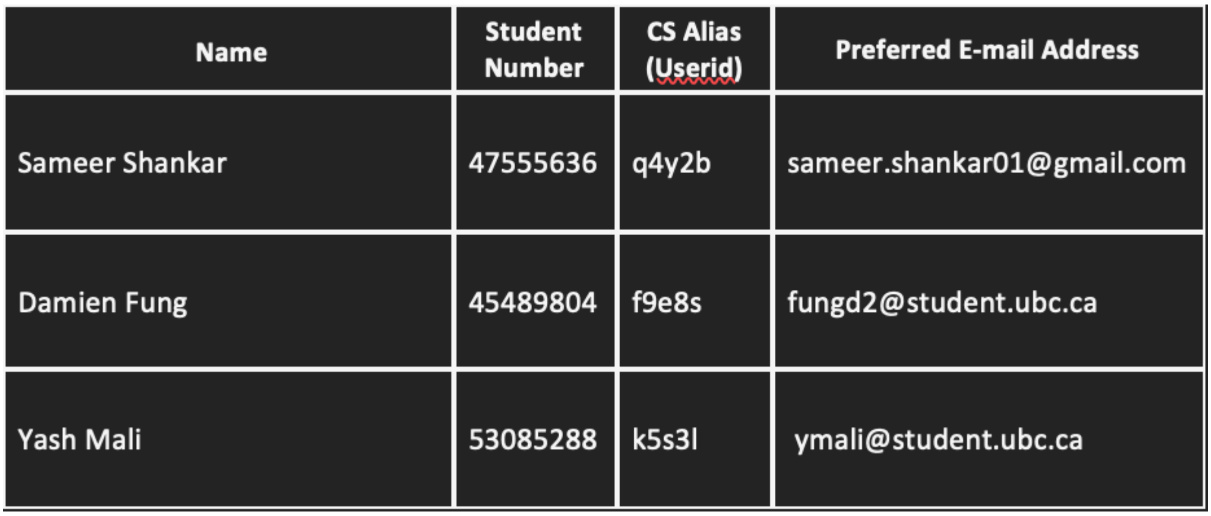
CPSC 304 Project Cover Page

Milestone #: 4

Date: 05/04/2024

Group Number: 23



Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Sameer Shankar	47555636	q4y2b	sameer.shankar01@gmail.com
Damien Fung	45489804	f9e8s	fungd2@student.ubc.ca
Yash Mali	53085288	k5s3l	ymali@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

Deliverable 2

2a. The script is of non-trivial size (filled with many examples per table)

2b. The script is runnable as is

Note: farmers_full is the SQL script which we would've used had there not been a space quota exceeded error (as we showed Julian on Friday). farmers_partial is what we ended up using so that space quota is not exceeded

Deliverable 3

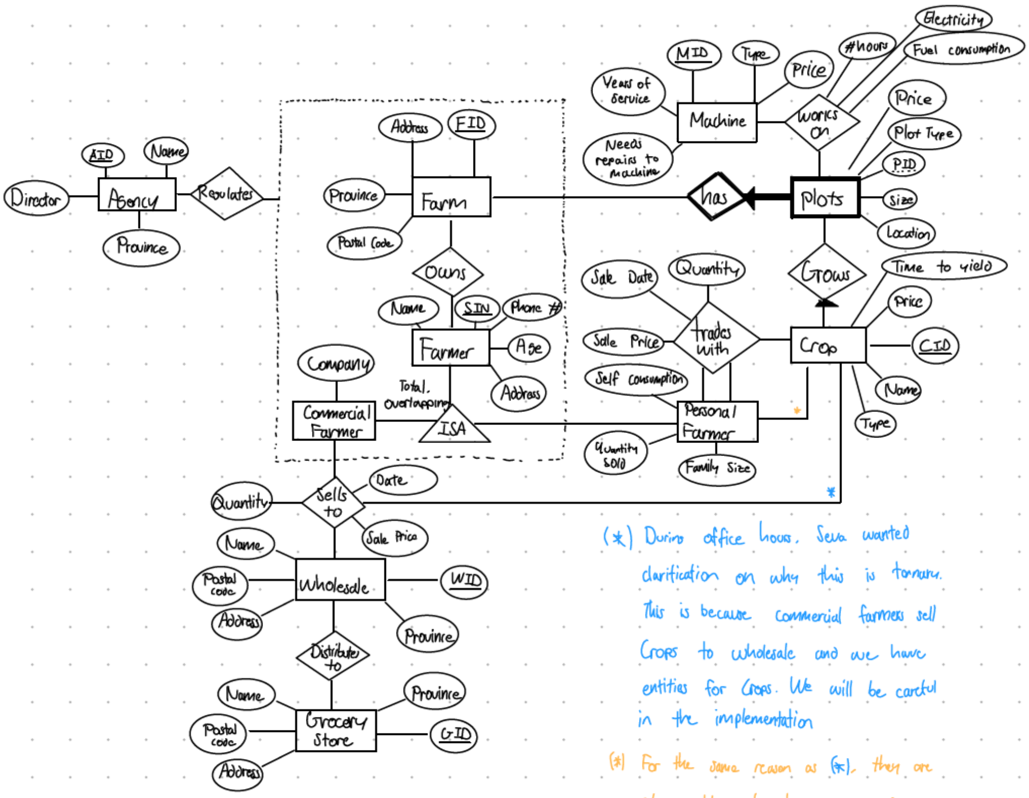
3a. The theme of this project is agricultural farming. The database is used to keep track of the agricultural distribution in a country. The farmer and their holdings (farms, plots, machines, etc.) are captured by the database. Furthermore, whom the farmers sell to (i.e., other personal/subsistence farmers or wholesales) and the overseeing regulatory agencies are also included. The main motivation/purpose of the project was to model the agricultural industry in a way that the Ministry of Agriculture (the parent regulatory body) can keep record of farming for a given country.

3bi. The final relational schema differed somewhat compared to the schema turned in previously; a lot of the TA's feedback has now been incorporated;

- sells_to functional dependencies primary keys were fixed (originally backwards)
- foreign key constraints (NOT NULL) were added to the SQL
- BCNF foreign key constraints were added
- ON DELETE CASCADE was added where necessary

Furthermore, some data types were corrected, SQL syntax was fixed and other important changes were made.

3ci. Final ER Diagram (just for reference)



(*) During office hours, Seva wanted clarification on why this is ternary. This is because commercial farmers sell Crops to wholesale and we have entities for Crops. We will be careful in the implementation

(*) For the same reason as (*), they are also able to trade produce for produce with other personal farmers. I.e. Bob gives Alice strawberries for blueberries

farm(fid: VARCHAR(20), address: VARCHAR(30))
farmAddress(address: VARCHAR(30), postal_code: CHAR(7))
farmProvince(province: CHAR(2), postal_code: CHAR(7))

farmer(sin: CHAR(9), name: VARCHAR(20), phone: INTEGER, age: INTEGER, address: VARCHAR(30))

owns(**fid: VARCHAR(20)**, **sin: CHAR(9)**)

commercial_farmer_nc(name: VARCHAR(20), phone: INTEGER, company: VARCHAR(20))
commercial_farmer(**sin: CHAR(9) NOT NULL**, name: VARCHAR(20), age: INTEGER, address: VARCHAR(30))
commercial_farmer_a(address: VARCHAR(30), company: VARCHAR(20))

sells_to(**wid: VARCHAR(20) NOT NULL**, **sin: CHAR(9) NOT NULL**, **cid: VARCHAR(20) NOT NULL**, sale_date: DATE)
sells_to_sale_date(quantity: INTEGER, sale_date: DATE)
sells_to_sale_quantity(quantity: INTEGER, sale_price: REAL)

WholesaleLocation(postal_code CHAR(7), province CHAR(2))
Wholesale(postal_code CHAR(7), wid VARCHAR(20), address VARCHAR(30), name VARCHAR(20))

GroceryLocation(postal_code CHAR(7), province CHAR(2))
Grocery(postal_code CHAR(7), gid VARCHAR(20), address VARCHAR(30), name VARCHAR(20))

AgencyLocation(name VARCHAR(20), province CHAR(2), director VARCHAR(20))
Agency(name VARCHAR(20), province CHAR(2), aid VARCHAR(20))

has_plots_location(location VARCHAR(20), size: VARCHAR(20), price: REAL)
has_plots_type(plot_type VARCHAR(20), location: VARCHAR(20))
has_plots(**fid: VARCHAR(20) NOT NULL**, p_i_d: VARCHAR(20), plot_type: VARCHAR(20))

trades_with(**farm_trader_A_SIN CHAR(9) NOT NULL**, **farm_trader_B_SIN CHAR(9) NOT NULL**, **cid VARCHAR(20) NOT NULL**, sale_date DATE)
trades_with_quantity(quantity INTEGER, sale_price REAL)
trades_with_sale_date(sale_date DATE, quantity INTEGER)

grows_crop_type(type VARCHAR(20), price REAL)
grows_crop(cid VARCHAR(20), name VARCHAR(20), type: VARCHAR(20), Time to Yield: REAL, **pid: VARCHAR(20) NOT NULL**, **fid: VARCHAR(20) NOT NULL**)
grows_crop_time(Time to Yield REAL, price REAL)

personal_farmer_family_size_sc(family_size: INTEGER, self_consumption: INTEGER)
personal_farmer_family_size_qs(family_size: INTEGER, quantity_sold: INTEGER)
personal_farmer(**sin: CHAR(9) NOT NULL**, name: VARCHAR(20), phone: INTEGER, age: INTEGER, address: VARCHAR(30), family_size: INTEGER)

works_on_num_hours_fc(num_hours: REAL, fuel_consumption: REAL)
works_on_num_hours_ec(num_hours: REAL, electricity_consumption: REAL)
works_on(**mid: VARCHAR(20) NOT NULL**, **pid: VARCHAR(20) NOT NULL**, **fid: VARCHAR(20) NOT NULL**, num_hours: REAL)

machine_type(type: VARCHAR(20), price: REAL)
machine_yos(years_of_service: INTEGER, needs_repair: BOOL)
machine(mid: VARCHAR(20), type: VARCHAR(20), years_of_service: INTEGER)

Regulates (**aid: VARCHAR(20), fid: VARCHAR(20), sin: CHAR(9)**)

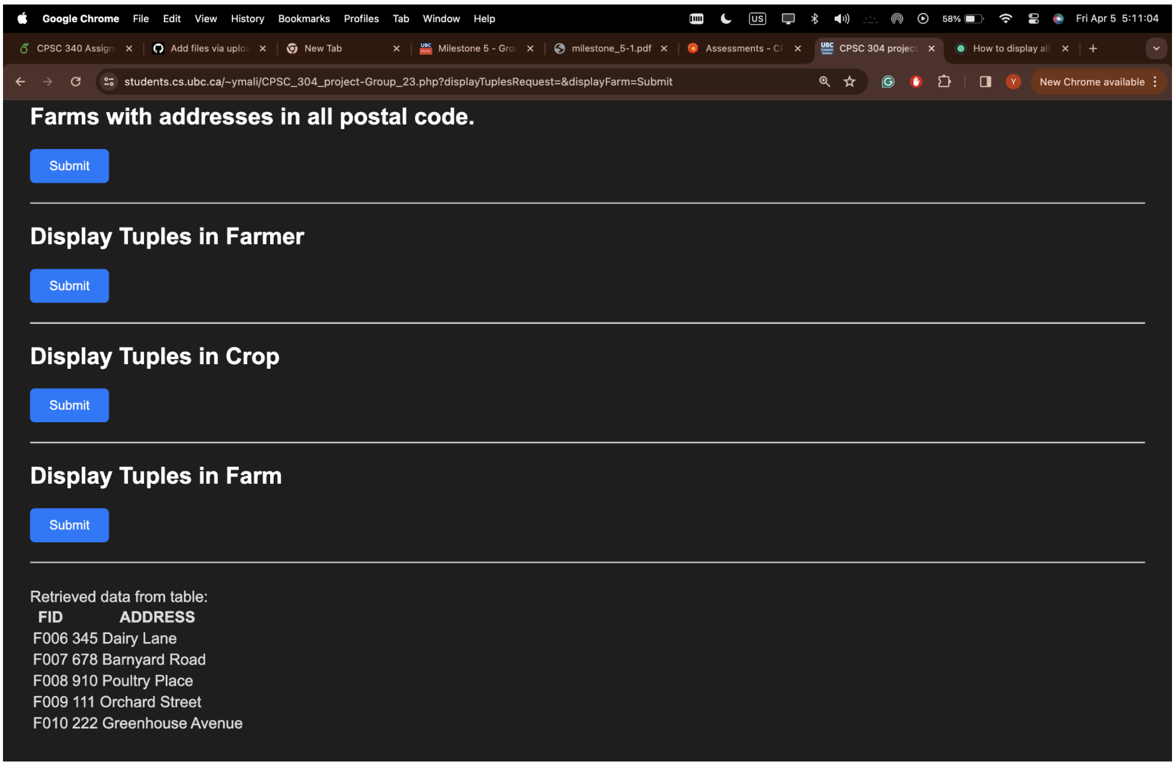
Distributes (**wid: VARCHAR(20), gid: VARCHAR(20)**)

3d. SQL queries and where they can be found in the code. They are all in GRADE_THIS.php.

INSERT Operation - Line 540
DELETE Operation - Line 551
UPDATE Operation - Line 525
Selection - Line 678
Projection - Line 575
Join - Line 559 to 563
Aggregation with Group By - Line 586
Aggregation with Having - Line 599
Nested Aggregation with Group By - Line 612 to 618
Division - Line 630 to 642

3e.

Before - The tuples in farm


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
58%
Fri Apr 5 5:11:04
CPSC 340 Assign
Add files via uplo
New Tab
Milestone 5 - Gro
milestone_5-1.pdf
Assessments - CF
CPSC 304 project
How to display all
students.cs.ubc.ca/~ymali/CPSC_304_project-Group_23.php?displayTuplesRequest=&displayFarm=Submit
New Chrome available

Farms with addresses in all postal code.

Submit

Display Tuples in Farmer

Submit

Display Tuples in Crop

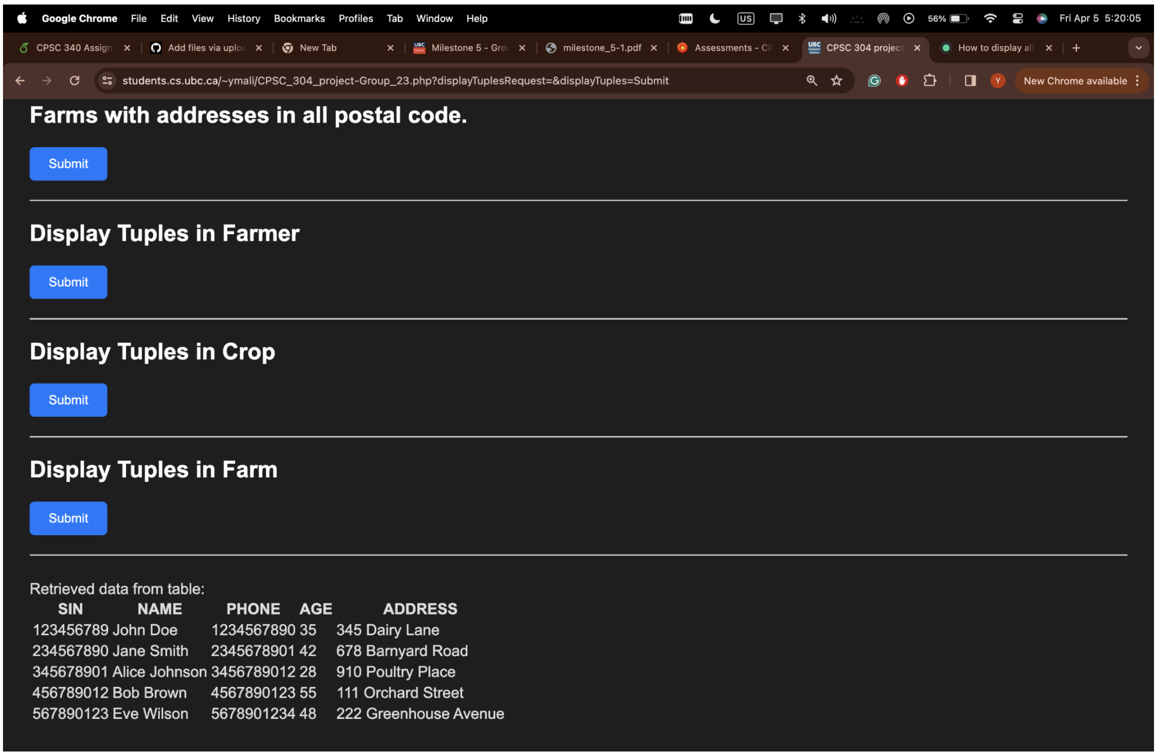
Submit

Display Tuples in Farm

Submit

```
Retrieved data from table:
 FID       ADDRESS
F006 345 Dairy Lane
F007 678 Barnyard Road
F008 910 Poultry Place
F009 111 Orchard Street
F010 222 Greenhouse Avenue
```

Before - The tuples in farmer


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
56%
Fri Apr 5 5:20:05
CPSC 340 Assign
Add files via uplo
New Tab
Milestone 5 - Gro
milestone_5-1.pdf
Assessments - CF
CPSC 304 project
How to display all
students.cs.ubc.ca/~ymali/CPSC_304_project-Group_23.php?displayTuplesRequest=&displayTuples=Submit
New Chrome available

Farms with addresses in all postal code.

Submit

Display Tuples in Farmer

Submit

Display Tuples in Crop

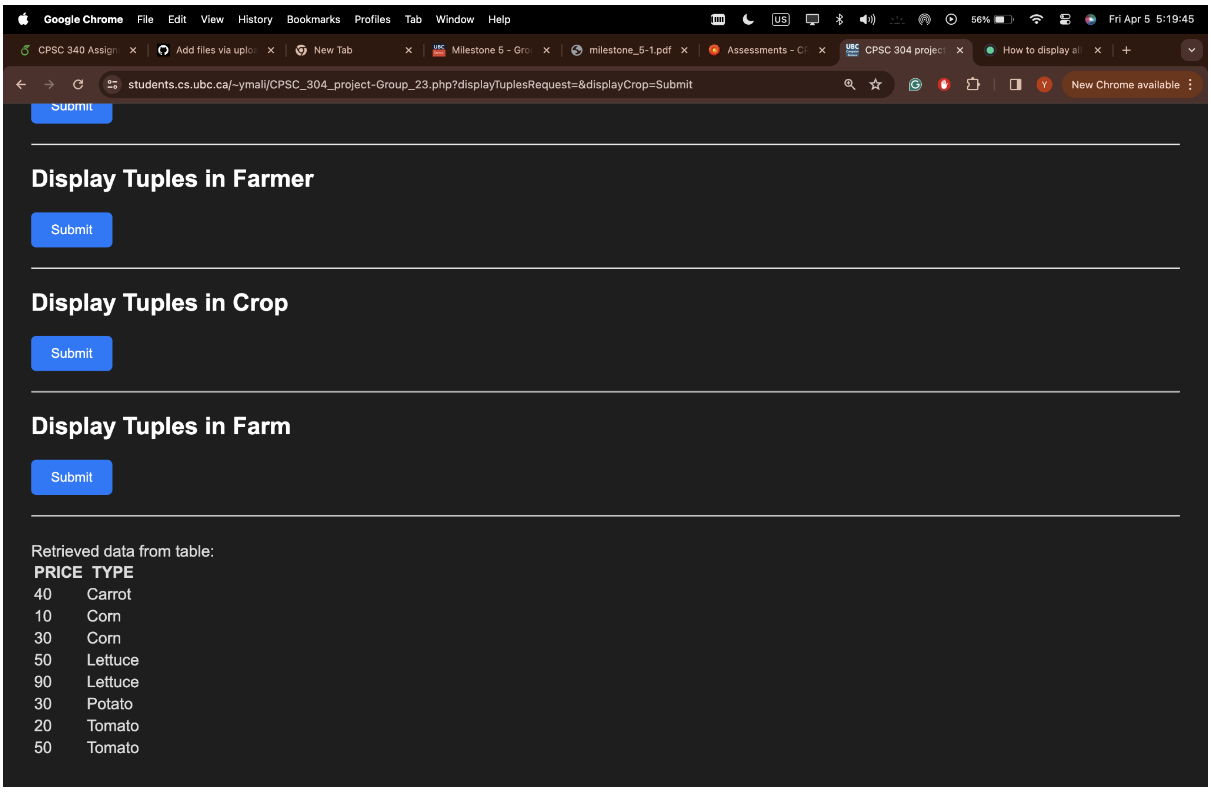
Submit

Display Tuples in Farm

Submit

```
Retrieved data from table:
    SIN          NAME         PHONE    AGE        ADDRESS
123456789 John Doe       1234567890 35   345 Dairy Lane
234567890 Jane Smith     2345678901 42   678 Barnyard Road
345678901 Alice Johnson 3456789012 28   910 Poultry Place
456789012 Bob Brown      4567890123 55   111 Orchard Street
567890123 Eve Wilson     5678901234 48   222 Greenhouse Avenue
```

Before - The tuples in grows_crop_type



students.cs.ubc.ca/~ymali/CPSC_304_project-Group_23.php?displayTuplesRequest=&displayCrop=Submit

Display Tuples in Farmer

Submit

Display Tuples in Crop

Submit

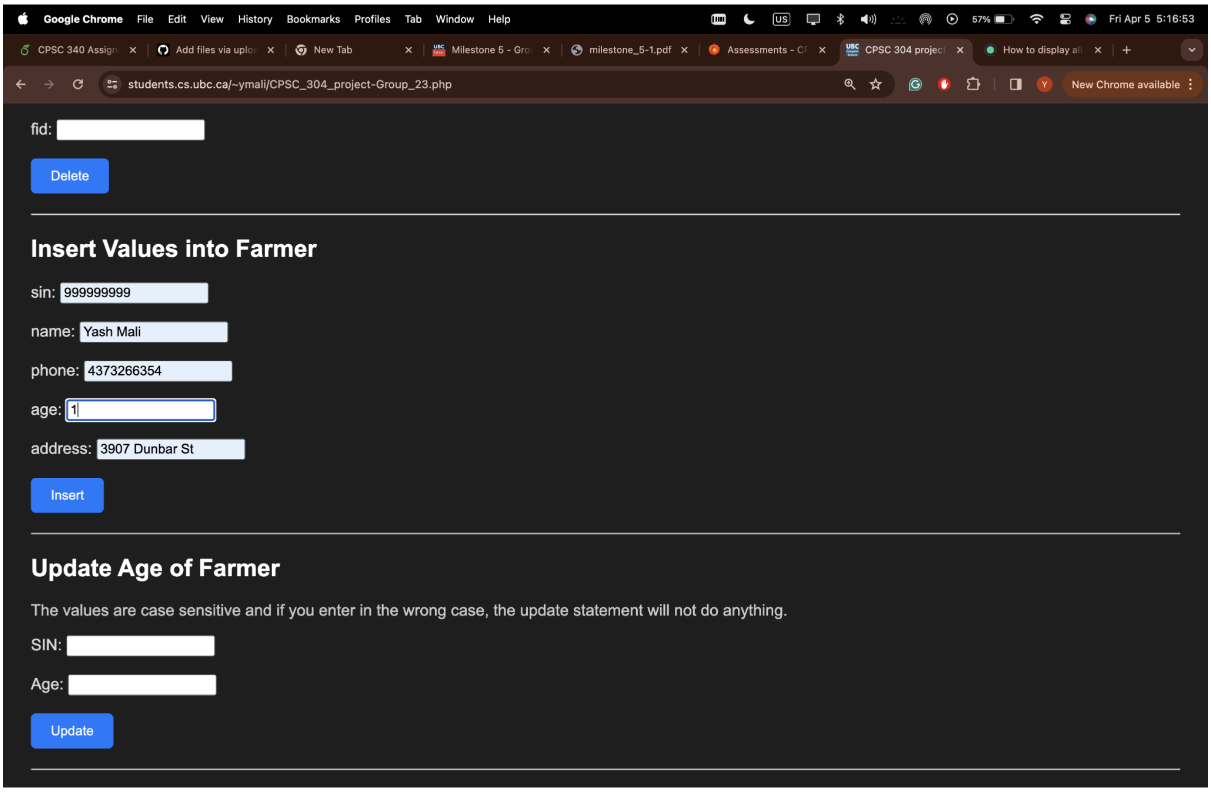
Display Tuples in Farm

Submit

Retrieved data from table:

PRICE	TYPE
40	Carrot
10	Corn
30	Corn
50	Lettuce
90	Lettuce
30	Potato
20	Tomato
50	Tomato

During - **Insert** into farmer


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Fri Apr 5 5:16:53
students.cs.ubc.ca/~ymali/CPSC_304_project-Group_23.php
New Chrome available

fid:

Delete

Insert Values into Farmer

sin: 999999999

name: Yash Mali

phone: 4373266354

age: 1

address: 3907 Dunbar St

Insert

Update Age of Farmer

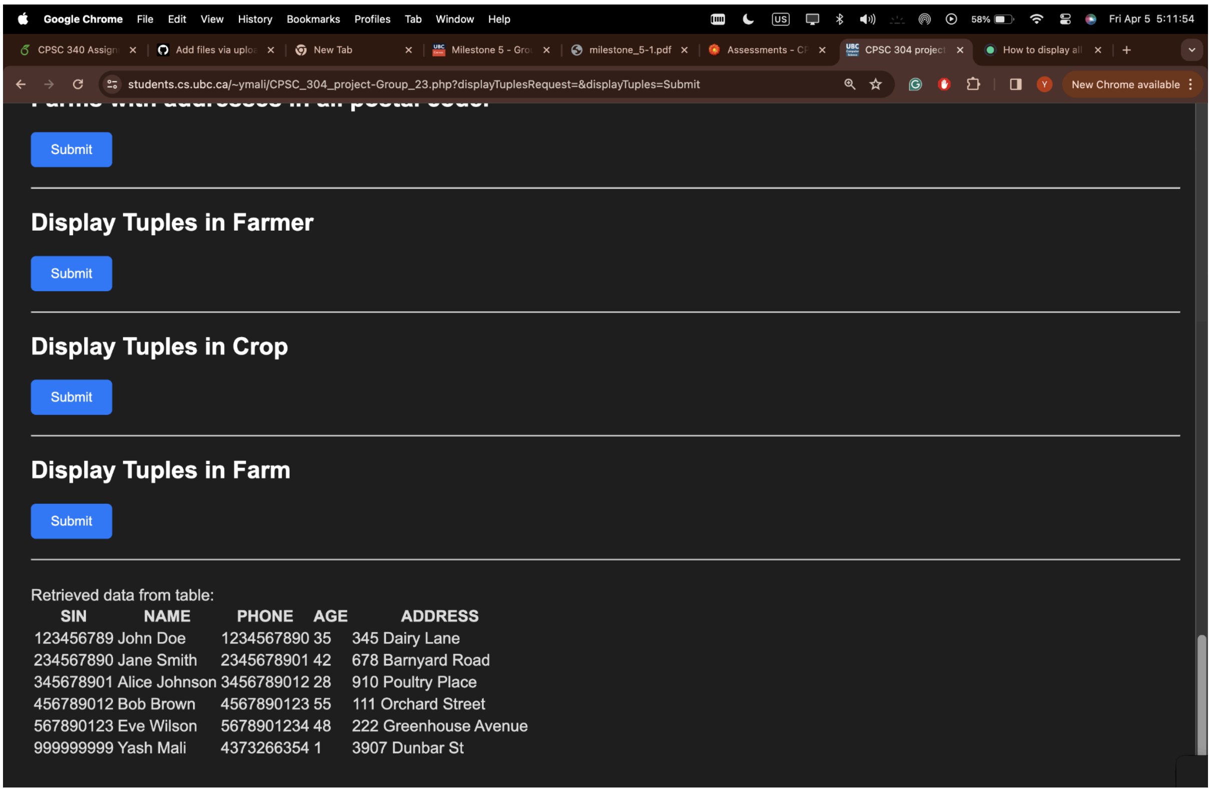
The values are case sensitive and if you enter in the wrong case, the update statement will not do anything.

SIN:

Age:

Update

After - **Insert** into farmer


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Fri Apr 5 5:11:54
students.cs.ubc.ca/~ymali/CPSC_304_project-Group_23.php?displayTuplesRequest=&displayTuples=Submit
New Chrome available

Submit

Display Tuples in Farmer

Submit

Display Tuples in Crop

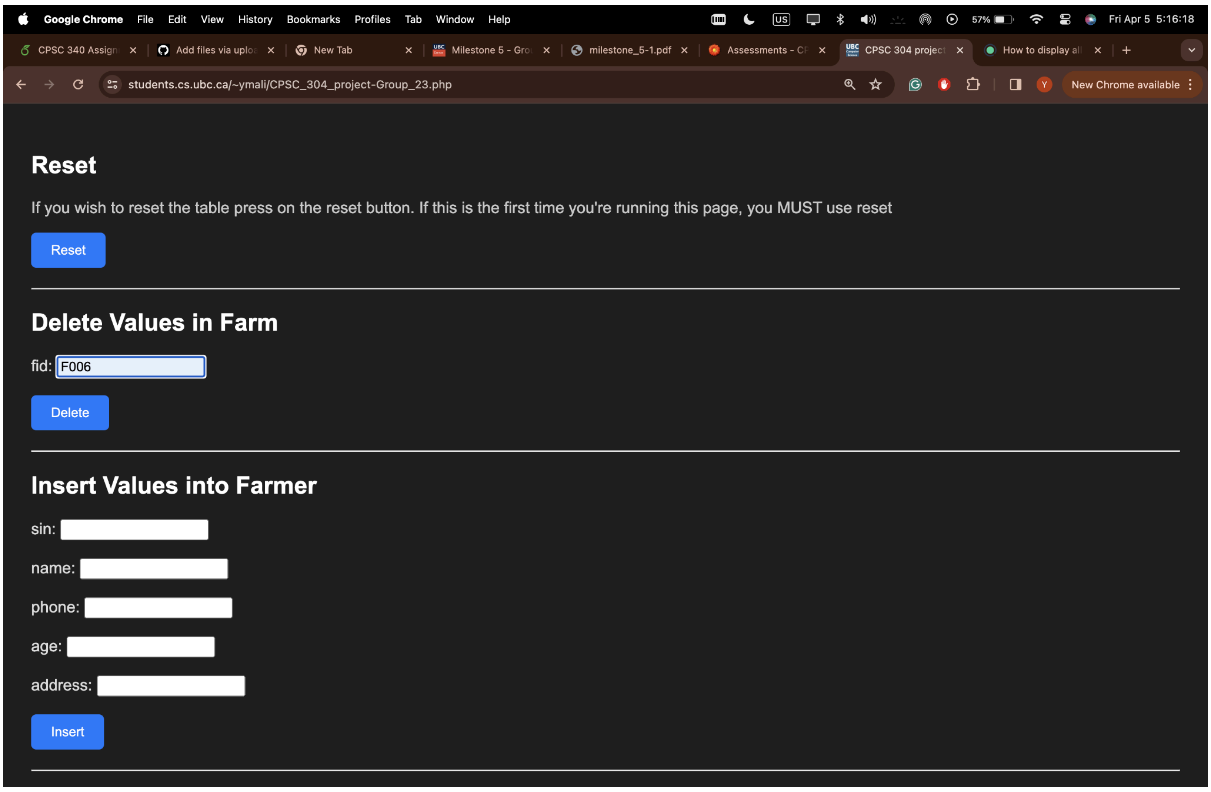
Submit

Display Tuples in Farm

Submit

```
Retrieved data from table:
    SIN         NAME        PHONE    AGE       ADDRESS
123456789 John Doe      1234567890 35    345 Dairy Lane
234567890 Jane Smith    2345678901 42    678 Barnyard Road
345678901 Alice Johnson 3456789012 28    910 Poultry Place
456789012 Bob Brown     4567890123 55    111 Orchard Street
567890123 Eve Wilson    5678901234 48    222 Greenhouse Avenue
999999999 Yash Mali     4373266354 1     3907 Dunbar St
```

During - **Delete** from farm



Reset

If you wish to reset the table press on the reset button. If this is the first time you're running this page, you MUST use reset

Reset

Delete Values in Farm

fid: F006

Delete

Insert Values into Farmer

sin:

name:

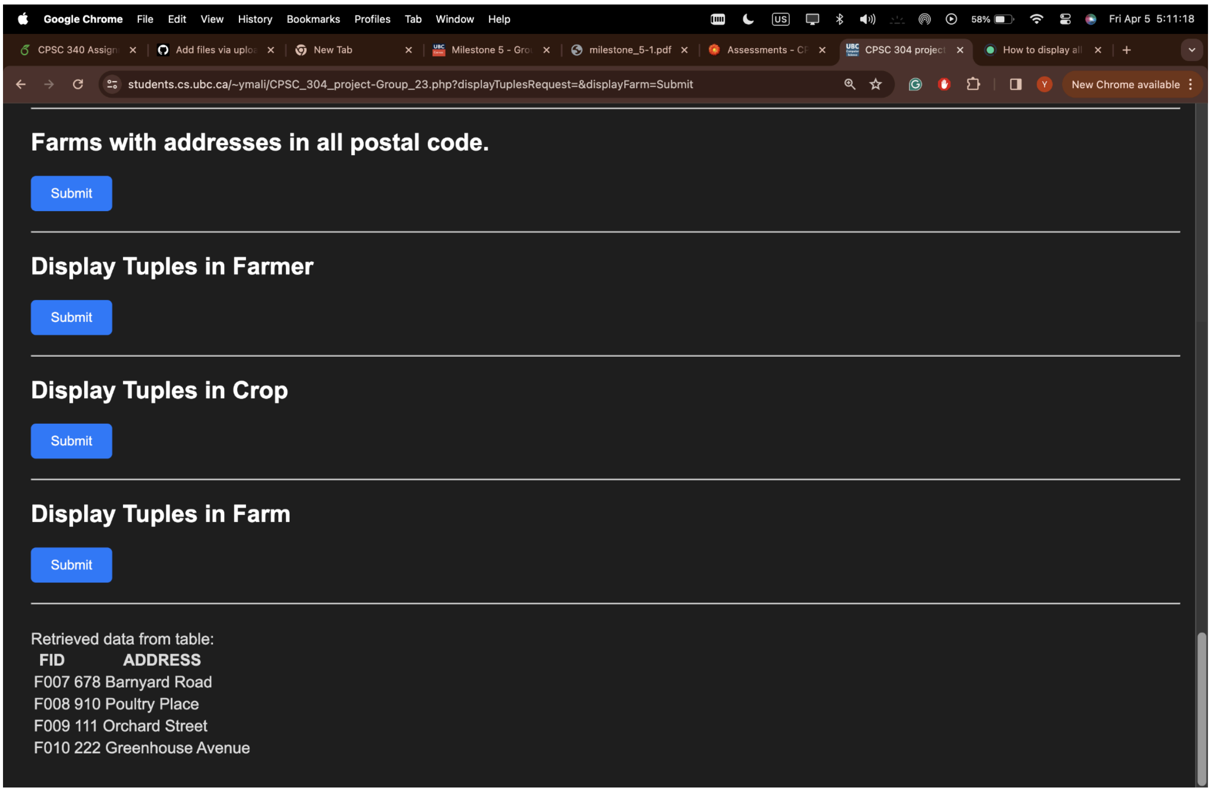
phone:

age:

address:

Insert

After - **Delete** from farm


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
58%
Fri Apr 5 5:11:18
CPSC 340 Assign
Add files via uplo
New Tab
Milestone 5 - Gro
milestone_5-1.pdf
Assessments - C
CPSC 304 project
How to display all
students.cs.ubc.ca/~ymali/CPSC_304_project-Group_23.php?displayTuplesRequest=&displayFarm=Submit
New Chrome available

Farms with addresses in all postal code.

Submit

Display Tuples in Farmer

Submit

Display Tuples in Crop

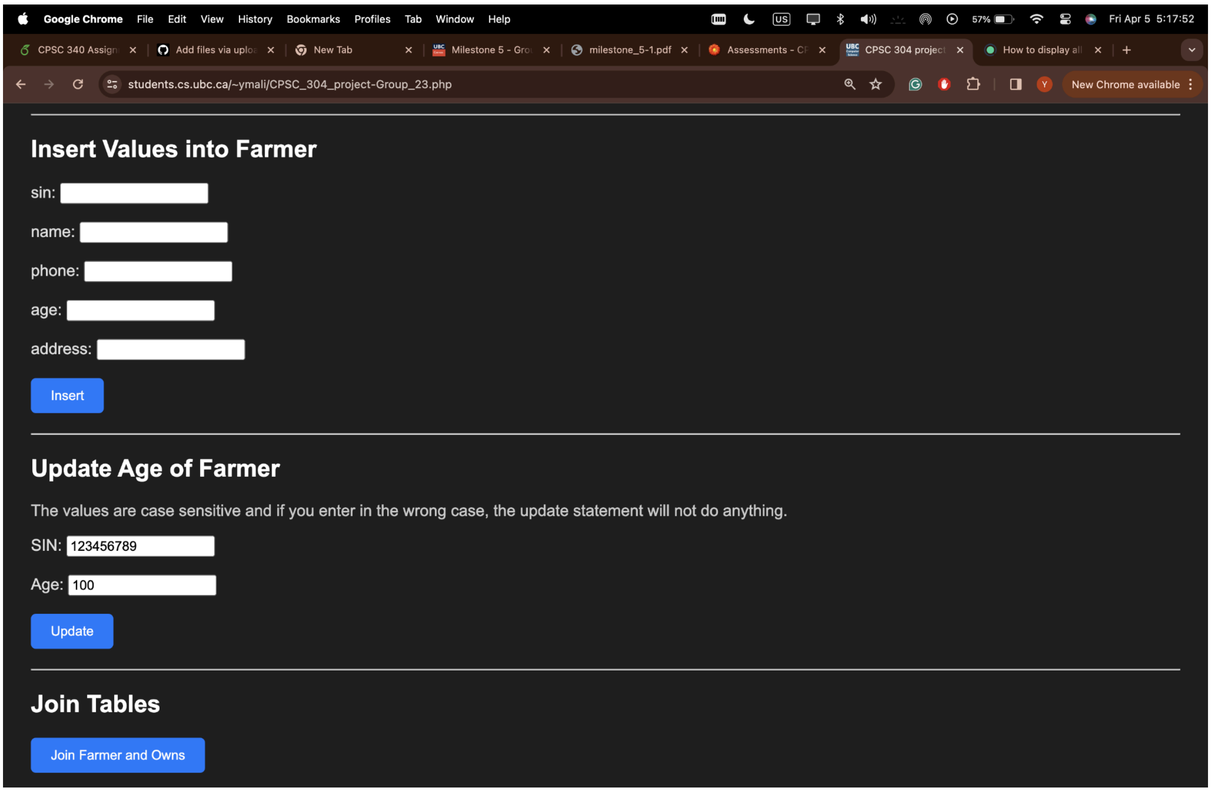
Submit

Display Tuples in Farm

Submit

```
Retrieved data from table:
  FID        ADDRESS
 F007 678 Barnyard Road
 F008 910 Poultry Place
 F009 111 Orchard Street
 F010 222 Greenhouse Avenue
```

During - **Update** age of farmer



Insert Values into Farmer

sin:

name:

phone:

age:

address:

Update Age of Farmer

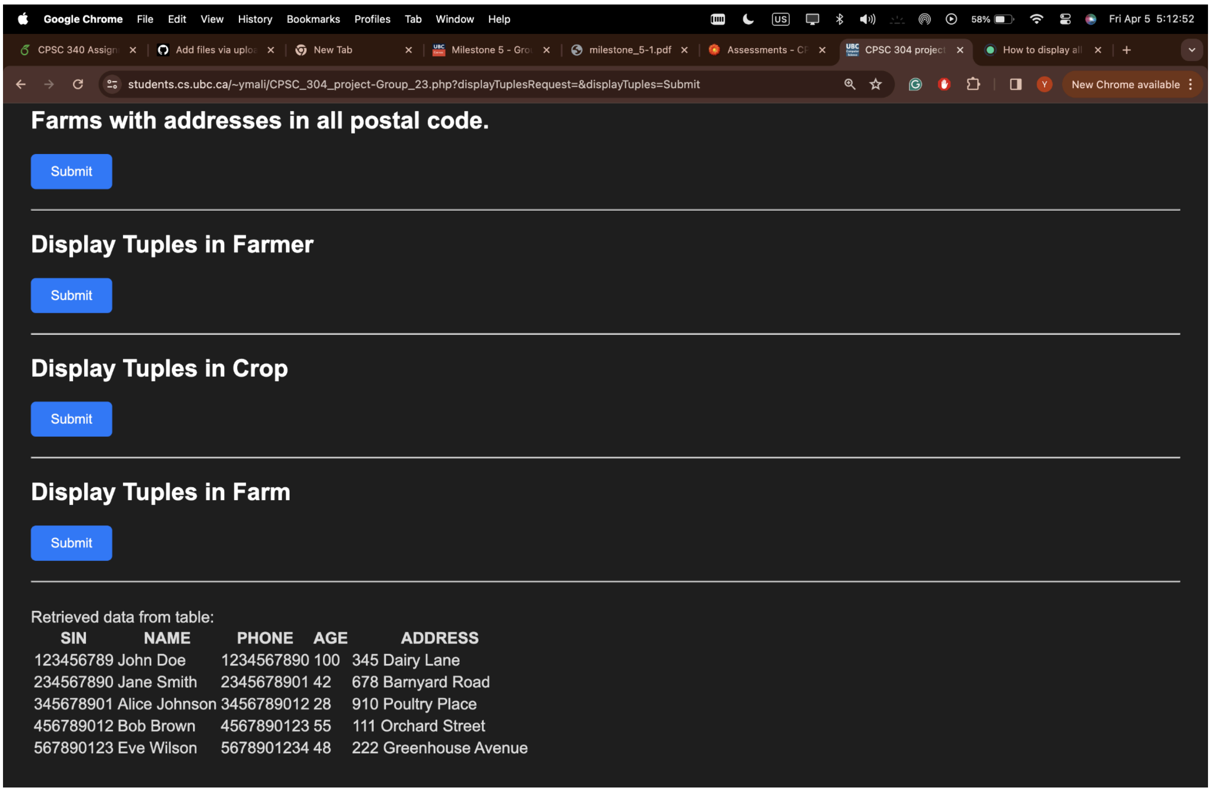
The values are case sensitive and if you enter in the wrong case, the update statement will not do anything.

SIN: 123456789

Age: 100

Join Tables

After - **Update** age of farmer



Farms with addresses in all postal code.

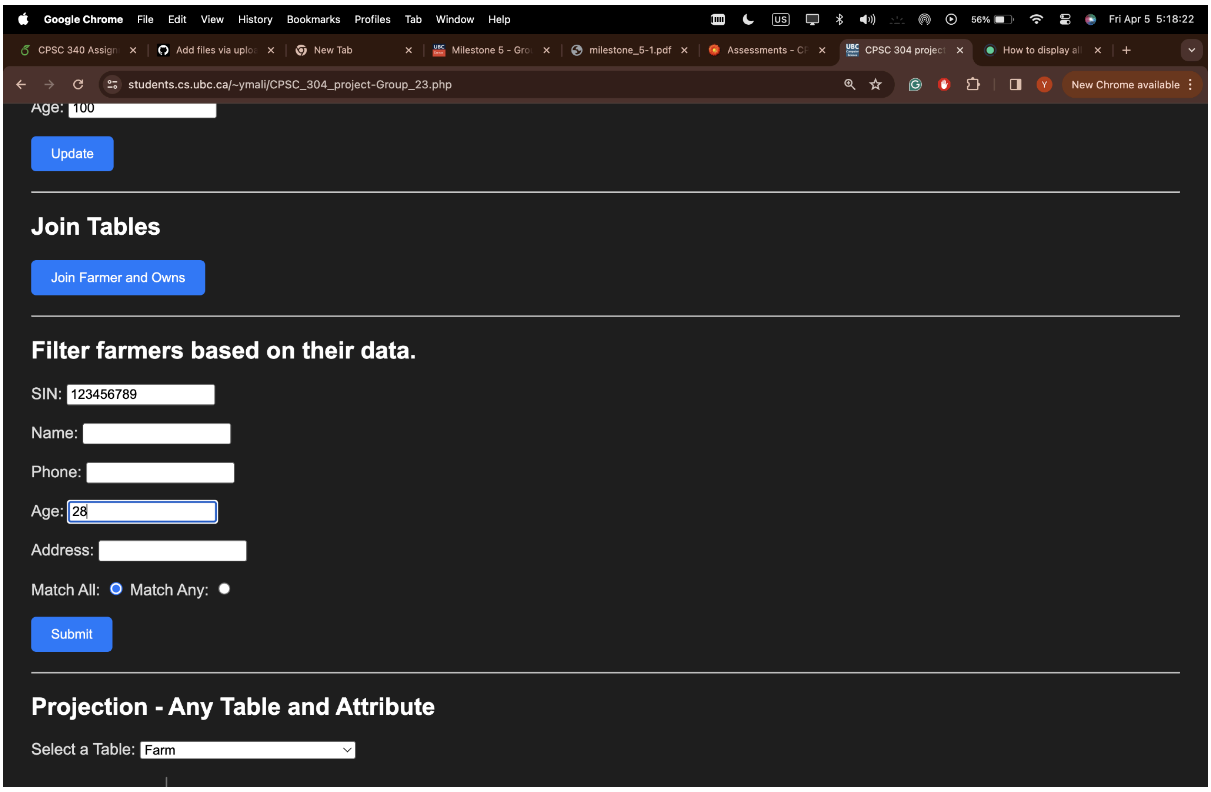
Display Tuples in Farmer

Display Tuples in Crop

Display Tuples in Farm

```
Retrieved data from table:
    SIN          NAME          PHONE    AGE        ADDRESS
123456789 John Doe        1234567890 100   345 Dairy Lane
234567890 Jane Smith      2345678901 42    678 Barnyard Road
345678901 Alice Johnson 3456789012 28    910 Poultry Place
456789012 Bob Brown       4567890123 55    111 Orchard Street
567890123 Eve Wilson      5678901234 48    222 Greenhouse Avenue
```

During - **Selection** on farmer using OR clause (SIN = 123456789 OR Age = 28)



students.cs.ubc.ca/~ymali/CPSC_304_project-Group_23.php

Age: 100

Update

Join Tables

Join Farmer and Owns

Filter farmers based on their data.

SIN: 123456789

Name:

Phone:

Age: 28

Address:

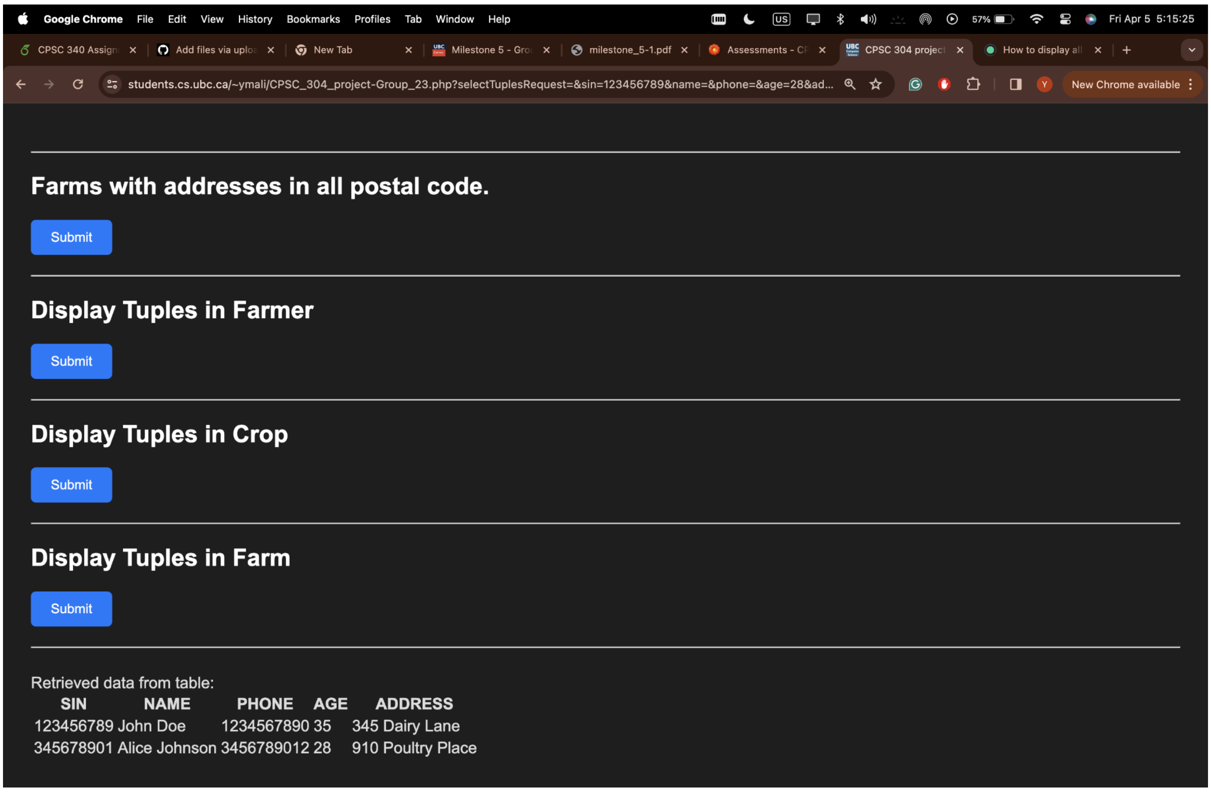
Match All: ● Match Any: ○

Submit

Projection - Any Table and Attribute

Select a Table: Farm

After - **Selection** on farmer using OR clause (SIN = 123456789 OR Age = 28)



students.cs.ubc.ca/~ymali/CPSC_304_project-Group_23.php?selectTuplesRequest=&sin=123456789&name=&phone=&age=28&ad...

Farms with addresses in all postal code.

Submit

Display Tuples in Farmer

Submit

Display Tuples in Crop

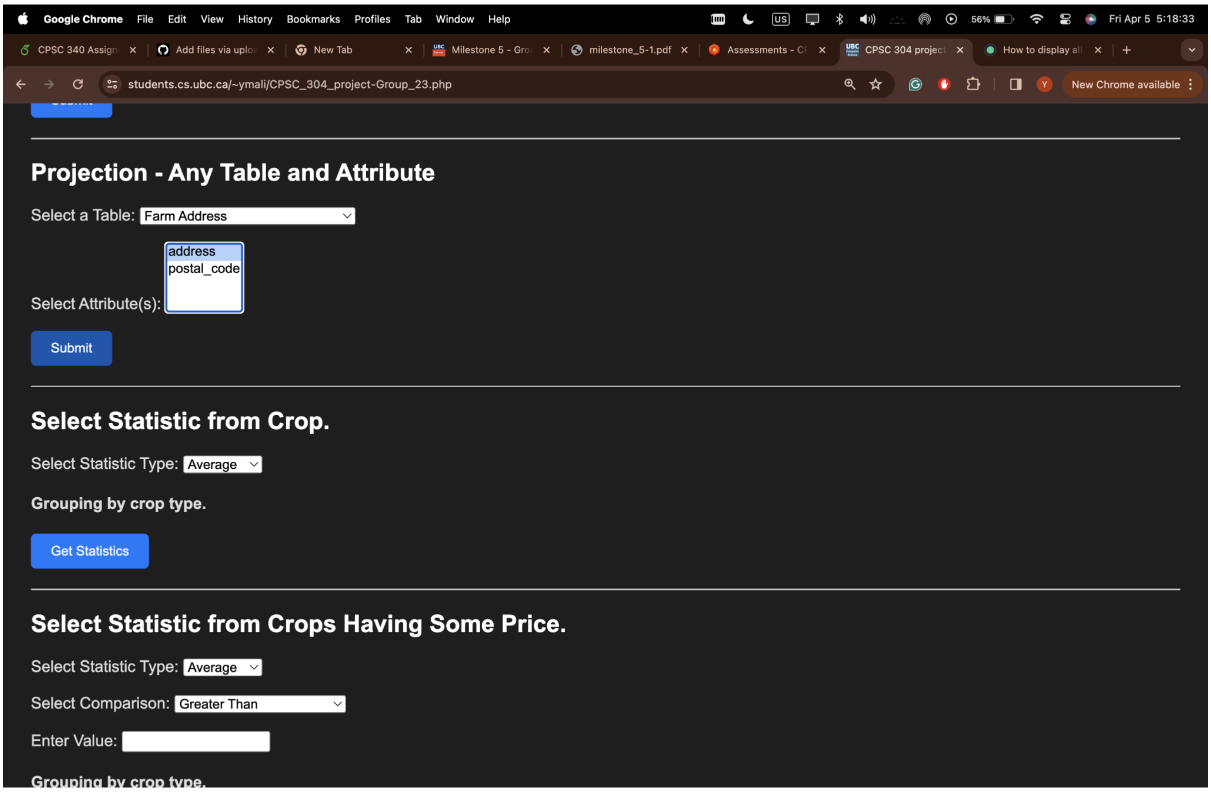
Submit

Display Tuples in Farm

Submit

```
Retrieved data from table:
    SIN          NAME         PHONE    AGE    ADDRESS
123456789 John Doe        1234567890 35    345 Dairy Lane
345678901 Alice Johnson 3456789012 28    910 Poultry Place
```

During - **Projection** address from FarmAddress



Projection - Any Table and Attribute

Select a Table: Farm Address

Select Attribute(s): address / postal_code

Submit

Select Statistic from Crop.

Select Statistic Type: Average

Grouping by crop type.

Get Statistics

Select Statistic from Crops Having Some Price.

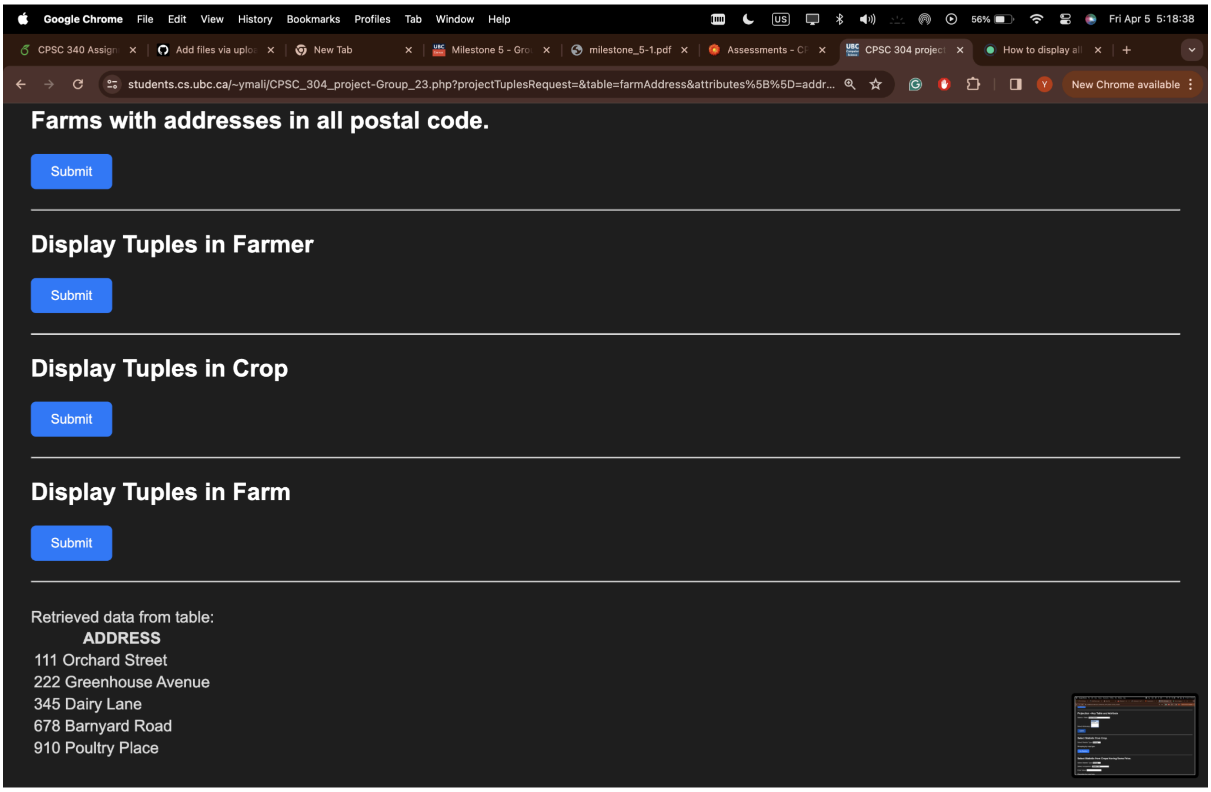
Select Statistic Type: Average

Select Comparison: Greater Than

Enter Value:

Grouping by crop type.

After - **Projection** address from FarmAddress


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Fri Apr 5 5:18:38
CPSC 340 Assign
Add files via uplo
New Tab
Milestone 5 - Gro
milestone_5-1.pdf
Assessments - C
CPSC 304 project
How to display al
students.cs.ubc.ca/~ymali/CPSC_304_project-Group_23.php?projectTuplesRequest=&table=farmAddress&attributes%5B%5D=addr...
New Chrome available

Farms with addresses in all postal code.

Submit

Display Tuples in Farmer

Submit

Display Tuples in Crop

Submit

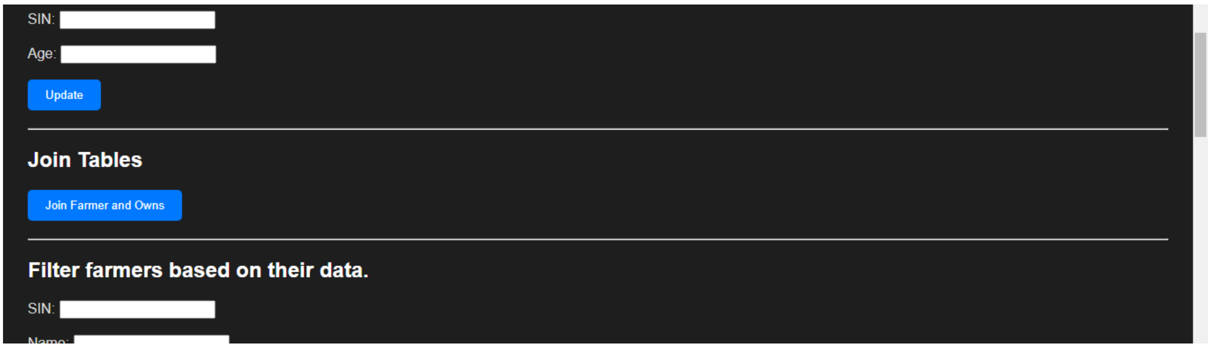
Display Tuples in Farm

Submit

Retrieved data from table:

ADDRESS
111 Orchard Street
222 Greenhouse Avenue
345 Dairy Lane
678 Barnyard Road
910 Poultry Place

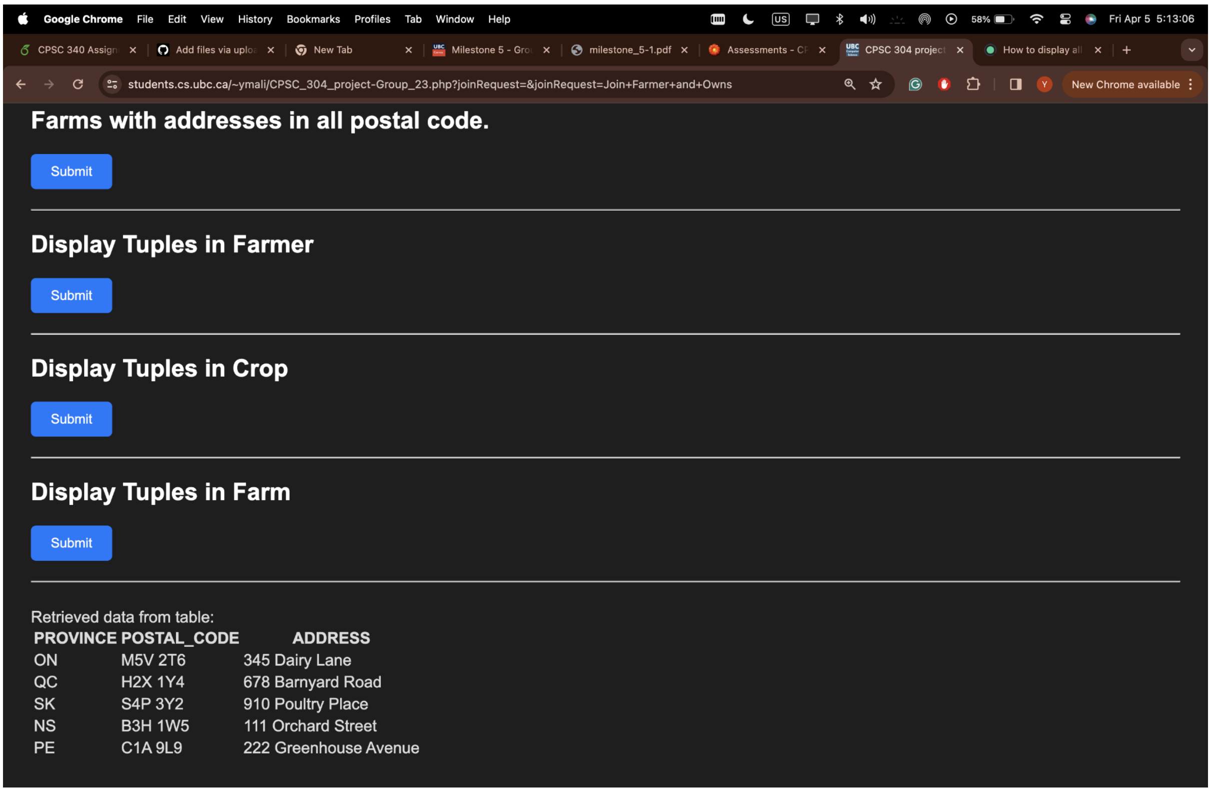
During - **Join** farmer and owns (Hard-coded, see line 559 to 563 in GRADE_THIS.php), and then project province, postal code and address



Join Tables

Filter farmers based on their data.

After - **Join** farmer and owns (Hard-coded, see line 559 to 563 in GRADE_THIS.php), and then project province, postal code and address



students.cs.ubc.ca/~ymali/CPSC_304_project-Group_23.php?joinRequest=&joinRequest=Join+Farmer+and+Owns

Farms with addresses in all postal code.

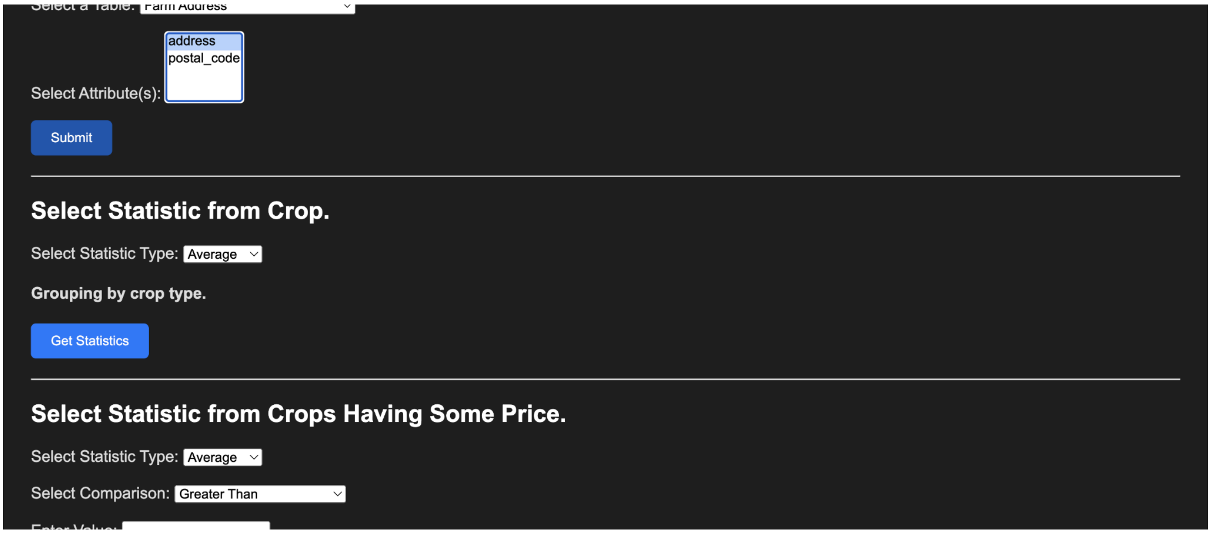
Display Tuples in Farmer

Display Tuples in Crop

Display Tuples in Farm

```
Retrieved data from table:
PROVINCE POSTAL_CODE      ADDRESS
ON       M5V 2T6     345 Dairy Lane
QC       H2X 1Y4     678 Barnyard Road
SK       S4P 3Y2     910 Poultry Place
NS       B3H 1W5     111 Orchard Street
PE       C1A 9L9     222 Greenhouse Avenue
```

During - **Aggregation** (AVG (PRICE)) **with Group By** crop type using grows_crop_type table



Select Attribute(s):

Select Statistic from Crop.

Select Statistic Type: Average

Grouping by crop type.

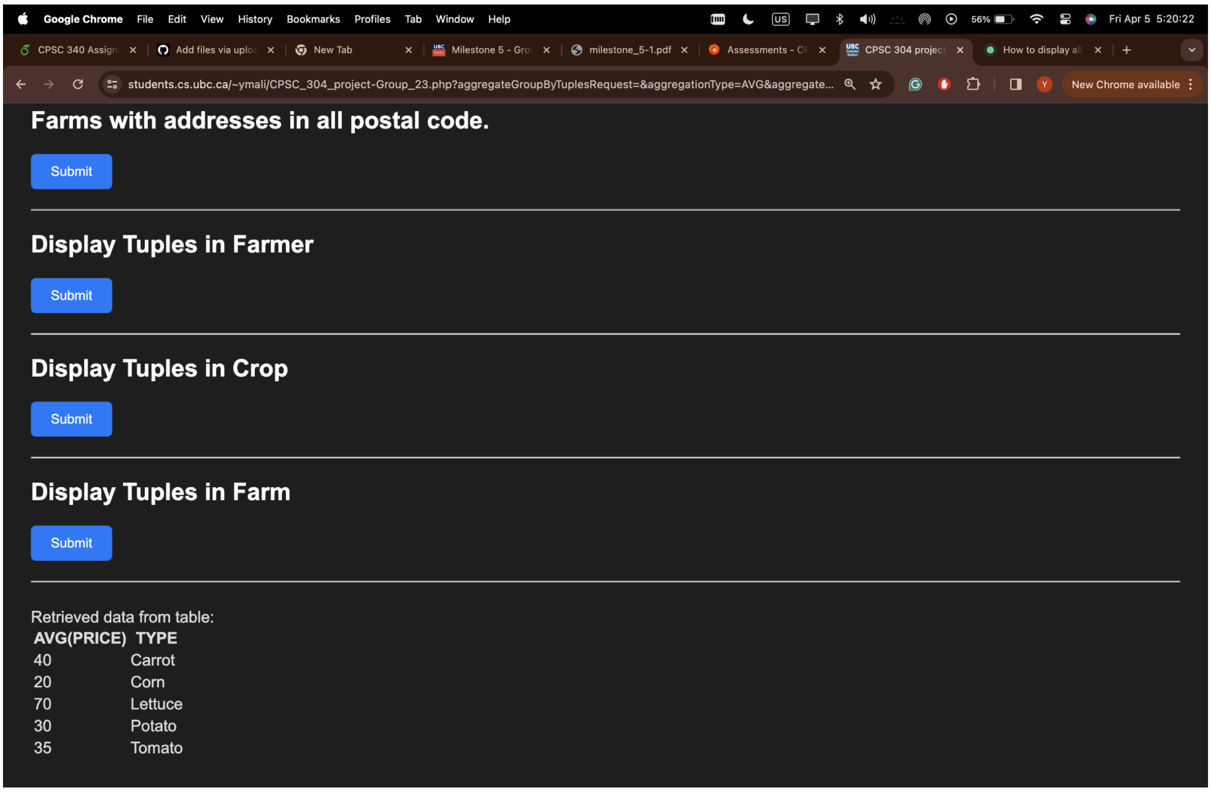
Select Statistic from Crops Having Some Price.

Select Statistic Type: Average

Select Comparison: Greater Than

Enter Value:

After - **Aggregation** (AVG (PRICE)) **with Group By** crop type using grows_crop_type table


Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
56%
Fri Apr 5 5:20:22
CPSC 340 Assign
Add files via uplo
New Tab
Milestone 5 - Gro
milestone_5-1.pdf
Assessments - C
CPSC 304 project
How to display al
students.cs.ubc.ca/~ymali/CPSC_304_project-Group_23.php?aggregateGroupByTuplesRequest=&aggregationType=AVG&aggregate...
New Chrome available
Submit
Submit
Submit
Submit

Farms with addresses in all postal code.

Display Tuples in Farmer

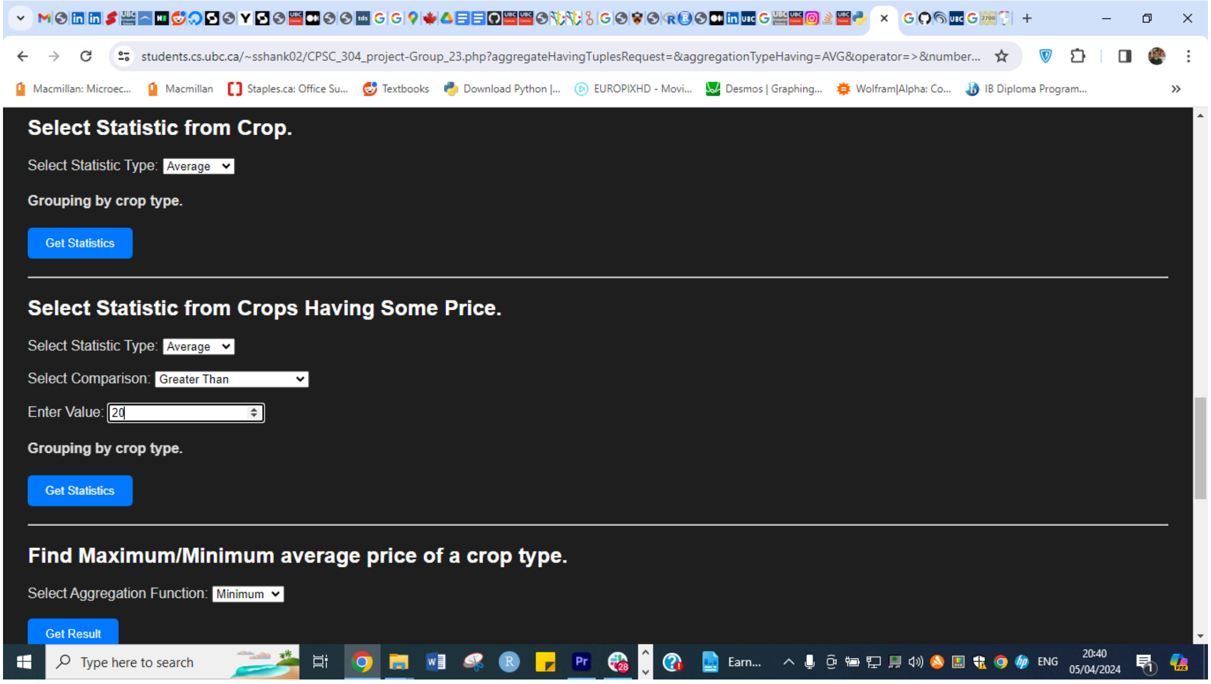
Display Tuples in Crop

Display Tuples in Farm

Retrieved data from table:

```
AVG(PRICE)  TYPE
40          Carrot
20          Corn
70          Lettuce
30          Potato
35          Tomato
```

During - **Aggregation** (AVG(PRICE)) **with Having** Price＞20 using grows_crop_type table



students.cs.ubc.ca/~sshank02/CPSC_304_project-Group_23.php?aggregateHavingTuplesRequest=&aggregationTypeHaving=AVG&operator=>&number...

Select Statistic from Crop.

Select Statistic Type: Average

Grouping by crop type.

Get Statistics

Select Statistic from Crops Having Some Price.

Select Statistic Type: Average

Select Comparison: Greater Than

Enter Value: 20

Grouping by crop type.

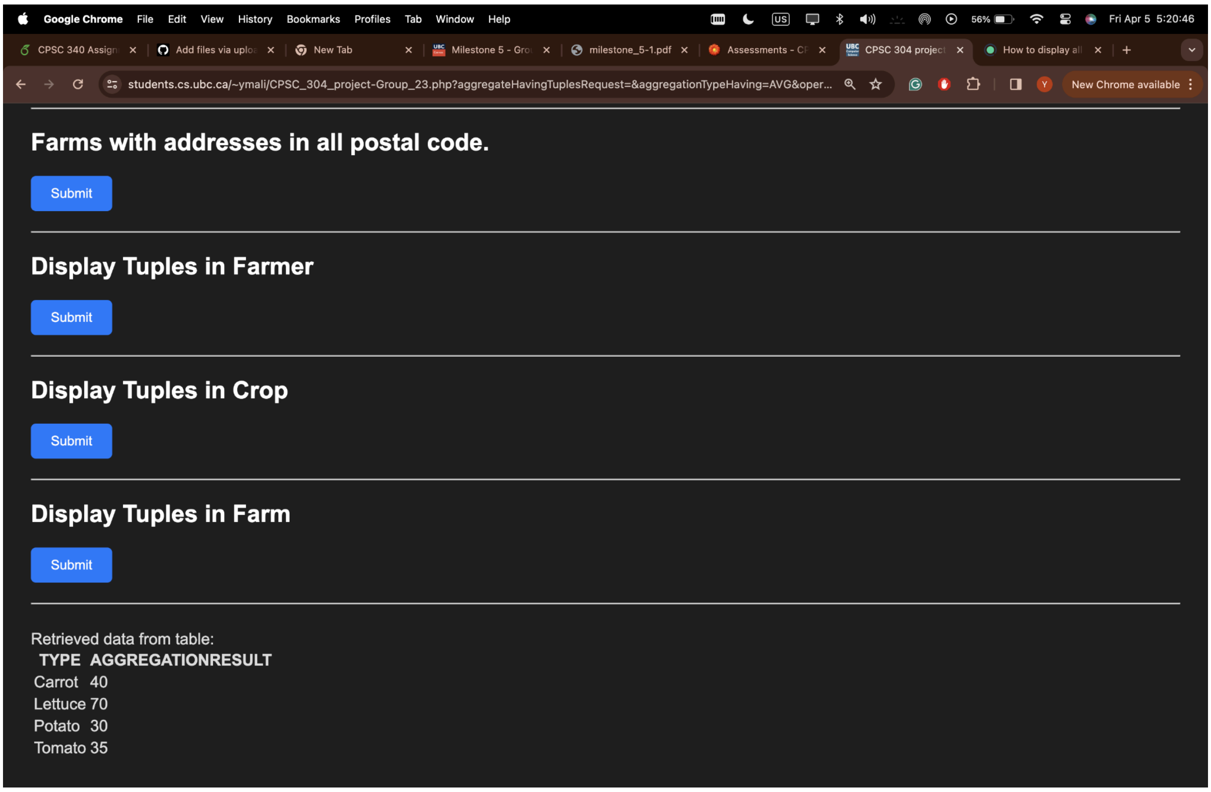
Get Statistics

Find Maximum/Minimum average price of a crop type.

Select Aggregation Function: Minimum

Get Result

After - **Aggregation** (AVG(PRICE)) **with Having** Price＞20 using grows_crop_type table



students.cs.ubc.ca/~ymali/CPSC_304_project-Group_23.php?aggregateHavingTuplesRequest=&aggregationTypeHaving=AVG&oper...

Farms with addresses in all postal code.

Submit

Display Tuples in Farmer

Submit

Display Tuples in Crop

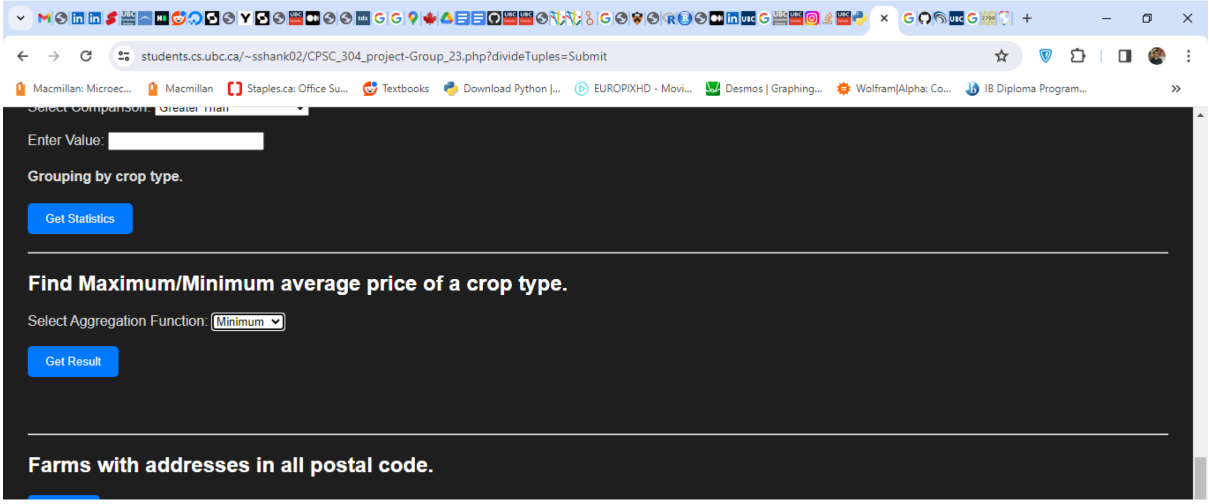
Submit

Display Tuples in Farm

Submit

```
Retrieved data from table:
 TYPE  AGGREGATIONRESULT
Carrot  40
Lettuce 70
Potato  30
Tomato 35
```

During - **Nested Aggregation** MIN(AVG(PRICE)) **with Group By** crop type using grows_crop_type table



Select Comparison: Greater Than

Enter Value:

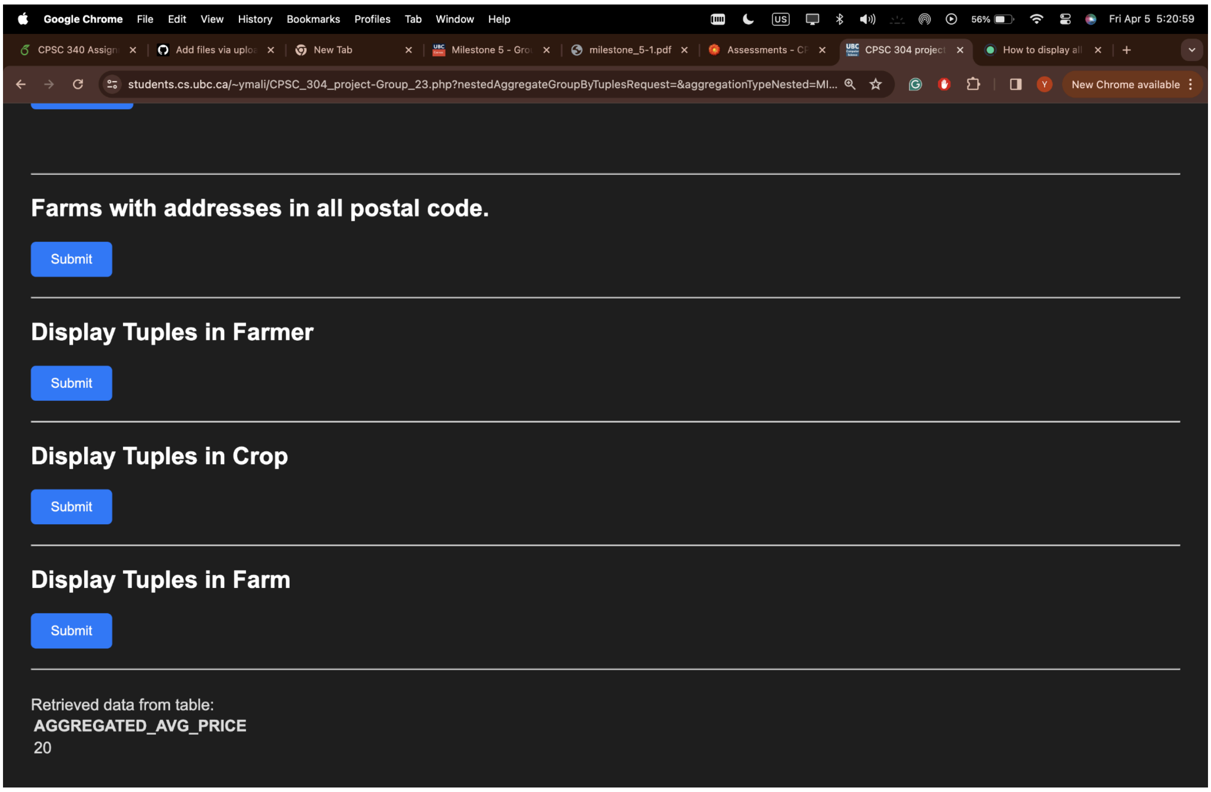
Grouping by crop type.

Find Maximum/Minimum average price of a crop type.

Select Aggregation Function: Minimum

Farms with addresses in all postal code.

After - **Nested Aggregation** MIN(AVG(PRICE)) **with Group By** crop type using grows_crop_type table



Farms with addresses in all postal code.

Submit

Display Tuples in Farmer

Submit

Display Tuples in Crop

Submit

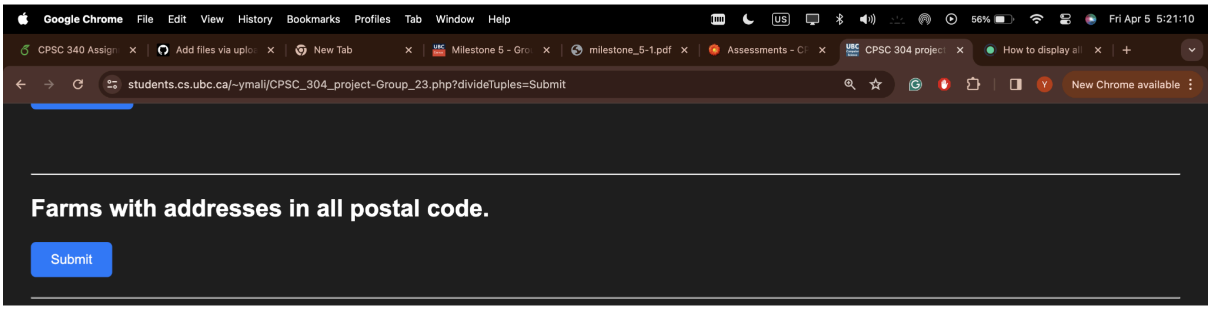
Display Tuples in Farm

Submit

Retrieved data from table:

AGGREGATED_AVG_PRICE
20

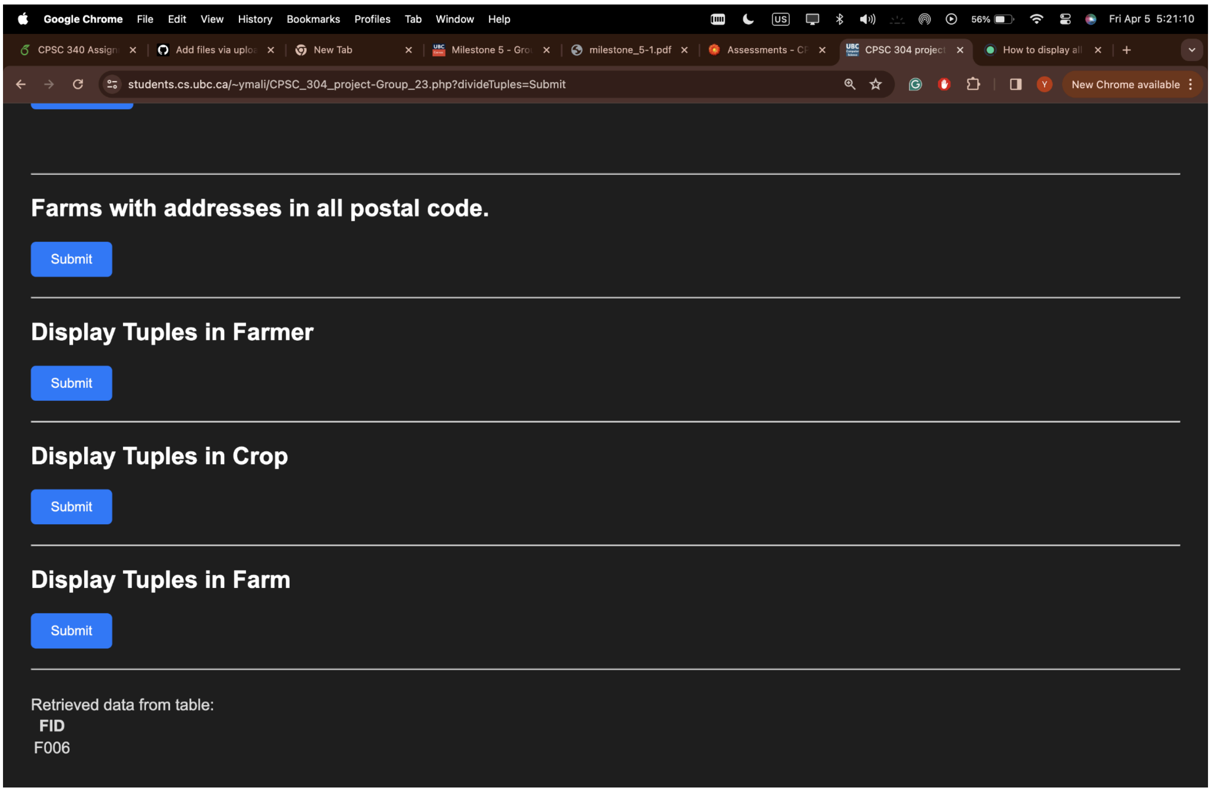
During - **Division**: Find farms that are owned by ALL farmers (Hard-coded, see line 630 to 642 in GRADE_THIS.php)



Farms with addresses in all postal code.

Note: The title for this is incorrect, it should say, “Farms owned by all farmers instead”. We ran out of time and couldn’t correct this.

After - **Division**: Find farms that are owned by ALL farmers (Hard-coded, see line 630 to 642 in GRADE_THIS.php)



Farms with addresses in all postal code.

Display Tuples in Farmer

Display Tuples in Crop

Display Tuples in Farm

```
Retrieved data from table:
 FID
 F006
```

Deliverable 4

4. ReadMe.md has been added to the GitHub page