (1)-> Return
The database has primary keys to make each table unique , and foreign keys to connect tables together.
It also has constraints to ensure data integrity. (e.g., NOT NULL, UNIQUE).

Database implementation:

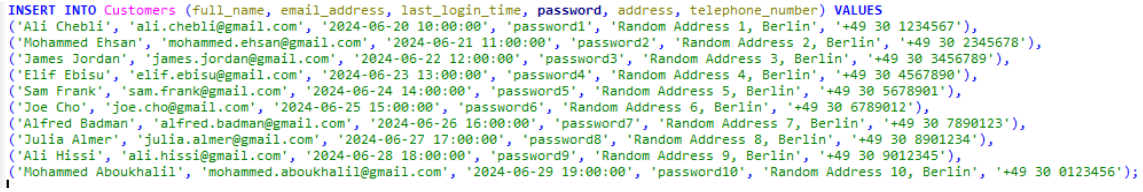
The data base was implemented using Maria db.



```
CREATE TABLE Customers (
    customer_id INT AUTO_INCREMENT PRIMARY KEY,
    full_name VARCHAR(255) NOT NULL,
    email_address VARCHAR(255) NOT NULL UNIQUE,
    last_login_time TIMESTAMP NULL DEFAULT NULL,
    password VARCHAR(255) NOT NULL,
    address TEXT,
    telephone_number VARCHAR(20)
);
CREATE TABLE Suppliers (
    supplier_id INT AUTO_INCREMENT PRIMARY KEY,
    supplier_name VARCHAR(255) NOT NULL,
    supplier_contact VARCHAR(255)
);
CREATE TABLE Products (
    product_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    supplier_id INT,
    product_description TEXT,
    price DECIMAL(10, 2) NOT NULL,
    product_category VARCHAR(255),
    image_url TEXT,
    tags TEXT, -- Changed from array to TEXT
    FOREIGN KEY (supplier_id) REFERENCES Suppliers(supplier_id)
);
CREATE TABLE Orders (
    order_id INT AUTO_INCREMENT PRIMARY KEY,
    customer_id INT,
    order_date TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    total_amount DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
);
CREATE TABLE Order_Items (
    order_item_id INT AUTO_INCREMENT PRIMARY KEY,
    order_id INT,
    product_id INT,
    quantity INT NOT NULL,
    price DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (order_id) REFERENCES Orders(order_id),
    FOREIGN KEY (product_id) REFERENCES Products(product_id)
);
CREATE TABLE Comments (
    comment_id INT AUTO_INCREMENT PRIMARY KEY,
    product_id INT,
    customer_id INT,
    comment_text TEXT,
    comment_date TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (product_id) REFERENCES Products(product_id),
    FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
);
CREATE TABLE Returns (
    return_id INT AUTO_INCREMENT PRIMARY KEY,
    order_item_id INT,
    return_date TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    return_reason TEXT,
    FOREIGN KEY (order_item_id) REFERENCES Order_Items(order_item_id)
);
```

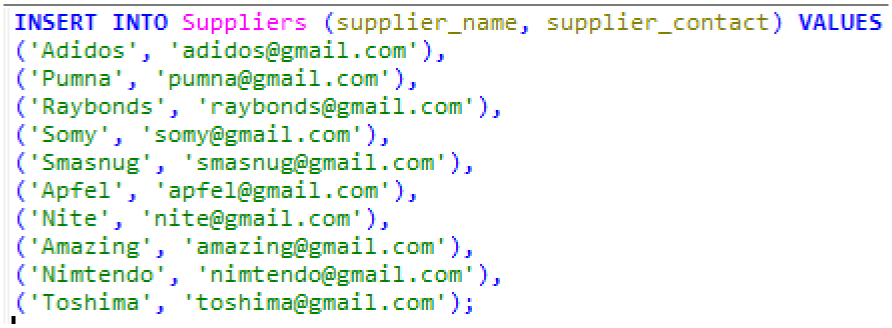
Database population:

Populating Customers table:



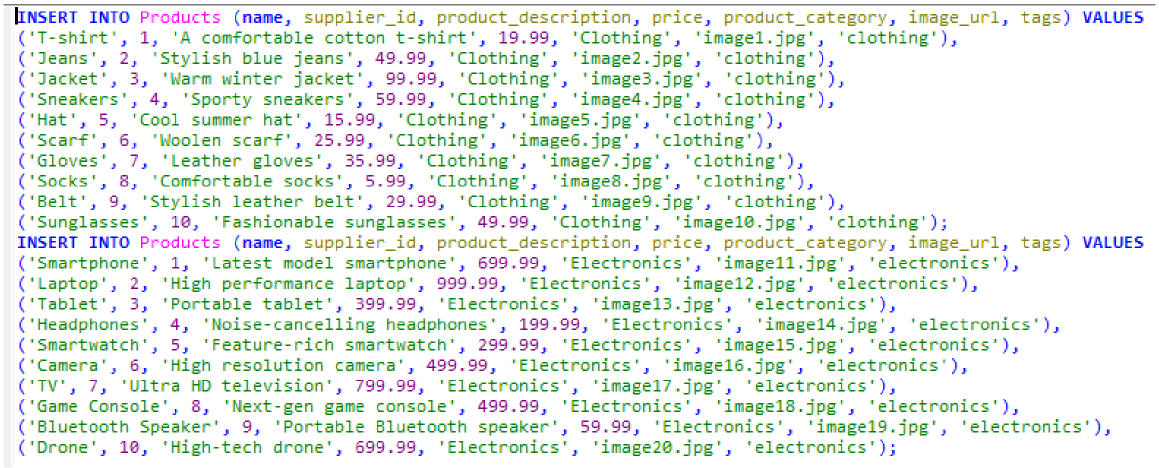
```
INSERT INTO Customers (full_name, email_address, last_login_time, password, address, telephone_number) VALUES
('Ali Chebli', 'ali.chebli@gmail.com', '2024-06-20 10:00:00', 'password1', 'Random Address 1, Berlin', '+49 30 1234567'),
('Mohammed Ehsan', 'mohammed.ehsan@gmail.com', '2024-06-21 11:00:00', 'password2', 'Random Address 2, Berlin', '+49 30 2345678'),
('James Jordan', 'james.jordan@gmail.com', '2024-06-22 12:00:00', 'password3', 'Random Address 3, Berlin', '+49 30 3456789'),
('Elif Ebisu', 'elif.ebisu@gmail.com', '2024-06-23 13:00:00', 'password4', 'Random Address 4, Berlin', '+49 30 4567890'),
('Sam Frank', 'sam.frank@gmail.com', '2024-06-24 14:00:00', 'password5', 'Random Address 5, Berlin', '+49 30 5678901'),
('Joe Cho', 'joe.cho@gmail.com', '2024-06-25 15:00:00', 'password6', 'Random Address 6, Berlin', '+49 30 6789012'),
('Alfred Badman', 'alfred.badman@gmail.com', '2024-06-26 16:00:00', 'password7', 'Random Address 7, Berlin', '+49 30 7890123'),
('Julia Almer', 'julia.almer@gmail.com', '2024-06-27 17:00:00', 'password8', 'Random Address 8, Berlin', '+49 30 8901234'),
('Ali Hissi', 'ali.hissi@gmail.com', '2024-06-28 18:00:00', 'password9', 'Random Address 9, Berlin', '+49 30 9012345'),
('Mohammed Aboukhalil', 'mohammed.aboukhalil@gmail.com', '2024-06-29 19:00:00', 'password10', 'Random Address 10, Berlin', '+49 30 0123456');
```

Populating suppliers table:



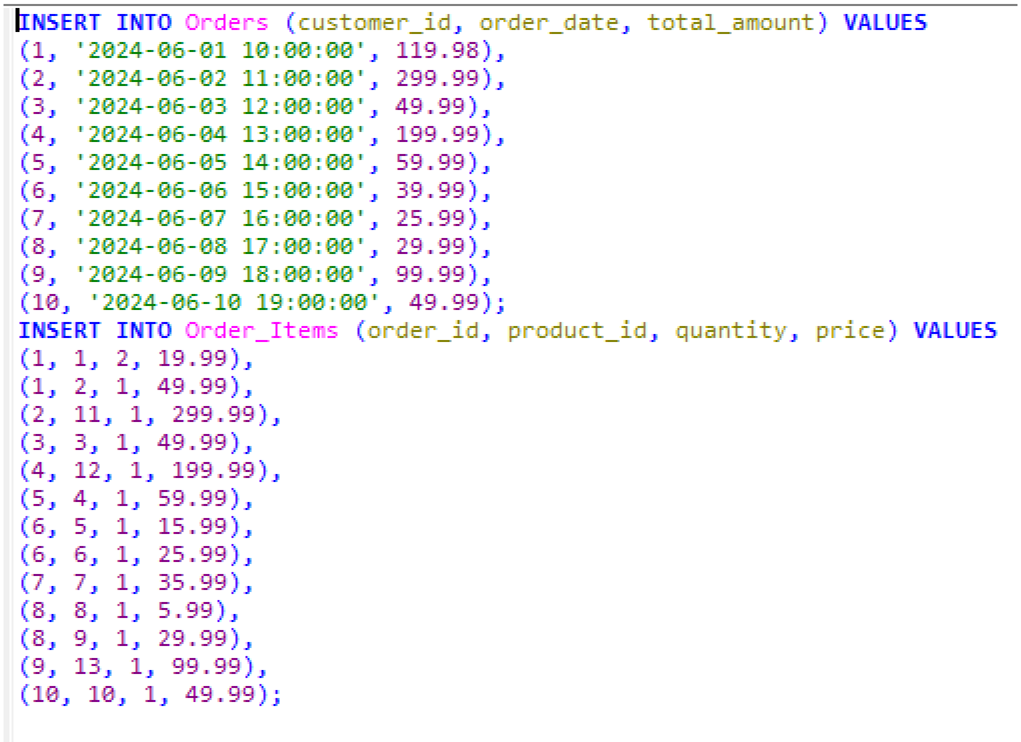
```
INSERT INTO Suppliers (supplier_name, supplier_contact) VALUES
('Adidos', 'adidos@gmail.com'),
('Pumna', 'pumna@gmail.com'),
('Raybonds', 'raybonds@gmail.com'),
('Somy', 'somy@gmail.com'),
('Smasnug', 'smasnug@gmail.com'),
('Apfel', 'apfel@gmail.com'),
('Nite', 'nite@gmail.com'),
('Amazing', 'amazing@gmail.com'),
('Nimtendo', 'nimtendo@gmail.com'),
('Toshima', 'toshima@gmail.com');
```

Populating Products table with 2 types of products (electronics and clothing’s):



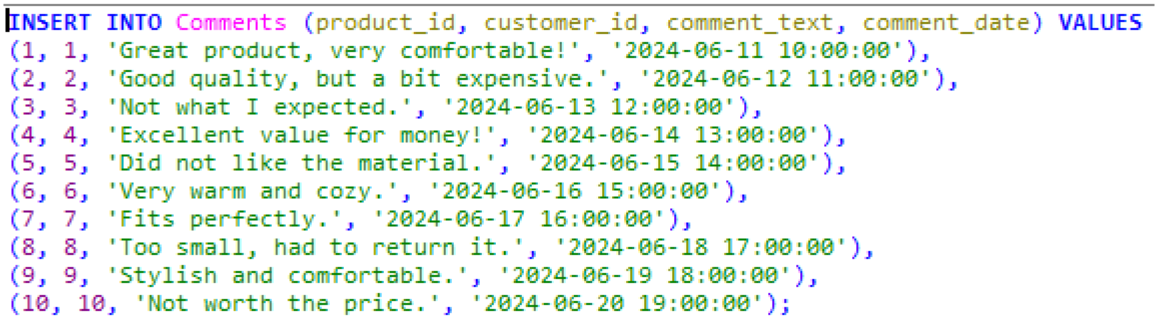
```
INSERT INTO Products (name, supplier_id, product_description, price, product_category, image_url, tags) VALUES
('T-shirt', 1, 'A comfortable cotton t-shirt', 19.99, 'Clothing', 'image1.jpg', 'clothing'),
('Jeans', 2, 'Stylish blue jeans', 49.99, 'Clothing', 'image2.jpg', 'clothing'),
('Jacket', 3, 'Warm winter jacket', 99.99, 'Clothing', 'image3.jpg', 'clothing'),
('Sneakers', 4, 'Sporty sneakers', 59.99, 'Clothing', 'image4.jpg', 'clothing'),
('Hat', 5, 'Cool summer hat', 15.99, 'Clothing', 'image5.jpg', 'clothing'),
('Scarf', 6, 'Woolen scarf', 25.99, 'Clothing', 'image6.jpg', 'clothing'),
('Gloves', 7, 'Leather gloves', 35.99, 'Clothing', 'image7.jpg', 'clothing'),
('Socks', 8, 'Comfortable socks', 5.99, 'Clothing', 'image8.jpg', 'clothing'),
('Belt', 9, 'Stylish leather belt', 29.99, 'Clothing', 'image9.jpg', 'clothing'),
('Sunglasses', 10, 'Fashionable sunglasses', 49.99, 'Clothing', 'image10.jpg', 'clothing');
INSERT INTO Products (name, supplier_id, product_description, price, product_category, image_url, tags) VALUES
('Smartphone', 1, 'Latest model smartphone', 699.99, 'Electronics', 'image11.jpg', 'electronics'),
('Laptop', 2, 'High performance laptop', 999.99, 'Electronics', 'image12.jpg', 'electronics'),
('Tablet', 3, 'Portable tablet', 399.99, 'Electronics', 'image13.jpg', 'electronics'),
('Headphones', 4, 'Noise-cancelling headphones', 199.99, 'Electronics', 'image14.jpg', 'electronics'),
('Smartwatch', 5, 'Feature-rich smartwatch', 299.99, 'Electronics', 'image15.jpg', 'electronics'),
('Camera', 6, 'High resolution camera', 499.99, 'Electronics', 'image16.jpg', 'electronics'),
('TV', 7, 'Ultra HD television', 799.99, 'Electronics', 'image17.jpg', 'electronics'),
('Game Console', 8, 'Next-gen game console', 499.99, 'Electronics', 'image18.jpg', 'electronics'),
('Bluetooth Speaker', 9, 'Portable Bluetooth speaker', 59.99, 'Electronics', 'image19.jpg', 'electronics'),
('Drone', 10, 'High-tech drone', 699.99, 'Electronics', 'image20.jpg', 'electronics');
```

Populating orders and order items tables:



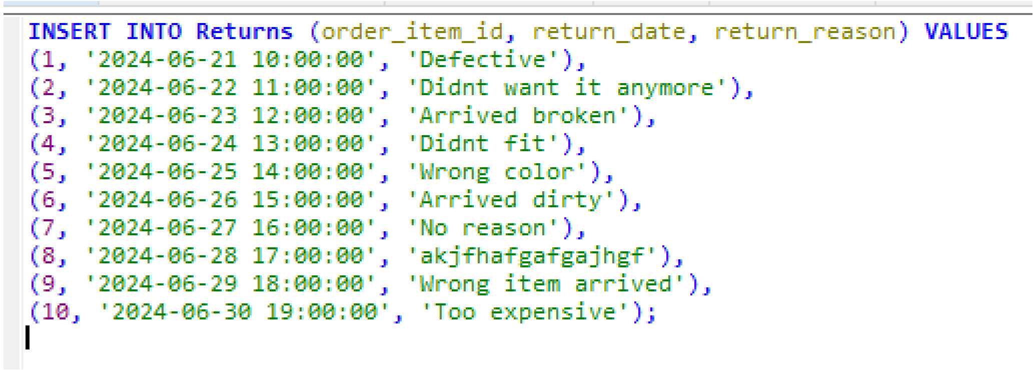
```sql
INSERT INTO Orders (customer_id, order_date, total_amount) VALUES
(1, '2024-06-01 10:00:00', 119.98),
(2, '2024-06-02 11:00:00', 299.99),
(3, '2024-06-03 12:00:00', 49.99),
(4, '2024-06-04 13:00:00', 199.99),
(5, '2024-06-05 14:00:00', 59.99),
(6, '2024-06-06 15:00:00', 39.99),
(7, '2024-06-07 16:00:00', 25.99),
(8, '2024-06-08 17:00:00', 29.99),
(9, '2024-06-09 18:00:00', 99.99),
(10, '2024-06-10 19:00:00', 49.99);
INSERT INTO Order_Items (order_id, product_id, quantity, price) VALUES
(1, 1, 2, 19.99),
(1, 2, 1, 49.99),
(2, 11, 1, 299.99),
(3, 3, 1, 49.99),
(4, 12, 1, 199.99),
(5, 4, 1, 59.99),
(6, 5, 1, 15.99),
(6, 6, 1, 25.99),
(7, 7, 1, 35.99),
(8, 8, 1, 5.99),
(8, 9, 1, 29.99),
(9, 13, 1, 99.99),
(10, 10, 1, 49.99);
```

Populating comments table:



```sql
INSERT INTO Comments (product_id, customer_id, comment_text, comment_date) VALUES
(1, 1, 'Great product, very comfortable!', '2024-06-11 10:00:00'),
(2, 2, 'Good quality, but a bit expensive.', '2024-06-12 11:00:00'),
(3, 3, 'Not what I expected.', '2024-06-13 12:00:00'),
(4, 4, 'Excellent value for money!', '2024-06-14 13:00:00'),
(5, 5, 'Did not like the material.', '2024-06-15 14:00:00'),
(6, 6, 'Very warm and cozy.', '2024-06-16 15:00:00'),
(7, 7, 'Fits perfectly.', '2024-06-17 16:00:00'),
(8, 8, 'Too small, had to return it.', '2024-06-18 17:00:00'),
(9, 9, 'Stylish and comfortable.', '2024-06-19 18:00:00'),
(10, 10, 'Not worth the price.', '2024-06-20 19:00:00');
```

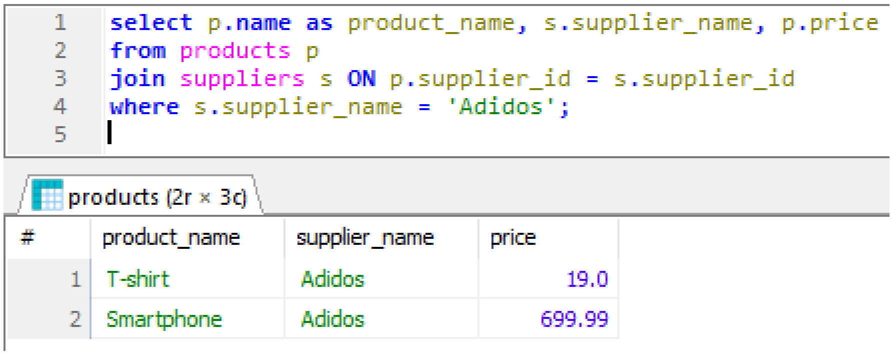
Populating returns table:



```sql
INSERT INTO Returns (order_item_id, return_date, return_reason) VALUES
(1, '2024-06-21 10:00:00', 'Defective'),
(2, '2024-06-22 11:00:00', 'Didnt want it anymore'),
(3, '2024-06-23 12:00:00', 'Arrived broken'),
(4, '2024-06-24 13:00:00', 'Didnt fit'),
(5, '2024-06-25 14:00:00', 'Wrong color'),
(6, '2024-06-26 15:00:00', 'Arrived dirty'),
(7, '2024-06-27 16:00:00', 'No reason'),
(8, '2024-06-28 17:00:00', 'akjfhafgafgajhgf'),
(9, '2024-06-29 18:00:00', 'Wrong item arrived'),
(10, '2024-06-30 19:00:00', 'Too expensive');
```

Queries:

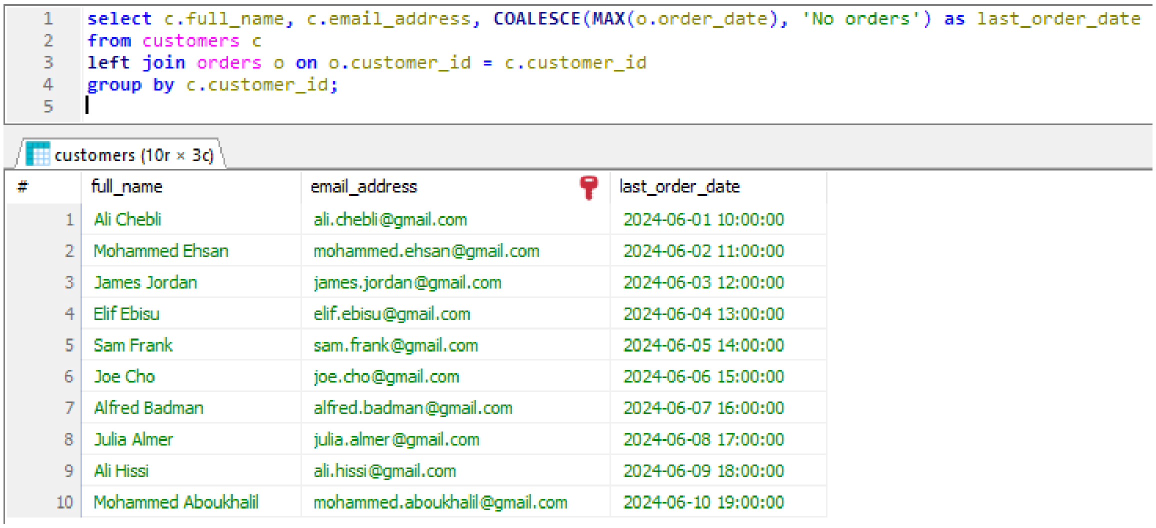
- Query to fetch product details supplied by a specific supplier:



```sql
select p.name as product_name, s.supplier_name, p.price
from products p
join suppliers s ON p.supplier_id = s.supplier_id
where s.supplier_name = 'Adidos';
```

#	product_name	supplier_name	price
1	T-shirt	Adidos	19.0
2	Smartphone	Adidos	699.99

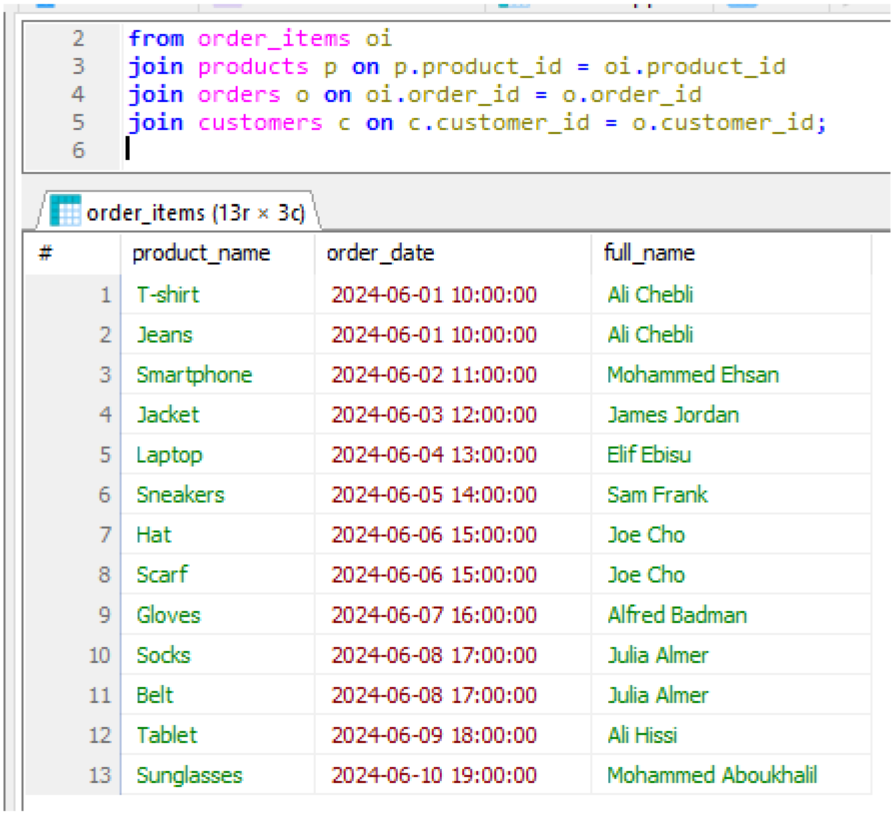
- Query to fetch customer details along with their last order date:



```sql
select c.full_name, c.email_address, COALESCE(MAX(o.order_date), 'No orders') as last_order_date
from customers c
left join orders o on o.customer_id = c.customer_id
group by c.customer_id;
```

#	full_name	email_address	last_order_date
1	Ali Chebli	ali.chebli@gmail.com	2024-06-01 10:00:00
2	Mohammed Ehsan	mohammed.ehsan@gmail.com	2024-06-02 11:00:00
3	James Jordan	james.jordan@gmail.com	2024-06-03 12:00:00
4	Elif Ebisu	elif.ebisu@gmail.com	2024-06-04 13:00:00
5	Sam Frank	sam.frank@gmail.com	2024-06-05 14:00:00
6	Joe Cho	joe.cho@gmail.com	2024-06-06 15:00:00
7	Alfred Badman	alfred.badman@gmail.com	2024-06-07 16:00:00
8	Julia Almer	julia.almer@gmail.com	2024-06-08 17:00:00
9	Ali Hissi	ali.hissi@gmail.com	2024-06-09 18:00:00
10	Mohammed Aboukhalil	mohammed.aboukhalil@gmail.com	2024-06-10 19:00:00

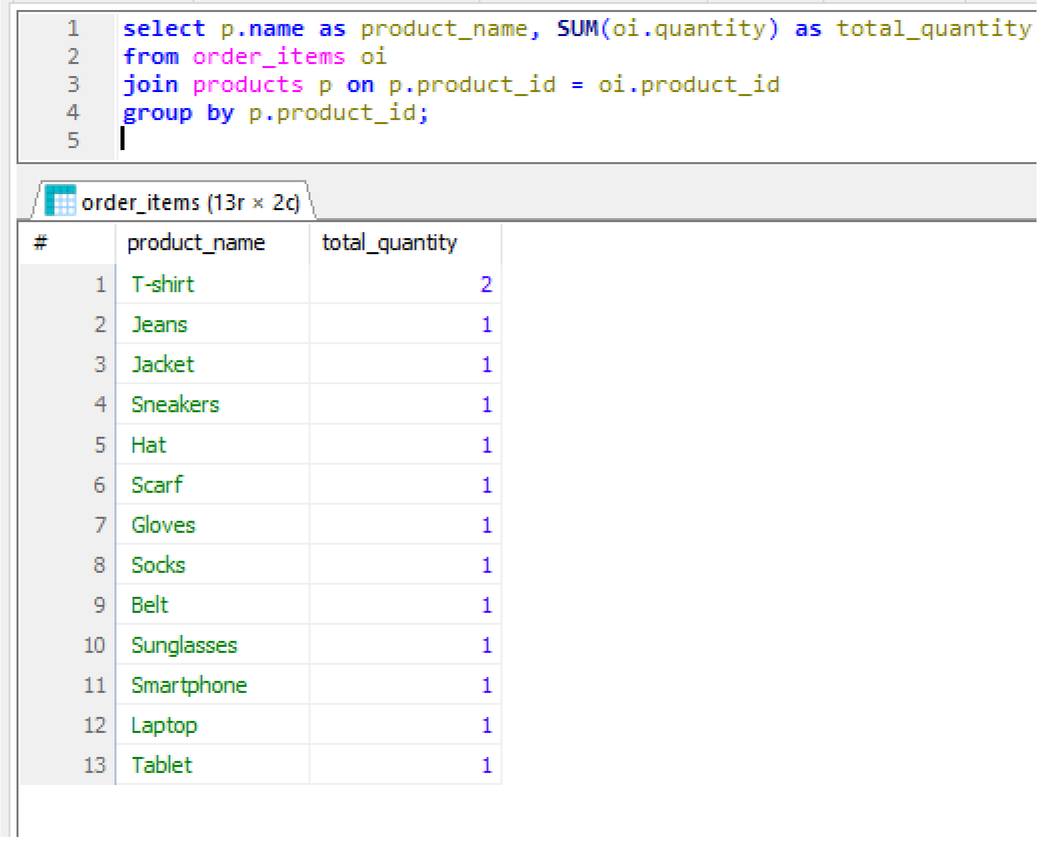
- Query to fetch product name, order date, and customer details for each order item:



```sql
from order_items oi
join products p on p.product_id = oi.product_id
join orders o on oi.order_id = o.order_id
join customers c on c.customer_id = o.customer_id;
```

#	product_name	order_date	full_name
1	T-shirt	2024-06-01 10:00:00	Ali Chebli
2	Jeans	2024-06-01 10:00:00	Ali Chebli
3	Smartphone	2024-06-02 11:00:00	Mohammed Ehsan
4	Jacket	2024-06-03 12:00:00	James Jordan
5	Laptop	2024-06-04 13:00:00	Elif Ebisu
6	Sneakers	2024-06-05 14:00:00	Sam Frank
7	Hat	2024-06-06 15:00:00	Joe Cho
8	Scarf	2024-06-06 15:00:00	Joe Cho
9	Gloves	2024-06-07 16:00:00	Alfred Badman
10	Socks	2024-06-08 17:00:00	Julia Almer
11	Belt	2024-06-08 17:00:00	Julia Almer
12	Tablet	2024-06-09 18:00:00	Ali Hissi
13	Sunglasses	2024-06-10 19:00:00	Mohammed Aboukhalil

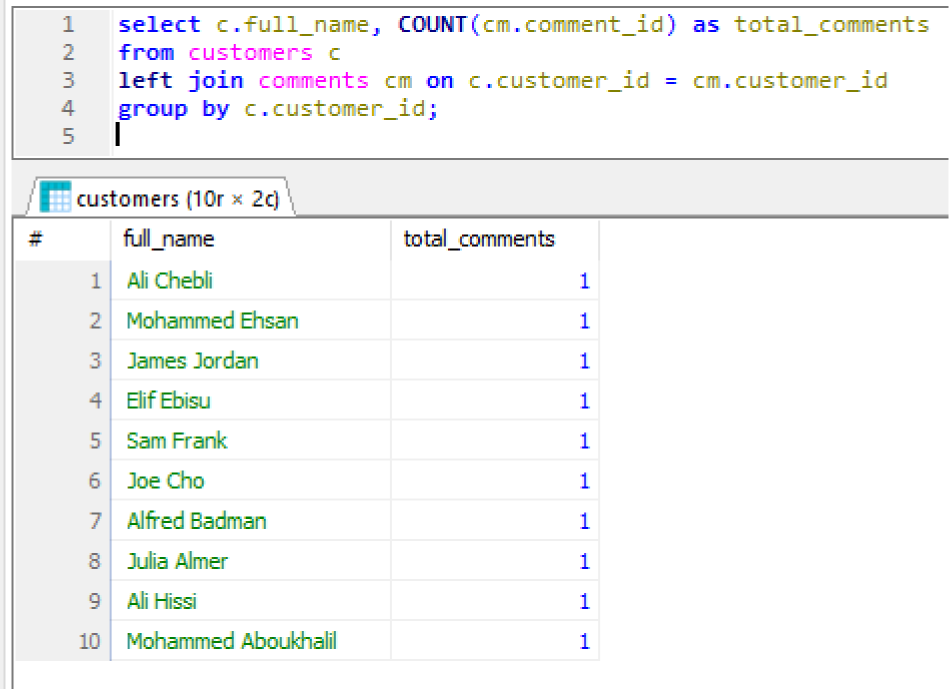
- Query to fetch product name and total quantity ordered:



```sql
select p.name as product_name, SUM(oi.quantity) as total_quantity
from order_items oi
join products p on p.product_id = oi.product_id
group by p.product_id;
```

#	product_name	total_quantity
1	T-shirt	2
2	Jeans	1
3	Jacket	1
4	Sneakers	1
5	Hat	1
6	Scarf	1
7	Gloves	1
8	Socks	1
9	Belt	1
10	Sunglasses	1
11	Smartphone	1
12	Laptop	1
13	Tablet	1

- Query to fetch customer names and the number of comments they have made:

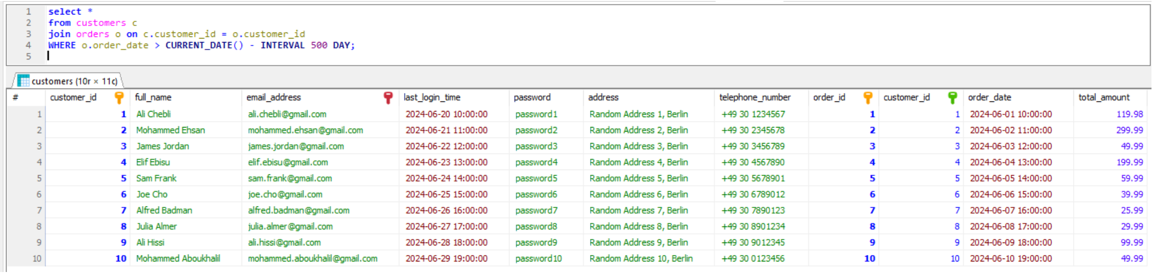


```sql
select c.full_name, COUNT(cm.comment_id) as total_comments
from customers c
left join comments cm on c.customer_id = cm.customer_id
group by c.customer_id;
```

customers (10r × 2c)

#	full_name	total_comments
1	Ali Chebli	1
2	Mohammed Ehsan	1
3	James Jordan	1
4	Elif Ebisu	1
5	Sam Frank	1
6	Joe Cho	1
7	Alfred Badman	1
8	Julia Almer	1
9	Ali Hissi	1
10	Mohammed Aboukhalil	1

- Query to fetch all orders placed in the last 500 days:

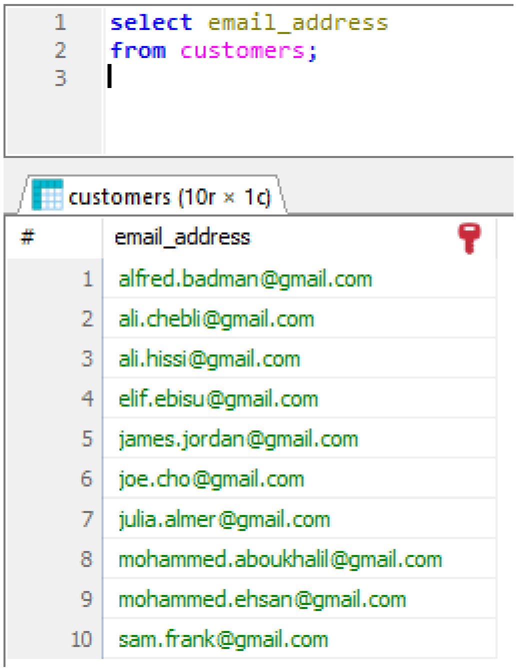


```sql
select *
from customers c
join orders o on c.customer_id = o.customer_id
WHERE o.order_date > CURRENT_DATE() - INTERVAL 500 DAY;
```

customers (10r × 11c)

#	customer_id	full_name	email_address	last_login_time	password	address	telephone_number	order_id	customer_id	order_date	total_amount
1	1	Ali Chebli	ali.chebli@gmail.com	2024-06-20 10:00:00	password1	Random Address 1, Berlin	+49 30 1234567	1	1	2024-06-01 10:00:00	119.98
2	2	Mohammed Ehsan	mohammed.ehsan@gmail.com	2024-06-21 11:00:00	password2	Random Address 2, Berlin	+49 30 2345678	2	2	2024-06-02 11:00:00	299.99
3	3	James Jordan	james.jordan@gmail.com	2024-06-22 12:00:00	password3	Random Address 3, Berlin	+49 30 3456789	3	3	2024-06-03 12:00:00	49.99
4	4	Elif Ebisu	elif.ebisu@gmail.com	2024-06-23 13:00:00	password4	Random Address 4, Berlin	+49 30 4567890	4	4	2024-06-04 13:00:00	199.99
5	5	Sam Frank	sam.frank@gmail.com	2024-06-24 14:00:00	password5	Random Address 5, Berlin	+49 30 5678901	5	5	2024-06-05 14:00:00	59.99
6	6	Joe Cho	joe.cho@gmail.com	2024-06-25 15:00:00	password6	Random Address 6, Berlin	+49 30 6789012	6	6	2024-06-06 15:00:00	39.99
7	7	Alfred Badman	alfred.badman@gmail.com	2024-06-26 16:00:00	password7	Random Address 7, Berlin	+49 30 7890123	7	7	2024-06-07 16:00:00	25.99
8	8	Julia Almer	julia.almer@gmail.com	2024-06-27 17:00:00	password8	Random Address 8, Berlin	+49 30 8901234	8	8	2024-06-08 17:00:00	29.99
9	9	Ali Hissi	ali.hissi@gmail.com	2024-06-28 18:00:00	password9	Random Address 9, Berlin	+49 30 9012345	9	9	2024-06-09 18:00:00	99.99
10	10	Mohammed Aboukhalil	mohammed.aboukhalil@gmail.com	2024-06-29 19:00:00	password10	Random Address 10, Berlin	+49 30 0123456	10	10	2024-06-10 19:00:00	49.99

- Query to fetch all customer email addresses:

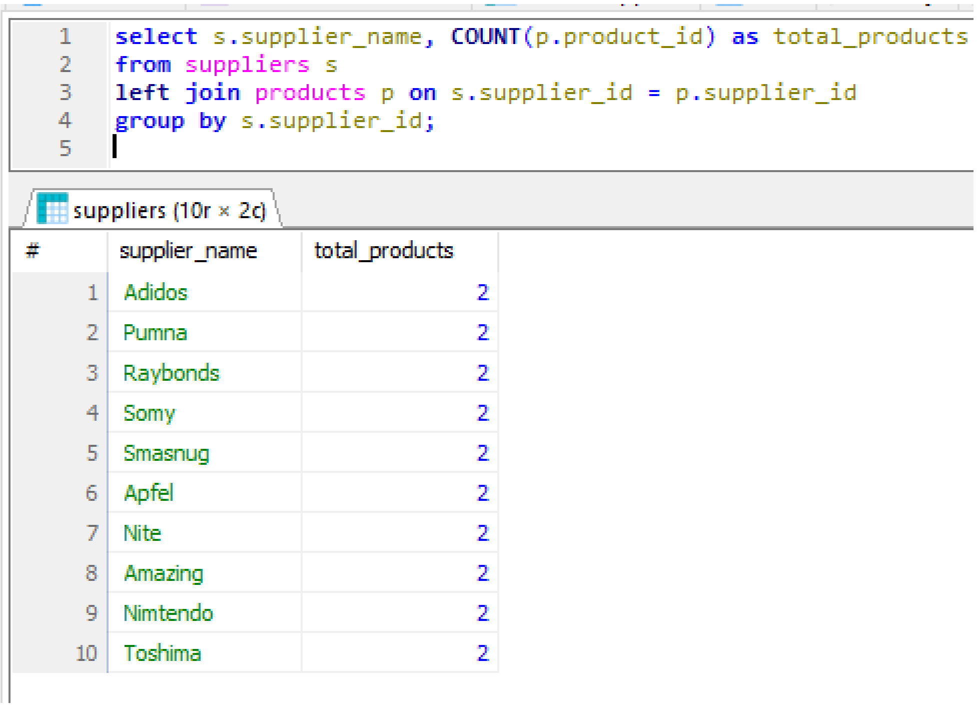


```sql
select email_address
from customers;
```

customers (10r × 1c)

#	email_address
1	alfred.badman@gmail.com
2	ali.chebli@gmail.com
3	ali.hissi@gmail.com
4	elif.ebisu@gmail.com
5	james.jordan@gmail.com
6	joe.cho@gmail.com
7	julia.almer@gmail.com
8	mohammed.aboukhalil@gmail.com
9	mohammed.ehsan@gmail.com
10	sam.frank@gmail.com

- Query to fetch supplier names and the number of products they supply:

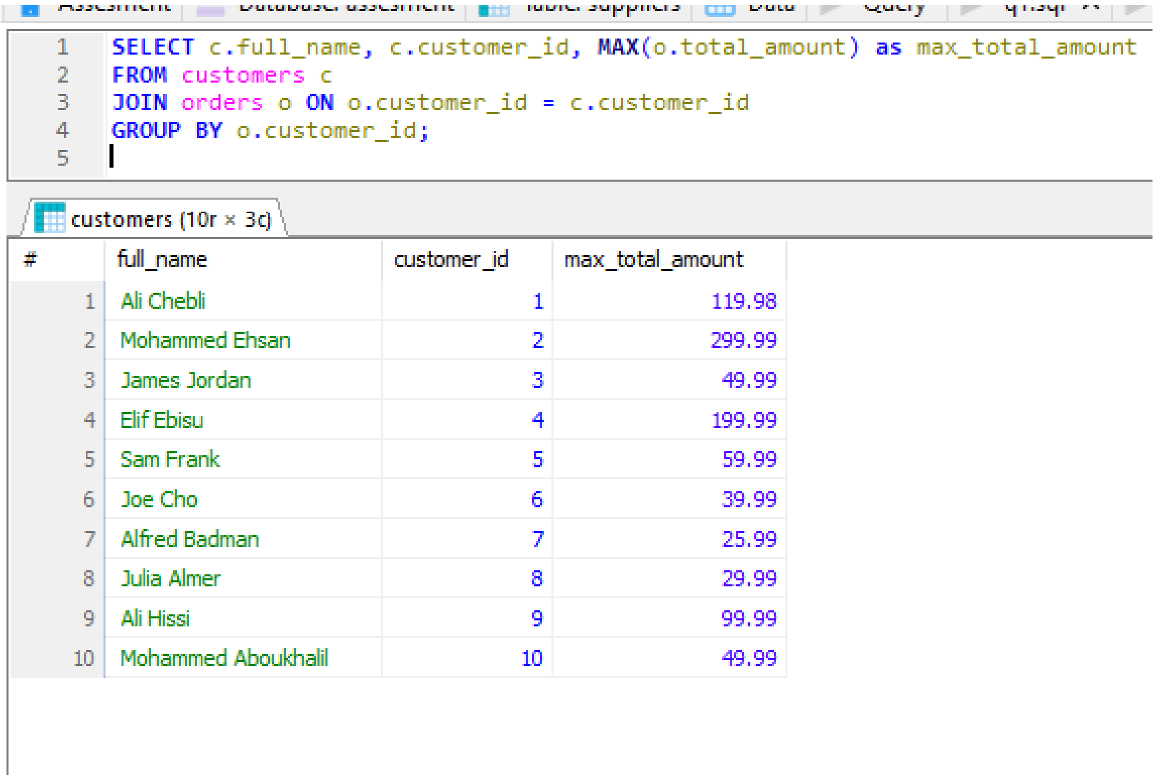


```
select s.supplier_name, COUNT(p.product_id) as total_products
from suppliers s
left join products p on s.supplier_id = p.supplier_id
group by s.supplier_id;
```

suppliers (10r × 2c)

#	supplier_name	total_products
1	Adidos	2
2	Pumna	2
3	Raybonds	2
4	Somy	2
5	Smasnug	2
6	Apfel	2
7	Nite	2
8	Amazing	2
9	Nimtendo	2
10	Toshima	2

- Query to fetch customers with their maximum order total amount:

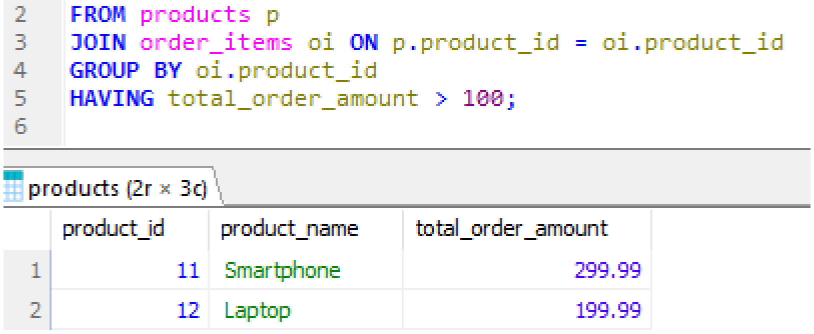


```
SELECT c.full_name, c.customer_id, MAX(o.total_amount) as max_total_amount
FROM customers c
JOIN orders o ON o.customer_id = c.customer_id
GROUP BY o.customer_id;
```

customers (10r × 3c)

#	full_name	customer_id	max_total_amount
1	Ali Chebli	1	119.98
2	Mohammed Ehsan	2	299.99
3	James Jordan	3	49.99
4	Elif Ebisu	4	199.99
5	Sam Frank	5	59.99
6	Joe Cho	6	39.99
7	Alfred Badman	7	25.99
8	Julia Almer	8	29.99
9	Ali Hissi	9	99.99
10	Mohammed Aboukhalil	10	49.99

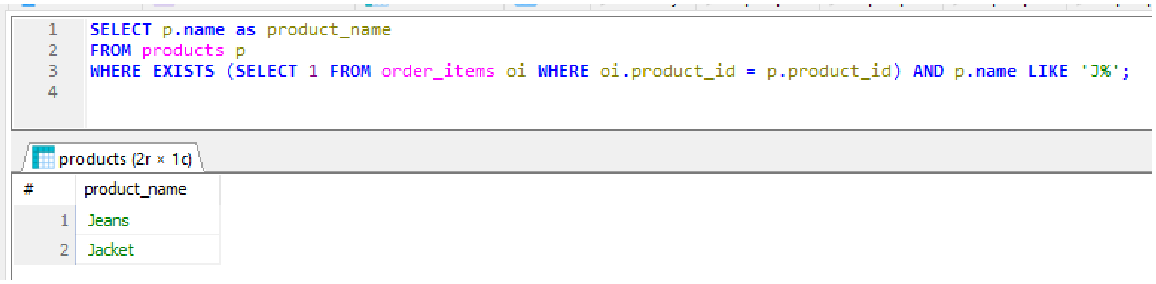
- Query to fetch products with total order amount greater than 100



```
FROM products p
JOIN order_items oi ON p.product_id = oi.product_id
GROUP BY oi.product_id
HAVING total_order_amount > 100;
```

	product_id	product_name	total_order_amount
1	11	Smartphone	299.99
2	12	Laptop	199.99

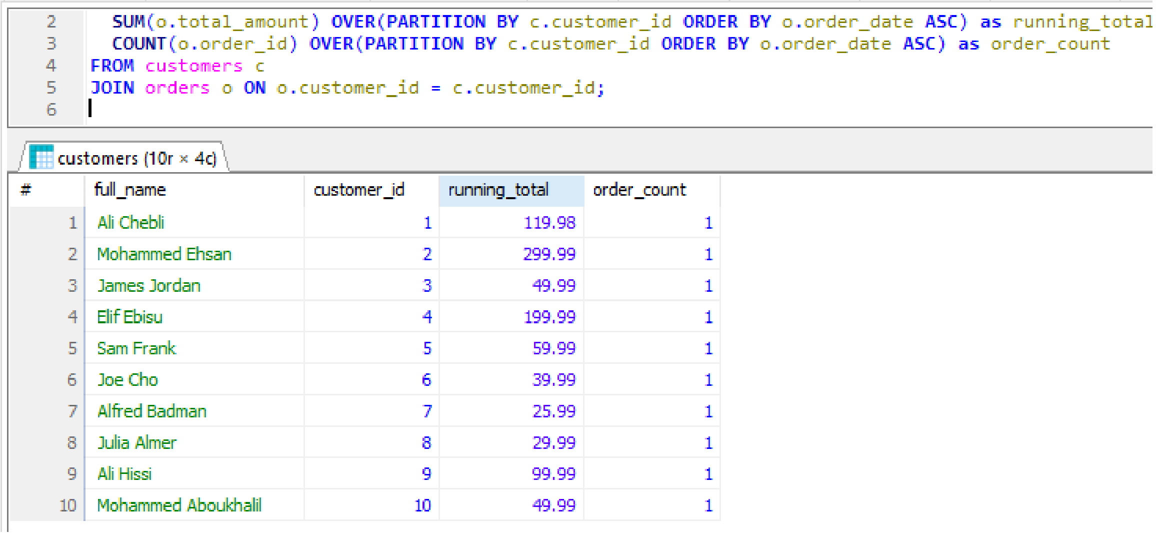
- Query to fetch products starting with 'J' that have been ordered:



```
SELECT p.name as product_name
FROM products p
WHERE EXISTS (SELECT 1 FROM order_items oi WHERE oi.product_id = p.product_id) AND p.name LIKE 'J%';
```

#	product_name
1	Jeans
2	Jacket

- Query to fetch customers with their order count and total amount, partitioned by order date:



```
   SUM(o.total_amount) OVER(PARTITION BY c.customer_id ORDER BY o.order_date ASC) as running_total
   COUNT(o.order_id) OVER(PARTITION BY c.customer_id ORDER BY o.order_date ASC) as order_count
FROM customers c
JOIN orders o ON o.customer_id = c.customer_id;
```

#	full_name	customer_id	running_total	order_count
1	Ali Chebli	1	119.98	1
2	Mohammed Ehsan	2	299.99	1
3	James Jordan	3	49.99	1
4	Elif Ebisu	4	199.99	1
5	Sam Frank	5	59.99	1
6	Joe Cho	6	39.99	1
7	Alfred Badman	7	25.99	1
8	Julia Almer	8	29.99	1
9	Ali Hissi	9	99.99	1
10	Mohammed Aboukhalil	10	49.99	1

- Project repository:
- https://github.com/Hadialishibli/Database-assesment.git