

**University of British Columbia**  
**Department of Computer Science**  
**CPSC 304 2019 W2**

**Group Project - Implementation of a Relational Database**

<b>Project Title:</b>	Designing a Restaurant Inventory Application
<b>Project Milestone:</b>	Milestone 4: Implementation

#	Student Name	Student Number	Email Address
1	Gerrit Van Esch	99978819	gerritvanesch@gmail.com
2	William Luo	23458145	williamluo555@gmail.com
3	Noor Elramley	55759154	noor.ramley@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above.

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

## 1. Project Description

The project consists of an inventory tracking and ordering system that communicates between Restaurants, Wholesalers, and Customers. Restaurants get their inventory from Wholesalers via inventory orders, then assemble menuitems using those ingredients and sell them to Customers via food orders. Each of the 3 types of users have their own login/registration system where they can control things relevant to their user types. Restaurants can manually restock inventory from wholesalers, or set up auto restocking rules so that when inventory dips below a specified level it will automatically trigger an order from the defined wholesalers. Finally, they can extract some aggregated details about food orders so that they can see which menu items are the most popular or are bringing in the most money.

In the process of implementing this project, we learned a lot about how to write certain types of SQL queries, but perhaps we learned even more about how we could have improved our design in the beginning to make our lives easier in the implementation phase. This includes items such as making sure our keys are as minimal as possible, and redesigning things in some areas to reduce the amount of keys; being smarter about how we defined our primary keys for things like OrderDuration (which effectively prevents us from submitting more than one order in the same second); and being more consistent about where we used different specifications for ingredients or possibly condensing it into a single table somehow.

Our project is quite a massive one, which we did not completely realize when we started. In some instances we let code quality standards slide because the focus was on the database, but it would have helped us quite a bit to keep it high and DRY. Additionally, we spent a lot of time on implementing types of queries which we had already implemented dozens of times in other parts of the project, so we may have benefited from spreading out our focus more and giving each rubric item a more equal amount of effort. Perhaps that could also have been foreseen a bit more and planned for, but that's not as relevant to this discussion.

We did come up with some neat techniques for sorting filtering, or adjusting the parameters of the reports for example. Once we started adding a little more JS into the picture, we got some nice results and the reports are quite fun to use.

Beyond this, the project gave us an opportunity to work on the aspects of SQL that aren't quite as common, or even use queries that were not covered in the course. For example, I have personally never used SQL division in any of my other projects, so it was nice to have a chance to learn something new and practice those skills.

## 2. Final Schema Differences

Notes:

Underline represents primary key

Bolded represents foreign key

Previous Schema:

Customer(UID: INTEGER, Name: VARCHAR(30), **AID**: INTEGER, PhoneNumber: **IN-TEGER**, EmailAddress: VARCHAR(40), Password: VARCHAR(64))

New Schema:

Customer(UID: INTEGER, Name: VARCHAR(30), **AID**: INTEGER, PhoneNumber: **VAR-CHAR(10)**, EmailAddress: VARCHAR(40), Password: VARCHAR(64))

Reason:

We changed PhoneNumber type to VARCHAR instead of INTEGER because integer will overflow on a 10 digit phone number.

Previous Schema:

Restaurant(UID: INTEGER, Name: VARCHAR(30), **AID**: INTEGER, PhoneNumber: **IN-TEGER**, EmailAddress: VARCHAR(40), Password: VARCHAR(64))

New Schema:

Restaurant(UID: INTEGER, Name: VARCHAR(30), **AID**: INTEGER, PhoneNumber: **VAR-CHAR(10)**, EmailAddress: VARCHAR(40), Password: VARCHAR(64))

Reason:

We changed PhoneNumber type to VARCHAR instead of INTEGER because integer will overflow on a 10 digit phone number.

Previous Schema:

Wholesaler(UID: INTEGER, Name: VARCHAR(30), **AID**: INTEGER, PhoneNumber: **IN-TEGER**, EmailAddress: VARCHAR(40), Password: VARCHAR(64))

New Schema:

Wholesaler(UID: INTEGER, Name: VARCHAR(30), **AID**: INTEGER, PhoneNumber: **VAR-CHAR(10)**, EmailAddress: VARCHAR(40), Password: VARCHAR(64))

Reason:

We changed PhoneNumber type to VARCHAR instead of INTEGER because integer will overflow on a 10 digit phone number.

Previous Schema:

AutoRestock(UID: INTEGER, AUID: INTEGER, INID: INTEGER, Amount: INTEGER, Threshold: INTEGER)

New Schema:

AutoRestock(UID: INTEGER, AUID: INTEGER, **WholesalerUID**: INTEGER, **IID**: IN-TEGER, **INID**: INTEGER, Amount: INTEGER, Threshold: INTEGER)

Reason:

We realized we needed to know which item the auto restocker should place an order on to restock a certain ingredient for a certain restaurant. Without this change, there would be multiple items it could select and it wouldn't be consistent.

### 3. List of all SQL Queries

Quick Links to Rubric Items:

- Insert operation: AutoRestock
- Update operation: Address
- Delete operation: AutoRestock
- Selection: Food order details
- Projection: Reports → Order Statistics
- Join Query: Food order history
- Aggregation Query: Orders → Counts above tables
- Nested Aggregation w/ Group By Query: Reports → Order Statistics
- Division Query: Reports → Customer Loyalty

Note: There may be more than one for each type of query.

Note: No output for all insert/update queries.

```
SELECT customer.Name AS CustomerName,
       customer.PhoneNumber,
       customer.EmailAddress,
       restaurant.Name AS RestaurantName,
       restaurant.UID,
       address.Street,
       address.HouseNumber,
       address.ApartmenNumber,
       address.PostalCode,
       postalcode.Province,
       postalcode.City,
       foodorder.OID,
       foodorder.OrderDate,
       foodorder.FulfilledDate,
       orderduration.OrderDuration
  FROM foodorder
 INNER JOIN orderduration ON foodorder.OrderDate=orderduration.OrderDate AND
 foodOrder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN restaurant ON restaurant.UID=foodorder.RestaurantUID
 INNER JOIN customer ON customer.UID=foodorder.CustomerUID
 INNER JOIN address ON customer.AID=address.AID
 INNER JOIN postalcode ON address.PostalCode=postalcode.PostalCode
 WHERE foodorder.OID=? AND foodorder.CustomerUID=?
```

Output:

**Order ID: 1**

**Restaurant: Japanese Ramen and Sushi Bar**

Order Date: 2020-03-25 08:00:00

Fulfilled Date: 2020-03-25 08:20:00

Order Duration: 00:20:00

**Customer Information**

Name: Sally

Phone Number: 604-644-7180

Email Address: sally@mail.com

Address: 1005-6335 Thunderbird Crescent, Vancouver, BC, V6T2G9

Apartment Number: 301

```
SELECT customer.Name AS CustomerName,
       customer.PhoneNumber,
       customer.EmailAddress,
       restaurant.Name AS RestaurantName,
       restaurant.UID,
       address.Street,
       address.HouseNumber,
       address.ApartmenNumber,
       address.PostalCode,
       postalcode.Province,
       postalcode.City,
       foodorder.OID,
       foodorder.OrderDate,
       foodorder.FulfilledDate,
       orderduration.OrderDuration
  FROM foodorder
 INNER JOIN orderduration ON foodorder.OrderDate=orderduration.OrderDate AND
 foodOrder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN restaurant ON restaurant.UID=foodorder.RestaurantUID
 INNER JOIN customer ON customer.UID=foodorder.CustomerUID
 INNER JOIN address ON customer.AID=address.AID
 INNER JOIN postalcode ON address.PostalCode=postalcode.PostalCode
 WHERE foodorder.OID=? AND foodorder.RestaurantUID=?
```

Output:

**Order ID: 1**

**Restaurant: Japanese Ramen and Sushi Bar**

Order Date: 2020-03-25 08:00:00

Fulfilled Date: 2020-03-25 08:20:00

Order Duration: 00:20:00

**Customer Information**

Name: Sally

Phone Number: 604-644-7180

Email Address: sally@mail.com

Address: 1005-6335 Thunderbird Crescent, Vancouver, BC, V6T2G9

Apartment Number: 301

```
SELECT foodorderitems.UID, foodorderitems.Amount, menuitem.Name, menuitem.Price,
menuitem.FoodType
FROM foodorderitems
INNER JOIN menuitem ON foodorderitems.UID=menuitem.UID AND
foodorderitems.IID=menuitem.IID
WHERE OID=?
```

Output:

#### Ordered Items

#	Item	Food Type	Quantity	Cost
1	Tuna Nigiri	Food	3	2.00
2	Tuna Sashimi	Food	2	3.00
3	Salmon Nigiri	Food	3	2.00
Total Cost: 18.00				

```
UPDATE orderduration SET FulfilledDate=?, OrderDuration=? WHERE OrderDate=?
```

```
SELECT * FROM inventoryitem WHERE UID=? AND IID=?
```

Output:

## Ribeye Steak 0.5lb

Amount: 1

\$16.99

```
SELECT Name FROM ingredient WHERE INID=?
```

Output:

**Contains:**

- Ribeye Steak

```
SELECT restaurant.Name AS RestaurantName,
       restaurant.PhoneNumber,
       restaurant.EmailAddress,
       wholesaler.Name AS WholesalerName,
       wholesaler.UID,
       address.Street,
       address.HouseNumber,
       address.ApartmenNumber,
       address.PostalCode,
       postalcode.Province,
       postalcode.City,
       inventoryorder.OID,
       inventoryorder.OrderDate,
       inventoryorder.FulfilledDate,
       orderduration.OrderDuration
  FROM inventoryorder
 INNER JOIN orderduration ON inventoryorder.OrderDate=orderduration.OrderDate AND
 inventoryorder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN restaurant ON restaurant.UID=inventoryorder.RestaurantUID
 INNER JOIN wholesaler ON wholesaler.UID=inventoryorder.WholesalerUID
 INNER JOIN address ON restaurant.AID=address.AID
 INNER JOIN postalcode ON address.PostalCode=postalcode.PostalCode
 WHERE inventoryorder.OID=? AND inventoryorder.WholesalerUID
```

Output:

**Order ID: 1**

**Wholesaler: Richmond Seafood Wholesale**

Order Date: 2020-03-25 15:30:00

Fulfilled Date: 2020-03-25 19:30:00

Order Duration: 04:00:00

**Restaurant Information**

Name: Japanese Ramen and Sushi Bar

Phone Number: 604-741-8500

Email Address: japanese@mail.com

Address: 257-3763 W 10th Ave, Vancouver, BC, V6R2G7

Apartment Number: -1

```
SELECT restaurant.Name AS RestaurantName,
       restaurant.PhoneNumber,
       restaurant.EmailAddress,
       wholesaler.Name AS WholesalerName,
       wholesaler.UID,
       address.Street,
       address.HouseNumber,
       address.ApartmenNumber,
       address.PostalCode,
       postalcode.Province,
       postalcode.City,
       inventoryorder.OID,
       inventoryorder.OrderDate,
       inventoryorder.FulfilledDate,
       orderduration.OrderDuration
  FROM inventoryorder
 INNER JOIN orderduration ON inventoryorder.OrderDate=orderduration.OrderDate AND
 inventoryorder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN restaurant ON restaurant.UID=inventoryorder.RestaurantUID
 INNER JOIN wholesaler ON wholesaler.UID=inventoryorder.WholesalerUID
 INNER JOIN address ON restaurant.AID=address.AID
 INNER JOIN postalcode ON address.PostalCode=postalcode.PostalCode
 WHERE inventoryorder.OID=? AND inventoryorder.RestaurantUID
```

Output:

**Order ID: 1**

**Wholesaler: Richmond Seafood Wholesale**

Order Date: 2020-03-25 15:30:00

Fulfilled Date: 2020-03-25 19:30:00

Order Duration: 04:00:00

**Restaurant Information**

Name: Japanese Ramen and Sushi Bar

Phone Number: 604-741-8500

Email Address: japanese@mail.com

Address: 257-3763 W 10th Ave, Vancouver, BC, V6R2G7

Apartment Number: -1

```
SELECT inventoryorderitems.Amount, inventoryitem.Name, inventoryitem.Cost
FROM inventoryorderitems
INNER JOIN inventoryitem ON inventoryorderitems.UID=inventoryitem.UID AND
inventoryorderitems.IID=inventoryitem.IID
WHERE OID=?
```

Output:

### Ordered Items

#	Item	Quantity	Cost
1	Whole Tuna	8	18.99
2	Whole Salmon	8	16.99
3	Atlantic Crab	4	10.99
4	Pacific Shrimp	2	12.99

Total Cost: 357.78

```
SELECT * FROM customer WHERE EmailAddress=?
```

Output:

### Sally's Profile

Customer

### Account Information

Edit

Your personal information is below.

Name: Sally

Email Address: sally@mail.com

Account Type: Customer

Password: \*\*\*\*\*

Phone Number: 604-644-7180

Street: 6335 Thunderbird Crescent

House Number: 1005

Apartment Number: 301

Postal Code: V6T2G9

Province: BC

City: Vancouver

```
SELECT * FROM restaurant WHERE EmailAddress=?
```

Output:

The image consists of two vertically stacked screenshots of a web-based application interface.

The top screenshot displays a profile page titled "Japanese Ramen and Sushi Bar's Profile". Below the title, it says "Restaurant".

The bottom screenshot displays an "Account Information" page. It includes a heading, a sub-heading "Your personal information is below.", and a table-like structure of data pairs:

Name: Japanese Ramen and Sushi Bar	Email Address: japanese@mail.com
Account Type: Restaurant	Password: *****
Phone Number: 604-741-8500	Street: 3763 W 10th Ave
House Number: 257	Apartment Number: -1
Postal Code: V6R2G7	Province: BC
City: Vancouver	

```
SELECT * FROM wholesaler WHERE EmailAddress=?
```

Output:

The screenshot shows a web application interface. At the top, a light blue header bar displays the title "Food Mart's Profile". Below this, a darker blue section is labeled "Wholesaler". The main content area has a light blue background and features a heading "Account Information" on the left. To the right of the heading is a green "Edit" button. Below the heading, a message says "Your personal information is below." The information is presented in two columns:

Name: Food Mart	Email Address: food@mail.com
Account Type: Wholesaler	Password: *****
Phone Number: 905-431-2202	Street: 202 Davenport Rd
House Number: 112	Apartment Number: -1
Postal Code: M5R1J2	Province: ON
City: Toronto	

```
SELECT * FROM address WHERE AID=?
```

Output: See above

```
SELECT * FROM postalcode WHERE PostalCode=?
```

Output: See above

```
INSERT INTO orderduration (OrderDate) VALUES (?)
```

```
INSERT INTO foodorder (CustomerUID, RestaurantUID, OrderDate, FulfilledDate) VALUES  
(?, ?, ?, ?)
```

```
INSERT INTO inventoryorder (RestaurantUID, WholesalerUID, OrderDate, FulfilledDate) VALUES  
(?, ?, ?, ?)
```

```
SELECT OID FROM foodorder WHERE CustomerUID=? AND RestaurantUID=? AND OrderDate=? AND  
FulfilledDate=?
```

Output: No output, used to retrieve OID for INSERT operation.

```
SELECT OID FROM inventoryorder WHERE RestaurantUID=? AND WholesalerUID=? AND OrderDate=? AND  
FulfilledDate=?
```

Output: No output, used to retrieve OID for INSERT operation.

```
INSERT INTO foodorderitems (OID, UID, IID, Amount) VALUES (?, ?, ?, ?)
```

```
INSERT INTO inventoryorderitems (OID, UID, IID, Amount) VALUES (?, ?, ?, ?)
```

```
SELECT * FROM menuitem WHERE UID=? AND IID=?
```

Output:

## Roast Vegetables

Food

\$2.99

```
SELECT * FROM menuitemingredient WHERE UID=? AND IID=?
```

Output: No output, used to retrieve ingredients from Ingredient.

```
SELECT Name FROM ingredient WHERE INID=?
```

Output:

### Contains:

- Carrot
- Green Pepper
- Red Pepper
- Asparagus

```
SELECT Name FROM restaurant WHERE UID=?
```

Output:

Restaurant Name  
Japanese Ramen and Sushi Bar

```
SELECT * FROM menuitem WHERE UID=?
```

Output:

Item 1:

Select an item

Select an item

Tuna Nigiri - \$2.00 - Food
Tuna Sashimi - \$3.00 - Food
Salmon Nigiri - \$2.00 - Food
Salmon Sashimi - \$3.00 - Food
Ebi Nigiri - \$2.50 - Food
Tamago Nigiri - \$1.75 - Food
Inari Nigiri - \$1.75 - Food
California Roll - \$3.75 - Food
Dragon Roll - \$8.50 - Food
Miso Ramen - \$9.50 - Food
Tonkatsu Ramen - \$9.50 - Food
Shoyu Ramen - \$9.50 - Food
Miso Soup - \$1.50 - Food
Green Tea - \$1.50 - Drink
Matcha Ice Cream - \$2.00 - Dessert
Mochi - \$2.00 - Dessert
Vanilla Ice Cream - \$2.00 - Dessert

```
SELECT Name FROM wholesaler WHERE UID=?
```

Output:

Wholesaler Name

Richmond Seafood Wholesale

```
SELECT * FROM inventoryitem WHERE UID=?
```

Output:

Item 1:

Select an item

Select an item

Whole Tuna - \$18.99 - Amount: 1  
Whole Salmon - \$16.99 - Amount: 1  
Atlantic Crab - \$10.99 - Amount: 1  
Atlantic Lobster - \$14.99 - Amount: 1  
Pacific Shrimp - \$12.99 - Amount: 20  
Whole Squid - \$6.99 - Amount: 1  
Whole Octopus - \$17.99 - Amount: 1  
Whole Halibut - \$11.99 - Amount: 1  
Whole Sea Bass - \$12.99 - Amount: 1  
Salmon Roe 0.5lb - \$5.99 - Amount: 1

```
SELECT count(foodorder.OID) FROM foodorder WHERE CustomerUID=?
```

Output:

Total Orders: 1

```
SELECT restaurant.Name,
       foodorder.OID,
       foodorder.OrderDate,
       foodOrder.FulfilledDate,
       orderduration.OrderDuration
  FROM foodorder
 INNER JOIN orderduration ON foodorder.OrderDate=orderduration.OrderDate AND
       foodOrder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN restaurant ON restaurant.UID=foodorder.RestaurantUID
 WHERE foodorder.CustomerUID=?
```

Output:

#	ID	Restaurant	Order Date	Fulfilled Date	Order Duration	
1	1	Japanese Ramen and Sushi Bar	2020-03-25 08:00:00	2020-03-25 08:20:00	00:20:00	<a href="#">View Order</a>
2	2	Italy Pasta Cafe	2020-03-26 10:00:00	2020-03-26 10:10:00	00:10:00	<a href="#">View Order</a>
3	4	Italy Pasta Cafe	2020-03-27 06:00:00	In Progress	-	<a href="#">View Order</a>
4	3	Burger Shack	2020-03-27 02:00:00	In Progress	-	<a href="#">View Order</a>
5	5	Burger Shack	2020-03-27 07:00:00	In Progress	-	<a href="#">View Order</a>

```
SELECT restaurant.Name,
       foodorder.OID,
       foodorder.OrderDate,
       foodOrder.FulfilledDate,
       orderduration.OrderDuration
  FROM foodorder
 INNER JOIN orderduration ON foodorder.OrderDate=orderduration.OrderDate AND
       foodOrder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN restaurant ON restaurant.UID=foodorder.RestaurantUID
 WHERE foodorder.CustomerUID=?
 ORDER BY OID DESC
```

Output:

#	ID ^	Restaurant	Order Date	Fulfilled Date	Order Duration	
1	5	Burger Shack	2020-03-27 07:00:00	In Progress	-	<button>View Order</button>
2	4	Italy Pasta Cafe	2020-03-27 06:00:00	In Progress	-	<button>View Order</button>
3	3	Burger Shack	2020-03-27 02:00:00	In Progress	-	<button>View Order</button>
4	2	Italy Pasta Cafe	2020-03-26 10:00:00	2020-03-26 10:10:00	00:10:00	<button>View Order</button>
5	1	Japanese Ramen and Sushi Bar	2020-03-25 08:00:00	2020-03-25 08:20:00	00:20:00	<button>View Order</button>

```
SELECT restaurant.Name,
       foodorder.OID,
       foodorder.OrderDate,
       foodOrder.FulfilledDate,
       orderduration.OrderDuration
  FROM foodorder
 INNER JOIN orderduration ON foodorder.OrderDate=orderduration.OrderDate AND
       foodOrder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN restaurant ON restaurant.UID=foodorder.RestaurantUID
 WHERE foodorder.CustomerUID=?
 ORDER BY OID ASC
```

Output:

#	ID	Restaurant	Order Date	Fulfilled Date	Order Duration	
1	1	Japanese Ramen and Sushi Bar	2020-03-25 08:00:00	2020-03-25 08:20:00	00:20:00	<a href="#">View Order</a>
2	2	Italy Pasta Cafe	2020-03-26 10:00:00	2020-03-26 10:10:00	00:10:00	<a href="#">View Order</a>
3	3	Burger Shack	2020-03-27 02:00:00	In Progress	-	<a href="#">View Order</a>
4	4	Italy Pasta Cafe	2020-03-27 06:00:00	In Progress	-	<a href="#">View Order</a>
5	5	Burger Shack	2020-03-27 07:00:00	In Progress	-	<a href="#">View Order</a>

```
SELECT restaurant.Name,
       foodorder.OID,
       foodorder.OrderDate,
       foodOrder.FulfilledDate,
       orderduration.OrderDuration
  FROM foodorder
 INNER JOIN orderduration ON foodorder.OrderDate=orderduration.OrderDate AND
       foodOrder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN restaurant ON restaurant.UID=foodorder.RestaurantUID
 WHERE foodorder.CustomerUID=?
 ORDER BY Name DESC
```

Output:

#	ID	Restaurant ^	Order Date	Fulfilled Date	Order Duration	
1	1	Japanese Ramen and Sushi Bar	2020-03-25 08:00:00	2020-03-25 08:20:00	00:20:00	<a href="#">View Order</a>
2	2	Italy Pasta Cafe	2020-03-26 10:00:00	2020-03-26 10:10:00	00:10:00	<a href="#">View Order</a>
3	4	Italy Pasta Cafe	2020-03-27 06:00:00	In Progress	-	<a href="#">View Order</a>
4	3	Burger Shack	2020-03-27 02:00:00	In Progress	-	<a href="#">View Order</a>
5	5	Burger Shack	2020-03-27 07:00:00	In Progress	-	<a href="#">View Order</a>

```
SELECT restaurant.Name,
       foodorder.OID,
       foodorder.OrderDate,
       foodOrder.FulfilledDate,
       orderduration.OrderDuration
  FROM foodorder
 INNER JOIN orderduration ON foodorder.OrderDate=orderduration.OrderDate AND
       foodOrder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN restaurant ON restaurant.UID=foodorder.RestaurantUID
 WHERE foodorder.CustomerUID=?
 ORDER BY Name ASC
```

Output:

#	ID	Restaurant ▾	Order Date	Fulfilled Date	Order Duration	
1	3	Burger Shack	2020-03-27 02:00:00	In Progress	-	<button>View Order</button>
2	5	Burger Shack	2020-03-27 07:00:00	In Progress	-	<button>View Order</button>
3	2	Italy Pasta Cafe	2020-03-26 10:00:00	2020-03-26 10:10:00	00:10:00	<button>View Order</button>
4	4	Italy Pasta Cafe	2020-03-27 06:00:00	In Progress	-	<button>View Order</button>
5	1	Japanese Ramen and Sushi Bar	2020-03-25 08:00:00	2020-03-25 08:20:00	00:20:00	<button>View Order</button>

```
SELECT restaurant.Name,
       foodorder.OID,
       foodorder.OrderDate,
       foodOrder.FulfilledDate,
       orderduration.OrderDuration
  FROM foodorder
 INNER JOIN orderduration ON foodorder.OrderDate=orderduration.OrderDate AND
       foodOrder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN restaurant ON restaurant.UID=foodorder.RestaurantUID
 WHERE foodorder.CustomerUID=?
 ORDER BY OrderDate DESC
```

Output:

#	ID	Restaurant	Order Date ▲	Fulfilled Date	Order Duration	
1	5	Burger Shack	2020-03-27 07:00:00	In Progress	-	<button>View Order</button>
2	4	Italy Pasta Cafe	2020-03-27 06:00:00	In Progress	-	<button>View Order</button>
3	3	Burger Shack	2020-03-27 02:00:00	In Progress	-	<button>View Order</button>
4	2	Italy Pasta Cafe	2020-03-26 10:00:00	2020-03-26 10:10:00	00:10:00	<button>View Order</button>
5	1	Japanese Ramen and Sushi Bar	2020-03-25 08:00:00	2020-03-25 08:20:00	00:20:00	<button>View Order</button>

```
SELECT restaurant.Name,
       foodorder.OID,
       foodorder.OrderDate,
       foodOrder.FulfilledDate,
       orderduration.OrderDuration
  FROM foodorder
 INNER JOIN orderduration ON foodorder.OrderDate=orderduration.OrderDate AND
       foodOrder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN restaurant ON restaurant.UID=foodorder.RestaurantUID
 WHERE foodorder.CustomerUID=?
 ORDER BY OrderDate ASC
```

Output:

#	ID	Restaurant	Order Date ▾	Fulfilled Date	Order Duration	
1	1	Japanese Ramen and Sushi Bar	2020-03-25 08:00:00	2020-03-25 08:20:00	00:20:00	<button>View Order</button>
2	2	Italy Pasta Cafe	2020-03-26 10:00:00	2020-03-26 10:10:00	00:10:00	<button>View Order</button>
3	3	Burger Shack	2020-03-27 02:00:00	In Progress	-	<button>View Order</button>
4	4	Italy Pasta Cafe	2020-03-27 06:00:00	In Progress	-	<button>View Order</button>
5	5	Burger Shack	2020-03-27 07:00:00	In Progress	-	<button>View Order</button>

```
SELECT restaurant.Name,
       foodorder.OID,
       foodorder.OrderDate,
       foodOrder.FulfilledDate,
       orderduration.OrderDuration
  FROM foodorder
 INNER JOIN orderduration ON foodorder.OrderDate=orderduration.OrderDate AND
       foodOrder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN restaurant ON restaurant.UID=foodorder.RestaurantUID
 WHERE foodorder.CustomerUID=?
 ORDER BY FulfilledDate DESC
```

Output:

#	ID	Restaurant	Order Date	Fulfilled Date ^	Order Duration	
1	2	Italy Pasta Cafe	2020-03-26 10:00:00	2020-03-26 10:10:00	00:10:00	<a href="#">View Order</a>
2	1	Japanese Ramen and Sushi Bar	2020-03-25 08:00:00	2020-03-25 08:20:00	00:20:00	<a href="#">View Order</a>
3	3	Burger Shack	2020-03-27 02:00:00	In Progress	-	<a href="#">View Order</a>
4	5	Burger Shack	2020-03-27 07:00:00	In Progress	-	<a href="#">View Order</a>
5	4	Italy Pasta Cafe	2020-03-27 06:00:00	In Progress	-	<a href="#">View Order</a>

```
SELECT restaurant.Name,
       foodorder.OID,
       foodorder.OrderDate,
       foodOrder.FulfilledDate,
       orderduration.OrderDuration
  FROM foodorder
 INNER JOIN orderduration ON foodorder.OrderDate=orderduration.OrderDate AND
       foodOrder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN restaurant ON restaurant.UID=foodorder.RestaurantUID
 WHERE foodorder.CustomerUID=?
 ORDER BY FulfilledDate ASC
```

Output:

#	ID	Restaurant	Order Date	Fulfilled Date ▾	Order Duration	
1	3	Burger Shack	2020-03-27 02:00:00	In Progress	-	<button>View Order</button>
2	5	Burger Shack	2020-03-27 07:00:00	In Progress	-	<button>View Order</button>
3	4	Italy Pasta Cafe	2020-03-27 06:00:00	In Progress	-	<button>View Order</button>
4	1	Japanese Ramen and Sushi Bar	2020-03-25 08:00:00	2020-03-25 08:20:00	00:20:00	<button>View Order</button>
5	2	Italy Pasta Cafe	2020-03-26 10:00:00	2020-03-26 10:10:00	00:10:00	<button>View Order</button>

```
SELECT restaurant.Name,
       foodorder.OID,
       foodorder.OrderDate,
       foodOrder.FulfilledDate,
       orderduration.OrderDuration
  FROM foodorder
 INNER JOIN orderduration ON foodorder.OrderDate=orderduration.OrderDate AND
       foodOrder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN restaurant ON restaurant.UID=foodorder.RestaurantUID
 WHERE foodorder.CustomerUID=?
 ORDER BY OrderDuration DESC
```

Output:

#	ID	Restaurant	Order Date	Fulfilled Date	Order Duration	▲
1	1	Japanese Ramen and Sushi Bar	2020-03-25 08:00:00	2020-03-25 08:20:00	00:20:00	<button>View Order</button>
2	2	Italy Pasta Cafe	2020-03-26 10:00:00	2020-03-26 10:10:00	00:10:00	<button>View Order</button>
3	3	Burger Shack	2020-03-27 02:00:00	In Progress	-	<button>View Order</button>
4	5	Burger Shack	2020-03-27 07:00:00	In Progress	-	<button>View Order</button>
5	4	Italy Pasta Cafe	2020-03-27 06:00:00	In Progress	-	<button>View Order</button>

```
SELECT restaurant.Name,
       foodorder.OID,
       foodorder.OrderDate,
       foodOrder.FulfilledDate,
       orderduration.OrderDuration
  FROM foodorder
 INNER JOIN orderduration ON foodorder.OrderDate=orderduration.OrderDate AND
       foodOrder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN restaurant ON restaurant.UID=foodorder.RestaurantUID
 WHERE foodorder.CustomerUID=?
 ORDER BY OrderDuration ASC
```

Output:

#	ID	Restaurant	Order Date	Fulfilled Date	Order Duration	
1	3	Burger Shack	2020-03-27 02:00:00	In Progress	-	<button>View Order</button>
2	5	Burger Shack	2020-03-27 07:00:00	In Progress	-	<button>View Order</button>
3	4	Italy Pasta Cafe	2020-03-27 06:00:00	In Progress	-	<button>View Order</button>
4	2	Italy Pasta Cafe	2020-03-26 10:00:00	2020-03-26 10:10:00	00:10:00	<button>View Order</button>
5	1	Japanese Ramen and Sushi Bar	2020-03-25 08:00:00	2020-03-25 08:20:00	00:20:00	<button>View Order</button>

```
SELECT wholesaler.Name,
       inventoryorder.OID,
       inventoryorder.OrderDate,
       inventoryorder.FulfilledDate,
       orderduration.OrderDuration
  FROM inventoryorder
 INNER JOIN orderduration ON inventoryorder.OrderDate=orderduration.OrderDate AND
 inventoryorder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN wholesaler ON wholesaler.UID=inventoryorder.WholesalerUID
 WHERE inventoryorder.RestaurantUID=?
```

Output:

#	ID	Wholesaler	Order Date	Fulfilled Date	Order Duration	
1	4	Canadian Butcher Store	2020-03-26 05:30:00	2020-03-26 09:10:00	03:40:00	<a href="#">View Order</a>
2	5	Canadian Butcher Store	2020-03-27 09:00:00	In Progress	-	<a href="#">View Order</a>

```
SELECT wholesaler.Name,
       inventoryorder.OID,
       inventoryorder.OrderDate,
       inventoryorder.FulfilledDate,
       orderduration.OrderDuration
  FROM inventoryorder
 INNER JOIN orderduration ON inventoryorder.OrderDate=orderduration.OrderDate AND
 inventoryorder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN wholesaler ON wholesaler.UID=inventoryorder.WholesalerUID
 WHERE inventoryorder.RestaurantUID=?
 ORDER BY OID DESC
```

Output:

#	ID ^	Wholesaler	Order Date	Fulfilled Date	Order Duration	
1	5	Canadian Butcher Store	2020-03-27 09:00:00	In Progress	-	<a href="#">View Order</a>
2	4	Canadian Butcher Store	2020-03-26 05:30:00	2020-03-26 09:10:00	03:40:00	<a href="#">View Order</a>

```
SELECT wholesaler.Name,
       inventoryorder.OID,
       inventoryorder.OrderDate,
       inventoryorder.FulfilledDate,
       orderduration.OrderDuration
  FROM inventoryorder
 INNER JOIN orderduration ON inventoryorder.OrderDate=orderduration.OrderDate AND
 inventoryorder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN wholesaler ON wholesaler.UID=inventoryorder.WholesalerUID
 WHERE inventoryorder.RestaurantUID=?
 ORDER BY OID ASC
```

Output:

#	ID ▾	Wholesaler	Order Date	Fulfilled Date	Order Duration	
1	4	Canadian Butcher Store	2020-03-26 05:30:00	2020-03-26 09:10:00	03:40:00	<a href="#">View Order</a>
2	5	Canadian Butcher Store	2020-03-27 09:00:00	In Progress	-	<a href="#">View Order</a>

```
SELECT wholesaler.Name,
       inventoryorder.OID,
       inventoryorder.OrderDate,
       inventoryorder.FulfilledDate,
       orderduration.OrderDuration
  FROM inventoryorder
 INNER JOIN orderduration ON inventoryorder.OrderDate=orderduration.OrderDate AND
 inventoryorder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN wholesaler ON wholesaler.UID=inventoryorder.WholesalerUID
 WHERE inventoryorder.RestaurantUID=?
 ORDER BY Name DESC
```

Output:

#	ID	Wholesaler ^	Order Date	Fulfilled Date	Order Duration	
1	4	Canadian Butcher Store	2020-03-26 05:30:00	2020-03-26 09:10:00	03:40:00	<a href="#">View Order</a>
2	5	Canadian Butcher Store	2020-03-27 09:00:00	In Progress	-	<a href="#">View Order</a>

```
SELECT wholesaler.Name,
       inventoryorder.OID,
       inventoryorder.OrderDate,
       inventoryorder.FulfilledDate,
       orderduration.OrderDuration
  FROM inventoryorder
 INNER JOIN orderduration ON inventoryorder.OrderDate=orderduration.OrderDate AND
 inventoryorder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN wholesaler ON wholesaler.UID=inventoryorder.WholesalerUID
 WHERE inventoryorder.RestaurantUID=?
 ORDER BY Name ASC
```

Output:

#	ID	Wholesaler ▾	Order Date	Fulfilled Date	Order Duration	
1	4	Canadian Butcher Store	2020-03-26 05:30:00	2020-03-26 09:10:00	03:40:00	<a href="#">View Order</a>
2	5	Canadian Butcher Store	2020-03-27 09:00:00	In Progress	-	<a href="#">View Order</a>

```
SELECT wholesaler.Name,
       inventoryorder.OID,
       inventoryorder.OrderDate,
       inventoryorder.FulfilledDate,
       orderduration.OrderDuration
  FROM inventoryorder
 INNER JOIN orderduration ON inventoryorder.OrderDate=orderduration.OrderDate AND
 inventoryorder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN wholesaler ON wholesaler.UID=inventoryorder.WholesalerUID
 WHERE inventoryorder.RestaurantUID=?
 ORDER BY OrderDate DESC
```

Output:

#	ID	Wholesaler	Order Date ▲	Fulfilled Date	Order Duration	
1	5	Canadian Butcher Store	2020-03-27 09:00:00	In Progress	-	<button>View Order</button>
2	4	Canadian Butcher Store	2020-03-26 05:30:00	2020-03-26 09:10:00	03:40:00	<button>View Order</button>

```
SELECT wholesaler.Name,
       inventoryorder.OID,
       inventoryorder.OrderDate,
       inventoryorder.FulfilledDate,
       orderduration.OrderDuration
  FROM inventoryorder
 INNER JOIN orderduration ON inventoryorder.OrderDate=orderduration.OrderDate AND
 inventoryorder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN wholesaler ON wholesaler.UID=inventoryorder.WholesalerUID
 WHERE inventoryorder.RestaurantUID=?
 ORDER BY OrderDate ASC
```

Output:

#	ID	Wholesaler	Order Date ▾	Fulfilled Date	Order Duration	
1	4	Canadian Butcher Store	2020-03-26 05:30:00	2020-03-26 09:10:00	03:40:00	<a href="#">View Order</a>
2	5	Canadian Butcher Store	2020-03-27 09:00:00	In Progress	-	<a href="#">View Order</a>

```
SELECT wholesaler.Name,
       inventoryorder.OID,
       inventoryorder.OrderDate,
       inventoryorder.FulfilledDate,
       orderduration.OrderDuration
  FROM inventoryorder
 INNER JOIN orderduration ON inventoryorder.OrderDate=orderduration.OrderDate AND
 inventoryorder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN wholesaler ON wholesaler.UID=inventoryorder.WholesalerUID
 WHERE inventoryorder.RestaurantUID=?
 ORDER BY FulfilledDate DESC
```

Output:

#	ID	Wholesaler	Order Date	Fulfilled Date ▲	Order Duration	
1	4	Canadian Butcher Store	2020-03-26 05:30:00	2020-03-26 09:10:00	03:40:00	<a href="#">View Order</a>
2	5	Canadian Butcher Store	2020-03-27 09:00:00	In Progress	-	<a href="#">View Order</a>

```
SELECT wholesaler.Name,
       inventoryorder.OID,
       inventoryorder.OrderDate,
       inventoryorder.FulfilledDate,
       orderduration.OrderDuration
  FROM inventoryorder
 INNER JOIN orderduration ON inventoryorder.OrderDate=orderduration.OrderDate AND
 inventoryorder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN wholesaler ON wholesaler.UID=inventoryorder.WholesalerUID
 WHERE inventoryorder.RestaurantUID=?
 ORDER BY FulfilledDate ASC
```

Output:

#	ID	Wholesaler	Order Date	Fulfilled Date ▾	Order Duration	
1	5	Canadian Butcher Store	2020-03-27 09:00:00	In Progress	-	<button>View Order</button>
2	4	Canadian Butcher Store	2020-03-26 05:30:00	2020-03-26 09:10:00	03:40:00	<button>View Order</button>

```
SELECT wholesaler.Name,
       inventoryorder.OID,
       inventoryorder.OrderDate,
       inventoryorder.FulfilledDate,
       orderduration.OrderDuration
  FROM inventoryorder
 INNER JOIN orderduration ON inventoryorder.OrderDate=orderduration.OrderDate AND
 inventoryorder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN wholesaler ON wholesaler.UID=inventoryorder.WholesalerUID
 WHERE inventoryorder.RestaurantUID=?
 ORDER BY OrderDuration DESC
```

Output:

#	ID	Wholesaler	Order Date	Fulfilled Date	Order Duration ^	
1	4	Canadian Butcher Store	2020-03-26 05:30:00	2020-03-26 09:10:00	03:40:00	<a href="#">View Order</a>
2	5	Canadian Butcher Store	2020-03-27 09:00:00	In Progress	-	<a href="#">View Order</a>

```
SELECT wholesaler.Name,
       inventoryorder.OID,
       inventoryorder.OrderDate,
       inventoryorder.FulfilledDate,
       orderduration.OrderDuration
  FROM inventoryorder
 INNER JOIN orderduration ON inventoryorder.OrderDate=orderduration.OrderDate AND
 inventoryorder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN wholesaler ON wholesaler.UID=inventoryorder.WholesalerUID
 WHERE inventoryorder.RestaurantUID=?
 ORDER BY OrderDuration ASC
```

Output:

#	ID	Wholesaler	Order Date	Fulfilled Date	Order Duration ▾	
1	5	Canadian Butcher Store	2020-03-27 09:00:00	In Progress	-	<a href="#">View Order</a>
2	4	Canadian Butcher Store	2020-03-26 05:30:00	2020-03-26 09:10:00	03:40:00	<a href="#">View Order</a>

```
SELECT UID FROM customer WHERE EmailAddress=?
```

Output: No output, used for email validation.

```
SELECT UID FROM restaurant WHERE EmailAddress=?
```

Output: No output, used for email validation.

```
SELECT UID FROM wholesaler WHERE EmailAddress=?
```

Output: No output, used for email validation.

```
INSERT INTO postalcode (PostalCode, Province, City) VALUES (?, ?, ?)
```

```
UPDATE address SET Street=?, HouseNumber=?, ApartmentNumber=?, PostalCode=? WHERE AID=?
```

```
SELECT count(foodorder.OID) FROM foodorder WHERE RestaurantUID=?
```

Output:

```
Total Received Orders: 2
```

```
SELECT customer.Name,
       foodorder.OID,
       foodorder.OrderDate,
       foodOrder.FulfilledDate,
       orderduration.OrderDuration
  FROM foodorder
 INNER JOIN orderduration ON foodorder.OrderDate=orderduration.OrderDate AND
       foodOrder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN customer ON customer.UID=foodorder.CustomerUID
 WHERE foodorder.RestaurantUID=?
```

Output:

#	ID	Customer	Order Date	Fulfilled Date	Order Duration	
1	8	John	2020-03-27 03:00:00	In Progress	-	<a href="#">View Order</a>
2	11	Frank	2020-03-27 05:00:00	In Progress	-	<a href="#">View Order</a>

```
SELECT customer.Name,
       foodorder.OID,
       foodorder.OrderDate,
       foodOrder.FulfilledDate,
       orderduration.OrderDuration
  FROM foodorder
 INNER JOIN orderduration ON foodorder.OrderDate=orderduration.OrderDate AND
       foodOrder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN customer ON customer.UID=foodorder.CustomerUID
 WHERE foodorder.RestaurantUID=?
 ORDER BY OID DESC
```

Output:

#	ID ▾	Customer	Order Date	Fulfilled Date	Order Duration	
1	11	Frank	2020-03-27 05:00:00	In Progress	-	<button>View Order</button>
2	8	John	2020-03-27 03:00:00	In Progress	-	<button>View Order</button>

```
SELECT customer.Name,
       foodorder.OID,
       foodorder.OrderDate,
       foodOrder.FulfilledDate,
       orderduration.OrderDuration
  FROM foodorder
 INNER JOIN orderduration ON foodorder.OrderDate=orderduration.OrderDate AND
       foodOrder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN customer ON customer.UID=foodorder.CustomerUID
 WHERE foodorder.RestaurantUID=?
 ORDER BY OID ASC
```

Output:

#	ID ▾	Customer	Order Date	Fulfilled Date	Order Duration	
1	8	John	2020-03-27 03:00:00	In Progress	-	<a href="#">View Order</a>
2	11	Frank	2020-03-27 05:00:00	In Progress	-	<a href="#">View Order</a>

```
SELECT customer.Name,
       foodorder.OID,
       foodorder.OrderDate,
       foodOrder.FulfilledDate,
       orderduration.OrderDuration
  FROM foodorder
 INNER JOIN orderduration ON foodorder.OrderDate=orderduration.OrderDate AND
       foodOrder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN customer ON customer.UID=foodorder.CustomerUID
 WHERE foodorder.RestaurantUID=?
 ORDER BY Name DESC
```

Output:

#	ID	Customer ▾	Order Date	Fulfilled Date	Order Duration	
1	8	John	2020-03-27 03:00:00	In Progress	-	<button>View Order</button>
2	11	Frank	2020-03-27 05:00:00	In Progress	-	<button>View Order</button>

```
SELECT customer.Name,
       foodorder.OID,
       foodorder.OrderDate,
       foodOrder.FulfilledDate,
       orderduration.OrderDuration
  FROM foodorder
 INNER JOIN orderduration ON foodorder.OrderDate=orderduration.OrderDate AND
       foodOrder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN customer ON customer.UID=foodorder.CustomerUID
 WHERE foodorder.RestaurantUID=?
 ORDER BY Name ASC
```

Output:

#	ID	Customer ▾	Order Date	Fulfilled Date	Order Duration	
1	11	Frank	2020-03-27 05:00:00	In Progress	-	<button>View Order</button>
2	8	John	2020-03-27 03:00:00	In Progress	-	<button>View Order</button>

```
SELECT customer.Name,
       foodorder.OID,
       foodorder.OrderDate,
       foodOrder.FulfilledDate,
       orderduration.OrderDuration
  FROM foodorder
 INNER JOIN orderduration ON foodorder.OrderDate=orderduration.OrderDate AND
       foodOrder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN customer ON customer.UID=foodorder.CustomerUID
 WHERE foodorder.RestaurantUID=?
 ORDER BY OrderDate DESC
```

Output:

#	ID	Customer	Order Date ▾	Fulfilled Date	Order Duration	
1	11	Frank	2020-03-27 05:00:00	In Progress	-	<a href="#">View Order</a>
2	8	John	2020-03-27 03:00:00	In Progress	-	<a href="#">View Order</a>

```
SELECT customer.Name,
       foodorder.OID,
       foodorder.OrderDate,
       foodOrder.FulfilledDate,
       orderduration.OrderDuration
  FROM foodorder
 INNER JOIN orderduration ON foodorder.OrderDate=orderduration.OrderDate AND
       foodOrder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN customer ON customer.UID=foodorder.CustomerUID
 WHERE foodorder.RestaurantUID=?
 ORDER BY OrderDate ASC
```

Output:

#	ID	Customer	Order Date ▾	Fulfilled Date	Order Duration	
1	8	John	2020-03-27 03:00:00	In Progress	-	<a href="#">View Order</a>
2	11	Frank	2020-03-27 05:00:00	In Progress	-	<a href="#">View Order</a>

```
SELECT customer.Name,
       foodorder.OID,
       foodorder.OrderDate,
       foodOrder.FulfilledDate,
       orderduration.OrderDuration
  FROM foodorder
 INNER JOIN orderduration ON foodorder.OrderDate=orderduration.OrderDate AND
       foodOrder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN customer ON customer.UID=foodorder.CustomerUID
 WHERE foodorder.RestaurantUID=?
 ORDER BY FulfilledDate DESC
```

Output:

#	ID	Customer	Order Date	Fulfilled Date ^	Order Duration	
1	8	John	2020-03-27 03:00:00	In Progress	-	<a href="#">View Order</a>
2	11	Frank	2020-03-27 05:00:00	In Progress	-	<a href="#">View Order</a>

```
SELECT customer.Name,
       foodorder.OID,
       foodorder.OrderDate,
       foodOrder.FulfilledDate,
       orderduration.OrderDuration
  FROM foodorder
 INNER JOIN orderduration ON foodorder.OrderDate=orderduration.OrderDate AND
       foodOrder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN customer ON customer.UID=foodorder.CustomerUID
 WHERE foodorder.RestaurantUID=?
 ORDER BY FulfilledDate ASC
```

Output:

#	ID	Customer	Order Date	Fulfilled Date ▾	Order Duration	
1	8	John	2020-03-27 03:00:00	In Progress	-	<button>View Order</button>
2	11	Frank	2020-03-27 05:00:00	In Progress	-	<button>View Order</button>

```
SELECT customer.Name,
       foodorder.OID,
       foodorder.OrderDate,
       foodOrder.FulfilledDate,
       orderduration.OrderDuration
  FROM foodorder
 INNER JOIN orderduration ON foodorder.OrderDate=orderduration.OrderDate AND
       foodOrder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN customer ON customer.UID=foodorder.CustomerUID
 WHERE foodorder.RestaurantUID=?
 ORDER BY OrderDuration DESC
```

Output:

#	ID	Customer	Order Date	Fulfilled Date	Order Duration ▲	
1	8	John	2020-03-27 03:00:00	In Progress	-	<a href="#">View Order</a>
2	11	Frank	2020-03-27 05:00:00	In Progress	-	<a href="#">View Order</a>

```
SELECT customer.Name,
       foodorder.OID,
       foodorder.OrderDate,
       foodOrder.FulfilledDate,
       orderduration.OrderDuration
  FROM foodorder
 INNER JOIN orderduration ON foodorder.OrderDate=orderduration.OrderDate AND
       foodOrder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN customer ON customer.UID=foodorder.CustomerUID
 WHERE foodorder.RestaurantUID=?
 ORDER BY OrderDuration ASC
```

Output:

#	ID	Customer	Order Date	Fulfilled Date	Order Duration ▾	
1	8	John	2020-03-27 03:00:00	In Progress	-	<button>View Order</button>
2	11	Frank	2020-03-27 05:00:00	In Progress	-	<button>View Order</button>

```
SELECT count(inventoryorder.OID) FROM inventoryorder WHERE WholesalerUID=?
```

Output:

Total Orders: 1

```
SELECT count(inventoryorder.OID) FROM inventoryorder WHERE RestaurantUID=?
```

Output:

Total Received Orders: 2

```
SELECT restaurant.Name,
       inventoryorder.OID,
       inventoryorder.OrderDate,
       inventoryorder.FulfilledDate,
       orderduration.OrderDuration
  FROM inventoryorder
 INNER JOIN orderduration ON inventoryorder.OrderDate=orderduration.OrderDate AND
 inventoryorder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN restaurant ON restaurant.UID=inventoryorder.RestaurantUID
 WHERE inventoryorder.WholesalerUID=?
```

Output:

#	ID	Customer	Order Date	Fulfilled Date	Order Duration	
1	2	Italy Pasta Cafe	2020-03-26 05:00:00	2020-03-26 08:40:00	03:40:00	<a href="#">View Order</a>
2	3	Burger Shack	2020-03-27 08:00:00	In Progress	-	<a href="#">View Order</a>

```
SELECT restaurant.Name,
       inventoryorder.OID,
       inventoryorder.OrderDate,
       inventoryorder.FulfilledDate,
       orderduration.OrderDuration
  FROM inventoryorder
 INNER JOIN orderduration ON inventoryorder.OrderDate=orderduration.OrderDate AND
 inventoryorder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN restaurant ON restaurant.UID=inventoryorder.RestaurantUID
 WHERE inventoryorder.WholesalerUID=?
 ORDER BY OID DESC
```

Output:

#	ID ^	Customer	Order Date	Fulfilled Date	Order Duration	
1	3	Burger Shack	2020-03-27 08:00:00	In Progress	-	<button>View Order</button>
2	2	Italy Pasta Cafe	2020-03-26 05:00:00	2020-03-26 08:40:00	03:40:00	<button>View Order</button>

```
SELECT restaurant.Name,
       inventoryorder.OID,
       inventoryorder.OrderDate,
       inventoryorder.FulfilledDate,
       orderduration.OrderDuration
  FROM inventoryorder
 INNER JOIN orderduration ON inventoryorder.OrderDate=orderduration.OrderDate AND
 inventoryorder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN restaurant ON restaurant.UID=inventoryorder.RestaurantUID
 WHERE inventoryorder.WholesalerUID=?
 ORDER BY OID ASC
```

Output:

#	ID ▾	Customer	Order Date	Fulfilled Date	Order Duration	
1	2	Italy Pasta Cafe	2020-03-26 05:00:00	2020-03-26 08:40:00	03:40:00	<button>View Order</button>
2	3	Burger Shack	2020-03-27 08:00:00	In Progress	-	<button>View Order</button>

```
SELECT restaurant.Name,
       inventoryorder.OID,
       inventoryorder.OrderDate,
       inventoryorder.FulfilledDate,
       orderduration.OrderDuration
  FROM inventoryorder
 INNER JOIN orderduration ON inventoryorder.OrderDate=orderduration.OrderDate AND
 inventoryorder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN restaurant ON restaurant.UID=inventoryorder.RestaurantUID
 WHERE inventoryorder.WholesalerUID=?
 ORDER BY Name DESC
```

Output:

#	ID	Customer ^	Order Date	Fulfilled Date	Order Duration	
1	2	Italy Pasta Cafe	2020-03-26 05:00:00	2020-03-26 08:40:00	03:40:00	<a href="#">View Order</a>
2	3	Burger Shack	2020-03-27 08:00:00	In Progress	-	<a href="#">View Order</a>

```
SELECT restaurant.Name,
       inventoryorder.OID,
       inventoryorder.OrderDate,
       inventoryorder.FulfilledDate,
       orderduration.OrderDuration
  FROM inventoryorder
 INNER JOIN orderduration ON inventoryorder.OrderDate=orderduration.OrderDate AND
 inventoryorder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN restaurant ON restaurant.UID=inventoryorder.RestaurantUID
 WHERE inventoryorder.WholesalerUID=?
 ORDER BY Name ASC
```

Output:

#	ID	Customer ▾	Order Date	Fulfilled Date	Order Duration	
1	3	Burger Shack	2020-03-27 08:00:00	In Progress	-	<button>View Order</button>
2	2	Italy Pasta Cafe	2020-03-26 05:00:00	2020-03-26 08:40:00	03:40:00	<button>View Order</button>

```
SELECT restaurant.Name,
       inventoryorder.OID,
       inventoryorder.OrderDate,
       inventoryorder.FulfilledDate,
       orderduration.OrderDuration
  FROM inventoryorder
 INNER JOIN orderduration ON inventoryorder.OrderDate=orderduration.OrderDate AND
 inventoryorder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN restaurant ON restaurant.UID=inventoryorder.RestaurantUID
 WHERE inventoryorder.WholesalerUID=?
 ORDER BY OrderDate DESC
```

Output:

#	ID	Customer	Order Date ▲	Fulfilled Date	Order Duration	
1	3	Burger Shack	2020-03-27 08:00:00	In Progress	-	<button>View Order</button>
2	2	Italy Pasta Cafe	2020-03-26 05:00:00	2020-03-26 08:40:00	03:40:00	<button>View Order</button>

```
SELECT restaurant.Name,
       inventoryorder.OID,
       inventoryorder.OrderDate,
       inventoryorder.FulfilledDate,
       orderduration.OrderDuration
  FROM inventoryorder
 INNER JOIN orderduration ON inventoryorder.OrderDate=orderduration.OrderDate AND
 inventoryorder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN restaurant ON restaurant.UID=inventoryorder.RestaurantUID
 WHERE inventoryorder.WholesalerUID=?
 ORDER BY OrderDate ASC
```

Output:

#	ID	Customer	Order Date ▾	Fulfilled Date	Order Duration	
1	2	Italy Pasta Cafe	2020-03-26 05:00:00	2020-03-26 08:40:00	03:40:00	<a href="#">View Order</a>
2	3	Burger Shack	2020-03-27 08:00:00	In Progress	-	<a href="#">View Order</a>

```
SELECT restaurant.Name,
       inventoryorder.OID,
       inventoryorder.OrderDate,
       inventoryorder.FulfilledDate,
       orderduration.OrderDuration
  FROM inventoryorder
 INNER JOIN orderduration ON inventoryorder.OrderDate=orderduration.OrderDate AND
 inventoryorder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN restaurant ON restaurant.UID=inventoryorder.RestaurantUID
 WHERE inventoryorder.WholesalerUID=?
 ORDER BY FulfilledDate DESC
```

Output:

#	ID	Customer	Order Date	Fulfilled Date ▲	Order Duration	
1	2	Italy Pasta Cafe	2020-03-26 05:00:00	2020-03-26 08:40:00	03:40:00	<button>View Order</button>
2	3	Burger Shack	2020-03-27 08:00:00	In Progress	-	<button>View Order</button>

```
SELECT restaurant.Name,
       inventoryorder.OID,
       inventoryorder.OrderDate,
       inventoryorder.FulfilledDate,
       orderduration.OrderDuration
  FROM inventoryorder
 INNER JOIN orderduration ON inventoryorder.OrderDate=orderduration.OrderDate AND
 inventoryorder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN restaurant ON restaurant.UID=inventoryorder.RestaurantUID
 WHERE inventoryorder.WholesalerUID=?
 ORDER BY FulfilledDate ASC
```

Output:

#	ID	Customer	Order Date	Fulfilled Date ▾	Order Duration	
1	3	Burger Shack	2020-03-27 08:00:00	In Progress	-	<button>View Order</button>
2	2	Italy Pasta Cafe	2020-03-26 05:00:00	2020-03-26 08:40:00	03:40:00	<button>View Order</button>

```
SELECT restaurant.Name,
       inventoryorder.OID,
       inventoryorder.OrderDate,
       inventoryorder.FulfilledDate,
       orderduration.OrderDuration
  FROM inventoryorder
 INNER JOIN orderduration ON inventoryorder.OrderDate=orderduration.OrderDate AND
 inventoryorder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN restaurant ON restaurant.UID=inventoryorder.RestaurantUID
 WHERE inventoryorder.WholesalerUID=?
 ORDER BY OrderDuration DESC
```

Output:

#	ID	Customer	Order Date	Fulfilled Date	Order Duration ▲	
1	2	Italy Pasta Cafe	2020-03-26 05:00:00	2020-03-26 08:40:00	03:40:00	<a href="#">View Order</a>
2	3	Burger Shack	2020-03-27 08:00:00	In Progress	-	<a href="#">View Order</a>

```
SELECT restaurant.Name,
       inventoryorder.OID,
       inventoryorder.OrderDate,
       inventoryorder.FulfilledDate,
       orderduration.OrderDuration
  FROM inventoryorder
 INNER JOIN orderduration ON inventoryorder.OrderDate=orderduration.OrderDate AND
 inventoryorder.FulfilledDate=orderduration.FulfilledDate
 INNER JOIN restaurant ON restaurant.UID=inventoryorder.RestaurantUID
 WHERE inventoryorder.WholesalerUID=?
 ORDER BY OrderDuration ASC
```

Output:

#	ID	Customer	Order Date	Fulfilled Date	Order Duration ▾	
1	3	Burger Shack	2020-03-27 08:00:00	In Progress	-	<button>View Order</button>
2	2	Italy Pasta Cafe	2020-03-26 05:00:00	2020-03-26 08:40:00	03:40:00	<button>View Order</button>

```
SELECT EmailAddress FROM customer WHERE EmailAddress=?
```

Output: No output, used for email validation.

```
SELECT EmailAddress FROM restaurant WHERE EmailAddress=?
```

Output: No output, used for email validation.

```
SELECT EmailAddress FROM wholesaler WHERE EmailAddress=?
```

Output: No output, used for email validation.

```
INSERT INTO postalcode (PostalCode, Province, City) VALUES (?, ?, ?)
```

```
INSERT INTO address (Street, HouseNumber, PostalCode) VALUES (?, ?, ?)
```

```
INSERT INTO address (Street, HouseNumber, ApartmentNumber, PostalCode) VALUES (?, ?, ?, ?, ?)
```

```
SELECT AID FROM address WHERE Street=? AND HouseNumber=? AND ApartmentNumber=? AND PostalC
```

Output: No output, used for creating customer, restaurant, and wholesaler accounts.

```
INSERT INTO customer (AID, Name, PhoneNumber, EmailAddress, Password) VALUES  
(?, ?, ?, ?, ?)
```

```
INSERT INTO restaurant (AID, Name, PhoneNumber, EmailAddress, Password) VALUES  
(?, ?, ?, ?, ?)
```

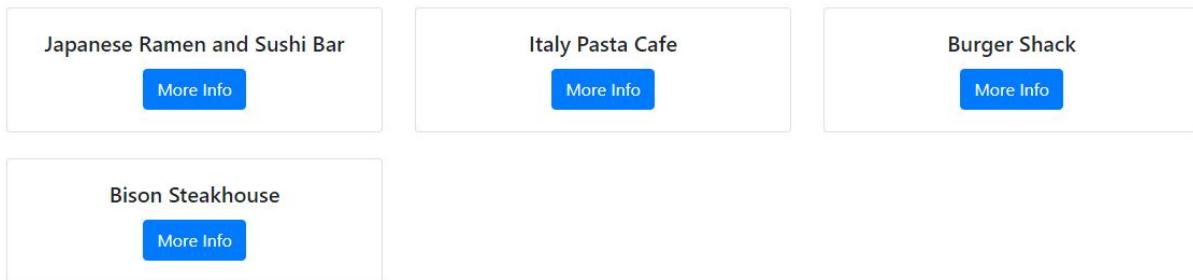
```
INSERT INTO wholesaler (AID, Name, PhoneNumber, EmailAddress, Password) VALUES  
(?, ?, ?, ?, ?)
```

```
SELECT * FROM restaurant WHERE UID=?
```

Output: No output, used to find wholesaler inventory items, address and postal code.

```
SELECT UID, Name FROM restaurant
```

Output: (UID stored in href)

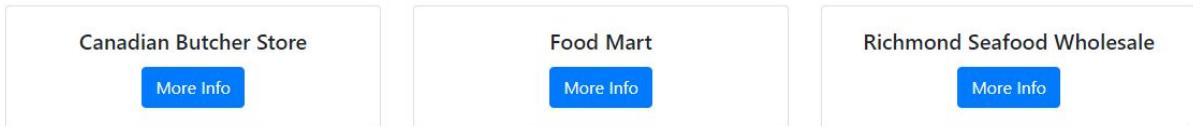


```
SELECT * FROM wholesaler WHERE UID=?
```

Output: No output, used to find wholesaler inventory items, address and postal code.

```
SELECT UID, Name FROM wholesaler
```

Output: (UID stored in href)



```
SELECT * FROM menuitem WHERE UID=?
```

Output:

<table border="1"><thead><tr><th>Food</th></tr></thead><tbody><tr><td>Tuna Nigiri</td></tr><tr><td>\$2.00</td></tr><tr><td><a href="#">More Info</a></td></tr></tbody></table>	Food	Tuna Nigiri	\$2.00	<a href="#">More Info</a>	<table border="1"><thead><tr><th>Food</th></tr></thead><tbody><tr><td>Tuna Sashimi</td></tr><tr><td>\$3.00</td></tr><tr><td><a href="#">More Info</a></td></tr></tbody></table>	Food	Tuna Sashimi	\$3.00	<a href="#">More Info</a>	<table border="1"><thead><tr><th>Food</th></tr></thead><tbody><tr><td>Salmon Nigiri</td></tr><tr><td>\$2.00</td></tr><tr><td><a href="#">More Info</a></td></tr></tbody></table>	Food	Salmon Nigiri	\$2.00	<a href="#">More Info</a>
Food														
Tuna Nigiri														
\$2.00														
<a href="#">More Info</a>														
Food														
Tuna Sashimi														
\$3.00														
<a href="#">More Info</a>														
Food														
Salmon Nigiri														
\$2.00														
<a href="#">More Info</a>														
<table border="1"><thead><tr><th>Food</th></tr></thead><tbody><tr><td>Salmon Sashimi</td></tr><tr><td>\$3.00</td></tr><tr><td><a href="#">More Info</a></td></tr></tbody></table>	Food	Salmon Sashimi	\$3.00	<a href="#">More Info</a>	<table border="1"><thead><tr><th>Food</th></tr></thead><tbody><tr><td>Ebi Nigiri</td></tr><tr><td>\$2.50</td></tr><tr><td><a href="#">More Info</a></td></tr></tbody></table>	Food	Ebi Nigiri	\$2.50	<a href="#">More Info</a>	<table border="1"><thead><tr><th>Food</th></tr></thead><tbody><tr><td>Tamago Nigiri</td></tr><tr><td>\$1.75</td></tr><tr><td><a href="#">More Info</a></td></tr></tbody></table>	Food	Tamago Nigiri	\$1.75	<a href="#">More Info</a>
Food														
Salmon Sashimi														
\$3.00														
<a href="#">More Info</a>														
Food														
Ebi Nigiri														
\$2.50														
<a href="#">More Info</a>														
Food														
Tamago Nigiri														
\$1.75														
<a href="#">More Info</a>														

```
SELECT * FROM menuitem WHERE UID=? AND Price>=?
```

Output:

The screenshot shows a user interface for searching menu items. At the top, there are sorting options: "Sort by price" (disabled), "Ascending ↑↓" (selected), "Descending ↓↑", "Min 5", "Max", and a green "Sort" button. Below these are filters: "Sort by type" (disabled), "Food" (selected), "Dessert", "Drink", "Everything", and a "Sort" button. A "Show All" link is also present. The main area displays five menu items in a grid:

Food	Food	Food
Original Hamburger \$6.99 <a href="#">More Info</a>	Cheeseburger \$6.99 <a href="#">More Info</a>	Original Wings \$7.99 <a href="#">More Info</a>
Hot Wings \$7.99 <a href="#">More Info</a>	Honey Garlic Wings \$7.99 <a href="#">More Info</a>	

```
SELECT * FROM menuitem WHERE UID=? AND Price<=?
```

Output:

The screenshot shows a user interface for searching menu items. At the top, there are sorting options: 'Sort by price' (gray), 'Ascending ↑↓' (blue), 'Descending ↑↓' (gray), 'Min' (gray input field), 'Max' (gray input field with value '1.99'), and a green 'Sort' button. Below these are category filters: 'Sort by type' (gray), 'Food 🍔' (gray), 'Dessert 🍰' (gray), 'Drink 🍹' (blue), 'Everything 🍔🍰🍹' (gray), and a green 'Sort' button. A 'Show All' button is also present. The main area displays five menu items in cards:

- Pepsi** (\$1.99) - Drink category
- Coca Cola** (\$1.99) - Drink category
- Sprite** (\$1.99) - Drink category
- A&W Root Beer** (\$1.99) - Drink category
- Ginger Ale** (\$1.99) - Drink category

Each item card has a blue 'More Info' button at the bottom.

```
SELECT * FROM menuitem WHERE UID=? AND Price<=? AND Price>=?
```

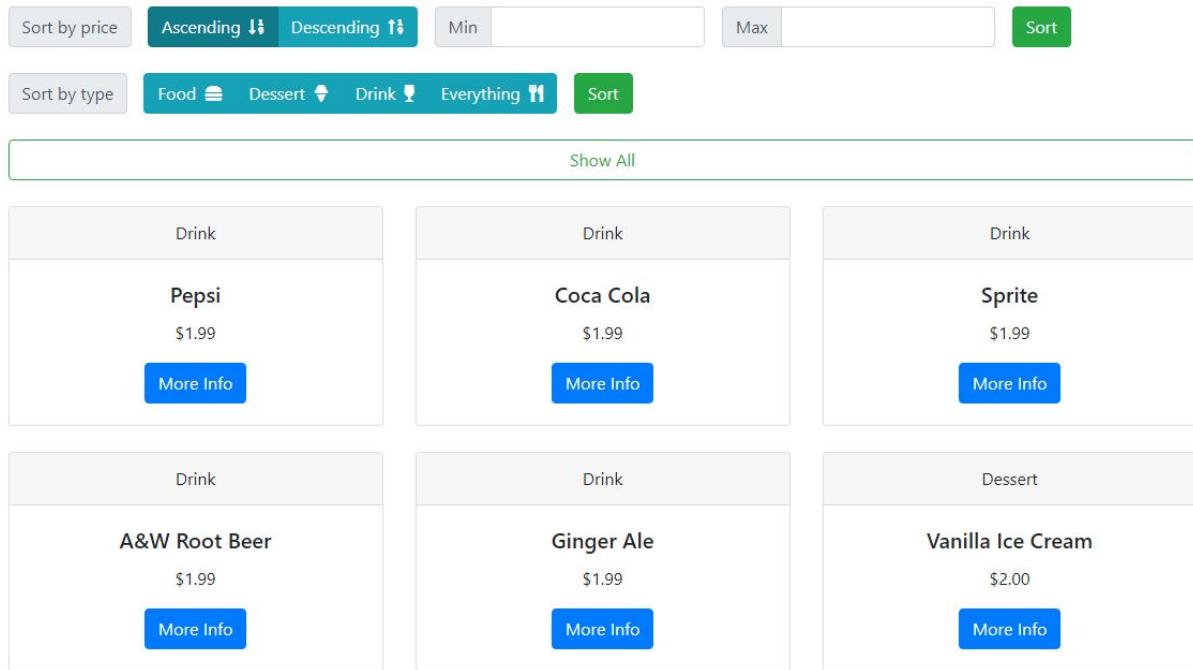
Output:

The screenshot shows a user interface for searching menu items. At the top, there are sorting options: "Sort by price" (gray), "Ascending" (blue), "Descending" (blue), and a search bar with "Min: 2" and "Max: 2.5". A green "Sort" button is to the right. Below this, there are category filters: "Sort by type" (gray), "Food" (gray), "Dessert" (blue), "Drink" (gray), "Everything" (gray), and another "Sort" button. A "Show All" link is also present. The main content area displays five menu items in cards:

Category	Item Name	Price	Action
Dessert	Vanilla Ice Cream	\$2.00	<a href="#">More Info</a>
Dessert	Chocolate Ice Cream	\$2.00	<a href="#">More Info</a>
Dessert	Strawberry Ice Cream	\$2.00	<a href="#">More Info</a>
Dessert	Ice Cream Sandwich	\$2.00	<a href="#">More Info</a>

```
SELECT * FROM menuitem WHERE UID=? ORDER BY Price ASC
```

Output:



The screenshot shows a user interface for searching menu items. At the top, there are sorting options: "Sort by price" (gray), "Ascending" (highlighted in teal), "Descending" (gray), "Min" (gray input field), "Max" (gray input field), and a green "Sort" button. Below these are filter buttons: "Sort by type" (gray), "Food" (gray), "Dessert" (gray), "Drink" (highlighted in teal), "Everything" (gray), and another "Sort" button. A "Show All" link is also present. The main area displays six menu items in a grid:

Category	Item Name	Price	Action
Drink	Pepsi	\$1.99	<a href="#">More Info</a>
Drink	Coca Cola	\$1.99	<a href="#">More Info</a>
Drink	Sprite	\$1.99	<a href="#">More Info</a>
Drink	A&W Root Beer	\$1.99	<a href="#">More Info</a>
Drink	Ginger Ale	\$1.99	<a href="#">More Info</a>
Dessert	Vanilla Ice Cream	\$2.00	<a href="#">More Info</a>

```
SELECT * FROM menuitem WHERE UID=? AND Price>=? ORDER BY Price ASC
```

Output:

The screenshot shows a user interface for searching menu items. At the top, there are several buttons: "Sort by price" (gray), "Ascending ↑↓" (blue), "Descending ↑↓" (blue), "Min" (gray) with the value "5", "Max" (gray) with an empty input field, and a green "Sort" button. Below these are buttons for "Sort by type": "Food" (gray), "Dessert" (gray), "Drink" (gray), and "Everything" (blue). There is also a "Sort" button. A "Show All" link is located above a grid of five menu items.

Food	Food	Food
Original Hamburger \$6.99 <a href="#">More Info</a>	Cheeseburger \$6.99 <a href="#">More Info</a>	Original Wings \$7.99 <a href="#">More Info</a>
Hot Wings \$7.99 <a href="#">More Info</a>	Honey Garlic Wings \$7.99 <a href="#">More Info</a>	

```
SELECT * FROM menuitem WHERE UID=? AND Price<=? ORDER BY Price ASC
```

Output:

The screenshot shows a user interface for searching menu items. At the top, there are buttons for "Sort by price" (gray), "Ascending" (green), "Descending" (green), and "Sort" (green). Below these are input fields for "Min" and "Max" with a value of "3". A "Sort" button is also present. Underneath, there are buttons for "Sort by type" (gray), "Food" (gray), "Dessert" (gray), "Drink" (gray), and "Everything" (green). A "Show All" button is located below the sorting controls. The main area displays six menu items in a grid:

Category	Item Name	Price	Action
Drink	Pepsi	\$1.99	<a href="#">More Info</a>
Drink	Coca Cola	\$1.99	<a href="#">More Info</a>
Drink	Sprite	\$1.99	<a href="#">More Info</a>
Drink	A&W Root Beer	\$1.99	<a href="#">More Info</a>
Drink	Ginger Ale	\$1.99	<a href="#">More Info</a>
Dessert	Ice Cream Sandwich	\$2.00	<a href="#">More Info</a>

```
SELECT * FROM menuitem WHERE UID=? AND Price<=? AND Price>=? ORDER BY Price ASC
```

Output:

The screenshot shows a user interface for searching menu items. At the top, there are sorting options: 'Sort by price' (disabled), 'Ascending ↑↓' (selected), 'Descending ↑↓' (disabled), and a 'Sort' button. Below these are filters for 'Min' (2.5) and 'Max' (4). There are also buttons for 'Sort by type' (disabled), 'Food' (selected), 'Dessert', 'Drink', and 'Everything'. A 'Show All' link is at the bottom of the filter section.

The results are displayed in four cards:

- Food**:
  - Original Fries**: \$2.99
  - Sweet Potato Fries**: \$3.99
- Dessert**:
  - Apple Pie**: \$2.99
- Food**:
  - Curly Fries**: \$3.99

Each card has a 'More Info' button at the bottom.

```
SELECT * FROM menuitem WHERE UID=? ORDER BY Price DESC
```

Output:

The screenshot shows a user interface for searching menu items. At the top, there are sorting options: 'Sort by price' (gray), 'Ascending ↓↑' (blue), 'Descending ↑↓' (gray), 'Min' (gray input field), 'Max' (gray input field), and a green 'Sort' button. Below these are filter buttons: 'Sort by type' (gray), 'Food' (gray), 'Dessert' (gray), 'Drink' (gray), and 'Everything' (gray). A 'Sort' button is also present here. A green 'Show All' button is located above a grid of six menu items.

Food	Food	Food
Honey Garlic Wings \$7.99 <a href="#">More Info</a>	Original Wings \$7.99 <a href="#">More Info</a>	Hot Wings \$7.99 <a href="#">More Info</a>
Original Hamburger \$6.99 <a href="#">More Info</a>	Cheeseburger \$6.99 <a href="#">More Info</a>	Onion Rings \$4.99 <a href="#">More Info</a>

```
SELECT * FROM menuitem WHERE UID=? AND Price>=? ORDER BY Price DESC
```

Output:

The screenshot shows a user interface for searching menu items. At the top, there are sorting options: 'Sort by price' (gray), 'Ascending ↓↑' (blue), 'Descending ↑↓' (gray), and a numeric input field 'Min 5 Max' followed by a 'Sort' button. Below these are category filters: 'Sort by type' (gray), 'Food' (blue), 'Dessert' (gray), 'Drink' (gray), and 'Everything' (gray). A 'Sort' button is also present here. A 'Show All' link is located above the results. The results are displayed in a grid of five boxes:

Food	Food	Food
Original Wings \$7.99 <a href="#">More Info</a>	Hot Wings \$7.99 <a href="#">More Info</a>	Honey Garlic Wings \$7.99 <a href="#">More Info</a>
Original Hamburger \$6.99 <a href="#">More Info</a>	Cheeseburger \$6.99 <a href="#">More Info</a>	

```
SELECT * FROM menuitem WHERE UID=? AND Price<=? ORDER BY Price DESC
```

Output:

<input type="button" value="Sort by price"/>	<input checked="" type="button" value="Ascending ⬇"/>	<input type="button" value="Descending ⬆"/>	<input type="text" value="Min"/>	<input type="text" value="Max"/>	<input type="text" value="3"/>	<input type="button" value="Sort"/>
<input type="button" value="Sort by type"/>	<input type="button" value="Food 🍔"/>	<input type="button" value="Dessert 🍰"/>	<input type="button" value="Drink 🥤"/>	<input type="button" value="Everything 🍔🍰🥤"/>	<input type="button" value="Sort"/>	
<a href="#">Show All</a>						
<div style="border: 1px solid #ccc; padding: 10px; margin-bottom: 10px;">Dessert  Apple Pie \$2.99 <a href="#" style="border: 1px solid #007bff; padding: 5px 15px; color: inherit; text-decoration: none;">More Info</a></div>		<div style="border: 1px solid #ccc; padding: 10px; margin-bottom: 10px;">Food  Original Fries \$2.99 <a href="#" style="border: 1px solid #007bff; padding: 5px 15px; color: inherit; text-decoration: none;">More Info</a></div>		<div style="border: 1px solid #ccc; padding: 10px; margin-bottom: 10px;">Dessert  Vanilla Ice Cream \$2.00 <a href="#" style="border: 1px solid #007bff; padding: 5px 15px; color: inherit; text-decoration: none;">More Info</a></div>		
<div style="border: 1px solid #ccc; padding: 10px; margin-bottom: 10px;">Dessert  Chocolate Ice Cream \$2.00 <a href="#" style="border: 1px solid #007bff; padding: 5px 15px; color: inherit; text-decoration: none;">More Info</a></div>		<div style="border: 1px solid #ccc; padding: 10px; margin-bottom: 10px;">Dessert  Strawberry Ice Cream \$2.00 <a href="#" style="border: 1px solid #007bff; padding: 5px 15px; color: inherit; text-decoration: none;">More Info</a></div>		<div style="border: 1px solid #ccc; padding: 10px; margin-bottom: 10px;">Dessert  Ice Cream Sandwich \$2.00 <a href="#" style="border: 1px solid #007bff; padding: 5px 15px; color: inherit; text-decoration: none;">More Info</a></div>		

```
SELECT * FROM menuitem WHERE UID=? AND Price<=? AND Price>=? ORDER BY Price DESC
```

Output:

Sort by price Ascending Descending Min 2.5 Max 5 Sort

Sort by type Food Dessert Drink Everything Show All

Food	Food	Food
Onion Rings \$4.99 More Info	Curly Fries \$3.99 More Info	Sweet Potato Fries \$3.99 More Info
Food	Dessert	
Original Fries \$2.99 More Info	Apple Pie \$2.99 More Info	

```
SELECT * FROM menuitem WHERE UID=? AND FoodType="Food"
```

Output:

Sort by price    Ascending    Descending    Min    Max    Sort

Sort by type    Food    Dessert    Drink    Everything    Sort

Show All

Food	Food	Food
Original Hamburger \$6.99 <a href="#">More Info</a>	Cheeseburger \$6.99 <a href="#">More Info</a>	Original Wings \$7.99 <a href="#">More Info</a>
Hot Wings \$7.99 <a href="#">More Info</a>	Honey Garlic Wings \$7.99 <a href="#">More Info</a>	Original Fries \$2.99 <a href="#">More Info</a>

```
SELECT * FROM menuitem WHERE UID=? AND FoodType="Dessert"
```

Output:

The screenshot shows a user interface for searching menu items. At the top, there are sorting options: 'Sort by price' (gray), 'Ascending ↑↓' (blue), 'Descending ↑↓' (blue), 'Min' (gray) with an input field, 'Max' (gray) with an input field, and a green 'Sort' button. Below these are filter buttons: 'Sort by type' (gray), 'Food' (gray), 'Dessert' (blue), 'Drink' (gray), 'Everything' (gray), and a green 'Sort' button. A 'Show All' link is also present. The main area displays five menu items in a grid:

	Dessert	Dessert	Dessert
	Vanilla Ice Cream \$2.00 <a href="#">More Info</a>	Chocolate Ice Cream \$2.00 <a href="#">More Info</a>	Strawberry Ice Cream \$2.00 <a href="#">More Info</a>
	Ice Cream Sandwich \$2.00 <a href="#">More Info</a>	Apple Pie \$2.99 <a href="#">More Info</a>	

```
SELECT * FROM menuitem WHERE UID=? AND FoodType="Drink"
```

Output:

The screenshot shows a user interface for a menu item search. At the top, there are sorting options: 'Sort by price' (disabled), 'Ascending' (selected), 'Descending', 'Min' input field (empty), 'Max' input field (empty), and a green 'Sort' button. Below these are filter buttons: 'Sort by type' (disabled), 'Food' (selected), 'Dessert' (disabled), 'Drink' (selected), 'Everything' (disabled), and another 'Sort' button. A 'Show All' link is also present. The main area displays five cards, each representing a drink:

- Pepsi** - Drink, \$1.99, More Info
- Coca Cola** - Drink, \$1.99, More Info
- Sprite** - Drink, \$1.99, More Info
- A&W Root Beer** - Drink, \$1.99, More Info
- Ginger Ale** - Drink, \$1.99, More Info

```
SELECT * FROM inventoryitem WHERE UID=?
```

Output:

Pork Chop 0.5lb \$1.99 Amount: 1 <a href="#">More Info</a>	Lamb Chop \$23.99 Amount: 16 <a href="#">More Info</a>	Whole Chicken \$6.99 Amount: 1 <a href="#">More Info</a>
Beef Ribs 2lb \$15.99 Amount: 1 <a href="#">More Info</a>	Pork Tenderloin 1lb \$6.99 Amount: 1 <a href="#">More Info</a>	Fillet Mignon 0.5lb \$22.99 Amount: 1 <a href="#">More Info</a>

```
SELECT * FROM inventoryitem WHERE UID=? AND Cost>=?
```

Output:

<a href="#">Sort by price</a>	<a href="#">Ascending ↗</a>	<a href="#">Descending ↘</a>	<input type="text" value="Min"/> 15	<input type="text" value="Max"/>	<a href="#">Sort</a>
<a href="#">Sort by name</a>	<a href="#">Ascending ↗</a>	<a href="#">Descending ↘</a>	<a href="#">Sort</a>	<a href="#">Show All</a>	
Ribeye Steak 0.5lb		New York Strip Loin 0.5lb		Lamb Chop	
\$16.99		\$18.99		\$23.99	
Amount: 1		Amount: 1		Amount: 16	
<a href="#">More Info</a>		<a href="#">More Info</a>		<a href="#">More Info</a>	
Beef Ribs 2lb		Fillet Mignon 0.5lb		Prime Rib 1lb	
\$15.99		\$22.99		\$24.99	
Amount: 1		Amount: 1		Amount: 1	
<a href="#">More Info</a>		<a href="#">More Info</a>		<a href="#">More Info</a>	

```
SELECT * FROM inventoryitem WHERE UID=? AND Cost<=?
```

Output:

<input type="button" value="Sort by price"/>	<input type="button" value="Ascending ⬇️"/>	<input type="button" value="Descending ⬆️"/>	<input type="text" value="Min"/>	<input type="text" value="Max"/>	<input type="text" value="15"/>	<input type="button" value="Sort"/>
<input type="button" value="Sort by name"/>	<input type="button" value="Ascending ⬇️"/>	<input type="button" value="Descending ⬆️"/>	<input type="button" value="Sort"/>	<a href="#">Show All</a>		
<b>Ground Beef 1lb</b> \$2.99 Amount: 1 <a href="#">More Info</a>		<b>Pork Chop 0.5lb</b> \$1.99 Amount: 1 <a href="#">More Info</a>		<b>Whole Chicken</b> \$6.99 Amount: 1 <a href="#">More Info</a>		
<b>Pork Tenderloin 1lb</b> \$6.99 Amount: 1 <a href="#">More Info</a>		<b>Chicken Breast</b> \$1.99 Amount: 1 <a href="#">More Info</a>		<b>Pork Belly 1lb</b> \$6.99 Amount: 1 <a href="#">More Info</a>		

```
SELECT * FROM inventoryitem WHERE UID=? AND Cost>=? AND Cost<=?
```

Output:

<input type="button" value="Sort by price"/>	<input type="button" value="Ascending ⬇"/> <input type="button" value="Descending ⬆"/>	<input type="button" value="Min"/> 5 <input type="button" value="Max"/> 20 <input type="button" value="Sort"/>
<input type="button" value="Sort by name"/>	<input type="button" value="Ascending ⬇"/> <input type="button" value="Descending ⬆"/>	<input type="button" value="Sort"/>
<a href="#">Show All</a>		
Ribeye Steak 0.5lb \$16.99 Amount: 1 <a href="#">More Info</a>	New York Strip Loin 0.5lb \$18.99 Amount: 1 <a href="#">More Info</a>	Whole Chicken \$6.99 Amount: 1 <a href="#">More Info</a>
Beef Ribs 2lb \$15.99 Amount: 1 <a href="#">More Info</a>	Pork Tenderloin 1lb \$6.99 Amount: 1 <a href="#">More Info</a>	Chicken Wings \$16.99 Amount: 20 <a href="#">More Info</a>

```
SELECT * FROM inventoryitem WHERE UID=? ORDER BY Cost ASC
```

Output:

The screenshot shows a user interface for managing inventory items. At the top, there are sorting options: 'Sort by price' (gray), 'Ascending ↓↑' (blue), 'Descending ↑↓' (green), input fields for 'Min' and 'Max' (gray), and a 'Sort' button (green). Below these are two more sets of buttons: 'Sort by name' (gray), 'Ascending ↓↑' (blue), 'Descending ↑↓' (green), and another 'Sort' button (green). A 'Show All' link is located above a green-bordered table. The table contains six rows, each representing an item with a 'More Info' button:

Pork Chop 0.5lb	Chicken Breast	Beef Bones 1lb
\$1.99 Amount: 1 <a href="#">More Info</a>	\$1.99 Amount: 1 <a href="#">More Info</a>	\$2.99 Amount: 1 <a href="#">More Info</a>
Ground Beef 1lb	Pork Belly 1lb	Pork Tenderloin 1lb
\$2.99 Amount: 1 <a href="#">More Info</a>	\$6.99 Amount: 1 <a href="#">More Info</a>	\$6.99 Amount: 1 <a href="#">More Info</a>

```
SELECT * FROM inventoryitem WHERE UID=? AND Cost>=? ORDER BY Cost ASC
```

Output:

<input type="button" value="Sort by price"/>	<input type="button" value="Ascending ⬆⬇"/>	<input type="button" value="Descending ⬇⬆"/>	<input type="button" value="Min"/> 15	<input type="button" value="Max"/>	<input type="button" value="Sort"/>
<input type="button" value="Sort by name"/>	<input type="button" value="Ascending ⬆⬇"/>	<input type="button" value="Descending ⬇⬆"/>	<input type="button" value="Sort"/>	<input type="button" value="Show All"/>	
Beef Ribs 2lb		Ribeye Steak 0.5lb		Chicken Wings	
\$15.99		\$16.99		\$16.99	
Amount: 1		Amount: 1		Amount: 20	
<input type="button" value="More Info"/>		<input type="button" value="More Info"/>		<input type="button" value="More Info"/>	
New York Strip Loin 0.5lb		Fillet Mignon 0.5lb		Lamb Chop	
\$18.99		\$22.99		\$23.99	
Amount: 1		Amount: 1		Amount: 16	
<input type="button" value="More Info"/>		<input type="button" value="More Info"/>		<input type="button" value="More Info"/>	

```
SELECT * FROM inventoryitem WHERE UID=? AND Cost<=? ORDER BY Cost ASC
```

Output:

The screenshot shows a user interface for searching inventory items. At the top, there are two rows of buttons: the first row includes "Sort by price" (gray), "Ascending" (blue), "Descending" (blue), "Min" (gray) with an input field containing an empty box, "Max" (gray) with an input field containing "5", and a green "Sort" button; the second row includes "Sort by name" (gray), "Ascending" (blue), "Descending" (blue), and another "Sort" button. Below these rows is a green "Show All" button. The main area displays five items in cards:

- Pork Chop 0.5lb**: \$1.99, Amount: 1, More Info button
- Chicken Breast**: \$1.99, Amount: 1, More Info button
- Ground Beef 1lb**: \$2.99, Amount: 1, More Info button
- Beef Bones 1lb**: \$2.99, Amount: 1, More Info button

```
SELECT * FROM inventoryitem WHERE UID=? AND Cost>=? AND Cost<=? ORDER BY Cost ASC
```

Output:

<input type="button" value="Sort by price"/>	<input type="button" value="Ascending ⬆⬇"/>	<input type="button" value="Descending ⬇⬆"/>	<input type="button" value="Min"/> 5	<input type="button" value="Max"/> 20	<input type="button" value="Sort"/>
<input type="button" value="Sort by name"/>	<input type="button" value="Ascending ⬆⬇"/>	<input type="button" value="Descending ⬇⬆"/>	<input type="button" value="Sort"/>	<a href="#">Show All</a>	
<b>Whole Chicken</b> \$6.99 Amount: 1 <a href="#">More Info</a>		<b>Pork Tenderloin 1lb</b> \$6.99 Amount: 1 <a href="#">More Info</a>		<b>Pork Belly 1lb</b> \$6.99 Amount: 1 <a href="#">More Info</a>	
<b>Beef Ribs 2lb</b> \$15.99 Amount: 1 <a href="#">More Info</a>		<b>Ribeye Steak 0.5lb</b> \$16.99 Amount: 1 <a href="#">More Info</a>		<b>Chicken Wings</b> \$16.99 Amount: 20 <a href="#">More Info</a>	

```
SELECT * FROM inventoryitem WHERE UID=? ORDER BY Cost DESC
```

Output:

<input type="button" value="Sort by price"/>	<input type="button" value="Ascending ⬇"/>	<input type="button" value="Descending ⬆"/>	<input type="button" value="Min"/>	<input type="text"/>	<input type="button" value="Max"/>	<input type="text"/>	<input type="button" value="Sort"/>
<input type="button" value="Sort by name"/>	<input type="button" value="Ascending ⬇"/>	<input type="button" value="Descending ⬆"/>	<input type="button" value="Sort"/>	<a href="#">Show All</a>			
<b>Prime Rib 1lb</b> \$24.99 Amount: 1 <a href="#">More Info</a>		<b>Lamb Chop</b> \$23.99 Amount: 16 <a href="#">More Info</a>		<b>Fillet Mignon 0.5lb</b> \$22.99 Amount: 1 <a href="#">More Info</a>			
<b>New York Strip Loin 0.5lb</b> \$18.99 Amount: 1 <a href="#">More Info</a>		<b>Ribeye Steak 0.5lb</b> \$16.99 Amount: 1 <a href="#">More Info</a>		<b>Chicken Wings</b> \$16.99 Amount: 20 <a href="#">More Info</a>			

```
SELECT * FROM inventoryitem WHERE UID=? AND Cost>=? ORDER BY Cost DESC
```

Output:

<input type="button" value="Sort by price"/>	<input type="button" value="Ascending ⬆⬇"/>	<input type="button" value="Descending ⬇⬆"/>	<input type="button" value="Min"/> 6	<input type="button" value="Max"/>	<input type="button" value="Sort"/>
<input type="button" value="Sort by name"/>	<input type="button" value="Ascending ⬆⬇"/>	<input type="button" value="Descending ⬇⬆"/>	<input type="button" value="Sort"/>	<a href="#">Show All</a>	
<b>Prime Rib 1lb</b> \$24.99 Amount: 1 <a href="#">More Info</a>		<b>Lamb Chop</b> \$23.99 Amount: 16 <a href="#">More Info</a>		<b>Fillet Mignon 0.5lb</b> \$22.99 Amount: 1 <a href="#">More Info</a>	
<b>New York Strip Loin 0.5lb</b> \$18.99 Amount: 1 <a href="#">More Info</a>		<b>Ribeye Steak 0.5lb</b> \$16.99 Amount: 1 <a href="#">More Info</a>		<b>Chicken Wings</b> \$16.99 Amount: 20 <a href="#">More Info</a>	

```
SELECT * FROM inventoryitem WHERE UID=? AND Cost<=? ORDER BY Cost DESC
```

Output:

<input type="button" value="Sort by price"/>	<input type="button" value="Ascending ⬇️"/>	<input type="button" value="Descending ⬆️"/>	<input type="button" value="Min"/>	<input type="text" value=""/>	<input type="button" value="Max"/>	<input type="text" value="20"/>	<input type="button" value="Sort"/>
<input type="button" value="Sort by name"/>	<input type="button" value="Ascending ⬇️"/>	<input type="button" value="Descending ⬆️"/>		<input type="button" value="Sort"/>			
<a href="#">Show All</a>							
<b>New York Strip Loin 0.5lb</b> \$18.99 Amount: 1 <a href="#">More Info</a>	<b>Ribeye Steak 0.5lb</b> \$16.99 Amount: 1 <a href="#">More Info</a>	<b>Chicken Wings</b> \$16.99 Amount: 20 <a href="#">More Info</a>					
<b>Beef Ribs 2lb</b> \$15.99 Amount: 1 <a href="#">More Info</a>	<b>Whole Chicken</b> \$6.99 Amount: 1 <a href="#">More Info</a>	<b>Pork Tenderloin 1lb</b> \$6.99 Amount: 1 <a href="#">More Info</a>					

```
SELECT * FROM inventoryitem WHERE UID=? AND Cost>=? AND Cost<=? ORDER BY Cost DESC
```

Output:

<input type="button" value="Sort by price"/>	<input type="button" value="Ascending ⬆⬇"/>	<input type="button" value="Descending ⬇⬆"/>	<input type="button" value="Min"/> 5	<input type="button" value="Max"/> 20	<input type="button" value="Sort"/>
<input type="button" value="Sort by name"/>	<input type="button" value="Ascending ⬆⬇"/>	<input type="button" value="Descending ⬇⬆"/>	<input type="button" value="Sort"/>	<a href="#">Show All</a>	
<b>New York Strip Loin 0.5lb</b> \$18.99 Amount: 1 <a href="#">More Info</a>		<b>Ribeye Steak 0.5lb</b> \$16.99 Amount: 1 <a href="#">More Info</a>		<b>Chicken Wings</b> \$16.99 Amount: 20 <a href="#">More Info</a>	
<b>Beef Ribs 2lb</b> \$15.99 Amount: 1 <a href="#">More Info</a>		<b>Whole Chicken</b> \$6.99 Amount: 1 <a href="#">More Info</a>		<b>Pork Tenderloin 1lb</b> \$6.99 Amount: 1 <a href="#">More Info</a>	

```
SELECT * FROM inventoryitem WHERE UID=? ORDER BY Name ASC
```

Output:

<input type="button" value="Sort by price"/>	<input type="button" value="Ascending ⬇"/> <input type="button" value="Descending ⬆"/>	<input type="text" value="Min"/> <input type="text" value="Max"/>	<input type="button" value="Sort"/>
<input type="button" value="Sort by name"/>	<input type="button" value="Ascending ⬇"/> <input type="button" value="Descending ⬆"/>	<input type="button" value="Sort"/>	
<a href="#">Show All</a>			
<b>Beef Bones 1lb</b> \$2.99 Amount: 1 <a href="#">More Info</a>	<b>Beef Ribs 2lb</b> \$15.99 Amount: 1 <a href="#">More Info</a>	<b>Chicken Breast</b> \$1.99 Amount: 1 <a href="#">More Info</a>	
<b>Chicken Wings</b> \$16.99 Amount: 20 <a href="#">More Info</a>	<b>Fillet Mignon 0.5lb</b> \$22.99 Amount: 1 <a href="#">More Info</a>	<b>Ground Beef 1lb</b> \$2.99 Amount: 1 <a href="#">More Info</a>	

```
SELECT * FROM inventoryitem WHERE UID=? ORDER BY Name DESC
```

Output:

<input type="button" value="Sort by price"/>	<input type="button" value="Ascending ⬆⬇"/>	<input type="button" value="Descending ⬇⬆"/>	<input type="text" value="Min"/>	<input type="text" value="Max"/>	<input type="button" value="Sort"/>
<input type="button" value="Sort by name"/>	<input type="button" value="Ascending ⬆⬇"/>	<input type="button" value="Descending ⬇⬆"/>	<input type="button" value="Sort"/>	<a href="#">Show All</a>	
<b>Whole Chicken</b> \$6.99 Amount: 1 <a href="#">More Info</a>		<b>Ribeye Steak 0.5lb</b> \$16.99 Amount: 1 <a href="#">More Info</a>		<b>Prime Rib 1lb</b> \$24.99 Amount: 1 <a href="#">More Info</a>	
<b>Pork Tenderloin 1lb</b> \$6.99 Amount: 1 <a href="#">More Info</a>		<b>Pork Chop 0.5lb</b> \$1.99 Amount: 1 <a href="#">More Info</a>		<b>Pork Belly 1lb</b> \$6.99 Amount: 1 <a href="#">More Info</a>	

```
DELETE FROM menuitem WHERE UID=? AND IID=?
```

```
DELETE FROM inventoryitem WHERE UID=? AND IID=?
```

```
SELECT MAX(menuitem.IID) FROM menuitem GROUP BY UID HAVING UID=?
```

Output: No output, used to find IID for INSERT.

```
SELECT MAX(inventoryitem.IID) FROM inventoryitem GROUP BY UID HAVING UID=?
```

Output: No output, used to find IID for INSERT.

```
SELECT INID FROM ingredient WHERE Name=?
```

Output: No output, used to find INID for INSERT.

```
INSERT INTO ingredient (Name) VALUES (?)
```

```
INSERT INTO inventoryitem (UID, IID, INID, Name, Cost, Amount) VALUES (?, ?, ?, ?, ?,  
".$Cost.", ?)
```

```
INSERT INTO menuitem (UID, IID, Name, Price, FoodType) VALUES (?, ?, ?, ?, ".$Price.", ?)
```

```
INSERT INTO menuitemingredient (UID, IID, INID, Amount) VALUES (?, ?, ?, ?, ?)
```

```
INSERT INTO autorestock (UID, AUID, WholesalerUID, IID, INID, Amount, Threshold)
VALUES ('$uid', '$auid', '$wholesaleruid', '$iid', '$inid', '$amount', '$threshold')
ON DUPLICATE KEY UPDATE Amount='\$amount', Threshold='\$threshold'
```

Output:

The image consists of two screenshots of a software application interface. The top screenshot shows a modal dialog titled "Add Rule". It contains three input fields: "Select Item" (with a dropdown menu showing "Select an item"), "Enter Threshold" (an empty input field), and "Enter Order Amount" (an empty input field). At the bottom are "Close" and "Save changes" buttons. The background shows a dark-themed dashboard with tabs for "Auto Restock", "Reports", and "Restaurants". The bottom screenshot shows a modal dialog titled "Edit Rule" for an item named "Beef Ribs 2lb (15.99 per 1) @ Canadian Butcher Store". It has the same three input fields: "Select Item" (set to the selected item), "Enter Threshold" (set to "1"), and "Enter Order Amount" (set to "15"). The "Save changes" button is visible at the bottom. The background shows a list of items with one item highlighted: "Beef Ribs" with "Threshold: 1", "To order: 15 of Beef Ribs 2lb (lots of 1) from Canadian Butcher Store", and "Current: 0". Below this are "Edit Rule" and "Delete Rule" buttons.

```
DELETE FROM autorestock WHERE UID = '$uid' AND AUID = '$auid'
```

Output:

Ribeye Steak	Beef Ribs	Avocado
<b>Threshold: 10</b>	<b>Threshold: 1</b>	<b>Threshold: 25</b>
To order: 15 of Ribeye Steak 0.5lb (lots of 1) from Canadian Butcher Store	To order: 15 of Beef Ribs 2lb (lots of 1) from Canadian Butcher Store	To order: 50 of Hass Avocado (lots of 1) from Food Mart
Current: 0	Current: 0	Current: 0
<a href="#">Edit Rule</a> <a href="#">Delete Rule</a>	<a href="#">Edit Rule</a> <a href="#">Delete Rule</a>	<a href="#">Edit Rule</a> <a href="#">Delete Rule</a>

```

$type = $_POST['type'] ?? 3;
$typetext = array("Food", "Dessert", "Drink", "%")[$type];

$grouping = $_POST['grouping'] ?? 0;
$groupingquery =
    array("GROUP BY c.Name", "GROUP BY m.Name", "GROUP BY m.FoodType")[$grouping];
$headerItem = array("c.Name", "m.Name", "m.FoodType")[$grouping];

$items = $mysqli->query("SELECT $headerItem as headerText,
SUM(Amount) as numItems, SUM(Amount * Price) as totalValue
FROM customer as c
LEFT JOIN foodorder f ON f.CustomerUID = c.UID
LEFT JOIN foodorderitems i ON f.OID = i.OID
LEFT JOIN menuitem m ON i.UID = m.UID AND i.IID = m.IID
WHERE m.FoodType LIKE '$typetext' AND f.RestaurantUID = '$uid'
" . $groupingquery);

```

## Order Statistics

View order statistics based on various groupings.

The screenshot shows a user interface for viewing order statistics. At the top, there are filtering options: 'Specify items:' with dropdowns for 'Food' (selected), 'Dessert', 'Drink', and 'Everything'. There are also buttons for 'Group by:' and sorting by 'Customer Name', 'Food Name', or 'Food Type'. A 'Show All' button is located below the filtering section. Below this, two customer profiles are displayed in boxes:

- John:** Items ordered: 3, Total value: 16.48
- Sally:** Items ordered: 6, Total value: 42.96

```
$typetext = array("Food", "Dessert", "Drink", "%")[$type];  
  
$items = $mysqli->query("SELECT * FROM customer as c  
WHERE NOT EXISTS (  
    SELECT m.IID FROM menuitem as m WHERE UID = '$uid'  
    AND FoodType LIKE '$typetext'  
    AND Price >= '$min' AND Price <= '$max'  
    AND m.IID NOT IN  
        (SELECT i.IID  
        FROM foodorder f, foodorderitems i  
        WHERE f.OID = i.OID AND f.CustomerUID = c.UID  
        AND i.UID = m.UID AND i.IID = m.IID)  
)");
```

## Customer Loyalty

Find out who has ordered all items with specified attributes.

Specify items:     Min \$  Max \$

[Show All](#)

John
john@mail.com
6046489001