King Saud University
College of Computer and Information Sciences
Department of Information Technology

IT222: Database Principles
2nd Semester 1445 H

Jarir

Phase # 3

Section #	NAME	ID
View Name: **customer**		
73033	*Rama Khalid Alomair*	*444200662*
	Alanoud Khalid Alshayea	*444200869*
	Sara Saleh Aloqiel	*444203016*
	Layan Al haider	*444200961*

Supervised By: T. Abeer Aldrees

Project Description:

Jarir bookstore is an offline and online retail store that offers electronics, books, office supplies and more. Jarir has been recognized as the market leader for consumer IT products. The project aspires to design a database for Jarir to manage and sustain data for online orders.

View Description:

The database will be insightful for customers, as it permits them to save personal information, view product details, view and track orders, enabling contact between the courier driver and the customer. This does not only save time but increases customer satisfaction.

Data Requirements:

Customer:

The customer purchases the goods, customers are tracked by unique IDs, additional information includes customer's name, phone number, email, address consisting of city, district and street. Customers may not necessarily have an order yet, but they may submit as many orders as they would like. Upon delivery, a customer would be contacted by a courier driver in order to receive their order.

Order:

Orders in reference to an expanded view of the placed online order. Orders are identified by an order number. They provide insight on order status, delivery fees, coupons if applied VAT summary, total price which is calculated by adding the products' prices to the VAT. Each order belongs to one customer, and it may consist of at least one product. Additionally, each order is picked by one courier driver.

Product:

Products are the goods the customer has purchased. Each product is identified by a different product number. Products carry names, warranty periods as well as prices and product types.

Driver:

The final process of ordering includes delivery, where the placed order is finally delivered. Delivery services require drivers to carry phones, each with a different driver's id, a phone number and a name. Drivers may deliver many orders a day, insinuating the need to contact the customers about submitting their parcels.

Transaction Requirements:

Data Entry:

- *Insert customer information.*
- *Insert a new order for a specific customer (product, coupon code).*

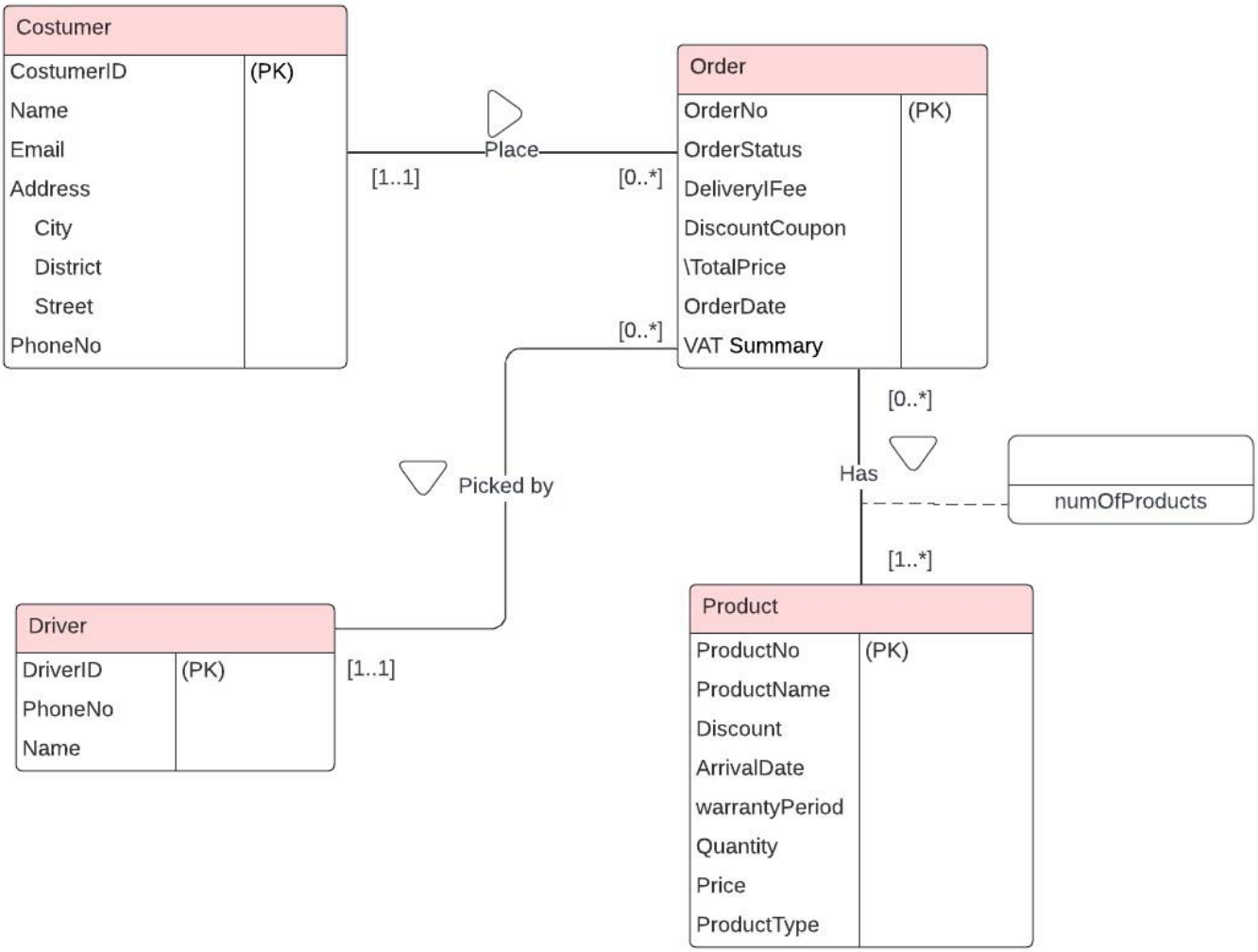
Data update/deletion:

- *Update customer's address.*
- *Update customer's phone number.*
- *Delete customer's information.*

Data Queries:

- *List products within specific type and price range.*
- *Sort products from highest price to lowest and vice versa.*
- *Retrieve customer's information.*
- *Display the last 5 orders for a specific customer.*
- *Display a specific order status.*
- *List products still under warranty (2 years from order date) for a specific customer.*
- *Display discounted products.*
- *List trending products by counting most ordered products and sort them from highest to lowest.*
- *Sort products from newest arrivals to oldest.*
- *Retrieve driver's phone number for a specific order.*
- *Display max and min product price within specific type.*

Global enhanced entity relationship diagram (EER):



Relational Schema:

Costumer (CostumerID , Name, Email, City, District, Street, PhoneNo)
Primary Key: CostumerID

Order (OrderNo, CostumerID, DriverID, OrderStatus, DeliveryIFee, DiscountCoupon, OrderDate, VAT Summary)

Primary Key: OrderNo
Foreign Key: CostumerID references Costumer (CostumerID)
Foreign Key: DriverID references Driver (DriverID)

Product (ProductNo, ProductName, ArrivalDate, Discount, warrantyPeriod, Quantity, Price, ProductType)
Primary Key: ProductNo

Has (OrderNo, ProductNo, numOfProducts)
Primary Key: OrderNo, ProductNo
Foreign Key: OrderNo references Order (OrderNo)
Foreign Key: ProductNo references Product (ProductNo)

Driver (DriverID, PhoneNo, Name)
Primary Key: DriverID

Data Dictionary showing description of all entities:

Entity Name	Description	Occurrence
Customer	The customer is the person that places an order.	A customer can place one, none or many orders
Order	An Order refers to a customer's request made through the online shop.	An order is placed by one and only one customer, has one or more products, and is picked by one and only one driver.
Product	Products are of many types, selected by the customer and added to an order.	Each product is in one, none or many orders.
Driver	The driver is the individual responsible for picking up an order and delivering it to a customer.	A driver picks one, none or many orders.

Data Dictionary showing description of all relationships:

Entity Name	Multiplicity	Relationship	Entity Name	Multiplicity
Customer	1..1	Place	Order	0.. *
Order	0.. *	Has	Product	1.. *
Order	0.. *	Picked by	Driver	1..1

Data Dictionary showing description of all attributes:

Entity Name	Attribute	Description	Data Type	Length	Nulls	Multi-Valued	Default Value	Range	PK
Customer	CustomerID	Uniquely identifies customers based on ID.	Varchar	10					Yes
	Name	Customer's name	Varchar	100					
	Email	Customer's email	Varchar	320					
	Address								
	City	Customer's address's city	Varchar	20					
	District	Customer's address's district	Varchar						
	Street	Customer's street address	Varchar						
	PhoneNo	Customer's phone number	Varchar	15					
Order	OrderNo	Uniquely identifies orders based on their number	Integer						Yes
	OrderStatus	Order's fulfillment status	Varchar					Confirmed/ delivered/ cancelled	
	DeliveryFee	Order's delivery fees	Decimal	6.2					
	DiscountCoupon	Discount if applied	Varchar		Yes				

	Total Price	Total price of order	Decimal	6.2					
	OrderDate	Date order was placed	Date						
	VAT summary	Order’s VAT summary	Decimal	6.2					
Driver	DriverID	Uniquely identifies drivers	Varchar	10					Yes
	PhoneNo	Driver’s phone number	Varchar	15					
	Name	Driver’s name	Varchar	100					
Product	ProductNo	Uniquely identifies products based on their number.	Integer	15					Yes
	ProductName	The product's name	Varchar	100					
	Discount	Discounted product price	Decimal	6.2	Yes				
	ArrivalDate	The date the product has been enlisted	Date						
	warrantyPeriod	The product’s warranty period	Integer		Yes				
	Quantity	The product's quantity	Integer						
	Price	The products price	Decimal	6.2					
	ProductType	The product’s type	Varchar	20					
Relation attribute	numOfProducts	The number of products ordered by the customer	Integer					1 - 49	

DB tables creation commands:

```
CREATE TABLE Product(
 ProductNo Numeric(15) ,
 ProductName Varchar(100) Not null,
 Discount Decimal(6,2),
 ArrivalDate Date Not null,
 warrantyPeriod Numeric,
 Quantity Numeric Not null,
 Price Decimal(6,2) Not null,
 ProductType Varchar(20) Not null,
 constraint product_pk primary key (ProductNo));

CREATE TABLE Driver(
 DriverID Varchar(10),
 PhoneNo Varchar(15) Not null,
 Name Varchar(100) Not null,
 constraint driver_pk primary key (DriverID));

CREATE TABLE Customer(
 CustomerID Varchar(10) ,
 Name Varchar(100) Not null,
 Email Varchar(320) Not null,
City Varchar(20),
District Varchar,
Street Varchar,
PhoneNo Varchar(15) Not null,
constraint customer_pk primary key (CustomerID));
```

```
CREATE TABLE Orders(
OrderNo Numeric Not null,
OrderStatus Varchar Not null,
  CHECK (OrderStatus IN('Confirmed','delivered','cancelled')),
DeliveryFee Decimal(6,2) Not null,
DiscountCoupon Varchar,
TotalPrice Decimal(6,2) Not null,
OrderDate Date Not null,
VATSummary Decimal(6,2) Not null,
ProductNo Numeric(15) ,
  CustomerID Varchar(10) ,
constraint Orders_pk primary key (OrderNo),
FOREIGN KEY(CustomerID)REFERENCES Customer(CustomerID) ON DELETE CASCADE ON UPDATE
CASCADE,
FOREIGN KEY(ProductNo)REFERENCES Product(ProductNo) ON DELETE CASCADE ON UPDATE
CASCADE);
```

```
CREATE TABLE Has
  (
    OrderNo numeric Not null,
    ProductNo numeric(15) Not null,
    numOfProducts numeric Not null
    CHECK(numOfProducts >= 1 AND numOfProducts <= 49),
    constraint Has_fk_Order FOREIGN KEY (OrderNo) REFERENCES Orders(OrderNo)ON DELETE
CASCADE ON UPDATE CASCADE,
    constraint Has_fk_Product FOREIGN KEY (ProductNo) REFERENCES Product(ProductNo) ON DELETE
CASCADE ON UPDATE CASCADE);
```

Data insertion commands:

```
INSERT INTO Product (ProductNo,ProductName,ArrivalDate,Quantity,Price,ProductType)
VALUES (1, ' Roco Pencil Set ', '2022-06-02', 60, 345.25, ' Pencil Set ');

INSERT INTO Product (ProductNo,ProductName,ArrivalDate,Quantity,Price,ProductType)
VALUES (2, ' Faber-Castle Pencil Set ', '2022-07-12', 100, 1900.25, ' Pencil Set ');

INSERT INTO Product (ProductNo,ProductName,ArrivalDate,Quantity,Price,ProductType)
VALUES (3, ' OOLY Lucky Star Charm Pencil Set ', '2022-07-12', 30, 862.5, ' Pencil Set ');

INSERT INTO Product
VALUES (4,'Asus laptop',0.36,'2023-09-06',2, 3 ,8097.50,'Laptops');

INSERT INTO Product
VALUES (5,'Sony soundbar',0.13,'2023-10-10',1, 5 ,6495.75,'Soundbar');

INSERT INTO Product
VALUES (6,'LG SK1 Soundbar',0.05,'2023-12-15',2, 3 ,1197.00,'Soundbar');

INSERT INTO Customer
VALUES ('1000000001', 'Malak Saleh','MalakS@gmail.com', 'Riyadh', 'Nakheel', '64', '+96651234567890');

INSERT INTO Customer
VALUES ('1000000002', 'Sarah Ahmad','SarahA@gmail.com', 'Jeddah', 'Alhamraa', '120', '+96651111222233');

INSERT INTO Orders
VALUES (464184,'delivered',29, NULL,40.5,'2024-4-6',5.28,1,'1000000001');

INSERT INTO Orders
VALUES (517164,'Confirmed',29, NULL,48.57,'2024-4-7',6.26,2,'1000000002');

INSERT INTO Driver
VALUES ('1220110013', ' +966505545188', ' Mohammad Rajab ');
```

```
INSERT INTO Driver
VALUES ('1220110075', ' +966535277106', ' Mustafa Talib ');

INSERT INTO Driver
VALUES ('1220110005', ' +966596482010', ' Basheer Sayyed ');


INSERT INTO Has
VALUES (464184, 1, 2);

INSERT INTO Has
VALUES (517164, 2, 1);
```

Data Queries commands and outputs:

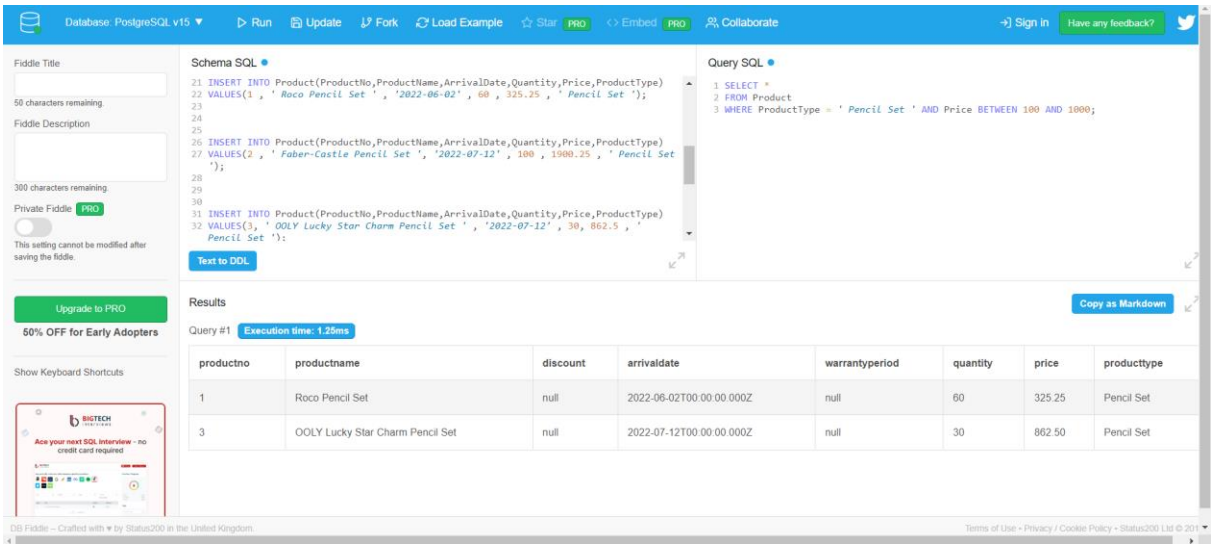
1) ***List products within specific type and price range.***

```
SELECT *

FROM Product

WHERE ProductType = ' Pencil Set ' AND Price BETWEEN 100 AND 1000;
```

Output:



2) ***Sort products from highest price to lowest and vice versa.***

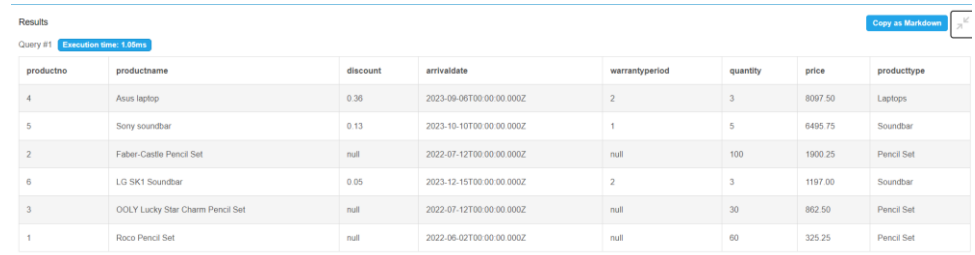
-Highest to lowest:

```
SELECT *

FROM Product

Order by Price DESC;
```

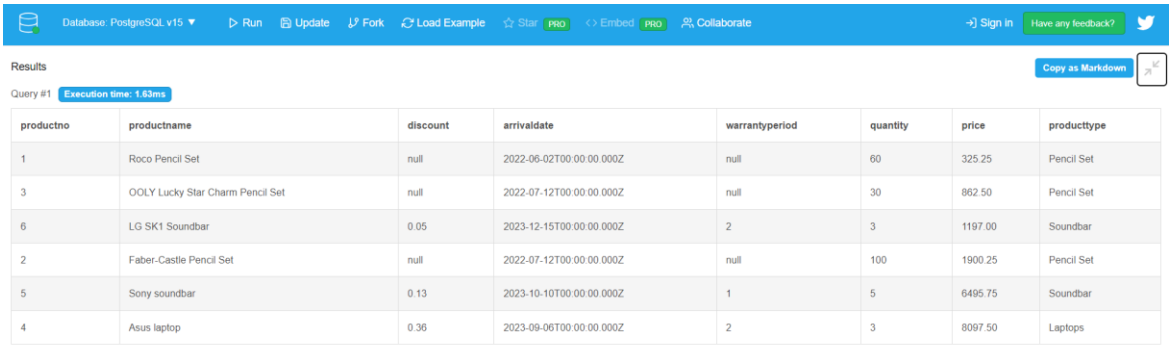
Output:



-lowest to highest:

```
SELECT *
FROM Product
Order by Price;
```

Output:



Results

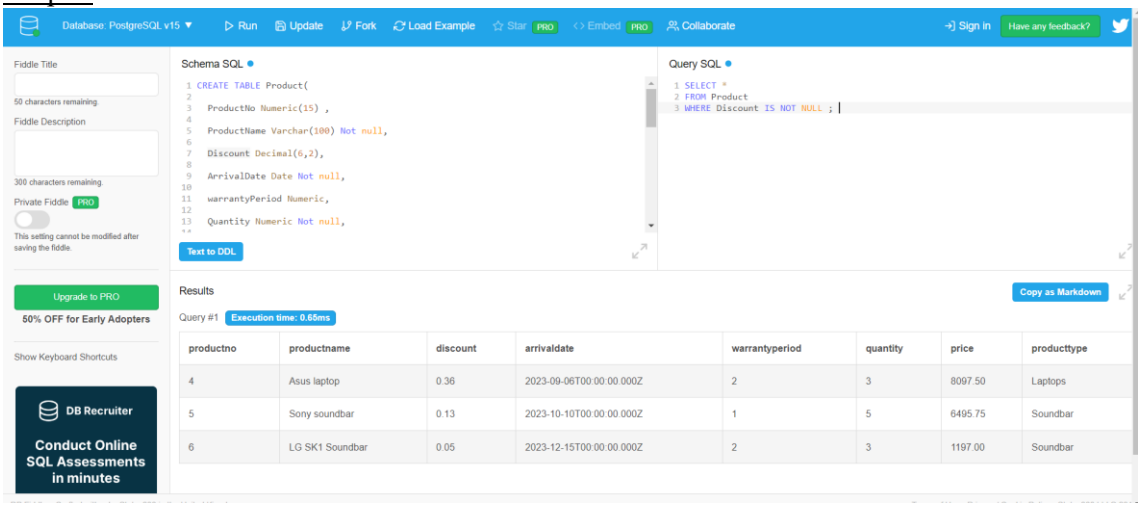
Query #1 Execution time: 1.63ms

productno	productname	discount	arrivaldate	warrantyperiod	quantity	price	producttype
1	Roco Pencil Set	null	2022-06-02T00:00:00.000Z	null	60	325.25	Pencil Set
3	OOLY Lucky Star Charm Pencil Set	null	2022-07-12T00:00:00.000Z	null	30	862.50	Pencil Set
6	LG SK1 Soundbar	0.05	2023-12-15T00:00:00.000Z	2	3	1197.00	Soundbar
2	Faber-Castle Pencil Set	null	2022-07-12T00:00:00.000Z	null	100	1900.25	Pencil Set
5	Sony soundbar	0.13	2023-10-10T00:00:00.000Z	1	5	6495.75	Soundbar
4	Asus laptop	0.36	2023-09-06T00:00:00.000Z	2	3	8097.50	Laptops

3) *Display discounted products.*

```
SELECT *
FROM Product
WHERE Discount IS NOT NULL;
```

Output:



Schema SQL

```
CREATE TABLE Product(

  ProductNo Numeric(15) ,

  ProductName Varchar(100) Not null,

  Discount Decimal(6,2),

  ArrivalDate Date Not null,

  warrantyPeriod Numeric,

  Quantity Numeric Not null,
```

Query SQL

```
SELECT *
FROM Product
WHERE Discount IS NOT NULL ;
```

Results

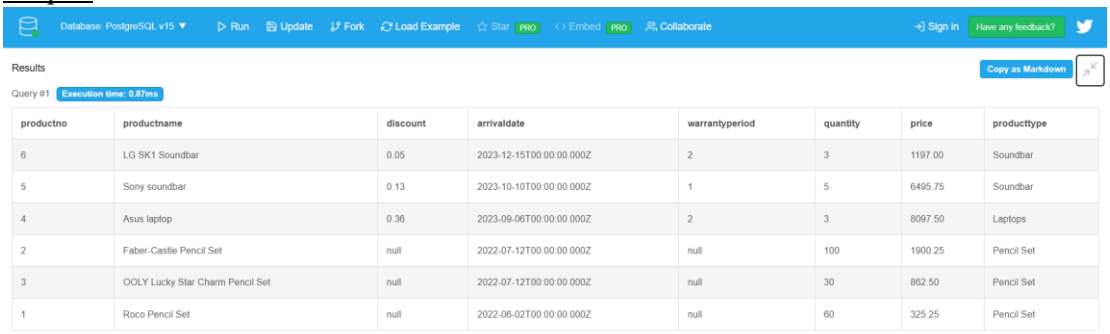
Query #1 Execution time: 0.65ms

productno	productname	discount	arrivaldate	warrantyperiod	quantity	price	producttype
4	Asus laptop	0.36	2023-09-06T00:00:00.000Z	2	3	8097.50	Laptops
5	Sony soundbar	0.13	2023-10-10T00:00:00.000Z	1	5	6495.75	Soundbar
6	LG SK1 Soundbar	0.05	2023-12-15T00:00:00.000Z	2	3	1197.00	Soundbar

4) *Sort products from newest arrivals to oldest.*

```
SELECT *
FROM Product
ORDER BY ArrivalDate DESC;
```

Output:



Results

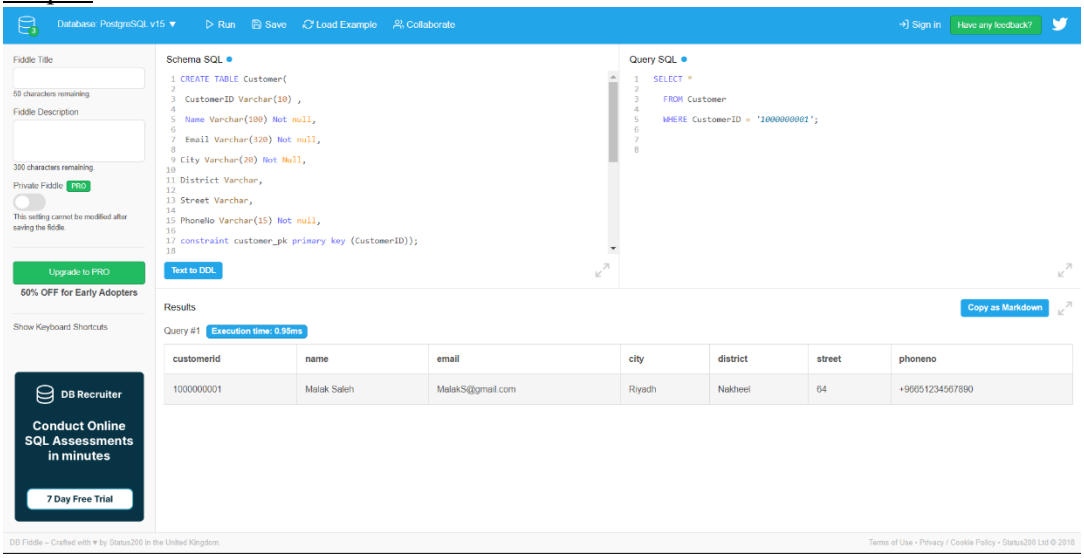
Query #1 Execution time: 0.87ms

productno	productname	discount	arrivaldate	warrantyperiod	quantity	price	producttype
6	LG SK1 Soundbar	0.05	2023-12-15T00:00:00.000Z	2	3	1197.00	Soundbar
5	Sony soundbar	0.13	2023-10-10T00:00:00.000Z	1	5	6495.75	Soundbar
4	Asus laptop	0.36	2023-09-06T00:00:00.000Z	2	3	8097.50	Laptops
2	Faber-Castle Pencil Set	null	2022-07-12T00:00:00.000Z	null	100	1900.25	Pencil Set
3	OOLY Lucky Star Charm Pencil Set	null	2022-07-12T00:00:00.000Z	null	30	862.50	Pencil Set
1	Roco Pencil Set	null	2022-06-02T00:00:00.000Z	null	60	325.25	Pencil Set

5) *Retrieve customer's information.*

```
SELECT *
FROM Customer
WHERE CustomerID = '1000000001';
```

Output:



Schema SQL

```
CREATE TABLE Customer(

 CustomerID Varchar(10) ,

 Name Varchar(100) Not null,

 Email Varchar(320) Not null,

City Varchar(20) Not Null,

District Varchar,

Street Varchar,

PhoneNo Varchar(15) Not null,

constraint customer_pk primary key (CustomerID));
```

Query SQL

```
SELECT *
  FROM Customer
  WHERE CustomerID = '1000000001';
```

Results

Query #1 Execution time: 0.95ms

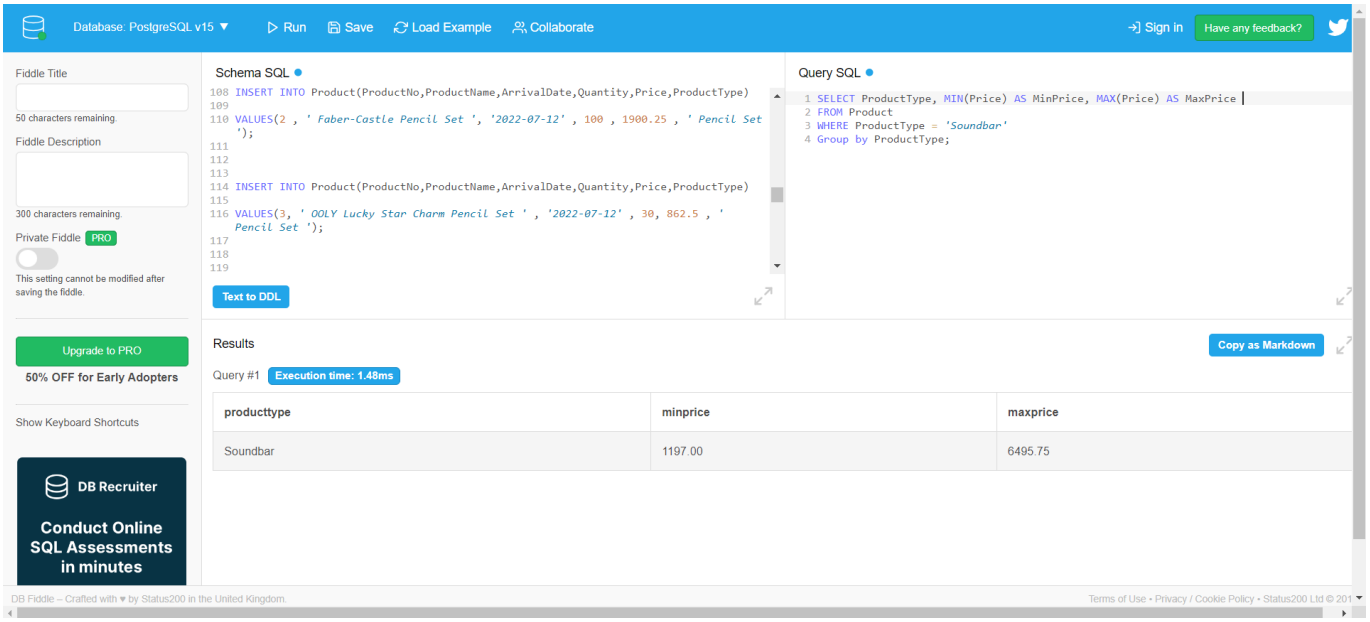
customerid	name	email	city	district	street	phoneno
1000000001	Malak Saleh	MalakS@gmail.com	Riyadh	Nakheel	64	+96651234567890

6) *Display a specific order status*

```
SELECT*
FROM orders
WHERE OrderStatus = 'Confirmed';
```

7) *Display max and min product price within specific type*

```
SELECT ProductType, MIN(Price) AS MinPrice, MAX(Price) AS MaxPrice
FROM Product
WHERE ProductType = 'Soundbar'
Group by ProductType;
```



Work Distribution:

NAME	ID	Percentage	WORK
Rama Khalid Alomair	*444200662*	*25 %*	*Description, view Description, Transaction,requirements review, EER, Relational schema, create table Product and Has, insert values to product table, implements queries related to product table*
Layan Alhaider	*444200961*	*25%*	*Description, view Description, Transaction,requirements review, EER , data dictionary , create table orders, insert values to orders table, implements queries related to orders table*
Sara Aloqiel	*444203016*	*25%*	*Project description, view description, data requirements, attribute data dictionary, create table Deriver, insert values to Deriver table, implements queries related to product table*
Alanoud Khalid Alshayea	*444200869*	*25%*	*Transaction requirements, entities data dictionary, create table Customer , insert values to Customer table, implements queries related to Customer table*