

# CPSC 304 Project Cover Page

Milestone #: 4

Date: April 5, 2023

Group Number: 98

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Matthew Ma	60034139	a2w0b	matthew.ma@alumni.ubc.ca
Colleen Rideout	38298097	colleenx	colleen.rideout@gmail.com
Seoyoung Kwon	63623466	seoykwon	kwonseoy@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 the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

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

# Project Description

## Database Specifications

The domain of this application is the shipping industry and involves logistics, parcel delivery, and parcel tracking. The project contains a database that a shipping company could use to manage its customers, orders and deliveries.

## Desired Functionality of the Database

The database stores information about customers, delivery drivers, and managers, as well as parcel and order information (such as tracking information, order transactions, and which drivers are delivering the orders with their tracking updates). Queries on the database allow new orders to be created, customers to track those orders, managers to dispatch orders, and drivers to update the tracking status of the delivery.

## What It Accomplished

We completed the infrastructure to create and update various ShippingUser entities in our Database. Managers are able to perform a handful of different searches to get useful information about customers, drivers, and orders.

## Differences in Schema

Our final schema contains the same entity and relationship tables as outlined in our diagram. In order to achieve a wider variety of meaningful queries, we adjusted some of the attributes of some of the entities as follows:

1. Customer table changes:
  - a. We removed the attribute “address” from customer, since it was redundant with its parent class (ISA relationship) ShippingUser.
  - b. We added the attribute “loyalty\_member,” which is an integer representing whether or not the customer is a member of the company’s loyalty program.
2. ShippingUser (User) table changes:
  - a. We changed the name of the entity “User” to “ShippingUser” due to semantic issues with SQL.
  - b. We changed the attribute “password” to “user\_password” due to semantic issues
  - c. We added the attribute address, since that is a common attribute between all ShippingUser entities.
3. Driver table changes:
  - a. We added the attribute “orders\_delivered” so we could include more meaningful queries in our application

4. ParcelOrder table renamed from Order table:
  - a. We changed the name of the Order table to ParcelOrder to give it more clarification.
5. Insurance table added:
  - a. We separated the attribute (insurance\_expiry) from the Vehicle entity for normalization deliverable.
6. Servicing table added:
  - a. We separated the attribute (next\_servicing) from the Vehicle entity for normalization deliverable.
7. ShippingUser table renamed from User table:
  - a. We changed the name of the User table to ShippingUser table due to semantic issues with SQL.
8. Parcel table changes:
  - a. We added the attribute delivery\_type (regular, overnight, expedited) so we could include more meaningful queries in our application
9. Vehicle table changes:
  - a. We removed the attributes next\_servicing and insurance\_expiry because it would be redundant with the Insurance table.

## Use Case Example

A use case for our project would be a customer wanting to have a package delivered from one location to another. This system will help the shipping company store needed information to organize and track the delivery of the parcel.

The customer can submit parcel information and an order request to the company. In return, a delivery transaction is priced based on the order details and delivery specifications. A manager assigns the order to be delivered by a driver, who will use a vehicle from a company branch to pick up and deliver the package. Throughout the delivery process, tracking information will be updated and available to the user. When the order has been delivered, the transaction will be complete and the customer will receive notification of its completion.

## Application Platform

Java Database Connectivity and Oracle will be the platform for our project. We used Java to build out the backend and frontend of the project, which allows the user to interact with the database.

## Queries:

## INSERT Operation

Query	Where in Code
INSERT INTO SHIPPINGUSER VALUES (username, userPassword, phone, email, firstName, lastName, address)  INSERT INTO CUSTOMER VALUES(username, isLoyaltyMember)	insertShippingUser(), insertCustomer() In database/DatabaseConnectionHandler.java

Before:

Customer Table:

	USERNAME	LOYALTY_MEMBER
1	mattma	0
2	olivek	1
3	greenc	0
4	colleenr	1
5	waltert	1

ShippingUser Table:

	USERNAME	USER_PASSWORD	PHONE	EMAIL	FIRST_NAME	LAST_NAME	ADDRESS
1	mattma	1234	2349020000	matt@gmail.com	Mattie	Lin	1010 UBC Boulevard, Richmond
2	olivek	4321	1020102222	olive@gmail.com	Ollie	Kiyono	3029 Banff Boulevard, Calgary
3	greenc	2345	1112223333	green@gmail.com	Green	Chun	9876 Curry Boulevard, West Vancouver
4	colleenr	0482	3942014726	colleen@gmail.com	Colleen	Rideout	2366 Main Mall, North Vancouver
5	waltert	6739	5603951093	walter@gmail.com	Khal	Tek	5960 Exchange Union, Vancouver
6	gamma	7588	5637851093	gamma@gmail.com	Steve	Velonic	2345 Kitsilano St, Vancouver
7	fairview	6238	6123765093	fairview@gmail.com	Jeremy	Guerin	9576 Safe Way, Surrey
8	emma	2634	498273647	emma@gmail.com	Emma	Cheng	5960 Professor Boulevard, Coquitlam
9	vanessa	2848	8376251938	vanessa@gmail.com	Vanessa	Riley	3948 Salt Lake, Coquitlam

During:

**Create Customer Account**

**Username:** student

**Password:** ....

**Phone:** 1234

**First Name:** student

**Last Name:** student

**Email:** student@student.com

**Address:** 1234 what way

OK Cancel

Quick Ship

Create Account Driver Login

After:

ShippingUser Table:

	USERNAME	USER_PASSWORD	PHONE	EMAIL	FIRST_NAME	LAST_NAME	ADDRESS
1	mattma	1234	2349020000	matt@gmail.com	Mattie	Lin	1010 UBC Boulevard, Richmond
2	olivek	4321	1020102222	olive@gmail.com	Ollie	Kiyono	3829 Banff Boulevard, Calgary
3	greenc	2345	1112223333	green@gmail.com	Green	Chun	9876 Curry Boulevard, West Vancouver
4	colleenr	0482	3942014726	colleen@gmail.com	Colleen	Rideout	2366 Main Mall, North Vancouver
5	waltert	6739	5603951093	walter@gmail.com	Khal	Tek	5960 Exchange Union, Vancouver
6	gamma	7588	5637851093	gamma@gmail.com	Steve	Velonic	2345 Kitsilano St, Vancouver
7	fairview	6238	6123765093	fairview@gmail.com	Jeremy	Guerin	9576 Safe Way, Surrey
8	emma	2634	498273647	emma@gmail.com	Emma	Cheng	5960 Professor Boulevard, Coquitlam
9	vanessa	2848	8376251938	vanessa@gmail.com	Vanessa	Riley	3948 Salt Lake, Coquitlam
10	student	1234	1234	student@student.com	student	student	1234

Customer Table:

	USERNAME	LOYALTY_MEMBER
1	mattma	0
2	olivek	1
3	greenc	0
4	colleenr	1
5	waltert	1
6	student	0

# DELETE Operation

Query	Where in Code
<pre>DELETE FROM Customer WHERE username = x  DELETE FROM SHIPPINGUSER WHERE username = x  x='student'</pre>	<pre>deleteShippingUser(), deleteCustomer() In database/DatabaseConnectionHandler.java</pre>

Before:

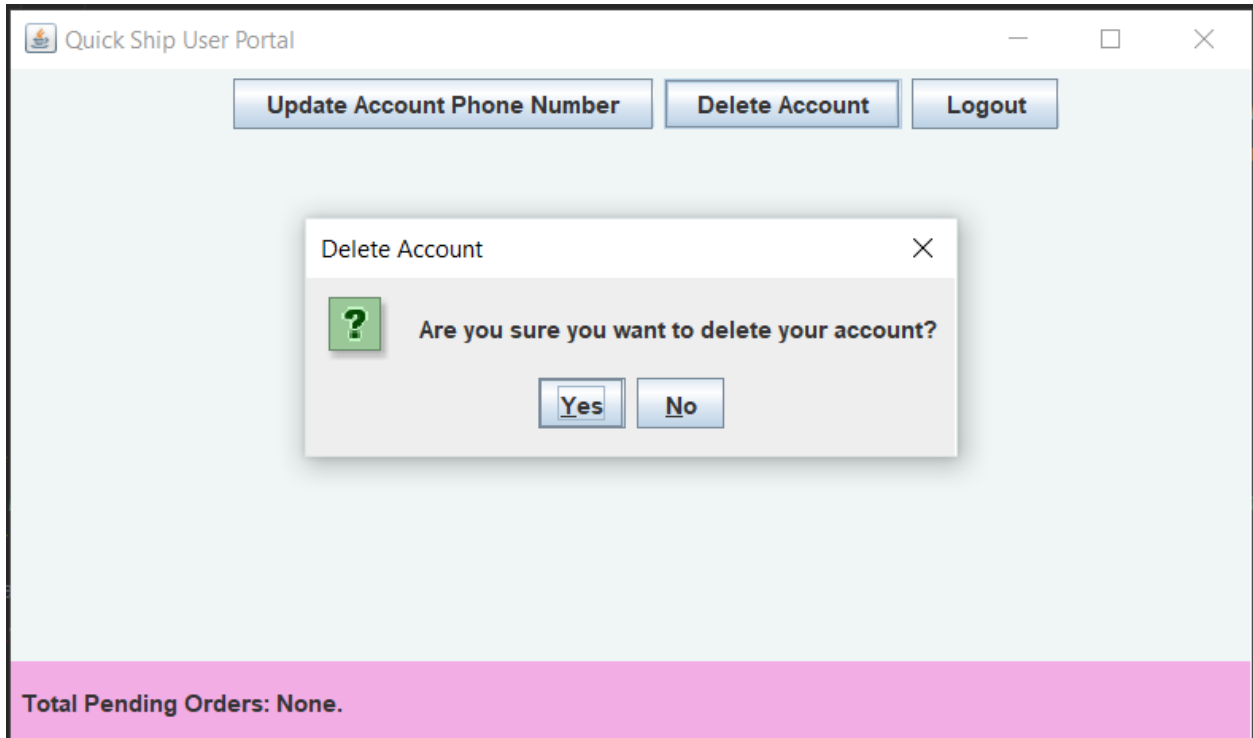
ShippingUser Table:

	USERNAME	USER_PASSWORD	PHONE	EMAIL	FIRST_NAME	LAST_NAME	ADDRESS
1	mattma	1234	2349020000	matt@gmail.com	Mattie	Lin	1010 UBC Boulevard, Richmond
2	olivek	4321	1020102222	olive@gmail.com	Ollie	Kiyono	3829 Banff Boulevard, Calgary
3	greenc	2345	12341234	green@gmail.com	Green	Chun	9876 Curry Boulevard, West Vancouver
4	colleenr	0482	3942014726	colleen@gmail.com	Colleen	Rideout	2366 Main Mall, North Vancouver
5	waltert	6739	5603951093	walter@gmail.com	Khal	Tek	5960 Exchange Union, Vancouver
6	gamma	7588	5637851093	gamma@gmail.com	Steve	Velonic	2345 Kitsilano St, Vancouver
7	fairview	6238	6123765093	fairview@gmail.com	Jeremy	Guerin	9576 Safe Way, Surrey
8	emma	2634	498273647	emma@gmail.com	Emma	Cheng	5960 Professor Boulevard, Coquitlam
9	vanessa	2848	8376251938	vanessa@gmail.com	Vanessa	Riley	3948 Salt Lake, Coquitlam
10	student	1234	1234	student@student.com	student	student	1234

Customer Table:

	USERNAME	LOYALTY_MEMBER
1	mattma	0
2	olivek	1
3	greenc	0
4	colleenr	1
5	waltert	1
6	student	0

During:



After:

ShippingUser Table:

	USERNAME	USER_PASSWORD	PHONE	EMAIL	FIRST_NAME	LAST_NAME	ADDRESS
1	mattma	1234	2349020000	matt@gmail.com	Mattie	Lin	1010 UBC Boulevard, Richmond
2	olivek	4321	1020102222	olive@gmail.com	Ollie	Kiyono	3829 Banff Boulevard, Calgary
3	greenc	2345	12341234	green@gmail.com	Green	Chun	9876 Curry Boulevard, West Vancouver
4	colleenr	0482	3942014726	colleen@gmail.com	Colleen	Rideout	2366 Main Mall, North Vancouver
5	waltert	6739	5603951093	walter@gmail.com	Khal	Tek	5960 Exchange Union, Vancouver
6	gamma	7588	5637851093	gamma@gmail.com	Steve	Velonic	2345 Kitsilano St, Vancouver
7	fairview	6238	6123765093	fairview@gmail.com	Jeremy	Guerin	9576 Safe Way, Surrey
8	emma	2634	498273647	emma@gmail.com	Emma	Cheng	5960 Professor Boulevard, Coquitlam
9	vanessa	2848	8376251938	vanessa@gmail.com	Vanessa	Riley	3948 Salt Lake, Coquitlam

Customer Table:

	USERNAME	LOYALTY_MEMBER
1	mattma	0
2	olivek	1
3	greenc	0
4	colleenr	1
5	waltert	1

# UPDATE Operation

Query	Where in Code
<pre>UPDATE SHIPPINGUSER SET phone=x WHERE username=y  x='greenc' y='12341234'</pre>	updatePhoneNumber() in database/DatabaseConnectionHandler.java

Before:

ShippingUser Table:

	USERNAME	USER_PASSWORD	PHONE	EMAIL	FIRST_NAME	LAST_NAME	ADDRESS
1	mattma	1234	2349020000	matt@gmail.com	Mattie	Lin	1010 UBC Boulevard, Richmond
2	olivek	4321	1020102222	olive@gmail.com	Ollie	Kiyono	3829 Banff Boulevard, Calgary
3	greenc	2345	1112223333	green@gmail.com	Green	Chun	9876 Curry Boulevard, West Vancouver
4	colleenr	0482	3942014726	colleen@gmail.com	Colleen	Rideout	2366 Main Mall, North Vancouver
5	waltert	6739	5603951093	walter@gmail.com	Khal	Tek	5960 Exchange Union, Vancouver
6	gamma	7588	5637851093	gamma@gmail.com	Steve	Velonic	2345 Kitsilano St, Vancouver
7	fairview	6238	6123765093	fairview@gmail.com	Jeremy	Guerin	9576 Safe Way, Surrey
8	emma	2634	498273647	emma@gmail.com	Emma	Cheng	5960 Professor Boulevard, Coquitlam
9	vanessa	2848	8376251938	vanessa@gmail.com	Vanessa	Riley	3948 Salt Lake, Coquitlam
10	student	1234	1234	student@student.com	student	student	1234

During:

Update

?

Please enter your phone number:

12341234

OK

Cancel

After:

	USERNAME	USER_PASSWORD	PHONE	EMAIL	FIRST_NAME	LAST_NAME	ADDRESS
1	mattma	1234	2349020000	matt@gmail.com	Mattie	Lin	1010 UBC Boulevard, Richmond
2	olivek	4321	1020102222	olive@gmail.com	Ollie	Kiyono	3829 Banff Boulevard, Calgary
3	greenc	2345	12341234	green@gmail.com	Green	Chun	9876 Curry Boulevard, West Vancouver
4	colleenr	0482	3942014726	colleen@gmail.com	Colleen	Rideout	2366 Main Mall, North Vancouver
5	waltert	6739	5603951093	walter@gmail.com	Khal	Tek	5960 Exchange Union, Vancouver
6	gamma	7588	5637851093	gamma@gmail.com	Steve	Velonic	2345 Kitsilano St, Vancouver
7	fairview	6238	6123765093	fairview@gmail.com	Jeremy	Guerin	9576 Safe Way, Surrey
8	emma	2634	498273647	emma@gmail.com	Emma	Cheng	5960 Professor Boulevard, Coquitlam
9	vanessa	2848	8376251938	vanessa@gmail.com	Vanessa	Riley	3948 Salt Lake, Coquitlam
10	student	1234	1234	student@student.com	student	student	1234



# SELECT Operation

Query	Where in Code
<pre>SELECT username FROM DRIVER WHERE salary BETWEEN x AND y</pre> x=20000 y=25000	getDriversBySalary() in: database/DatabaseConnectionHandler.java

Before:

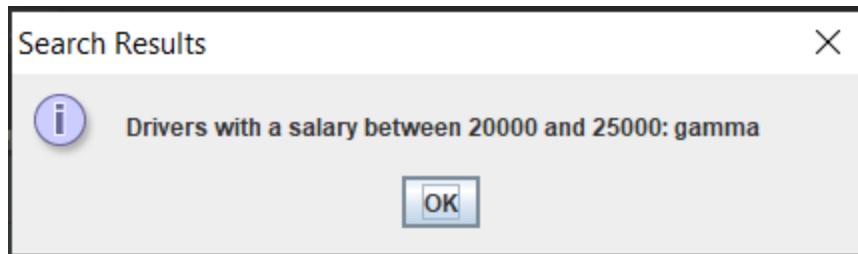
Driver table:

	USERNAME	LICENCE_NUMBER	SALARY	ORDERS_DELIVERED
1	mattma	ABC1234	30000	100
2	waltert	BNG0987	12000	12
3	gamma	MDK2391	20000.5	50
4	fairview	W0E1001	15000	20
5	olivek	TDS2841	35000.5	150

During:

The screenshot shows the 'Quick Ship User Portal' interface. At the top, there are several buttons: 'Update Account Phone Number', 'Delete Account', 'Search Drivers by Salary', and 'Get Driver Information'. Below these, there are more buttons: 'Search Drivers by Salary & Orders', 'Request Driver Information', and 'Logout'. A modal dialog titled 'Search Drivers' is open in the center. It contains a green question mark icon and the text 'Please specify a salary range to search.' Below this text are two input fields: 'From:' with the value '20000' and 'To:' with the value '25000'. At the bottom of the dialog are 'OK' and 'Cancel' buttons. At the bottom of the portal window, there is a section for 'Request Customers with Following Delivery Types:' with checkboxes for 'Regular', 'Overnight', 'Expedited', and 'All Types', followed by a 'Search' button.

After:



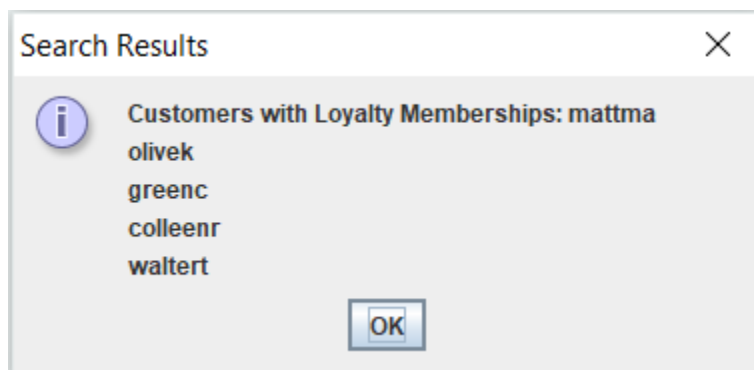
(There are no changes to any tables for this query)

Before:

	USERNAME	LOYALTY_MEMBER
1	mattma	0
2	olivek	1
3	greenc	0
4	colleenr	1
5	waltert	1

During:

After:



	USERNAME	LOYALTY_MEMBER
1	mattma	0
2	olivek	1
3	greenc	0
4	colleenr	1
5	waltert	1

# Query Join Operation

Query	Where in Code
<pre>SELECT * FROM DRIVER d, SHIPPINGUSER s WHERE d.username=s.username AND d.username=x</pre> <p>x= is specified by the user</p>	getDriverInfo() in: database/DatabaseConnectionHandler.java

Before:

Drivers table:

	USERNAME	LICENCE_NUMBER	SALARY	ORDERS_DELIVERED
1	mattma	ABC1234	30000	100
2	waltert	BNG0987	12000	12
3	gamma	MDK2391	20000.5	50
4	fairview	W0E1001	15000	20
5	olivek	TDS2841	35000.5	150

ShippingUser table:

	USERNAME	USER_PASSWORD	PHONE	EMAIL	FIRST_NAME	LAST_NAME	ADDRESS
1	mattma	1234	2349020000	matt@gmail.com	Mattie	Lin	1010 UBC Boulevard, Richmond
2	olivek	4321	1020102222	olive@gmail.com	Ollie	Kiyono	3829 Banff Boulevard, Calgary
3	greenc	2345	1112223333	green@gmail.com	Green	Chun	9876 Curry Boulevard, West Vancouver
4	colleenr	0482	3942014726	colleen@gmail.com	Colleen	Rideout	2366 Main Mall, North Vancouver
5	waltert	6739	5603951093	walter@gmail.com	Khal	Tek	5960 Exchange Union, Vancouver
6	gamma	7588	5637851093	gamma@gmail.com	Steve	Velonic	2345 Kitsilano St, Vancouver
7	fairview	6238	6123765093	fairview@gmail.com	Jeremy	Guerin	9576 Safe Way, Surrey
8	emma	2634	498273647	emma@gmail.com	Emma	Cheng	5960 Professor Boulevard, Coquitlam
9	vanessa	2848	8376251938	vanessa@gmail.com	Vanessa	Riley	3948 Salt Lake, Coquitlam

During:

Quick Ship User Portal

Update Account Phone Number Delete Account Search Drivers by Salary Get Driver Information

Search Drivers by Salary & Orders Request Driver Information Logout

Driver Info

Please enter the driver's username:

mattma

OK Cancel

Request Customers with Following Delivery Types: ☐ Regular ☐ Overnight ☐ Expedited ☐ All Types Search

After:

Search Results

first name: Mattie  
last name: Lin  
licence number: ABC1234  
orders delivered: 100

OK

(No tables changed from this query)

## Query Aggregation Group By Operation

Query	Where in Code
SELECT delivery_type, COUNT(*) from PARCEL GROUP BY delivery_type	countParcelsByDeliveryType() in: database/DatabaseConnectionHandler.java

Before:

Parcel Table:

	PID	PARCEL_SIZE	TRACKING_NUMBER	WEIGHT	DELIVERY_TYPE
1	PNAC56F2	12	TN1234	2	Regular
2	PN0B7869	6	TN1234	1	Regular
3	PNPP1111	10	TN6789	3.5	Expedited
4	9876543210	3	1234567890	1.5	Regular
5	38382919299	3	0123456789	1.5	Overnight
6	2222222222	3	1111111111	1.5	Expedited

During:

Quick Ship User Portal

Update Account Phone Number Delete Account Search Drivers by Salary Get Driver Information

Search Drivers by Salary & Orders Request Driver Information Logout

**Count Parcels By Delivery Type**

Request Customers with Following Delivery Types: ☐ Regular ☐ Overnight ☐ Expedited ☐ All Types Search

After:

Search Results

**delivery type: Expedited count: 2**  
**delivery type: Overnight count: 1**  
**delivery type: Regular count: 3**

OK

(No Tables were changed from this query)

# Query Aggregation with Having Operation

Query	Where in Code
<pre>SELECT orders_delivered,COUNT(*) FROM DRIVER GROUP BY orders_delivered HAVING COUNT(*) &gt; x</pre> x=0	getDriversByDeliveries() in: database/DatabaseConnectionHandler.java

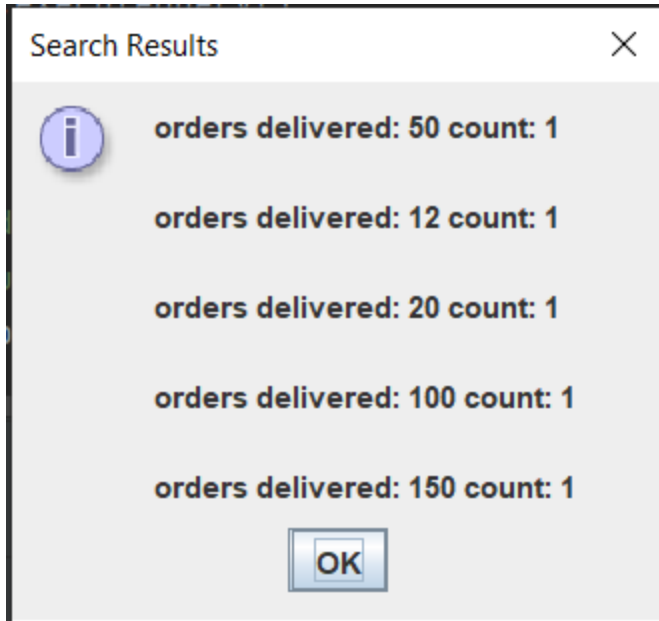
Before:

	USERNAME	LICENCE_NUMBER	SALARY	ORDERS_DELIVERED
1	mattma	ABC1234	30000	100
2	waltert	BNG0987	12000	12
3	gamma	MDK2391	20000.5	50
4	fairview	W0E1001	15000	20
5	olivek	TDS2841	35000.5	150

During:

The screenshot shows the 'Quick Ship User Portal' interface. At the top, there are several buttons: 'Update Account Phone Number', 'Delete Account', 'Search Drivers by Salary', 'Get Driver Information', 'Search Drivers by Salary & Orders', 'Request Driver Information', and 'Logout'. A modal dialog titled 'Search Drivers' is open in the foreground. It contains a green question mark icon, the text 'How many orders do you want to group by?', and a label 'Orders Delivered:' followed by a text input field containing the value '0'. At the bottom of the dialog are 'OK' and 'Cancel' buttons.

After:



(No tables were changed from this query)

## Query Nest Aggregation with Group By Operation

Query	Where in Code
<pre>SELECT P.username, COUNT(*) as total_orders FROM Places P GROUP BY P.username HAVING COUNT(*) &gt; ( SELECT AVG(total_orders) FROM ( SELECT C.username, COUNT(*) as total_orders FROM Places C GROUP BY C.username) )</pre>	<p>groupCustomersByOrdersPlaced() in: database/DatabaseConnectionHandler.java</p>

Before:

Places Table:

	USERNAME	TRACKING_NUMBER
1	mattma	1234567890
2	mattma	0123456789
3	mattma	1111111111

During:

Quick Ship User Portal

Update Account Phone Number   Delete Account   Search Drivers by Salary   Get Driver Information

Search Drivers by Salary & Orders   Request Driver Information   Logout

Count Parcels By Delivery Type   **Group Customer By Orders Placed**

Request Customers with Following Delivery Types: ☐ Regular ☐ Overnight ☐ Expedited ☐ All Types   Search

After:

Search Results

OK

(No tables were changed from this query)

## Query Division Operation

Query	Where in Code
<pre>SELECT p.username FROM Customer p INNER JOIN Places pl ON p.username =</pre>	<pre>getCustomersOrderingEveryDeliveryType() in: database/DatabaseConnectionHandler.java</pre>



```

pl.username INNER JOIN ParcelOrder o ON
pl.tracking_number = o.tracking_number
INNER JOIN HasDeliveryType d ON
o.tracking_number = d.tracking_number
GROUP BY p.username
HAVING COUNT(DISTINCT d.delivery_type)
= (SELECT COUNT(*) FROM DeliveryType);

```

Before:

Parcel Order:

	TRACKING_NUMBER	DESTINATION
1	TN1234	Vancouver
2	TN6789	Richmond
3	TN4321	Surrey
4	TN9876	Richmond
5	TN1111	Vancouver
6	1234567890	New York
7	0123456789	New York
8	1111111111	New York

Parcel Table:

	PID	PARCEL_SIZE	TRACKING_NUMBER	WEIGHT	DELIVERY_TYPE
1	PNAC56F2	12	TN1234	2	Regular
2	PN0B7869	6	TN1234	1	Regular
3	PNPP1111	10	TN6789	3.5	Expedited
4	9876543210	3	1234567890	1.5	Regular
5	38382919299	3	0123456789	1.5	Overnight
6	2222222222	3	1111111111	1.5	Expedited

Has Delivery Type Table:

	DELIVERY_TYPE	TRACKING_NUMBER
1	Regular	1234567890
2	Overnight	0123456789
3	Expedited	1111111111

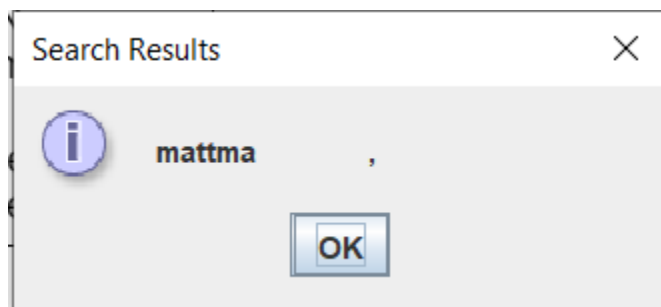
Places Table:

	USERNAME	TRACKING_NUMBER
1	mattma	1234567890
2	mattma	0123456789
3	mattma	1111111111

During:

Request Customers with Following Delivery Types: ☐ Regular ☐ Overnight ☐ Expedited ☒ All Types

After:



(No tables were changed from this query)