Repository link:

https://github.students.cs.ubc.ca/CPSC304-2022W-T1/project f5k3b i7n2b t3q7m

Description of the final project

The project allows users to retrieve information about food delivery platforms and their relevant data, for example restaurants, users, promotions, etc. Specific tasks that our project accomplishes are:

- Retrieving, inserting, deleting, and updating user addresses.
- Retrieving restaurants by filtering on rates, promotion discount code, platform.
 - Restaurants with a rating above a given threshold
 - Maximum rating for restaurant types with average rating higher than a given number
 - All restaurants on a given platform
 - All restaurants registered on all platforms
- Retrieving delivery persons' location, id, name, vehicle info.
- Retrieving platform-specific promotion and user balance details.
 - Minimum and maximum discounts per platform
 - Maximum user balance for each platform with more than one user

Final schema

Our final schema is almost the same as our previously submitted schema, except that it denormalizes Restaurant to only use one table, to make queries easier to write.

Platform(platformName:string)

User_1(username:string, userID:int, account_balance:float, platformName:string)

UserAddress(streetAddress:string, city:string, country:string, addressType:string, userID:int)

Restaurant(name:string, restaurantID:int, rating:int, location:string, type:string)

FoodItem(name:string, restaurantID:int)

DeliveryPerson(name:string, id:int, currentLocation:string)

Vehicle2(model:string, make:string)

Vehicle1(number plate:string, model:string)

Promotion(promotionID:int, discountPercentage:float, platform:string)

PlatformWidePromotion(redeemCode:int, promotionID:int)

RestaurantSpecificDiscount(restaurantID:int, promotionID:int)

UsesVehicle (deliveryPersonID:int, number_plate:string)

WorksFor(deliveryPersonID:int, platform:string)

Hosts (platform:string, restaurantID:int)

Offers (restaurantID:int, promotionID:int)

Screenshots of data

Platform

PLATFORMNAME
DoorDash Fantuan Grubhub SkipTheDishes UberEats

User_1

USERNAME	USERID	ACCOUNT_BALANCE	PLATFORMNAME
foodlover2	1001	43.75	DoorDash
bluesweater	1002	2.54	UberEats
catperson	1003	20	DoorDash
sunflower34	1004	78.88	DoorDash
abcdef	1005	0	SkipTheDishes
paperkite	1006	13.99	UberEats

UserAddress

STREETADDRESS	CITY	COUNTRY	ADDRESSTYPE	USERID
1 Apple Dr	Vancouver	Canada	delivery	1001
2 Banana St	Vancouver	Canada	billing	1002
3 Cherry Rd	Vancouver	Canada	delivery	1003
4 Donut Dr	Vancouver	Canada	delivery	1004
5 Eclair St	Vancouver	Canada	delivery	1005
6 Fig Rd	Vancouver	Canada	billing	1005
7 Grape Dr	Vancouver	Canada	delivery	1006

Restaurant

NAME	RESTAURANTID	RATING LOCATION	ТҮРЕ
McDonalds	5001	3 15 Fries St	fast food
Kung Fu Noodle	5002	4 25 Panda St	restaurant
Yunshang Rice Noodle	5003	5 35 Noodle St	restaurant
Nori	5004	5 45 Seaweed St	restaurant
Chatime	5005	5 55 Tea St	bubble tea

FoodItem

NAME	RESTAURANTID
chicken rice noodle	5003
chicken wings	5001
fried rice	5002
fries	5001
hamburger	5001

DeliveryPerson

NAME	ID CURRENTLOCATION
John Doe	4001 14 Hello St
Jane Deer	4002 24 High St
Jeremy Don	4003 34 Hay Rd
Jack Donald	4004 44 Greetings Ln
Jill Dew	4005 54 Salutations St

Vehicle2

MODEL	MAKE
Elantra	Hyundai
Everest	Ford
Civic	Honda
Camry	Toyota
X5	BMW

Vehicle1

NUMBER_PLA MODEL				
123 ABC	Elantra			
124 ABC	Everest			
125 ABC	Civic			
126 ABC	Camry			
127 ABC	X5			

Promotion

PROMOTIONID	DISCOUNTPERCENTAGE	PLATFORM
2001	.15	DoorDash
2002	.26	UberEats
2003	.37	SkipTheDishes
2004	.48	Fantuan
2005	.59	Grubhub
2006	.25	DoorDash
2007	.36	UberEats
2008	.47	SkipTheDishes
2009	.58	Fantuan
2010	.69	Grubhub

PlatformWidePromotion

REDEEMCODE	PROMOTIONID
111000	2001
222000	2002
333000	2003
444000	2004
555000	2005

RestaurantSpecificDiscount

RESTAURANTID	PROMOTIONID
5001	2006
5002	2007
5003	2008
5004	2009
5005	2010

UsesVehicle

WorksFor

```
DELIVERYPERSONID PLATFORM

4001 DoorDash

4002 UberEats

4003 DoorDash

4004 UberEats

4005 SkipTheDishes
```

Hosts

PLATFORM	RESTAURANTID
DoorDash	5001
Fantuan	5001
Fantuan	5003
Fantuan	5004
Grubhub	5001
Grubhub	5005
SkipTheDishes	5001
UberEats	5001
UberEats	5002

Offers

RESTAURANTID	PROMOTIONID
5001	2006
5002	2007
5003	2008
5004	2009
5005	2010

Queries

INSERT operation

- File: data_retrive_helper.php, function: insertAddress()

```
INSERT into UserAddress
VALUES($newStreet, $newCity, $newCountry, $addressType, $ID)
```

DELETE operation

- File: data_retrive_helper.php, function: deleteAddress()

```
DELETE FROM UserAddress
WHERE userID = $ID AND
    streetAddress = $oldStreet AND
    city = $oldCity AND
    country = $oldCountry
```

UPDATE operation

- File: data_retrive_helper.php, function: changeAddressType()

```
UPDATE UserAddress
SET addressType = $addressType
WHERE userID = $ID AND
    streetAddress = $oldStreet AND
    city = $oldCity AND
    country = $oldCountry
```

Projection operations

- File: data_retrive_helper.php, function: retriveAllDiscountCode()

SELECT promotionID

FROM RestaurantSpecificDiscount

- File: data_retrive_helper.php, function: retriveAllDiliveryPersonName()

SELECT name

FROM DeliveryPerson

Selection operations

- File: data_retrive_helper.php, function: restaurantsFilter()

SELECT restaurantID, name, rating FROM Restaurant WHERE rating > \$rate

- File: data_retrive_helper.php, function: retrieveDiscountCodeFromRestaurants()

SELECT promotionID
FROM RestaurantSpecificDiscount
WHERE restaurantID = \$restaurantID

- File: data_retrive_helper.php, function: retriveCarNumberPlate()

SELECT number_plate FROM UsesVehicle WHERE deliveryPersonID = \$personID

- File: data_retrive_helper.php, function: retriveCarModel()

SELECT model FROM Vehicle WHERE number_plate = \$num_plate

File: data_retrive_helper.php, function: retriveUserAddress()

SELECT streetAddress, city, country, addressType FROM UserAddress
WHERE userID = \$ID

Join

- File: restaurant.php, function: handleRestPlatRequest()

SELECT name, restaurantID FROM Hosts NATURAL JOIN Restaurant WHERE platform='\$platformName'

Aggregation – GROUP BY

- File: promotion.php, function: handlePromoMinMaxRequest()

SELECT p.platform, \$aggregateOp (p.discountPercentage)
FROM Promotion p
GROUP BY p.platform

Aggregation - HAVING

File: restaurant.php, function: handleMaxRatingGroupedRequest()

SELECT R.type, CAST(AVG(R.rating) AS DECIMAL(10,2)) AS avg_rating, MAX(R.rating) FROM Restaurant R
GROUP BY R.type
HAVING AVG(R.rating) > \$ratingThreshold

Nested aggregation – GROUP BY

File: promotion.php, function: handleMaxBalanceRequest()

)

Screen shots for the GUI after running each query

INSERT operation

Before insertion, the user with id = 1001 has two addresses as below.

	STREETADDRESS	CITY	COUNTRY	ADDRESSTYPE
ı	New St.	Toronto	Ontario	company
	STREETADDRESS New St. 1 Apple Dr	Vancouver	Canada	delivery

After insertion, the insert success message is printed, and now the user 1001's address becomes the picture below.

INSERT into UserAddress VALUES('American St.', 'Seattle', 'U.S.', 'home', 1001)

No results found.

INSERT SUCCESSFULLY.

STREETADDRESS	CITY	COUNTRY	ADDRESSTYPE
New St.	Toronto	Ontario	company
1 Apple Dr	Vancouver	Canada	delivery
American St.	Seattle	U.S.	home

DELETE operation

Before insertion, the user with id = 1001 has two addresses as below.

STREETADDRESS	CITY	COUNTRY	ADDRESSTYPE
New St.	Toronto	Ontario	company
1 Apple Dr	Vancouver	Canada	delivery
American St.	Seattle	U.S.	home

After deletion, the delete success message is printed, and now the user 1001's address becomes the picture below.

DELETE FROM UserAddress WHERE userID = 1001 AND streetAddress = 'New St.' AND city = 'Toronto' AND country = 'Ontario'

No results found.

DELETE SUCCESSFULLY.

STREETADDRESS CITY COUNTRY ADDRESSTYPE

1 Apple Dr Vancouver Canada delivery

American St. Seattle U.S. home

UPDATE operation

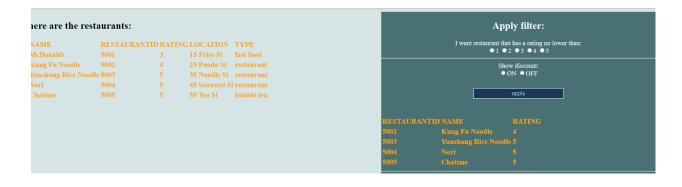
Before updating, the user with id = 1001's U.S. address has a type "home."

After updating the address type, the success message is printed and now the U.S. address type is updated to "company."



Selection

The value chosen is 4 and OFF. The table on the left is the table before the buttonclick and the table in the right is the one produces after the button is clicked.

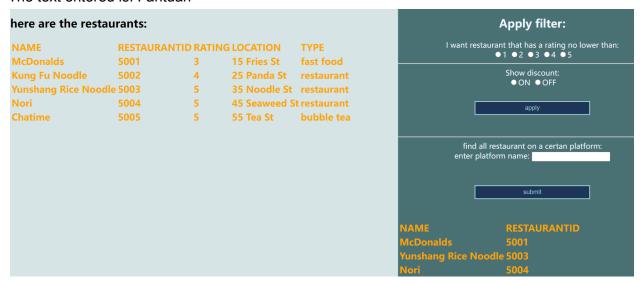


Projection



Join

The text entered is: Fantuan

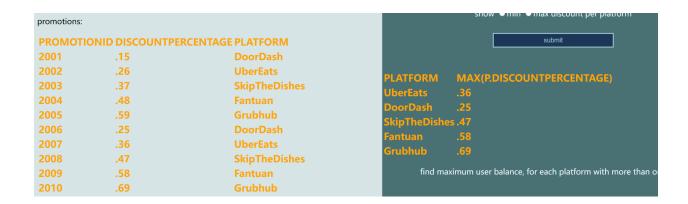


Aggregation - GROUP BY

After clicking the min:



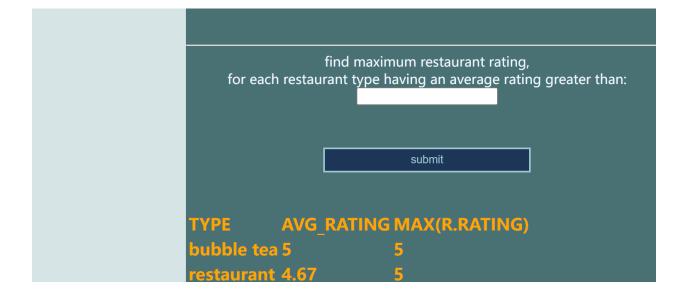
After clicking the max:



Aggregation – HAVING

here are the restaurants: NAME RESTAURANTID RATING LOCATION TYPE **McDonalds** 15 Fries St fast food 5001 25 Panda St restaurant **Kung Fu Noodle** 5002 **Yunshang Rice Noodle 5003** 5 35 Noodle St restaurant 5 **45 Seaweed St restaurant** Nori 5004 Chatime 5005 55 Tea St bubble tea

After clicking (the value entered here is 3):



Nested aggregation – GROUP BY



Division

Before clicking:

JRANTID RATII 3	NG LOCATION	TYPE
3	1E Evice CA	
	15 Fries St	fast food
4	25 Panda St	restaurant
5	35 Noodle St	t restaurant
5	45 Seaweed	St restaurant
5	55 Tea St	bubble tea
5	55 Tea St	bubl

Result given after clicking

