



CPSC 304 Project Cover Page

Milestone #: **4**

Date: **April 5, 2023**

Group #: **45**

Name	Student Number	CS ID	Email
Flora Zhou	17977596	h5t4s	flozhou@student.ubc.ca
Jasvir Sandhu	26638189	b8f8u	jsand01@students.cs.ubc.ca
Payam Forouzandeh	51597292	l5n2p	payamfz@student.ubc.ca

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

Our database project models a public transit system and tries to capture necessary information about transit lines, vehicles, commuters and employees. It allows for administrative supervision, rider account and payments management, and real-time schedule information for client-side applications.

In our application, users may perform a multitude of operations from registering for a new UserAccount to keep track of their transit card details, obtaining information about different entities in our database, and more.

Fixes in CREATE TABLE statements:

5. Line_Has_Station

- order changed to 'stationOrder' because order is a reserved keyword

7. Card_Used_At_Station:

- DATETIME type changed to TIMESTAMP

9. Supervises_Line:

- name changed to lineName
- Operation Control Centre changed to OperationControlCenter

27. Services:

- date renamed to serviceDate because date is a reserved keyword

Line_Has_Station:

- incorrect foreign key datatype for lineName

StationLine_Scheduled_For_Timing

- Foreign Key should have referenced Line_Has_Station

Drives:

- Foreign key mistakenly references Vehicle and Line separately whereas it should reference Operates_In

New_Vehicle

- moved the content of Bus, Ferry, Train relations to Vehicle

All.

- removed ON UPDATE CASCADE since Oracle does not support it
- changed TIME datatype to TIMESTAMP or DATE because there is no TIME datatype in Oracle
- changed BOOLEAN datatype to NUMBER(1) with value 0/1 since BOOLEAN not in Oracle
- Removed the precision from the INT/INTEGER data types or changed them to NUMBER

Schema changes:

5. Line_Has_Station

- order changed to 'stationOrder' because order is a reserved keyword

27. Services:

- date renamed to serviceDate because date is a reserved keyword
- The relation is to make it easier
- Added ServiceID as a key since a technician can perform multiple services on the same vehicle

29. Drives:

- VehicleID foreign key references Bus instead of Vehicle to make it easier

New_Vehicle:

- Added new Vehicle relation as parent of different types and moved the content of Bus, Ferry, Train relations to Vehicle so that the relations can reference this table alone

Bus, Ferry, Train:

- changed these relations to only reference the Vehicle table.

Driver, Engineer, Technician, Administrator:

- Removed the redundant Employee attributes since they are already in the Employee table.

Employee_Age:

- Incorporated it into Employee using the dateOfBirth attribute for better performance. Age can be calculated from DoB by functions.

Employee_Works_Under

- Combined with OperationControlCenter_Works_Under
- Schedule datatype required longer length of string

Service_Cost

- Added a serviceID so not to use description as foreign ID.

Pass_Loads_To

- cardID was missing from the Primary Key. The primary key is now (type, cardID)

Queries:

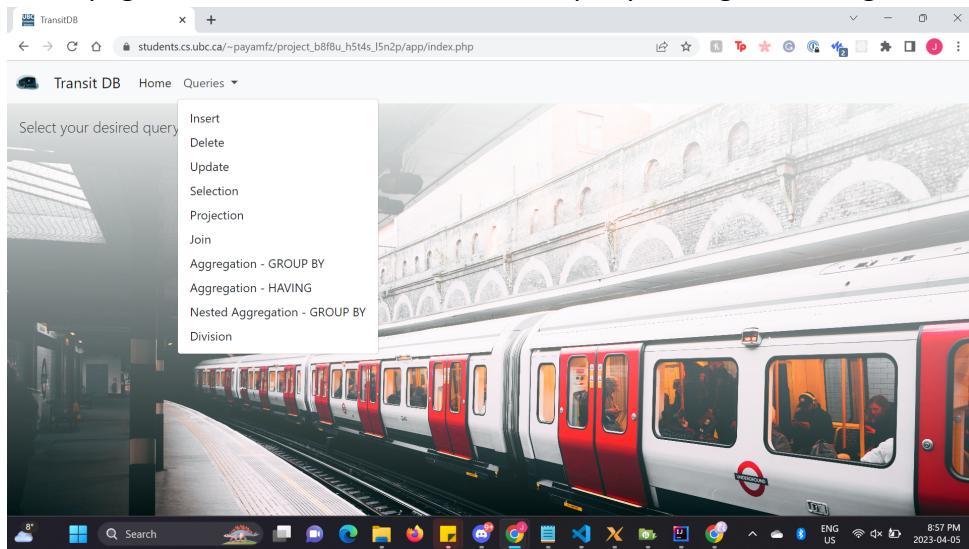
- INSERT : Jasvir - UserAccount & UserAccount_Registers (register new account)
- DELETE : Flora - UserAccount, Card, and Pass
- UPDATE : Jasvir - UserAccount (update password)
- Selection : Jasvir - StationLine_Scheduled_For_Timing (weekend/holiday lines)
- Projection : Flora - Project data from any table, any column
- Join : Payam - Line->Drives->Driver (Name of all drivers who work on a given line)
- Aggregation with GROUP BY : Flora - Line_Has_Station & Station (find the number of stations that are part of a line)
- Aggregation with HAVING : Flora - Line_Has_Station (find the total amount of transit lines that pass through a station, only for stations that are passed through more than one line)
- Nested aggregation with GROUP BY : Jasvir - Pass_Loads_To (most purchased pass type)
- Division : Payam - Vehicle & Services (technicians who have serviced all the buses)

SQL Queries and Location:

Query	Function name	File name	Line Number
INSERT	handleInsertRequest()	oracle-test.php	324-325
DELETE	handleTupleDeletion()	oracle-test.php	423-429
UPDATE	handleUpdateRequest()	oracle-test.php	281
Selection	handleCountRequest()	oracle-test.php	337
Projection	handleProjectTableNamesRequest() handleProjectionColumns()	oracle-test.php	350 384
Join		q_join.php	55-61
Aggregation with Group By	handleGroupByQuery()	oracle-test.php	443-446
Aggregation with Having	handleHavingQuery()	oracle-test.php	471-474
Nested Aggregation with Group By	handleNestedAggregation()	oracle-test.php	501
Division		q_division.php	34-43

Query Demos:

Home page - user can choose their desired query through the navigation bar



INSERT Operation

Before:

The screenshot shows a web browser window titled "TransitDB". The URL is "students.cs.ubc.ca/~payamfz/project_b8fbu_h5t4s_l5n2p/app/pages/q_insert.php". The page contains an "Insert Query" section with a "Register new user" form. The form has fields for "Username" and "Password", both currently empty. Below the form is a table with columns: #, Username, Password, Registration Date, and Center ID. The table lists five existing users:

#	Username	Password	Registration Date	Center ID
1	flozhou	bbtenthusiast	01-FEB-23	A8H6G2H837
2	raymondng	bestprofever1	15-FEB-23	HW7NK2YD8K
3	jsand01	b0nfl3g3nd	16-FEB-23	HW7NK2YD8K
4	payamfz	anythingEnthusiast1	17-FEB-23	H388HDLNaz
5	ansonchung	bestTAever	19-FEB-23	7H20JD6ZGA

During:

The screenshot shows the same web browser window. The "Username" field now contains "insertdemo" and the "Password" field contains "testing123". A red circle highlights the "Register" button. The table below still shows the same five users as before.

#	Username	Password	Registration Date	Center ID
1	flozhou	bbtenthusiast	01-FEB-23	A8H6G2H837
2	raymondng	bestprofever1	15-FEB-23	HW7NK2YD8K
3	jsand01	b0nfl3g3nd	16-FEB-23	HW7NK2YD8K
4	payamfz	anythingEnthusiast1	17-FEB-23	H388HDLNaz
5	ansonchung	bestTAever	19-FEB-23	7H20JD6ZGA

After:

The screenshot shows the same web browser window. The "Insert Query" section is empty. The table now includes a new row at the top, highlighted with a red box, representing the newly inserted user:

#	Username	Password	Registration Date	Center ID
1	insertdemo	testing123	05-APR-23	A8H6G2H837
2	flozhou	bbtenthusiast	01-FEB-23	A8H6G2H837
3	raymondng	bestprofever1	15-FEB-23	HW7NK2YD8K
4	jsand01	b0nfl3g3nd	16-FEB-23	HW7NK2YD8K
5	payamfz	anythingEnthusiast1	17-FEB-23	H388HDLNaz
6	ansonchung	bestTAever	19-FEB-23	7H20JD6ZGA

DELETE Operation

Before:

The screenshot shows a web application interface with two tables: 'Cards' and 'Passes'.
Cards Table:

#	Card ID	Stored Value
1	10C154982E	.75
2	31333C2090	20
3	519552E6F3	13.6

Passes Table:

#	Pass Type	Card ID	Purchase Date
1	Monthly	10C154982E	01-MAR-23
2	Monthly	31333C2090	01-MAR-23
3	Monthly	519552E6F3	01-MAR-23

During:

The screenshot shows the 'Transit DB' application with a 'Deletion Query' form and a 'Cards' table.
Deletion Query Form:

Type a Card ID below to be deleted.
Notice that the passes loaded to that card will be deleted as well.

Card ID: **10C154982E** (highlighted with a red oval)

Delete

Cards Table:

#	Card ID	Stored Value
1	10C154982E	.75
2	31333C2090	20
3	519552E6F3	13.6

After:

The screenshot shows the 'Transit DB' application after the deletion of Card ID 10C154982E.
Deletion Query Form:

Type a Card ID below to be deleted.
Notice that the passes loaded to that card will be deleted as well.

Card ID:

Delete

Cards Table:

#	Card ID	Stored Value
1	31333C2090	20
2	519552E6F3	13.6

Passes Table:

#	Pass Type	Card ID	Purchase Date
1	Monthly	31333C2090	01-MAR-23
2	Monthly	519552E6F3	01-MAR-23

UPDATE Operation

Before:

The screenshot shows a web application window titled "Update Query". Inside, there's a "Change password" section with fields for "Username" (empty), "Old Password" (empty), and "New Password" (empty). Below this is a table with columns: #, Username, Password, Registration Date, and Center ID. The data is as follows:

#	Username	Password	Registration Date	Center ID
1	insertdemo	testing123	05-APR-23	A8H6G2H837
2	flozhou	bbtenthustiast	01-FEB-23	A8H6G2H837
3	raymondng	bestprofever1	15-FEB-23	HW7NK2YD8K
4	jsand01	bconlegend	16-FEB-23	HW7NK2YD8K

During:

The screenshot shows the same "Update Query" page. The "Old Password" field now contains "*****" and has a red circle drawn around it. The "New Password" field also contains "*****". A red arrow points to the "Password" column of the first row in the table, indicating the target for the update.

#	Username	Password	Registration Date	Center ID
1	insertdemo	testing123	05-APR-23	A8H6G2H837
2	flozhou	bbtenthustiast	01-FEB-23	A8H6G2H837
3	raymondng	bestprofever1	15-FEB-23	HW7NK2YD8K
4	jsand01	bconlegend	16-FEB-23	HW7NK2YD8K

After:

The screenshot shows the final state of the "Update Query" page. The "Old Password" field is empty again. The "New Password" field contains "helloworld". A red circle highlights the "Password" column of the first row in the table, which now shows "helloworld".

#	Username	Password	Registration Date	Center ID
1	insertdemo	helloworld	05-APR-23	A8H6G2H837
2	flozhou	bbtenthustiast	01-FEB-23	A8H6G2H837
3	raymondng	bestprofever1	15-FEB-23	HW7NK2YD8K
4	jsand01	bconlegend	16-FEB-23	HW7NK2YD8K

Selection Operation

Before:

The screenshot shows a web browser window titled "TransitDB". The URL is "students.cs.ubc.ca/~payamflz/project_b8ff0u_h514s_l5n2p/app/pages/q_selection.php". The page has a header "Transit DB Home Queries". Below it is a "Selection Query" section with the heading "Super-active lines!". It says "Filter the transit lines by their availability over weekends or holidays." There are two checkboxes: "Active on Weekends" (unchecked) and "Active on Holidays" (unchecked). A blue "Filter" button is present. Below this is a table with columns "# Line name". The data rows are: 1 CanadaLine, 2 SeaBus, 3 Bus4, 4 MillenniumLine. At the bottom of the browser window, the taskbar shows various icons and the date/time as 8:38 PM 2023-04-05.

During:

This screenshot is similar to the previous one, showing the "Selection Query" page for "Super-active lines!". The "Active on Weekends" checkbox is now checked and highlighted with a red oval. The "Active on Holidays" checkbox remains unchecked. The table below shows the same four lines: CanadaLine, SeaBus, Bus4, and MillenniumLine. The browser's taskbar and system tray are visible at the bottom.

After:

This screenshot shows the results of the selection operation. The "Active on Weekends" checkbox is checked and highlighted with a red oval. The "Active on Holidays" checkbox is unchecked. The table now only lists the two lines that were active on weekends: CanadaLine and MillenniumLine. The browser's taskbar and system tray are visible at the bottom.

Join Operation

Before:

The screenshot shows a web browser window titled "TransitDB". The URL is "students.cs.ubc.ca/~payamfz/project_b8f8u_h5t4s_l5n2p/app/pages/q_join.php". The page has a header "Transit DB" and "Home" and "Queries" links. Below the header is a section titled "Join Query" with a sub-section "Drivers of a transit line". It says "Select a transit line to see the name and email of the drivers operating in that line." There is a dropdown menu labeled "Line" with "Select..." and a list of transit lines: Bus14, Bus4, Bus44, Canadoline, Expoline, MillenniumLine, and SeaBus. A "Submit" button is at the bottom. Below the dropdown is a table with columns "#", "Driver Name", and "Driver Email".



During:

The screenshot shows the same "Join Query" page. The "Line" dropdown now has "Select..." highlighted. A list of transit lines is displayed: Bus14, Bus4, Bus44, Canadoline, Expoline, MillenniumLine, and SeaBus. The "Bus44" option is currently selected, indicated by a blue highlight. A "Submit" button is at the bottom.



After:

The screenshot shows the results of the search. The "Line" dropdown now has "Bus44" selected. The table below shows two rows of data: one for Romeo Glijuvanni with email glijuvanni@gmail.com and another for Willy Joseph with email joseph@gmail.com. A red oval highlights the first row of the table.

#	Driver Name	Driver Email
1	Romeo Glijuvanni	glijuvanni@gmail.com
	Willy Joseph	joseph@gmail.com



Division Operation

Before:

The screenshot shows a web browser window titled "TransitDB". The URL is "students.cs.ubc.ca/~payamfz/project_b8f8u_h5t4s_l5n2p/app/pages/q_division.php". The page is titled "Division Query" and contains a section for "Experienced technicians". The instructions say: "Find the name and salary of the technicians who have serviced all the buses." Below this is a "Find" button. A table below the button has columns "#", "Technician Name", and "Technician Salary". The table is currently empty.

During:

User clicks Find button.

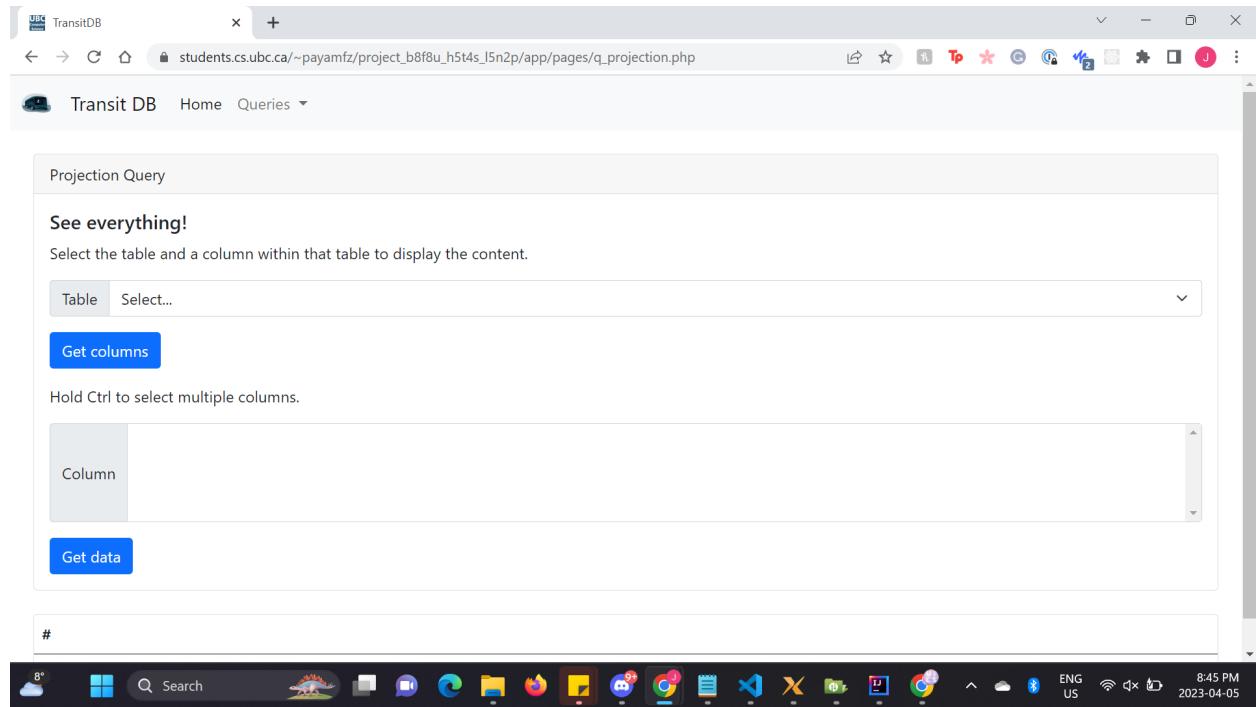
After:

The screenshot shows the same web browser window after the user has clicked the "Find" button. The table now displays two rows of data. The first row has #1, Technician Name "Anna Rembert", and Technician Salary "51000". The second row has #2, Technician Name "Oscar Arthur", and Technician Salary "60000". A red oval highlights the entire table area.

#	Technician Name	Technician Salary
1	Anna Rembert	51000
2	Oscar Arthur	60000

Projection Operation

Before:



TransitDB

students.cs.ubc.ca/~payamfz/project_b8f8u_h5t4s_l5n2p/app/pages/q_projection.php

Transit DB Home Queries

Projection Query

See everything!

Select the table and a column within that table to display the content.

Table Select...

Get columns

Hold Ctrl to select multiple columns.

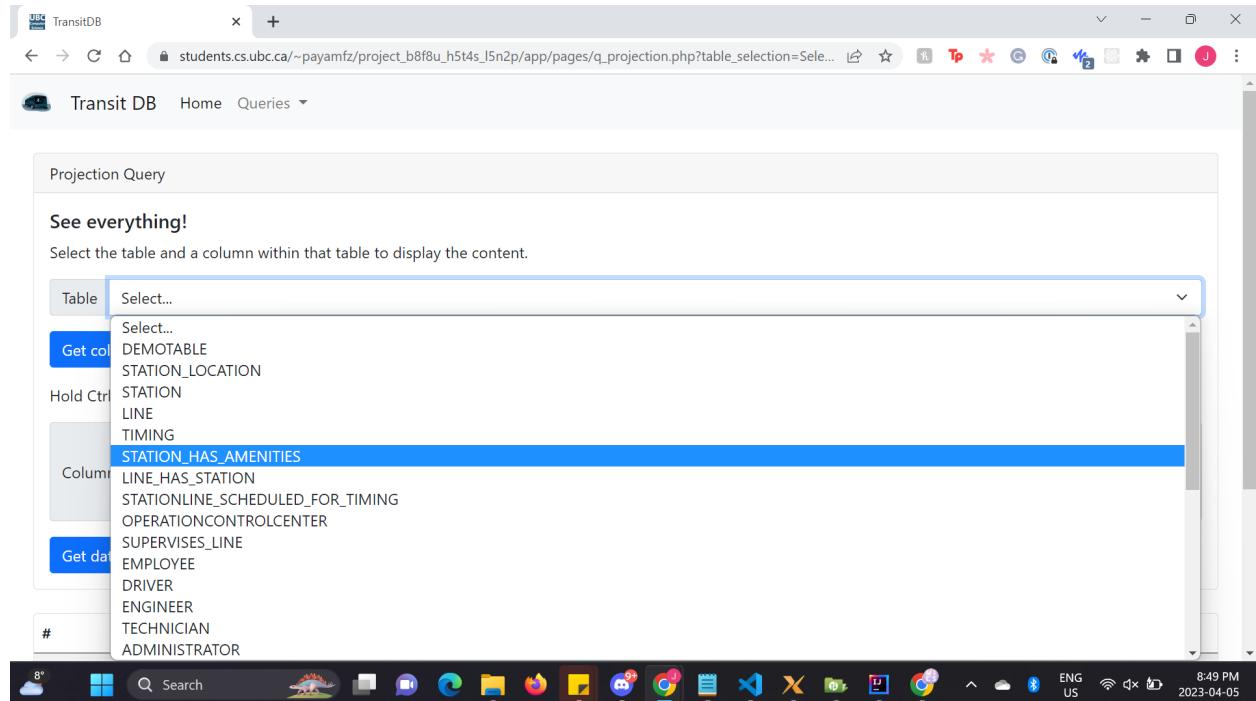
Column

Get data

#

8:45 PM 2023-04-05

During:



TransitDB

students.cs.ubc.ca/~payamfz/project_b8f8u_h5t4s_l5n2p/app/pages/q_projection.php?table_selection=Sele...

Transit DB Home Queries

Projection Query

See everything!

Select the table and a column within that table to display the content.

Table Select...

Get columns

Hold Ctrl

Column

Get data

#

STATION_HAS_AMENITIES

DEMOTABLE
STATION_LOCATION
STATION
LINE
TIMING

LINE_HAS_STATION
STATIONLINE_SCHEDULED_FOR_TIMING
OPERATIONCONTROLCENTER
SUPERVISES_LINE

EMPLOYEE
DRIVER
ENGINEER
TECHNICIAN
ADMINISTRATOR

8:49 PM 2023-04-05

TransitDB

students.cs.ubc.ca/~payamfz/project_b8f8u_h5t4s_l5n2p/app/pages/q_projection.php?table_selection=STA...

Projection Query

See everything!

Select the table and a column within that table to display the content.

Table Select...

Get columns

Hold Ctrl to select multiple columns.

Column	AMENITYNAME STATIONID COUNT DESCRIPTION
--------	--

Get data

#

8° Search 9:21 PM 2023-04-05

TransitDB

students.cs.ubc.ca/~payamfz/project_b8f8u_h5t4s_l5n2p/app/pages/q_projection.php?table_selection=Sele...

Get columns

Hold Ctrl to select multiple columns.

Column

Get data

#	AMENITYNAME	STATIONID	DESCRIPTION
1	Starbucks	456	Coffeeshop
2	Subway	456	Fast Food
3	Ticket Booth	456	Purchase transit tickets
4	Vending Machine	105	Fast Food
5	Ticket Booth	105	Purchase transit tickets

8° Search 9:22 PM 2023-04-05

Aggregation with Group By Operation

Before:

The screenshot shows a Microsoft Edge browser window with the title "TransitDB". The address bar contains the URL "students.cs.ubc.ca/~payamfz/project_b8f8u_h5t4s_l5n2p/app/pages/q_groupAggregation.php". The main content area is titled "Aggregation Query with GROUP BY" and contains a section titled "Stations in Line" with the sub-instruction "For each transit line, this will find the number of stations that are part of that line." Below this is a blue "Find" button. At the bottom is a table with three columns: "#", "Line Name", and "Station Count". The table currently has one row with data: #1, Line Name Bus4, and Station Count 1.

During:

User clicks Find button.

After:

The screenshot shows the same Microsoft Edge browser window after the user has clicked the "Find" button. The table now displays six rows of data, each representing a transit line and its station count. The first five lines (Bus4, CanadaLine, ExpoLine, SeaBus, Bus14) have a station count of 1, while the MillenniumLine has a station count of 2. A red oval highlights the entire table area.

#	Line Name	Station Count
1	Bus4	1
2	CanadaLine	1
3	ExpoLine	1
4	SeaBus	1
5	Bus14	2
6	MillenniumLine	1

Aggregation with Having Operation

Before:

The screenshot shows a web browser window titled "TransitDB" with the URL "students.cs.ubc.ca/~payamfz/project_b8f8u_h5t4s_l5n2p/app/pages/q_havingAggregation.php". The page displays an "Aggregation Query with HAVING" section for "Busy Stations". The query description states: "For each station, find the total number of transit lines that pass through them. Only show those stations that more than one line pass through them." Below this is a "Find" button. A table header is shown with columns "#", "Station ID", and "Lines passing through". The taskbar at the bottom shows various application icons and the date/time "2023-04-05 9:02 PM".

During:

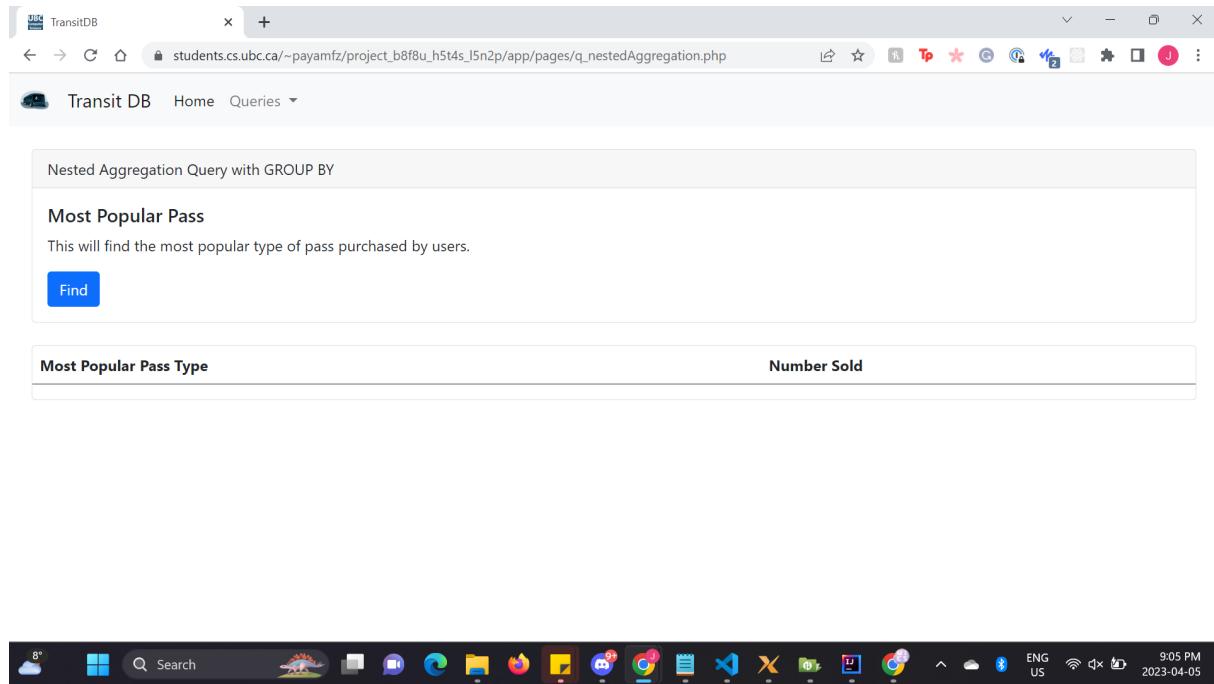
User clicks Find button.

After:

The screenshot shows the same web browser window after the "Find" button was clicked. The results table now contains two rows: one for Station ID 456 with 3 lines passing through, and another for Station ID 123 with 2 lines passing through. Both rows are highlighted with a red rounded rectangle. The taskbar at the bottom shows various application icons and the date/time "2023-04-05 9:02 PM".

Nested Aggregation with Group By Operation

Before:



Nested Aggregation Query with GROUP BY

Most Popular Pass

This will find the most popular type of pass purchased by users.

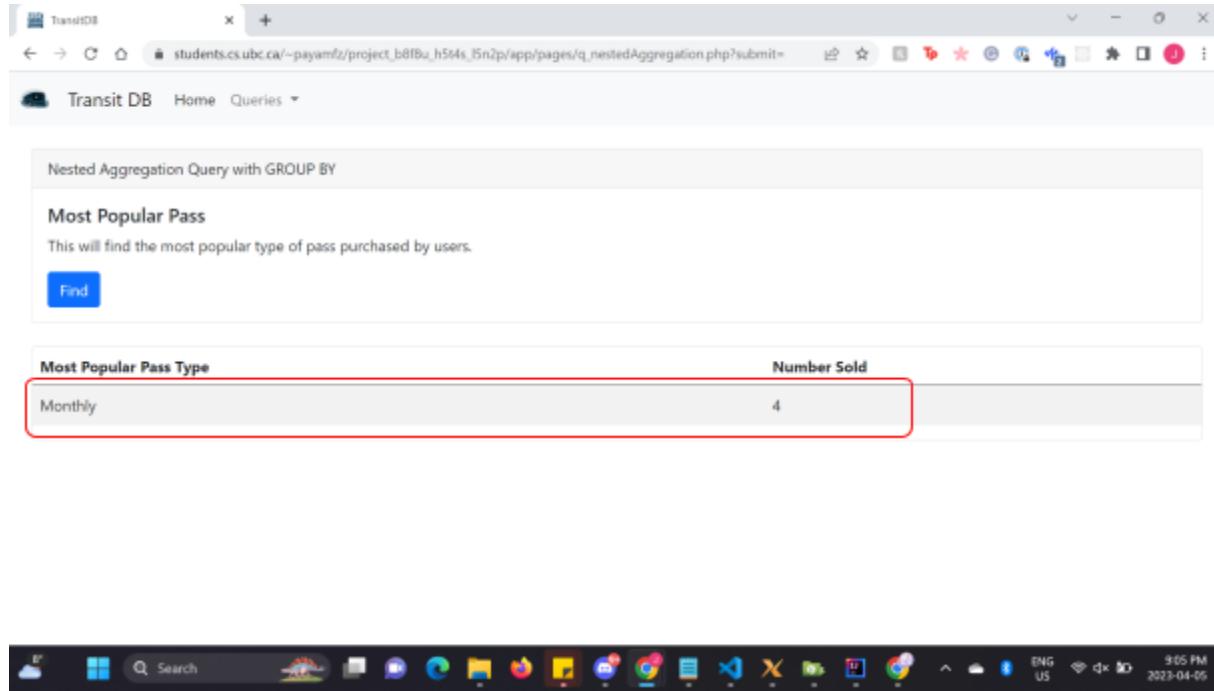
Find

Most Popular Pass Type	Number Sold
Monthly	4

During:

User clicks Find button.

After:



Nested Aggregation Query with GROUP BY

Most Popular Pass

This will find the most popular type of pass purchased by users.

Find

Most Popular Pass Type	Number Sold
Monthly	4