PROJECT REPORT



DATA MANAGEMENT

Group 9:

Chengyuan Huang(cxh173830)

Aritri Biswas (axb171230)

Gopesh Vijayvergiya(gxv170830)

Cynthia Sharon (cxp170920)

Ashi Sharma(axs173132)

Contents

[ABSTRACT 3](#_Toc499847466)

[GRUBHUB: 3](#_Toc499847467)

[ENTITY RELATIONSHIP DIAGRAM (ERD) 3](#_Toc499847468)

[ERD EXPLANATION 4](#_Toc499847469)

[TABLES CREATED 5](#_Toc499847470)

[User 5](#_Toc499847471)

[Orders 5](#_Toc499847472)

[Restaurant 6](#_Toc499847473)

[Cuisine 6](#_Toc499847474)

[Speciality 6](#_Toc499847475)

[Entrée Details 6](#_Toc499847476)

[Menu 6](#_Toc499847477)

[Orderline 6](#_Toc499847478)

[Delivery 7](#_Toc499847479)

[Payment 7](#_Toc499847480)

[Credit Card 7](#_Toc499847481)

[Paypal 7](#_Toc499847482)

[Cash 7](#_Toc499847483)

[Checkout 7](#_Toc499847484)

[Bill 8](#_Toc499847485)

[Tip 8](#_Toc499847486)

[Working hours 8](#_Toc499847487)

[Schedule 8](#_Toc499847488)

[QUERIES 8](#_Toc499847489)

[NEW FEATURES 12](#_Toc499847490)

# ABSTRACT

The objective of this project is to create a database for the Grubhub app. The purpose is to implement all the concepts covered in the class and to create a database that explains all the features of the app.

# GRUBHUB:

Grubhub is an [online and mobile food-ordering](https://en.wikipedia.org/wiki/Online_food_ordering) company that connects dinners with local restaurants. Based in [Chicago](https://en.wikipedia.org/wiki/Chicago), the company has more than 9.18 million active dinners, and more than 55,000 restaurant partners in over 1,100 cities across the [United States](https://en.wikipedia.org/wiki/United_States) and the [United Kingdom](https://en.wikipedia.org/wiki/United_Kingdom)

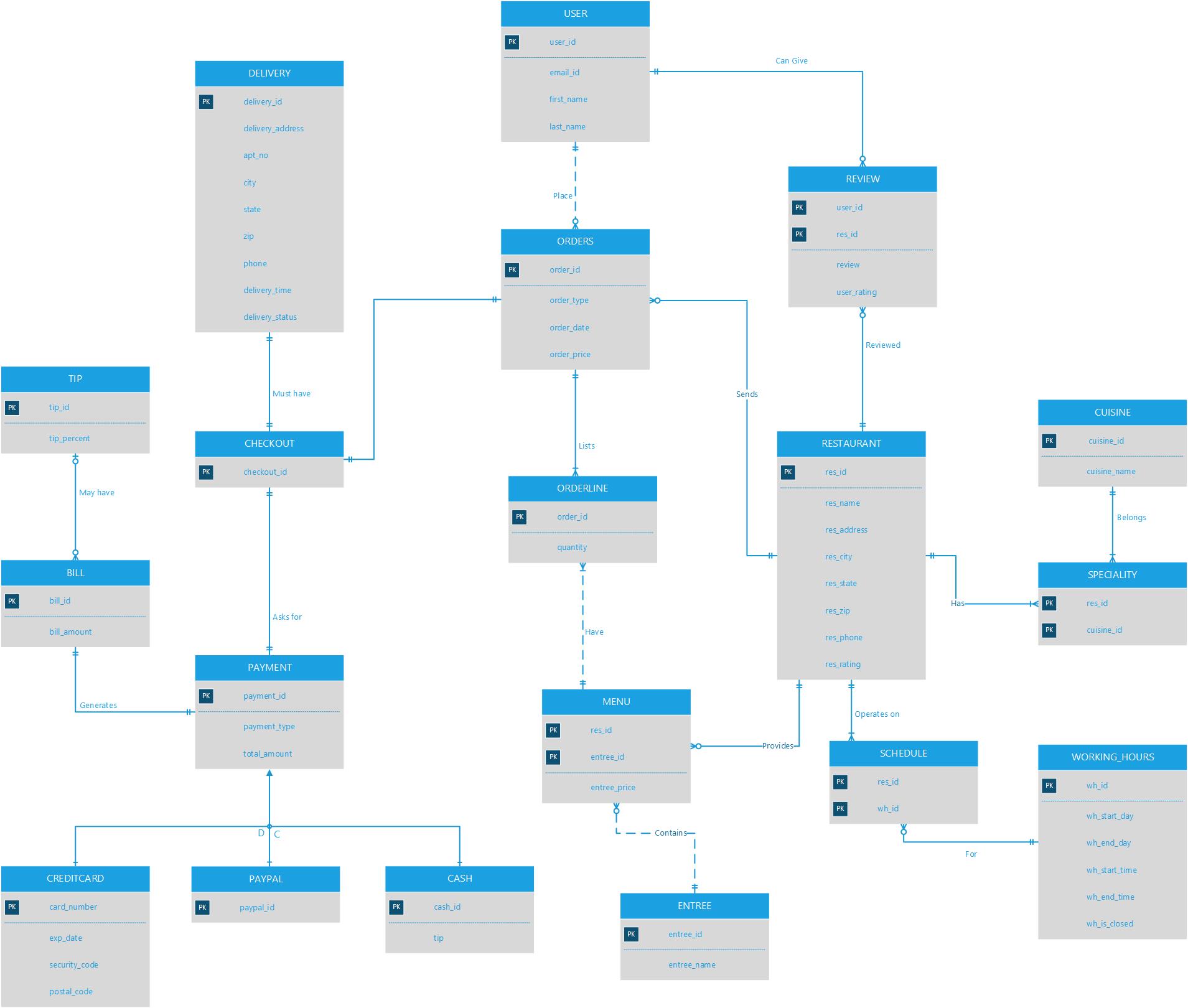
In 1999, Seamless was founded by two lawyers, Jason Finger and an associate, fed up with out-of-date paper menus. In 2004, Grubhub was founded by Matthew Maloney and Michael Evans, two web developers looking for an alternative to paper menus.

In 2013, Grubhub and [Seamless](https://en.wikipedia.org/wiki/Seamless_(company)) merged. The combined organization, Grubhub, went public in April 2014 and trades on the New York Stock Exchange (NYSE) under the ticker symbol "GRUB"

# ENTITY RELATIONSHIP DIAGRAM (ERD)

An entity-relationship diagram (ERD) is a graphical representation of an information system that shows the relationship between people, objects, places, concepts or events within that system. An ERD is a [data modelling](http://searchdatamanagement.techtarget.com/definition/data-modeling) technique that can help define business processes and can be used as the foundation for a [relational database](http://searchsqlserver.techtarget.com/definition/relational-database).

We have used Microsoft Visio to construct the ERD Diagram. The GRUBHUB ERD below shows the relationships of entity sets stored in a database and the relationships between them.



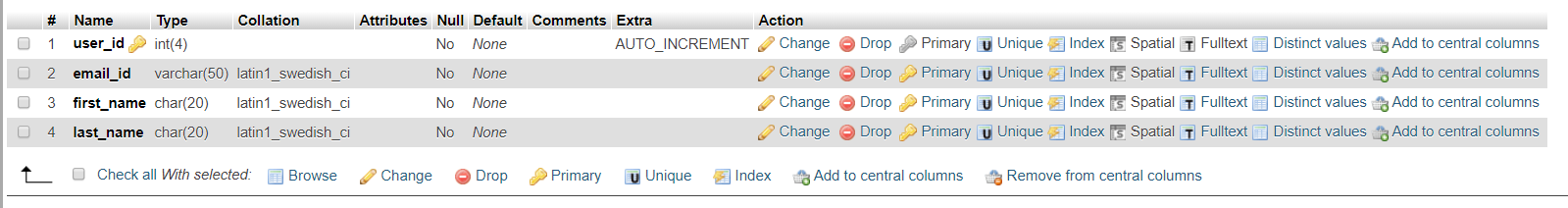
# ERD EXPLANATION

The Grubhub ERD has 17 entities and 3 Subtype entities. The 17 Entities are User, order, Delivery, Tip, Review, Checkout, Orderline, Restaurant, Cuisine, Menu, Speciality, Schedule, Working hours, Entrée, Payment, Bill. The Subtype entities are Credit card, PayPal, Cash.

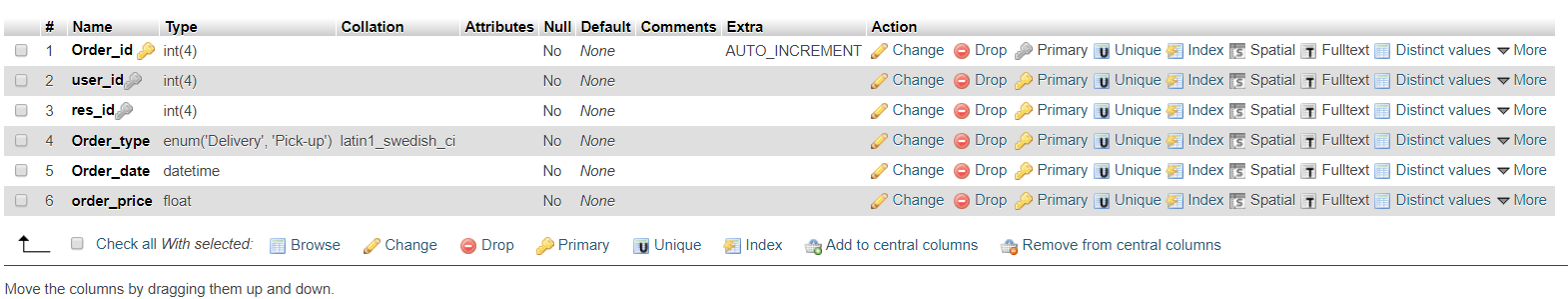
* The important entity of the Grubhub app is the User and the Restaurant. The User entity has attributes such as User\_id, Email\_id, First name, Last name with the User\_id as the primary key. The user can order in no or more restaurant. The User can give no or more reviews and ratings about a restaurant.
* The Restaurant has Restaurant id, Restaurant name, Restaurant Address, Restaurant Phone as its attributes. The Restaurant can have one or more speciality cuisine. The restaurant also has specific working hours specified in its schedule.
* The Order entity has attributes such as Order id , Order type, Order Date with Order Id as the primary key. The order can be placed by only one user and the each order is associated with only one restaurant. The restaurant can take one or many orders.The orders have one checkout id at one time. The order can have one or more orderlines.
* The delivery entity is created with the attributes such as Delivery\_id as the primary key, delivery\_address, delivery\_time, delivery\_status, and delivery\_phone. The Delivery can have only one check out at a time.
* The Payment entity is created with 3 sub type entity. The user can choose to pay in three methods. Credit card, PayPal, Cash. The Payment is associated with only one checkout or only one bill at a time at a time.
* The bill amount may or may not have tip. The bill amount entity has bill\_id, bill\_amount.
* The tip entity has attributes such as tip\_id and tip\_percent.
* The Entrée Entity has a entrée id and entrée price. The entrée can be in more than one menu. But the menu should have at least one entrée. The menu should be connected with only one restaurant.
* The orderline entity has quantity attribute associated with it. The ordeline can be from only one menu and can have only one order id in it.

# TABLES CREATED

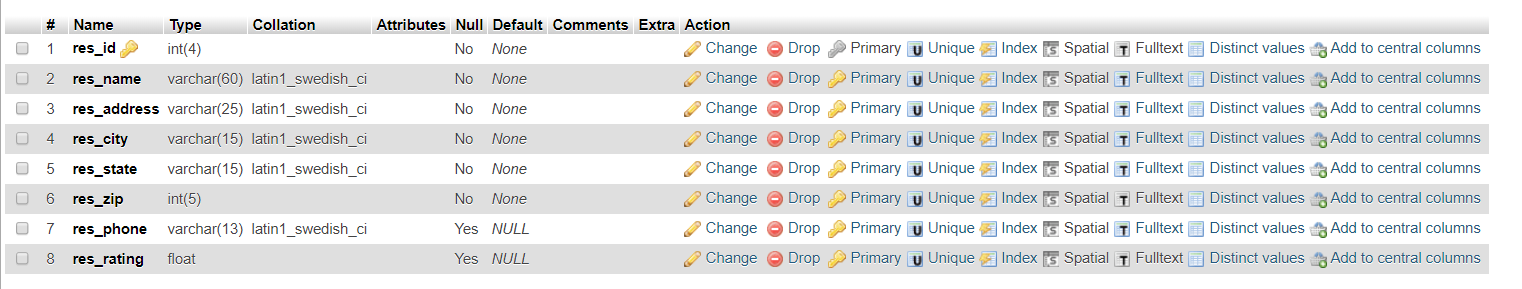
## User



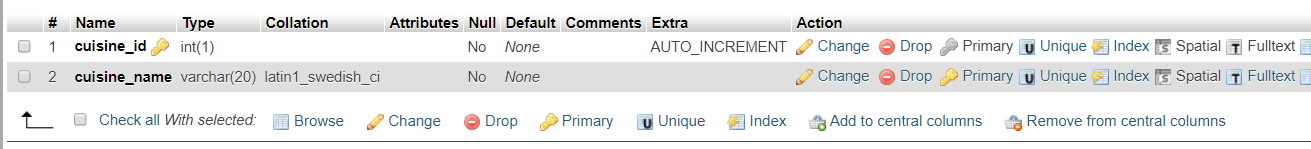
## Orders



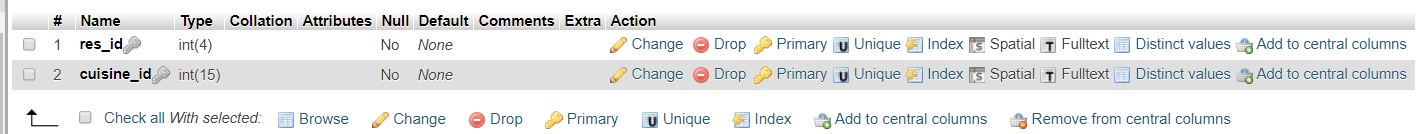
## Restaurant



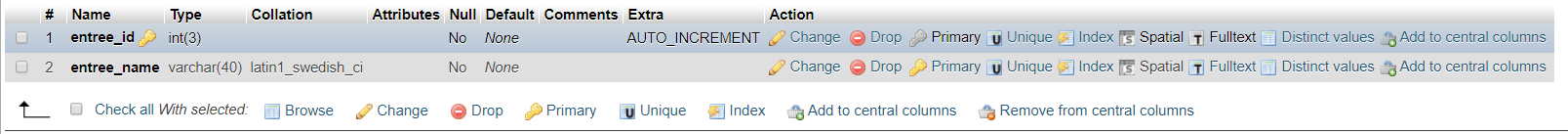
## Cuisine



## Speciality



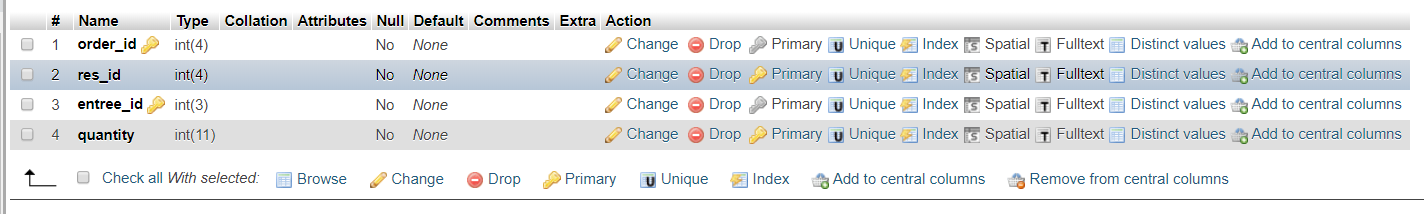
## Entrée Details



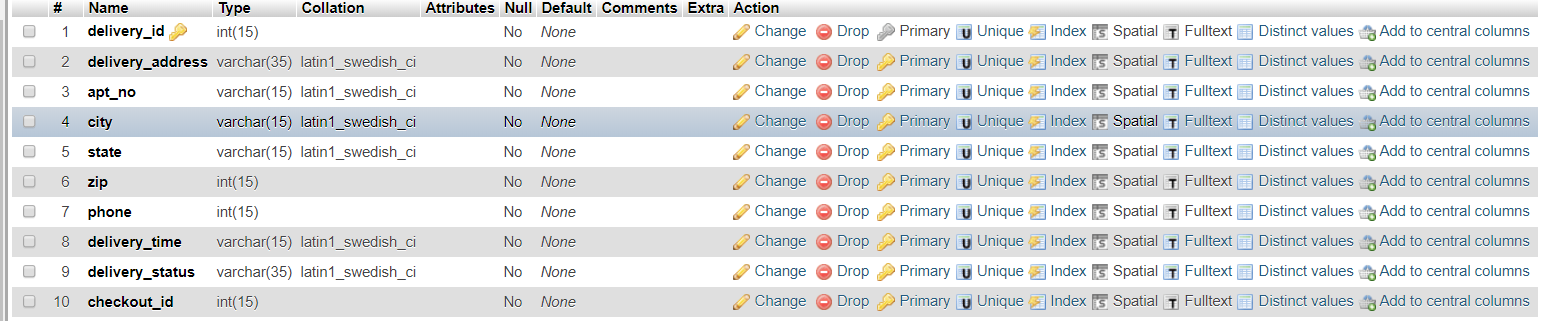
## Menu

## 

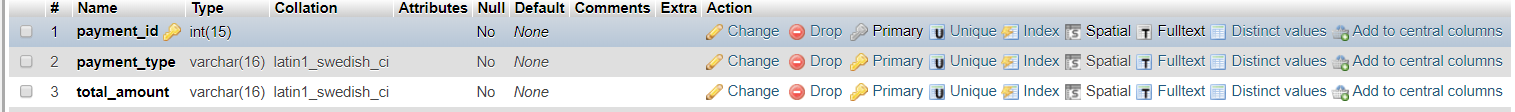
## Orderline



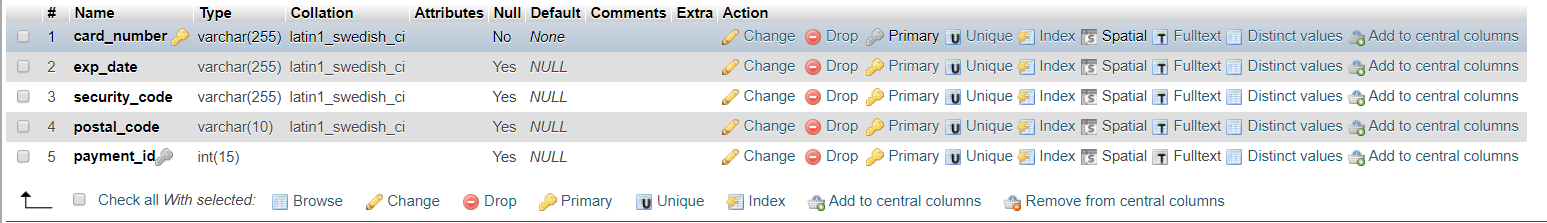
## Delivery



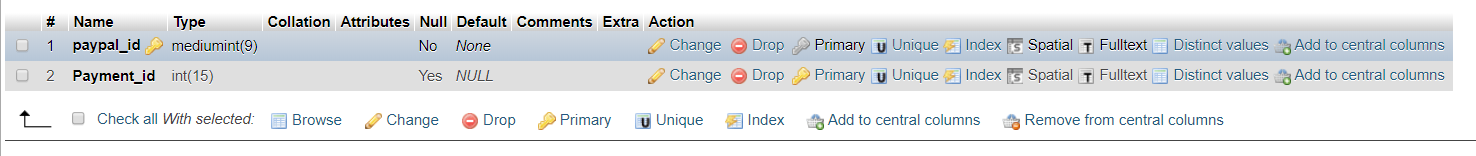
## Payment



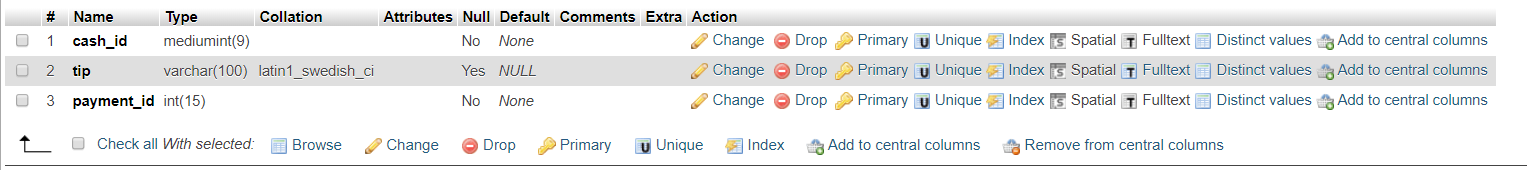
## Credit Card



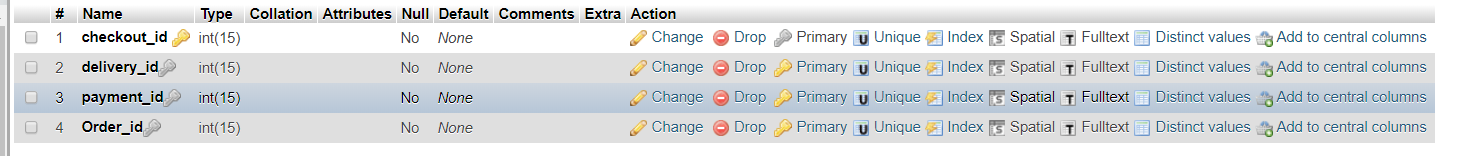
## Paypal



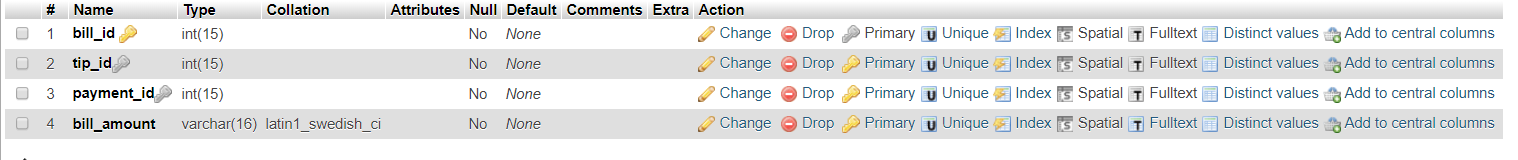
## Cash



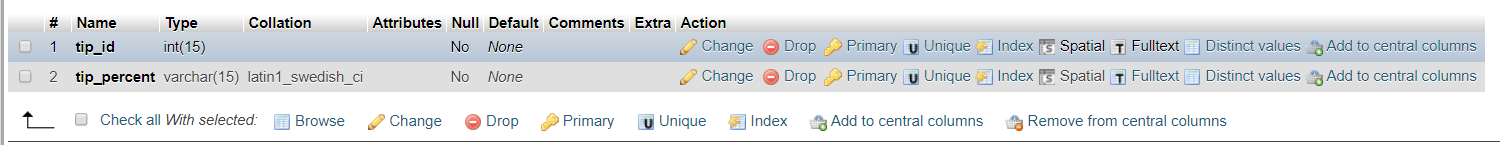
## Checkout



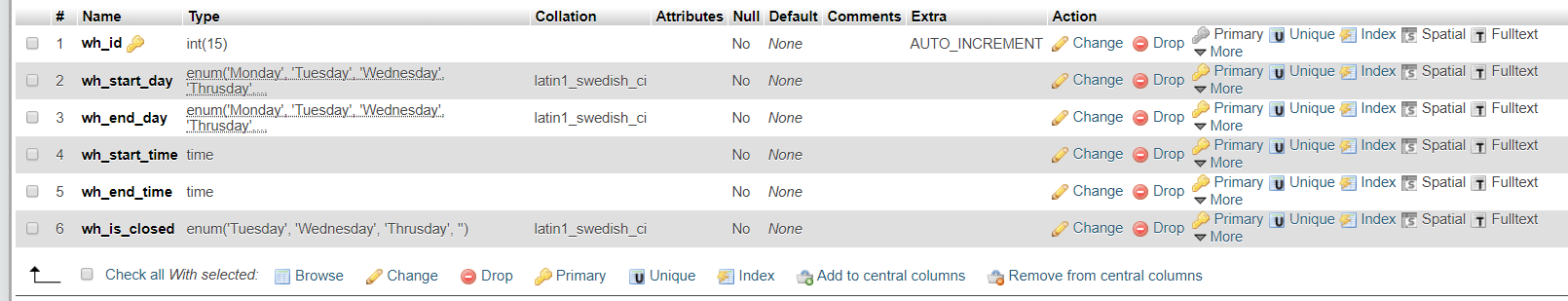
## Bill



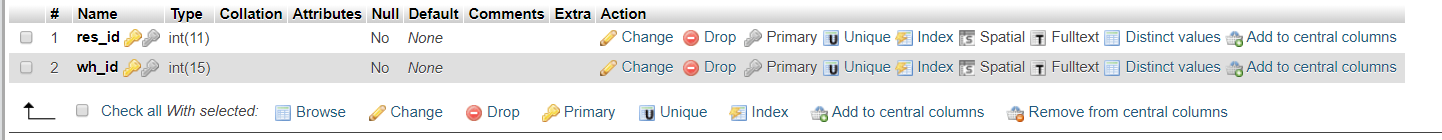
## Tip



## Working hours



## Schedule



# QUERIES

* List all the restaurants with THAI cuisine

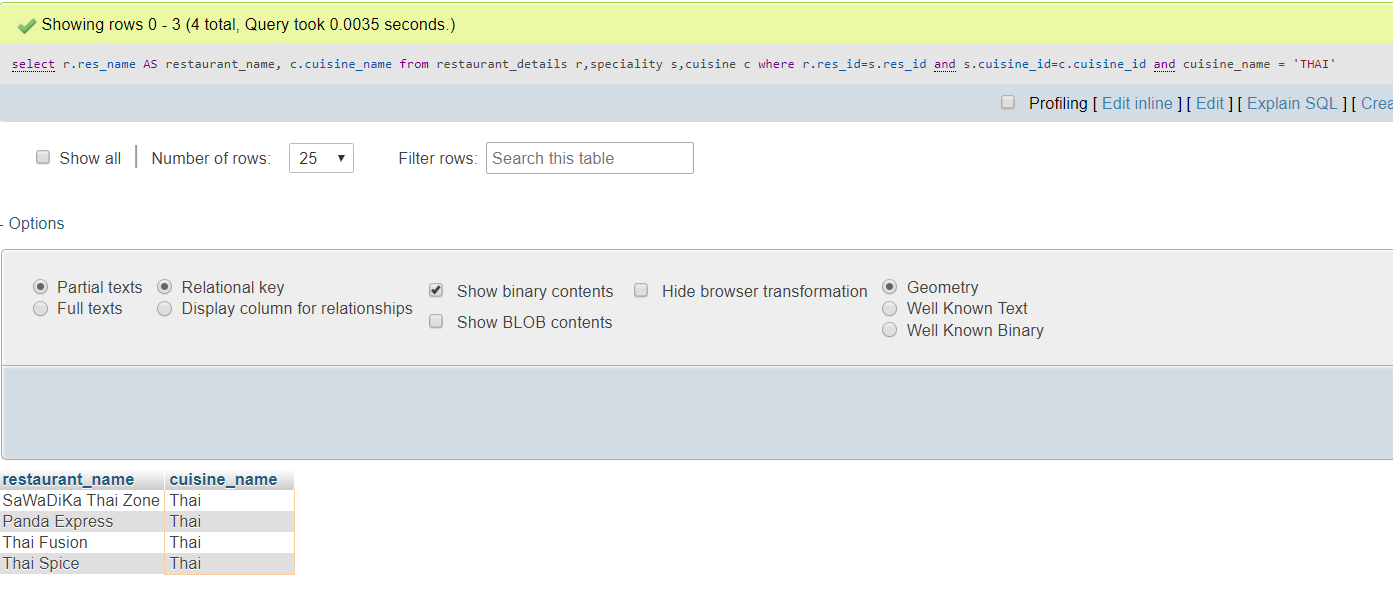
select r.res\_name AS restaurant\_name, c.cuisine\_name

from restaurant\_details r,speciality s,cuisine c

where r.res\_id=s.res\_id

and s.cuisine\_id=c.cuisine\_id

and cuisine\_name = 'THAI'



* List all the restaurants that are closed in Wednesdays or Tuesdays

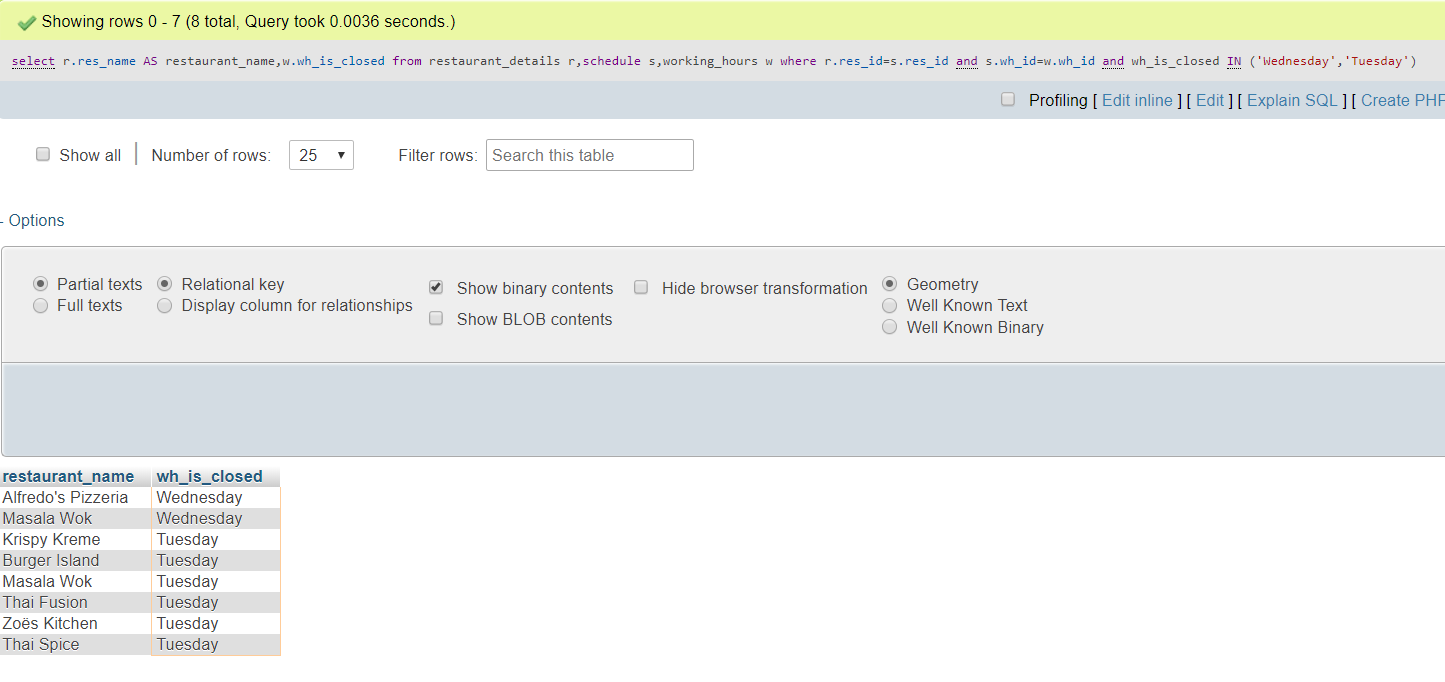
select r.res\_name AS restaurant\_name,w.wh\_is\_closed

from restaurant\_details r,schedule s,working\_hours w

where r.res\_id=s.res\_id

and s.wh\_id=w.wh\_id

and wh\_is\_closed IN ('Wednesday','Tuesday')



* Users who paid by cash

SELECT u.first\_name, u.last\_name, p.payment\_type

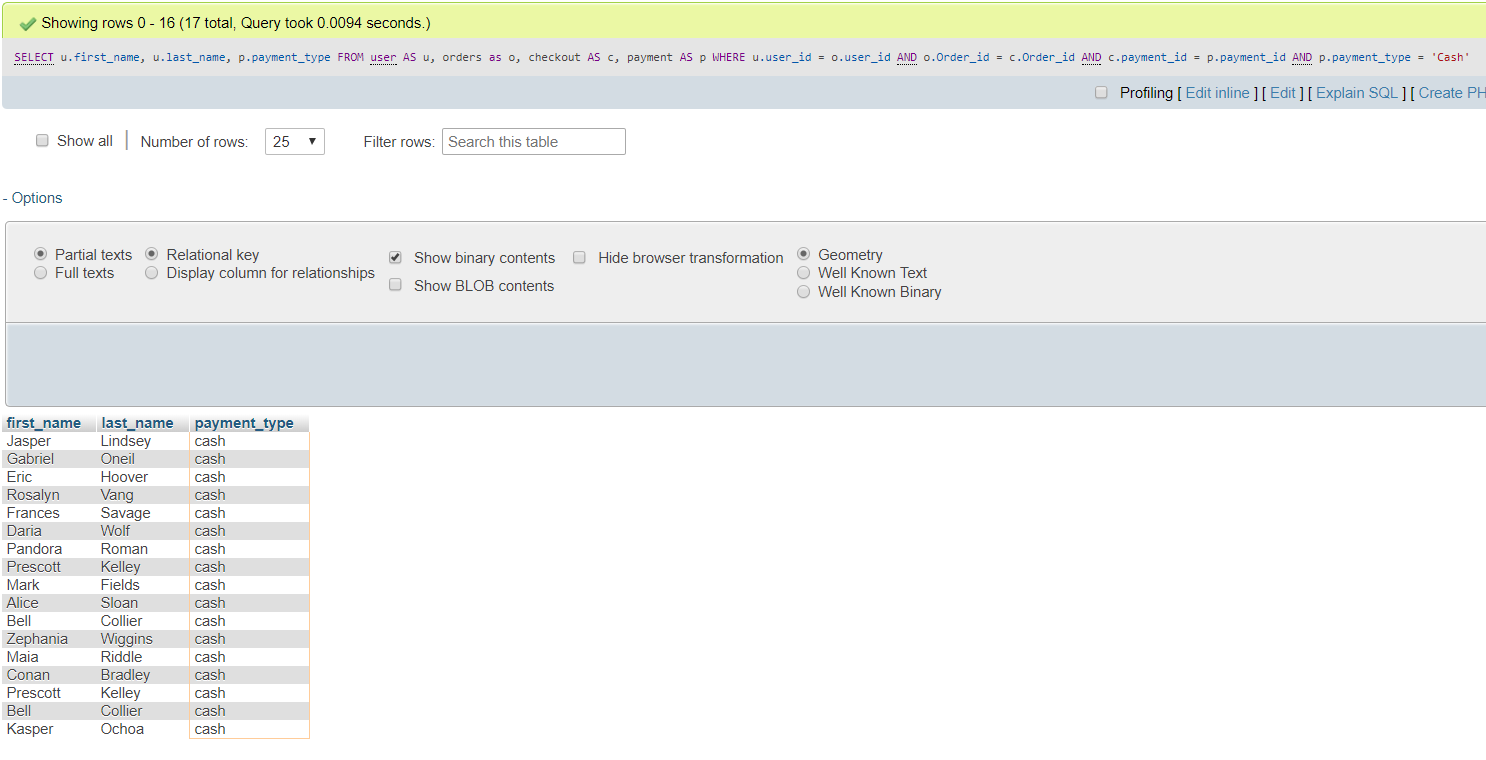
FROM user AS u, orders as o, checkout AS c, payment AS p

WHERE u.user\_id = o.user\_id

AND o.Order\_id = c.Order\_id

AND c.payment\_id = p.payment\_id

AND p.payment\_type = 'Cash'



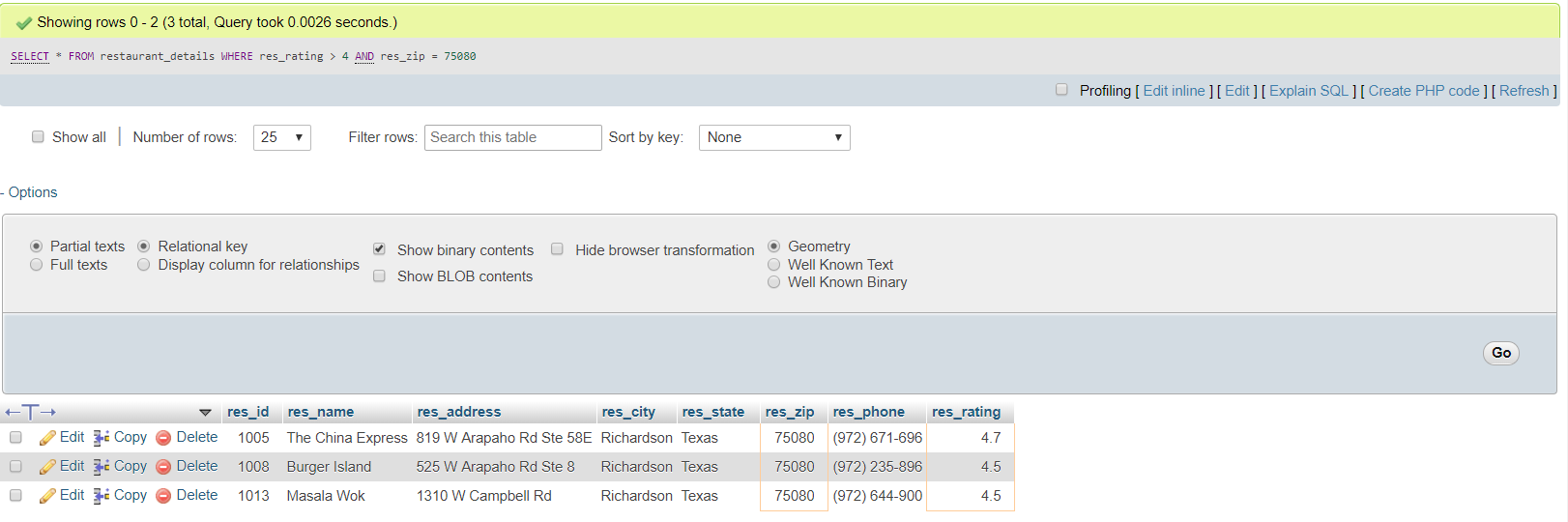
* List restaurant details having rating greater than 4 and belongs to zip code 75080

SELECT \*

FROM restaurant\_details

WHERE res\_rating > 4

AND res\_zip = 75080



* List of users who are having order on the way

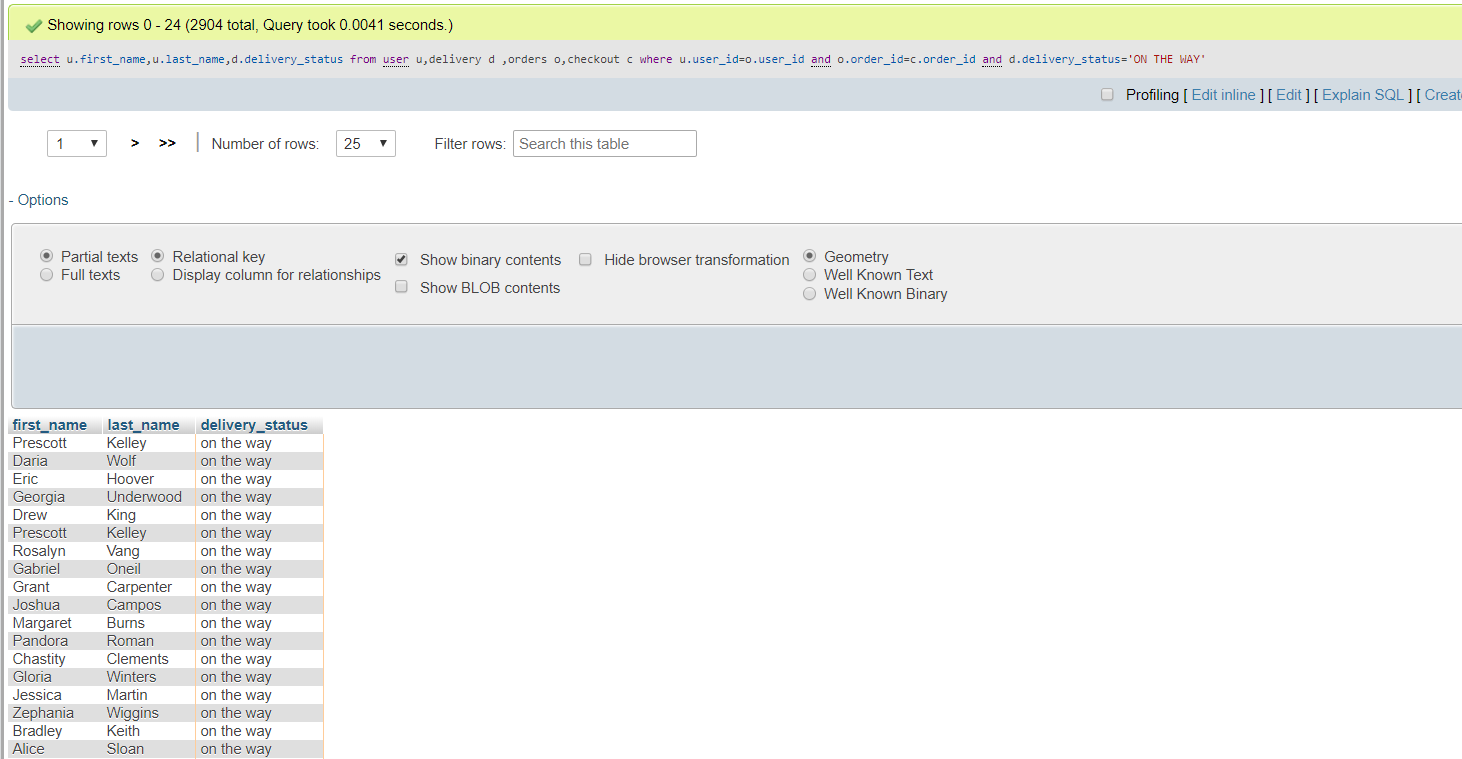
select u.first\_name,u.last\_name,d.delivery\_status

from user u,delivery d ,orders o,checkout c

where u.user\_id=o.user\_id

and o.order\_id=c.order\_id

and d.delivery\_status='ON THE WAY';



* Person who paid with highest tip

select u.first\_name,u.last\_name,t.tip\_percent

from user u,orders o,checkout c, payment p,bill b,tip t

where u.user\_id=o.user\_id

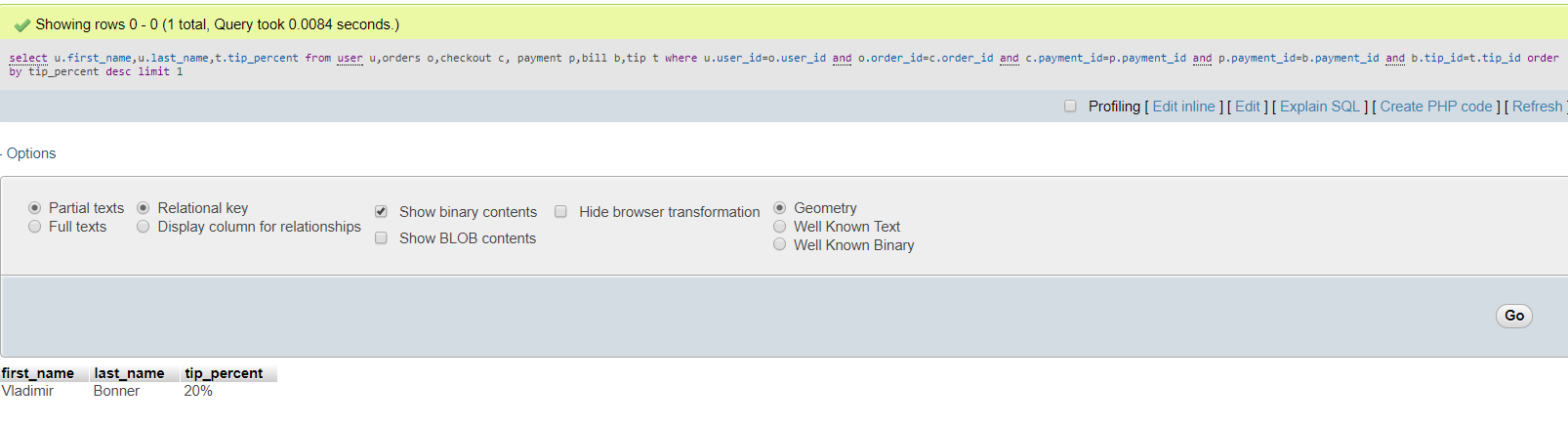
and o.order\_id=c.order\_id

and c.payment\_id=p.payment\_id

and p.payment\_id=b.payment\_id

and b.tip\_id=t.tip\_id

order by tip\_percent desc limit 1;



* City having highest number of restaurants

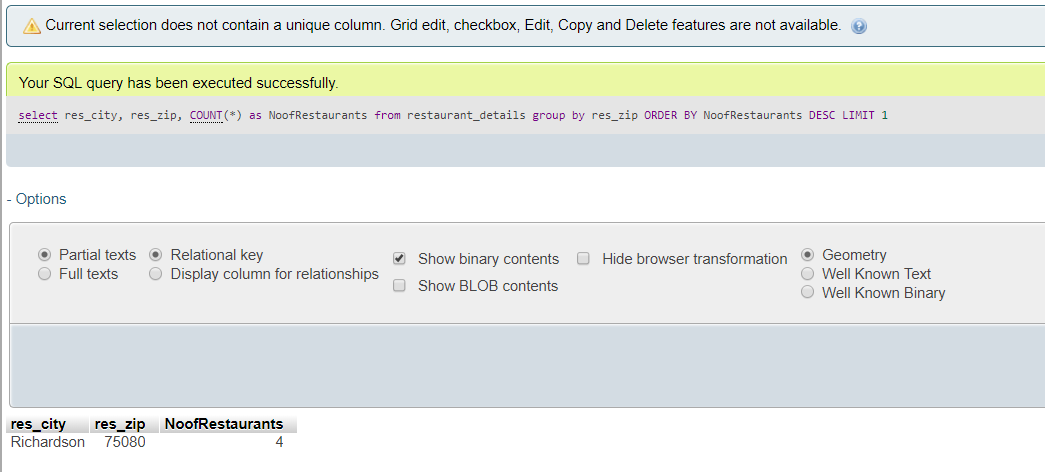
select res\_city, res\_zip, COUNT(\*) as NoofRestaurants

from restaurant\_details

group by res\_zip

ORDER BY NoofRestaurants DESC

LIMIT 1;



* The number, name of each kind of cuisine which get ‘5’ rate

SELECT c.cuisine\_name, COUNT(a.res\_rating) AS number

FROM restuarant\_details AS a, speciality AS b, cuisine AS c

WHERE a.res\_id=b.res\_id AND b.cuisine\_id=c.cuisine\_id

AND a.res\_rating='5'

GROUP BY c.cuisine\_name

* User having the highest number of orders

SELECT u.first\_name, u.last\_name, COUNT(order\_id) AS NumberOfOrders

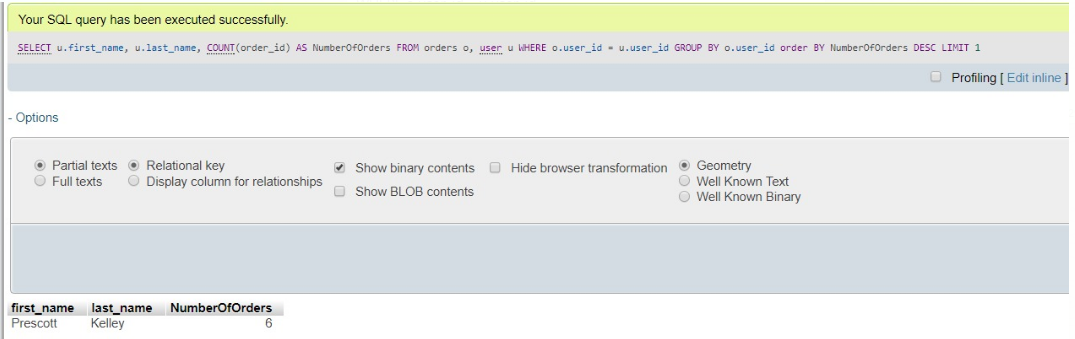
FROM orders o, user u

WHERE o.user\_id = u.user\_id

GROUP BY o.user\_id

order BY NumberOfOrders

DESC LIMIT 1;



# 

# NEW FEATURES

* There are lot of options to pay using smart device like the apple payment and Samsung payment.
* We have decided to include the same in the grub hub feature, where the customer has the option to pay using apple or Samsung money. We have included this in the cash on delivery option.
* The Grub hub delivery man can have the scanning device by which the customer can pay through apple or Samsung money.
* We have added sub type entities for the cash entity. Hence cash is the parent entity and apple and Samsung are the child entity. Apple pay has the apple id and the Samsung pay has the Samsung id.

