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

IT222: Database Principles

Jan Burger

Phase # 3

Section #	NAME	ID
View Name: customer		
56385	*Raseel Aldawish*	*443203036*
56385	*Norah Aljedai*	*443200841*
56385	*Doaa Abdul hakim Aldobai*	*443203882*
56385	*Ghadah suod alismail*	*443200501*
56385	*Raghad Ahmed Rawih Hassan*	*443204743*

Supervised By: *Abeer Aldrees*

Project Description:

Jan Burger is an online restaurant that offers its services and promotions through mobile phone applications or websites. The Customers can easily choose the dishes they want to have and add them to the order basket This project aims to develop a database for Jan burger restaurants The purpose of Jan burger DB is to maintain the data that is used & generated to support the online restaurant for the clients

View Description:

This view revolves around the customer he/she can order meals, the customer can customize their orders by adding special notes or modifying the ingredients according to their preferences

Data Requirements:

Customer:
A customer is the person who orders the meal from the restaurant application . It has a Name, address, Email, Phone Number, password, and identified by Customer ID. Each customer has zero or many orders.

Orders:
An order is a customer request to buy meals from the restaurant. It has a Date, Status, TotalPrice and identified by Order ID. Each order is owned by one and only one customer.

Branch:

A location that is a franchise in a restaurant business chain. The branch has an address, contact No. and a unique branch number. Each branch provides multiple item options.

Item:

An item is the food or drink that can be ordered in the restaurant. It has price, description, extras, calories, type, size, name, and unique item number. Each item is included in zero or many orders and provided by at least one branch.

Transaction Requirements:

Data Entry:

1- Customer can enter his name.
2- Customer can enter his address.
3- Customer can enter his phone number.
4- Customer can enter his email.

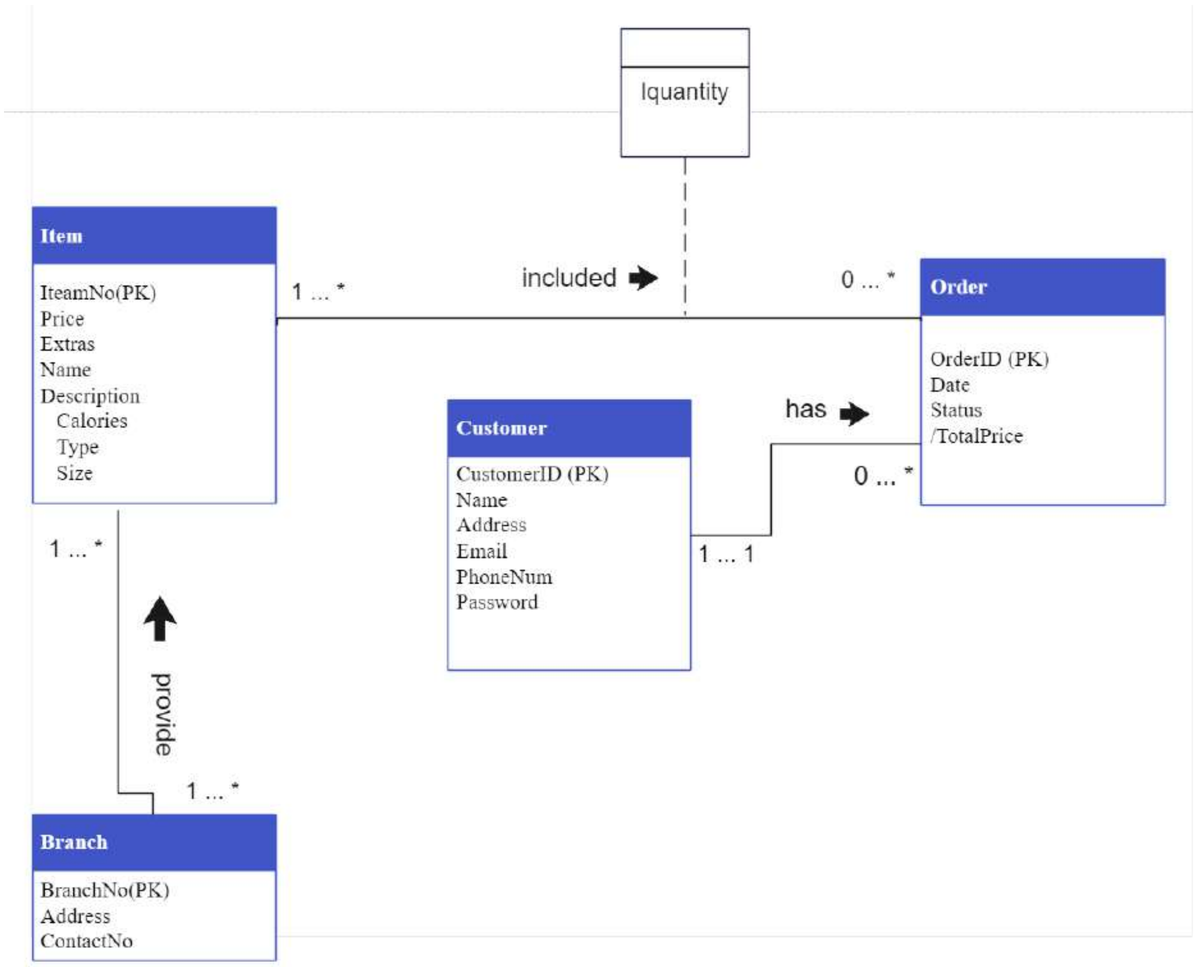
Data update/deletion:

1- Customer can update/delete address.
2- Customer can update his account password.

Data Queries:

1. Display customer info.
2. Display all items.
3. Display all orders.
4. Display items from the lowest to highest Price.
5. Display type side dishes items.
6. Display items based on specific size.
7. Display item by specific name.
8. Display orders from the oldest to newest.
9. Display completed status order.
10. Display items from the lowest to highest calorie.

Global enhanced entity relationship diagram (EER):



Relational Schema:

Item (<u>ItemNo</u>, Price, Extras, Name, Calories, type, size)
Primary Key: ItemNo.

Branch (<u>BranchNo</u>, Address, ContactNo)
Primary Key: BranchNo.

Provide (<u>ItemNo</u>, <u>BranchNo)</u>
Primary Key: BranchNo, ItemNo.
Foreign Key: BranchNo references Branch (BranchNo).
Foreign Key: ItemNo references Item (ItemNo).

Orders (<u>OrderID</u>, Date, Status, CustomerID)
Primary Key: OrderID.
Foreign Key: customerID references Customer (customerID).

Included (<u>OrderID</u> , <u>ItemNo</u> , Quantity)
Primary Key: OrderID, ItemNo.
Foreign Key: OrderID references Orders (OrderID).
Foreign Key: ItemNo references Item (ItemNo).

Customer (<u>customerID</u>, Name, Address, Email, phoneNum, Password)
Primary Key: customerID

Data Dictionary showing description of all entities:

Entity Name	Description	Occurrence
Item	Food or drink that can be ordered in the restaurant.	-Each item is included in zero or many orders. -Each item is provided by at least one branch.
Customer	A person who orders the meal from the restaurant application	Each customer has zero or many orders.
Branch	A franchise location of a restaurant business chain.	Each branch provides multiple item options.
Orders	A customer request to buy meals from the restaurant.	-Each order is owned by one and only one customer - Each order includes one or many items.

Data Dictionary showing description of all relationships:

Entity Name	Multiplicity	Relationship	Entity Name	Multiplicity
Customer	1..1	has	Orders	0..*
Item	1..*	Included	Orders	0..*
Branch	1..*	provide	Item	1..*

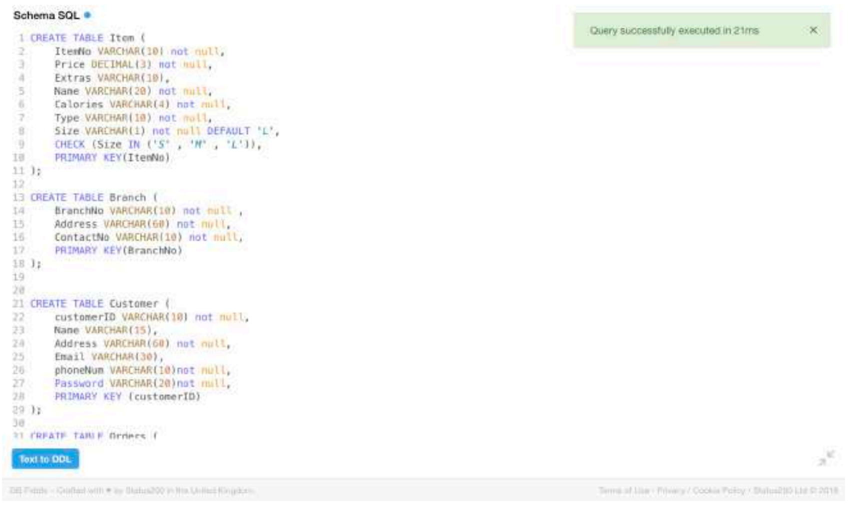
Data Dictionary showing description of all attributes:

Entity Name	Attribute	Description	Data Type	Length	Nulls	Multi-Valued	Default Value	Range	PK
Customer	CustomerID	ID uniquely identifies the customer	VARCHAR	10					YES
	Name	The name of the customer	VARCHAR	15	Yes				
	Address	Address of the customer	VARCHAR	60					
	Email	Email of the customer	VARCHAR	30	YES				
	PhoneNum	Phone number of the customer	VARCHAR	10					
	Password	The password of the customer	VARCHAR	20					
Item	ItemNo	Number of the item uniquely identifies the item	VARCHAR	10					YES
	Name	The name of the item	VARCHAR	20					
	Price	The cost of the item	DECIMAL	3					
	Extras	The extras the can be added to the item	VARCHAR	10	YES				
	Description								
	Calories	the calories of item	VARCHAR	4					
	Type	the type of item	VARCHAR	10					
	Size	the size of item	CHARACTER	1			L	S,M,L	

	BranchNo	The number of branches uniquely identifies branch	VARCHAR	10					YES
Branch	Address	The address of the branch	VARCHAR	60					
	ContactNo	The contact number	VARCHAR	10					
	OrderID	The order ID, which uniquely identifies the order	VARCHAR	10					YES
Orders	TotalPrice	The total price of the order	DECIMAL	4					
	Date	The date of the order	DATE						
	Status	The status of the order	VARCHAR	10	YES				

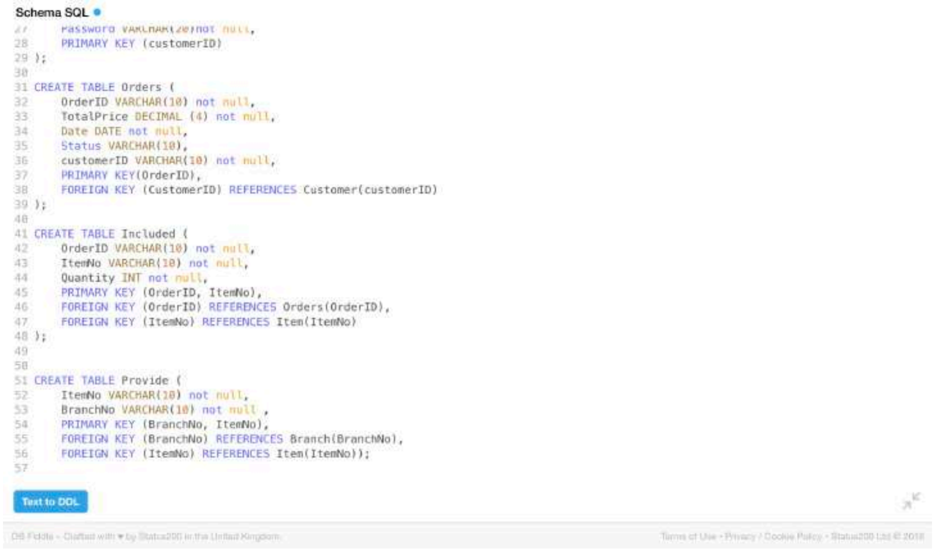
DB tables creation commands:

```
CREATE TABLE Item (
   ItemNo VARCHAR(10) not null,
   Price DECIMAL(3) not null,
   Extras VARCHAR(10),
   Name VARCHAR(20) not null,
   Calories VARCHAR(4) not null,
   Type VARCHAR(10) not null,
   Size VARCHAR(1) not null DEFAULT 'L',
   CHECK (Size IN ('S' , 'M' , 'L')),
   PRIMARY KEY(ItemNo)
);

CREATE TABLE Branch (
   BranchNo VARCHAR(10) not null ,
   Address VARCHAR(60) not null,
   ContactNo VARCHAR(10) not null,
   PRIMARY KEY(BranchNo)
);


CREATE TABLE Customer (
   customerID VARCHAR(10) not null,
   Name VARCHAR(15),
   Address VARCHAR(60) not null,
   Email VARCHAR(30),
   phoneNum VARCHAR(10)not null,
   Password VARCHAR(20)not null,
   PRIMARY KEY (customerID)
);

CREATE TABLE Orders (
   OrderID VARCHAR(10) not null,
   TotalPrice DECIMAL (4) not null,
   Date DATE not null,
   Status VARCHAR(10),
   customerID VARCHAR(10) not null,
   PRIMARY KEY(OrderID),
   FOREIGN KEY (CustomerID) REFERENCES Customer(customerID)
);

CREATE TABLE Included (
   OrderID VARCHAR(10) not null,
   ItemNo VARCHAR(10) not null,
   Quantity INT not null,
   PRIMARY KEY (OrderID, ItemNo),
   FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
   FOREIGN KEY (ItemNo) REFERENCES Item(ItemNo)
);


CREATE TABLE Provide (
   ItemNo VARCHAR(10) not null,
   BranchNo VARCHAR(10) not null ,
   PRIMARY KEY (BranchNo, ItemNo),
   FOREIGN KEY (BranchNo) REFERENCES Branch(BranchNo),
   FOREIGN KEY (ItemNo) REFERENCES Item(ItemNo));
```


Schema SQL

Query successfully executed in 21ms

```
CREATE TABLE Item (
    ItemNo VARCHAR(10) not null,
    Price DECIMAL(3) not null,
    Extras VARCHAR(10),
    Name VARCHAR(20) not null,
    Calories VARCHAR(4) not null,
    Type VARCHAR(10) not null,
    Size VARCHAR(1) not null DEFAULT 'L',
    CHECK (Size IN ('S' , 'M' , 'L')),
    PRIMARY KEY(ItemNo)
);

CREATE TABLE Branch (
    BranchNo VARCHAR(10) not null ,
    Address VARCHAR(60) not null,
    ContactNo VARCHAR(10) not null,
    PRIMARY KEY(BranchNo)
);


CREATE TABLE Customer (
    customerID VARCHAR(10) not null,
    Name VARCHAR(15),
    Address VARCHAR(60) not null,
    Email VARCHAR(30),
    phoneNum VARCHAR(10)not null,
    Password VARCHAR(20)not null,
    PRIMARY KEY (customerID)
);

CREATE TABLE Orders (
```

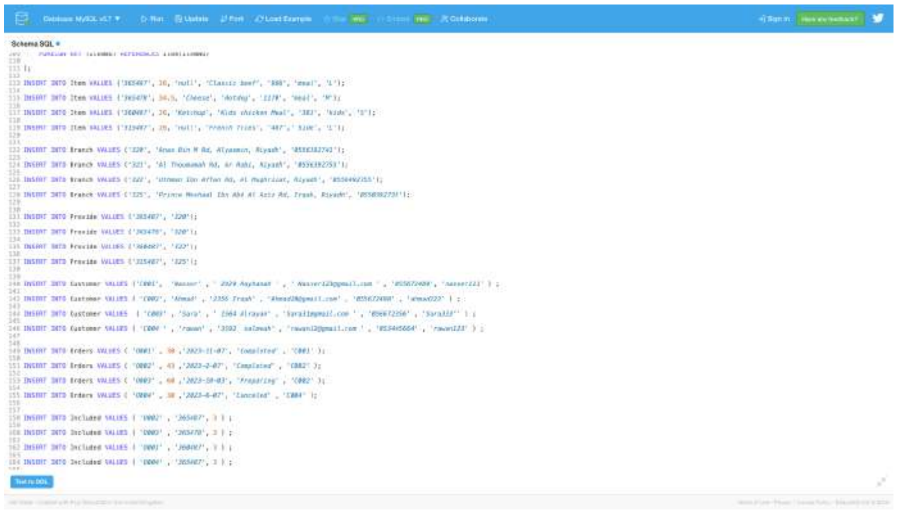

Schema SQL

```
    Password VARCHAR(20)not null,
    PRIMARY KEY (customerID)
);

CREATE TABLE Orders (
    OrderID VARCHAR(10) not null,
    TotalPrice DECIMAL (4) not null,
    Date DATE not null,
    Status VARCHAR(10),
    customerID VARCHAR(10) not null,
    PRIMARY KEY(OrderID),
    FOREIGN KEY (CustomerID) REFERENCES Customer(customerID)
);

CREATE TABLE Included (
    OrderID VARCHAR(10) not null,
    ItemNo VARCHAR(10) not null,
    Quantity INT not null,
    PRIMARY KEY (OrderID, ItemNo),
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    FOREIGN KEY (ItemNo) REFERENCES Item(ItemNo)
);


CREATE TABLE Provide (
    ItemNo VARCHAR(10) not null,
    BranchNo VARCHAR(10) not null ,
    PRIMARY KEY (BranchNo, ItemNo),
    FOREIGN KEY (BranchNo) REFERENCES Branch(BranchNo),
    FOREIGN KEY (ItemNo) REFERENCES Item(ItemNo));
```

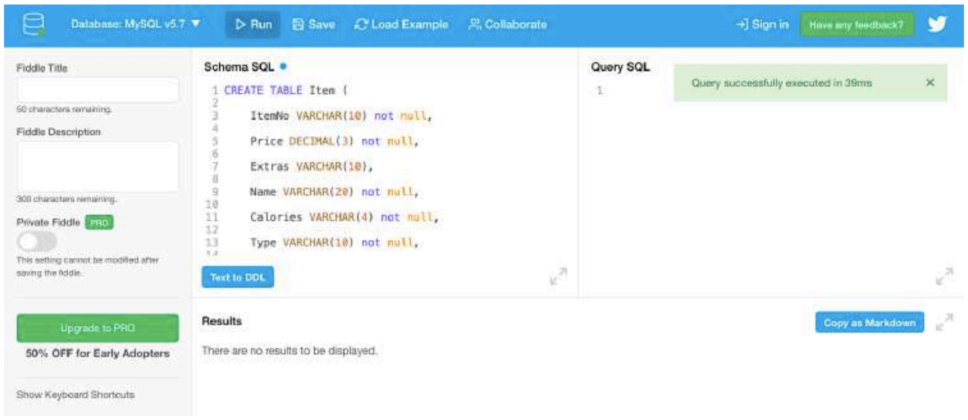
Data insertion commands:

```
INSERT INTO Item VALUES ('365487', 26, 'null', 'Classic beef', '986', 'Meal', 'L');
INSERT INTO Item VALUES ('365478', 34.5, 'Cheese', 'Hotdog', '1170', 'Meal', 'M');
INSERT INTO Item VALUES ('360487', 26, 'Ketchup', 'Kids chicken Meal', '383', 'kids', 'S');
INSERT INTO Item VALUES ('315487', 26, 'null', 'French fries', '487','Side', 'L');


INSERT INTO Branch VALUES ('320', 'Anas Bin M Rd, Alyasmin, Riyadh', '0556392741');
INSERT INTO Branch VALUES ('321', 'Al Thoumamah Rd, Ar Rabi, Riyadh', '0556392755');
INSERT INTO Branch VALUES ('322', 'Uthman Ibn Affan Rd, Al Mughrizat, Riyadh',
'0556492755');
INSERT INTO Branch VALUES ('325', 'Prince Meshaal Ibn Abd Al Aziz Rd, Irqah, Riyadh',
'0550392735');


INSERT INTO Provide VALUES ('365487', '320');
INSERT INTO Provide VALUES ('365478', '320');
INSERT INTO Provide VALUES ('360487', '322');
INSERT INTO Provide VALUES ('315487', '325');


INSERT INTO Customer VALUES ('C001',  'Nasser' , ' 2929 Rayhanah ' , ' Nasser123@gmail.com
' , '055672489', 'nasser111' ) ;
INSERT INTO Customer VALUES ( 'C002', 'Ahmad' , '2356 Irqah' , 'Ahmad20@gmail.com' ,
'055672498' , 'ahmad222' ) ;
INSERT INTO Customer VALUES  ( 'C003' , 'Sara' , ' 1564 Alrayan' , 'Sara31@gmail.com ' ,
'056672356' , 'Sara333’' ) ;
INSERT INTO Customer VALUES ( 'C004 ' , 'rawan' , '3592  salmeah' , 'rawan12@gmail.com ' ,
'053445664' , 'rawan123' ) ;


INSERT INTO Orders VALUES ( 'O001' , 30 ,'2023-11-07', 'Completed' , 'C001' );
INSERT INTO Orders VALUES ( 'O002' , 43 ,'2023-2-07', 'Completed' , 'C002' );
INSERT INTO Orders VALUES ( 'O003' , 60 ,'2023-10-03', 'Preparing' , 'C002' );
INSERT INTO Orders VALUES ( 'O004' , 30 ,'2023-6-07', 'Canceled' , 'C004' );
```

INSERT INTO Included VALUES ('O002' , '365487', 3) ;
INSERT INTO Included VALUES ('O003' , '365478', 3) ;
INSERT INTO Included VALUES ('O001' , '360487', 3) ;
INSERT INTO Included VALUES ('O004' , '365487', 3) ;


Schema SQL

```
);

INSERT INTO Item VALUES ('365487', 30, 'null', 'Classic beef', '880', 'meal', 'L');

INSERT INTO Item VALUES ('365478', 34.5, 'Cheese', 'Hotdog', '1170', 'meal', 'M');

INSERT INTO Item VALUES ('360487', 30, 'Ketchup', 'Kids chicken Meal', '382', 'kids', 'S');

INSERT INTO Item VALUES ('315487', 20, 'null', 'French fries', '487', 'side', 'L');

INSERT INTO Branch VALUES ('320', 'Anas Bin M Rd, Alyasmin, Riyadh', '0558382740');

INSERT INTO Branch VALUES ('321', 'Al Thoomamah Rd, Ar Rabi, Riyadh', '0556382753');

INSERT INTO Branch VALUES ('322', 'Uthman Ibn Affan Rd, Al Maghrizat, Riyadh', '0550492355');

INSERT INTO Branch VALUES ('325', 'Prince Meshaal Ibn Abd Al Aziz Rd, Irqah, Riyadh', '0558382731');

INSERT INTO Provide VALUES ('365487', '320');

INSERT INTO Provide VALUES ('365478', '320');

INSERT INTO Provide VALUES ('360487', '322');

INSERT INTO Provide VALUES ('315487', '325');

INSERT INTO Customer VALUES ('C001',  'Nasser' , ' 2929 Ayshanah ' , ' Nasser123@gmail.com ' , '055672400', 'nasser111' ) ;

INSERT INTO Customer VALUES ( 'C002', 'Ahmad' , '2356 Irqah' , 'Ahmad20@gmail.com' , '0556724000' , 'ahmad222' ) ;

INSERT INTO Customer VALUES  ( 'C003' , 'Sara' , ' 1564 Alrayan' , 'Sara11@gmail.com ' , '0566672256' , 'Sara333'' ) ;

INSERT INTO Customer VALUES ( 'C004 ' , 'rawan' , '3592  salmah' , 'rawan12@gmail.com ' , '0554456664' , 'rawan123' ) ;

INSERT INTO Orders VALUES ( 'O001' , 30 ,'2023-11-07', 'Completed' , 'C001' );

INSERT INTO Orders VALUES ( 'O002' , 43 ,'2023-2-07', 'Completed' , 'C002' );

INSERT INTO Orders VALUES ( 'O003' , 60 ,'2023-10-03', 'Preparing' , 'C002' );

INSERT INTO Orders VALUES ( 'O004' , 30 ,'2023-6-07', 'Canceled' , 'C004' );

INSERT INTO Included VALUES ( 'O002' , '365487', 3 ) ;

INSERT INTO Included VALUES ( 'O003' , '365478', 3 ) ;

INSERT INTO Included VALUES ( 'O001' , '360487', 3 ) ;

INSERT INTO Included VALUES ( 'O004' , '365487', 3 ) ;
```



Schema SQL

```
CREATE TABLE Item (

    ItemNo VARCHAR(10) not null,

    Price DECIMAL(3) not null,

    Extras VARCHAR(10),

    Name VARCHAR(20) not null,

    Calories VARCHAR(4) not null,

    Type VARCHAR(10) not null,
```

Query SQL

Query successfully executed in 39ms

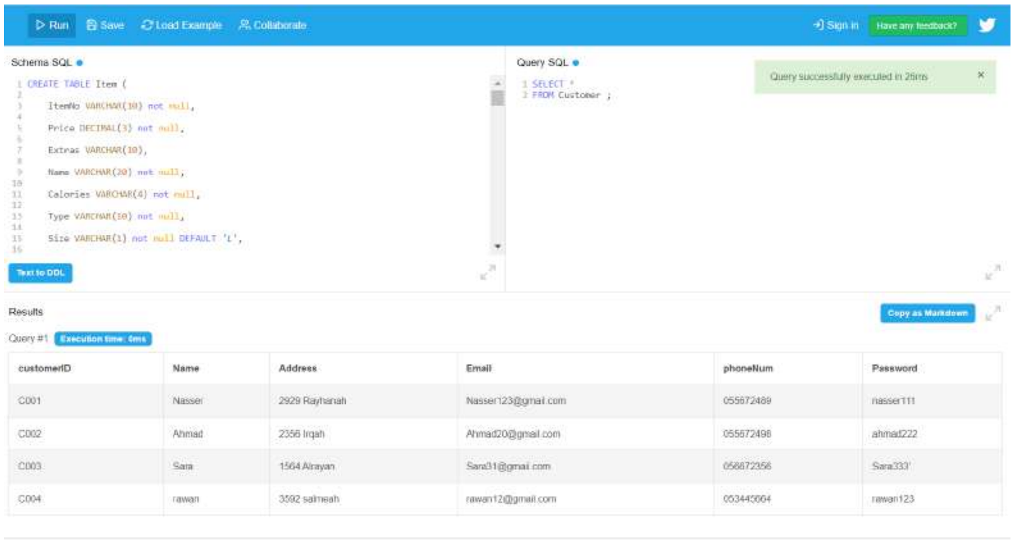
Results

There are no results to be displayed.

Data Queries commands and outputs:

1) Display customer info.

SELECT *
FROM Customer ;



```
CREATE TABLE Item (

    ItemNo VARCHAR(10) not null,

    Price DECIMAL(3) not null,

    Extras VARCHAR(10),

    Name VARCHAR(20) not null,

    Calories VARCHAR(4) not null,

    Type VARCHAR(10) not null,

    Size VARCHAR(1) not null DEFAULT 'L',

```

```
SELECT *
FROM Customer ;
```

Query successfully executed in 26ms

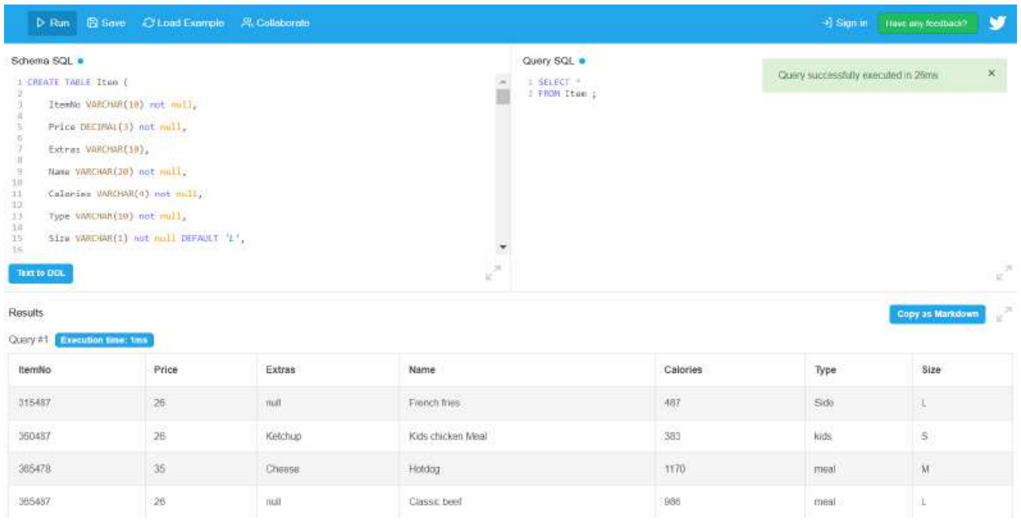
Results

Query #1 Execution time: 0ms

customerID	Name	Address	Email	phoneNum	Password
C001	Nasser	2929 Rayhanah	Nasser123@gmail.com	055672489	nasser111
C002	Ahmad	2356 Irqah	Ahmad20@gmail.com	055672498	ahmad222
C003	Sara	1564 Alrayan	Sara31@gmail.com	056672356	Sara333'
C004	rawan	3592 salmeah	rawan12@gmail.com	053445664	rawan123

2) Display all items.

SELECT *
FROM Item ;



```
CREATE TABLE Item (

    ItemNo VARCHAR(10) not null,

    Price DECIMAL(3) not null,

    Extras VARCHAR(10),

    Name VARCHAR(20) not null,

    Calories VARCHAR(4) not null,

    Type VARCHAR(10) not null,

    Size VARCHAR(1) not null DEFAULT 'L',

```

```
SELECT *
FROM Item ;
```

Query successfully executed in 26ms

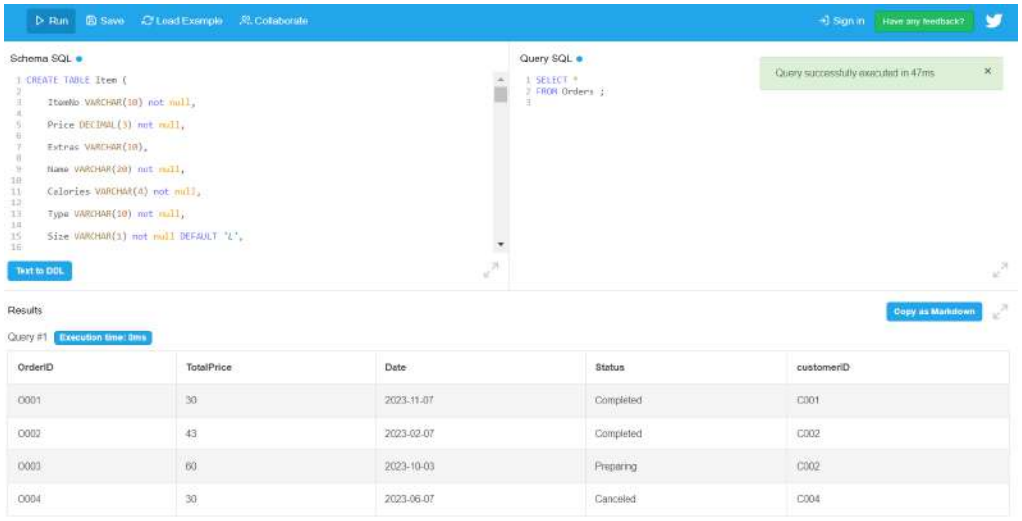
Results

Query #1 Execution time: 1ms

ItemNo	Price	Extras	Name	Calories	Type	Size
315487	26	null	French fries	487	Side	L
360487	26	Ketchup	Kids chicken Meal	383	kids	S
365478	35	Cheese	Hotdog	1170	meal	M
365487	26	null	Classic beef	986	meal	L

3) **Display all orders.**

SELECT *
FROM Orders ;



```
CREATE TABLE Item (

    ItemNo VARCHAR(10) not null,

    Price DECIMAL(3) not null,

    Extras VARCHAR(10),

    Name VARCHAR(20) not null,

    Calories VARCHAR(4) not null,

    Type VARCHAR(10) not null,

    Size VARCHAR(1) not null DEFAULT 'L',

```

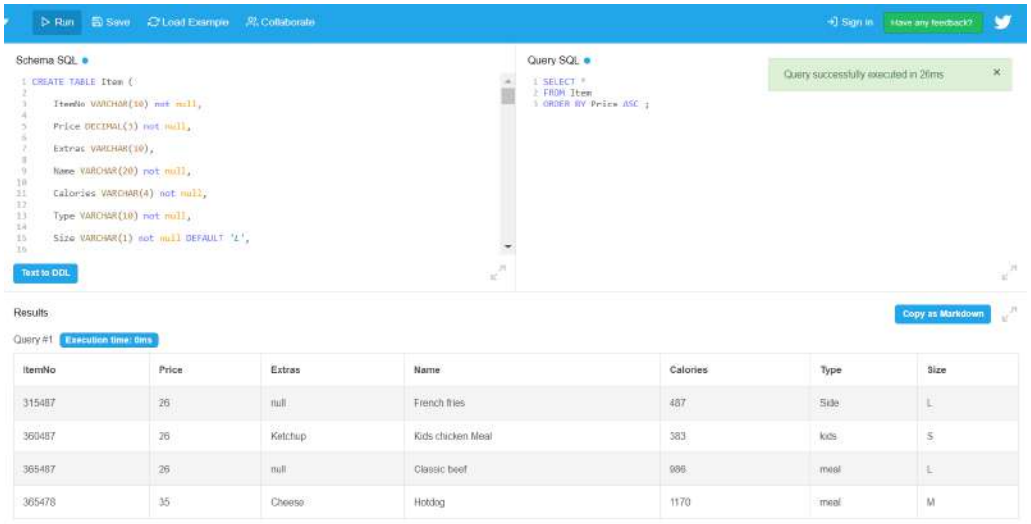
```
SELECT *
FROM Orders ;
```

Query successfully executed in 47ms

OrderID	TotalPrice	Date	Status	customerID
O001	30	2023-11-07	Completed	C001
O002	43	2023-02-07	Completed	C002
O003	60	2023-10-03	Preparing	C002
O004	30	2023-06-07	Canceled	C004

4) **Display items from the lowest to highest Price.**

SELECT *
FROM Item
ORDER BY Price ASC ;



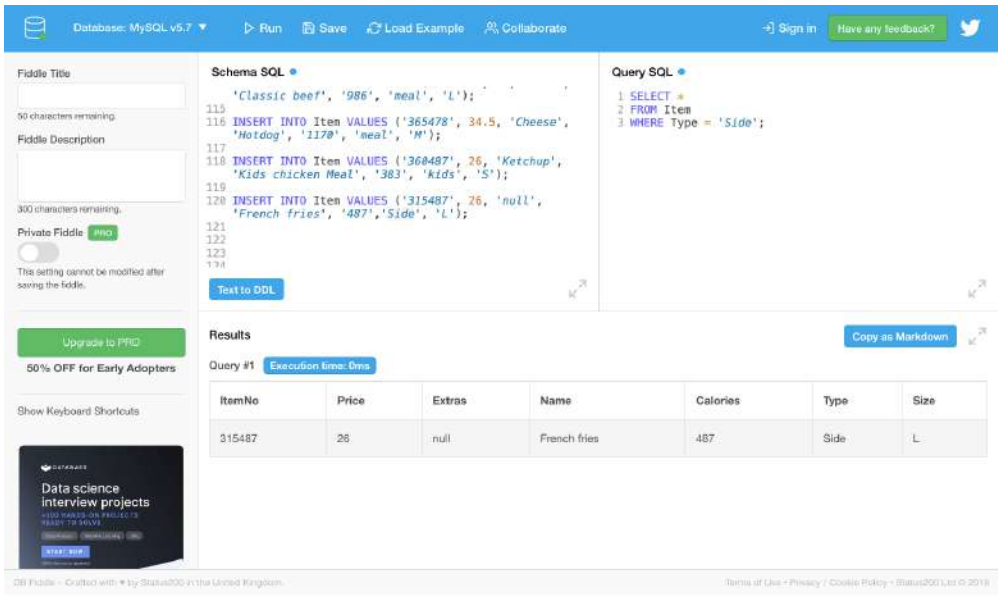
```
CREATE TABLE Item (

    ItemNo VARCHAR(10) not null,

    Price DECIMAL(3) not null,

    Extras VARCHAR(10),

    Name VARCHAR(20) not null,

    Calories VARCHAR(4) not null,

    Type VARCHAR(10) not null,

    Size VARCHAR(1) not null DEFAULT 'L',

```

```
SELECT *
FROM Item
ORDER BY Price ASC ;
```

Query successfully executed in 26ms

ItemNo	Price	Extras	Name	Calories	Type	Size
315487	26	null	French fries	487	Side	L
360487	26	Ketchup	Kids chicken Meal	383	kids	S
365487	26	null	Classic beef	986	meal	L
365478	35	Cheese	Hotdog	1170	meal	M

5) **Display type side dishes items.**
SELECT *
FROM Item
WHERE Type = ‘Side’;



Schema SQL

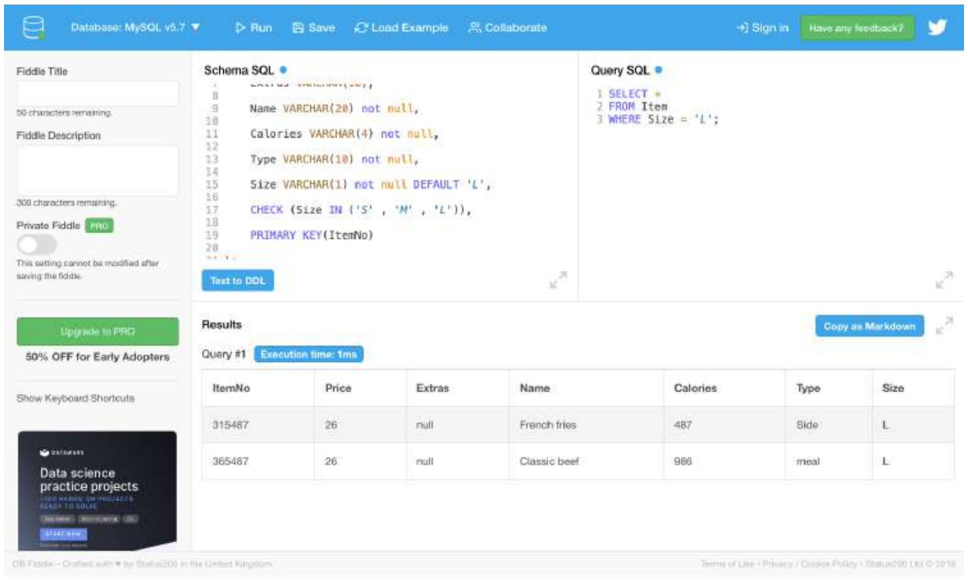
```
    'Classic beef', '986', 'meal', 'L');

INSERT INTO Item VALUES ('365478', 34.5, 'Cheese',
    'Hotdog', '1170', 'meal', 'M');

INSERT INTO Item VALUES ('360487', 26, 'Ketchup',
    'Kids chicken Meal', '383', 'kids', 'S');

INSERT INTO Item VALUES ('315487', 26, 'null',
    'French fries', '487','Side', 'L');
```

Query SQL

```
SELECT *
FROM Item
WHERE Type = 'Side';
```

ItemNo	Price	Extras	Name	Calories	Type	Size
315487	26	null	French fries	487	Side	L

6) **Display items based on specific size.**
SELECT *
FROM Item
WHERE Size = ‘L’;



Schema SQL

```
    Name VARCHAR(20) not null,

    Calories VARCHAR(4) not null,

    Type VARCHAR(10) not null,

    Size VARCHAR(1) not null DEFAULT 'L',

    CHECK (Size IN ('S' , 'M' , 'L')),

    PRIMARY KEY(ItemNo)
```

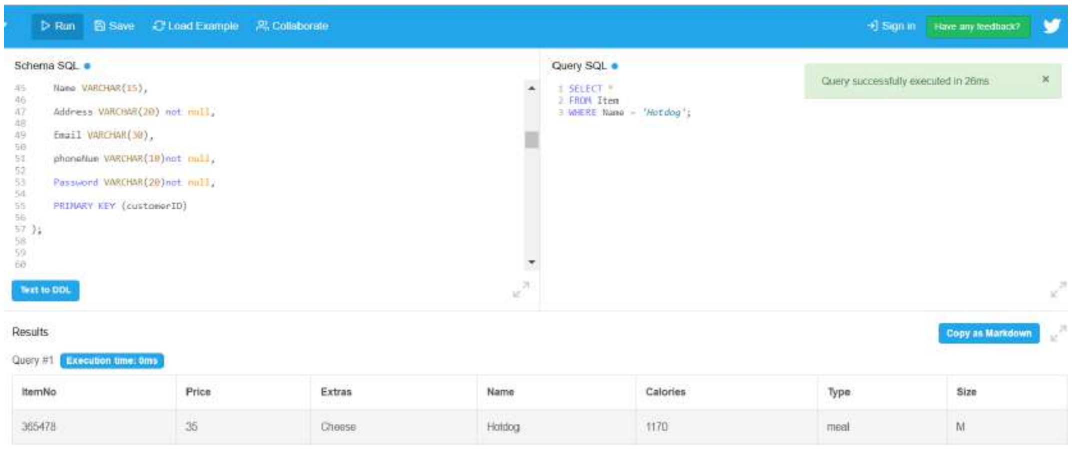
Query SQL

```
SELECT *
FROM Item
WHERE Size = 'L';
```

ItemNo	Price	Extras	Name	Calories	Type	Size
315487	26	null	French fries	487	Side	L
365487	26	null	Classic beef	986	meal	L

7) Display item by specific name.

SELECT *
FROM Item
WHERE Name = ‘Hotdog’ ;



```
    Name VARCHAR(15),

    Address VARCHAR(20) not null,

    Email VARCHAR(30),

    phoneNum VARCHAR(10)not null,

    Password VARCHAR(20)not null,

    PRIMARY KEY (customerID)

);
```

```
SELECT *
FROM Item
WHERE Name = 'Hotdog';
```

Query successfully executed in 26ms

Results

Query #1 Execution time: 0ms

ItemNo	Price	Extras	Name	Calories	Type	Size
365478	35	Cheese	Hotdog	1170	meal	M

Work Distribution:

NAME	ID	Percentage	WORK
Raseel Aldawish	*443203036*	*100%*	*-Data Requirements: branch & item.* *- Data dictionary for entities.* *-Insertion commands: Item, Branch, and Provide.*
Raghad Ahmed Hassan	*443204743*	*100%*	*-Transaction requirements: Data entry, Data update/deletion, Data queries* *-Order, included, and customer schemes.*
Doaa Abdul hakim	*443203882*	*100%*	*-Data Requirements: customer order* *-Data Dictionary for relationships* *- Data Queries commands and outputs*
Norah Nasser aljedai	*443200841*	*100%*	-Global enhanced entity relationship diagram *-Data Dictionary showing description of all attributes* *-Insertion commands: Customer ,Order and Included*
Ghadah Suod Alismail	*443200501*	*100%*	*-Project description, view description* *-Schemes: Item and branch* *- Data Queries commands and outputs*