Group 7: Project Deliverable 2

(Cao) Yifan

(Dawar) Ravi

(Feng) Wei

(Mohanty) Rutwika

(Sharma) Chirag

MIS 6326.001, Data Management  
Professor Vivek Arora  
November 30th, 2017

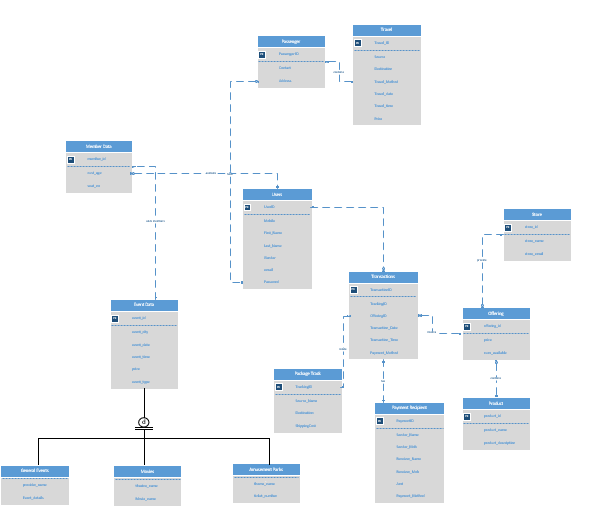
PayTM is one of the most popular mobile applications in India and has revolutionized the e-commerce industry by taking it a step further. Rivalling the likes of Amazon and Flipkart, PayTM has also emerged to be one of the biggest financial players in the industry.

PayTM is designed to be multi-dimensional. It contains various products for the users to choose from. Shopping, Travelling, paying off utility bills, prepaid mobile recharge, etc. are among some of the services that is provided by the application.

In this project, we will be covering four major aspects of the PayTM application. These aspects include the following elements:

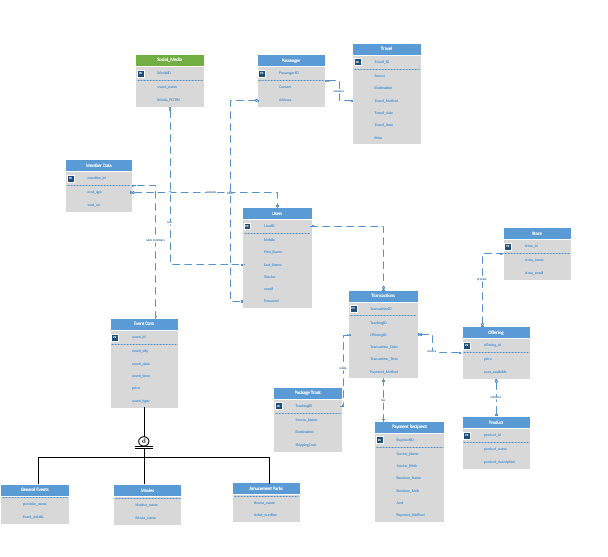
1. Shopping.
2. Payments/Banking.
3. Travelling.
4. Booking Events.

**PayTM Entity Relationship Diagram**



**PayTM Enitity Relationship Diagram With Added Feature**

We have added ‘Social Media’ feature to the application. Using the social media feature, user can share the details of the event on Facebook, Instagram or any other social media. The ‘Social Media’ table is linked to ‘Users’ table. The primary key of the table is MediaID and the foreign key is UserID.



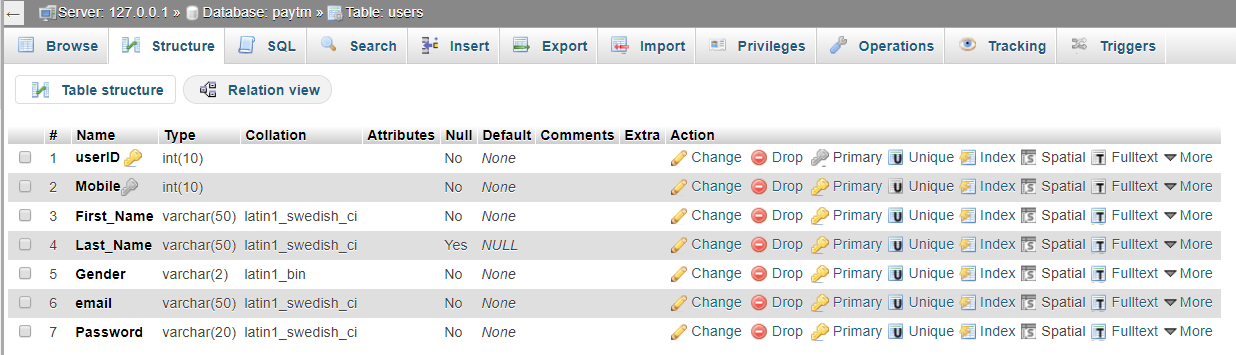
**TABLES**

**Table Name:** Users

The shopping section includes a universal ‘User’ class. The primary purpose of the ‘User’ class will be to maintain data of all the users in the system and fetch the details wherever required.

**Primary Key:** User ID

**Attributes:** Mobile |First\_Name |Last\_Name |Gender|Email|Password

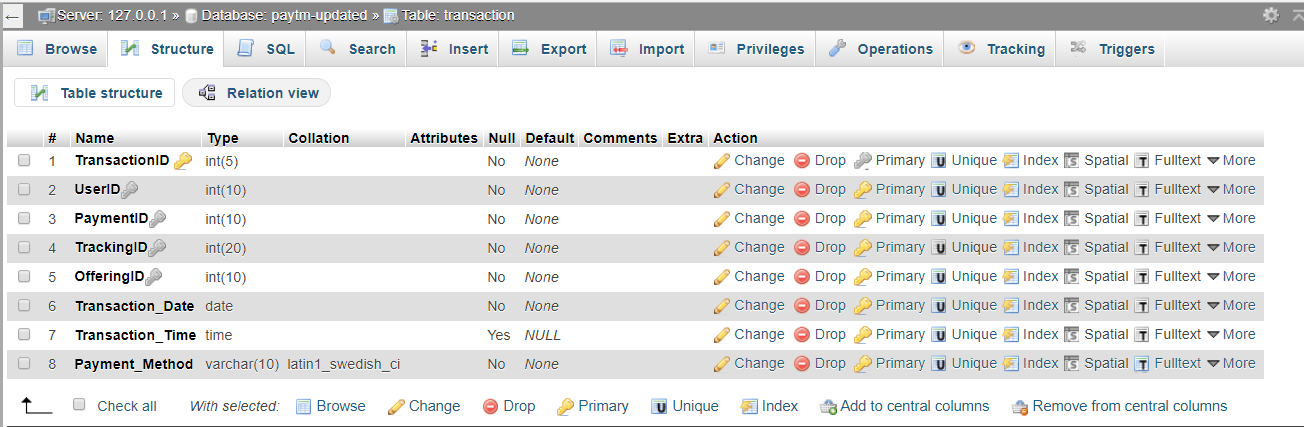


**Table Name:** Transactions

‘Transaction’ class has Transaction ID as a primary key, with other attributes being Transaction amount, transaction date, transaction time and payment method. The primary purpose of the ‘Transaction’ class is to maintain data about the transactions performed by user.

**Primary Key:** TransactionID

**Attributes:** Tracking ID|Offering ID|Transaction\_Date|Transaction\_Time|Payment\_Method

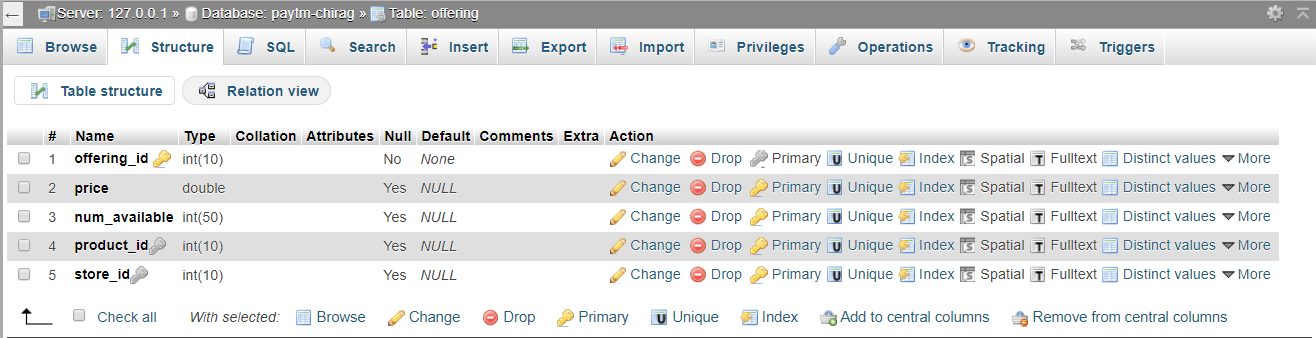


**Table Name:** Offering

‘Offering’ includes a unique offering ID, price and number of available items. The primary purpose of the ‘Offering’ class is to infer the product availability in quantity and price of the product. It also helps in determining stores that provide these products.

**Primary Key:** offering ID

**Attributes:** price | num\_available

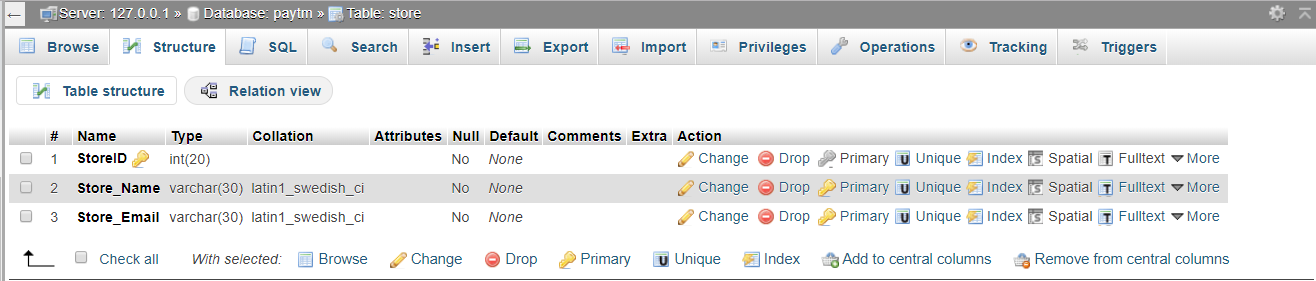


**Table Name:** Store

The purpose of this class is to maintain a list of vendors that can be searched throughout the database.

**Primary Key:** store ID

**Attributes:** store\_name |store\_email

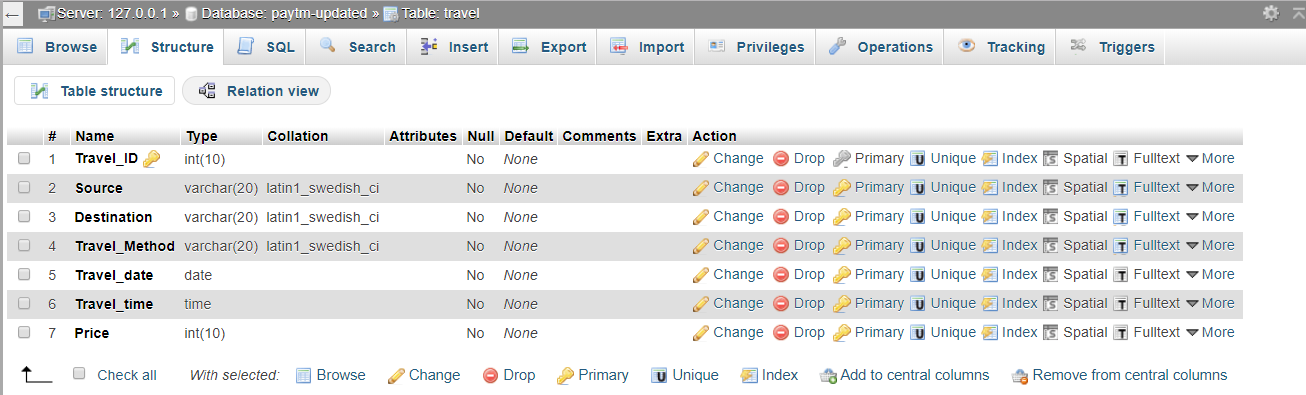


**Table Name:** Travel

The purpose of this class is to maintain details of the travel, travel method that can be flight, bus or train and ticket price of the travel made by the user.

**Primary Key:** Travel\_ID

**Attributes:** Source location |Destination Location |Travel Method |Date |Time |Price

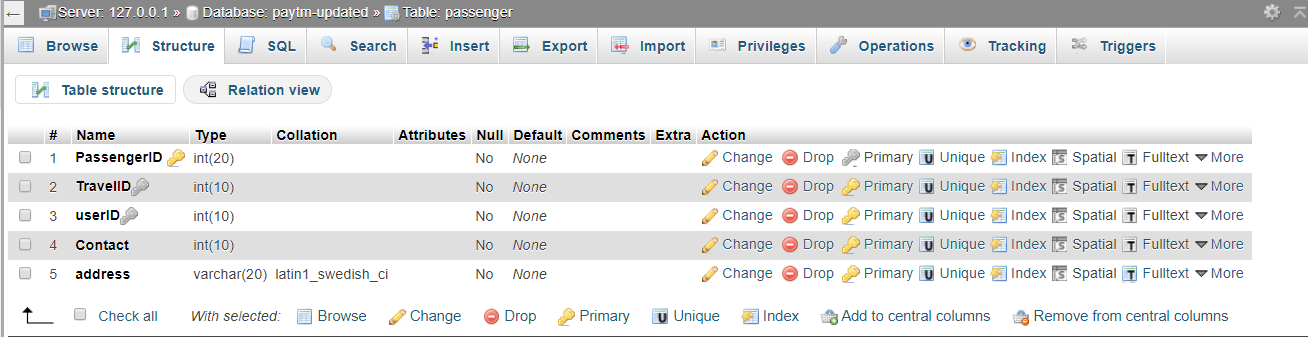


**Table Name:** Passenger

This class keeps track of all the passengers that are going to be travelling from source to destination. This class is also important as it can identify each passenger and link them to their government issued ID. A user can have multiple passengers and a passenger can only be associated to one user.

**Primary Key:** Passenger ID

**Attributes:** Contact|Address

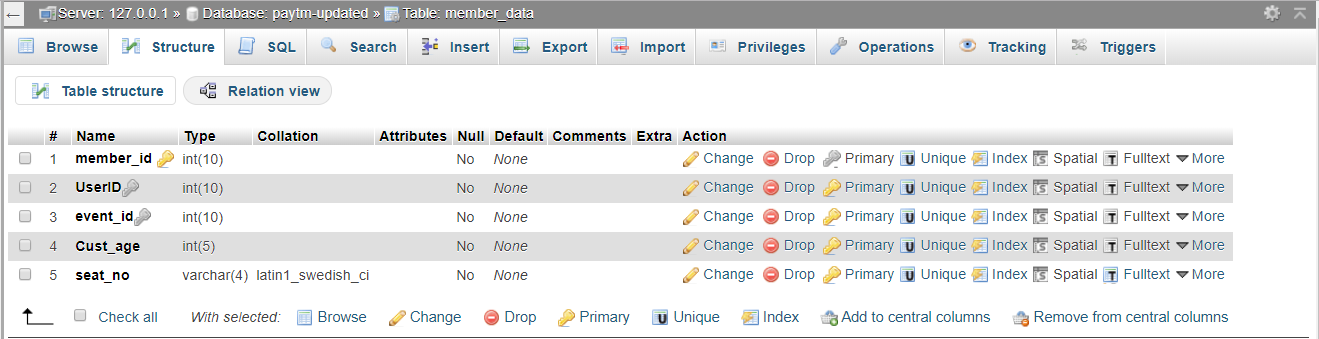


**Table Name:** Member Data

This class keeps track of all the members that are going to the desired event. This class is also important, as it can identify each member and link them to their government issued ID. A user can have multiple members and a member can only be associated to one user.

**Primary Key:** member\_id

**Attributes:** Cust\_age |seat\_no



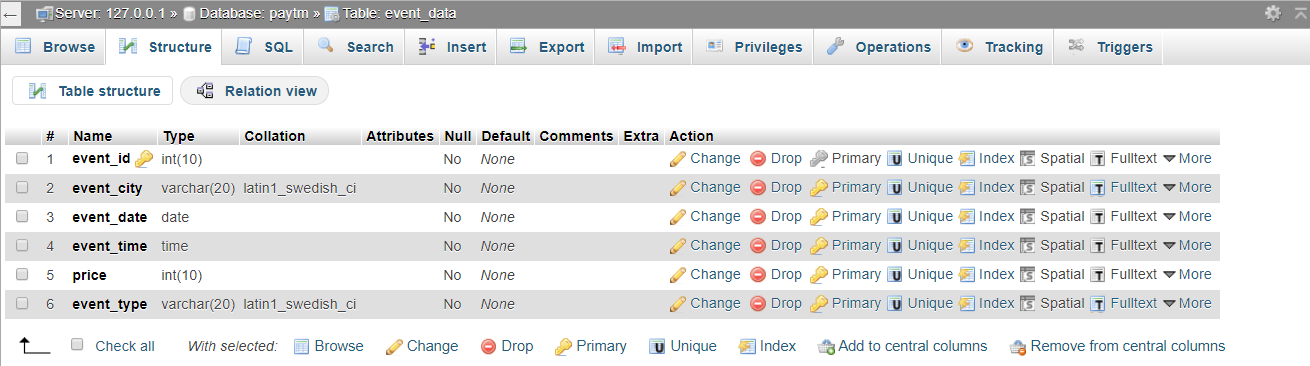
**Table Name:** Event Data

The purpose of this class is to maintain details of the event, event type that can be general events, movies or amusement parks as well as the ticket price of the event.

‘Event Data’ will be a parent class and will consist of four different sub-classes or child classes:

**Primary Key:** event\_id

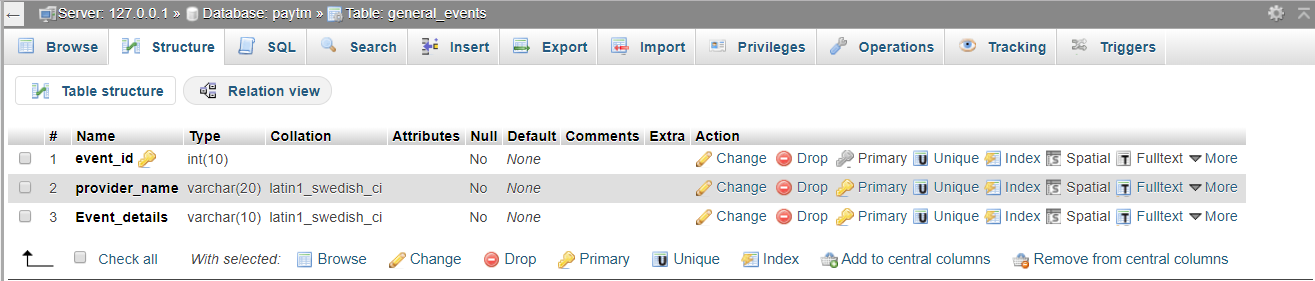
**Attributes:** event\_city |event\_date |event\_time |price |event\_type



**Table Name:** General Events

‘General Events’ is a child class of ‘Event data’. The purpose of this class is to maintain details of the general events.

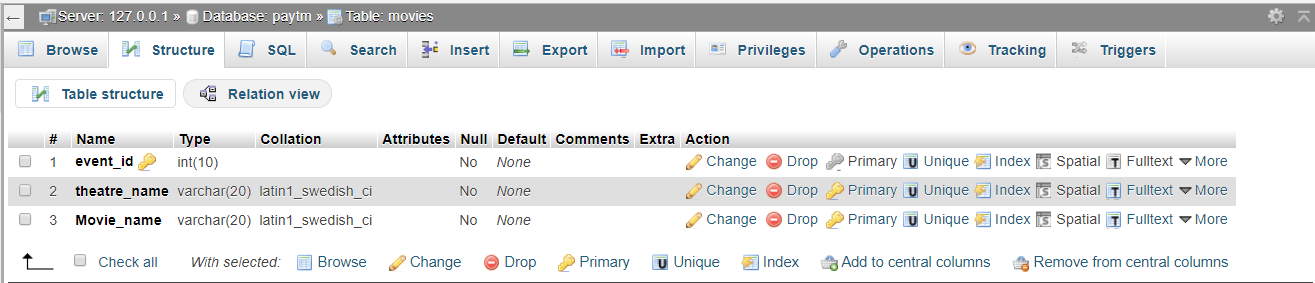
**Attributes:** Provider\_name | Event\_details



**Table Name:** Movies

‘Movies’ is a child class of ‘Event data’. The purpose of this class is to maintain details of the movies for which user can book tickets.

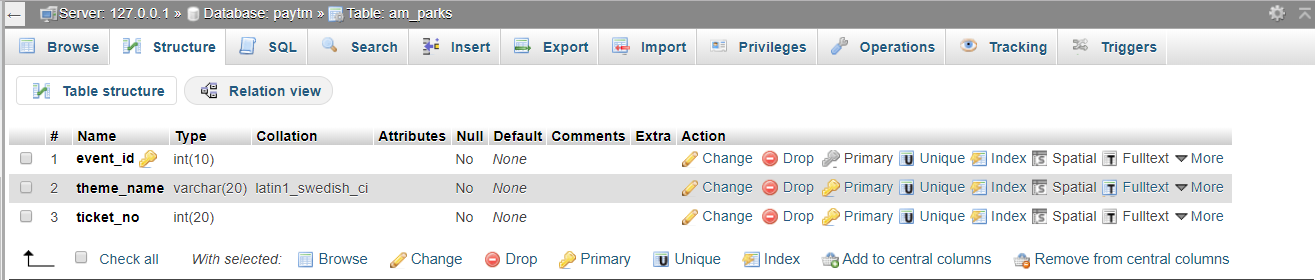
**Attributes:** Theatre\_name | Movie\_name



**Table Name:** Amusement Parks

‘Amusement Parks’ is a child class of ‘Event data’. The purpose of this class is to maintain details of the amusement parks.

**Attributes:** Theme\_name | Ticket\_number

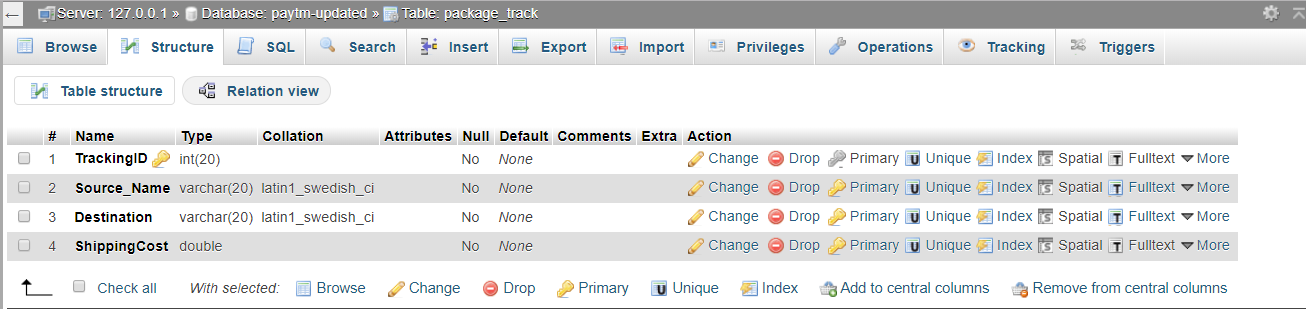


**Table Name:** Package\_Track

‘Package Track’ is created to keep track of all the order shipments that have been initiated by the user.

**Primary Key:** TrackingID

**Attributes:** Source\_name|destination |ShippingCost

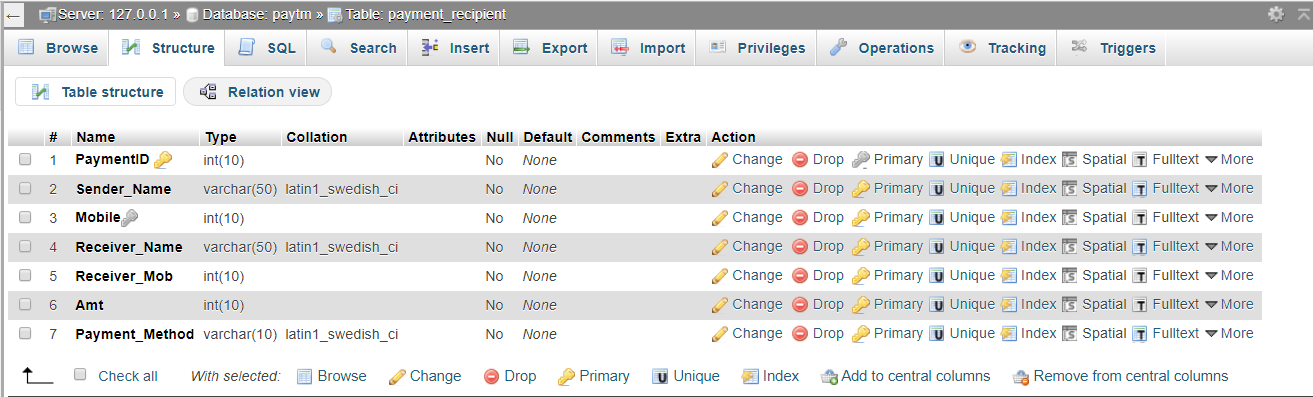


**Table Name:** Payment Recipient

‘Payment Recipient’ class keeps track of the senders, receivers and the amount that is being transferred from one node to another, so that it can be used as a payment log and can be fetched by either party as per their requirements. A user can send money to multiple recipients.

**Primary Key:** PaymentID

**Attributes:** Sender\_Name | Sender\_Mob | Receiver\_Name | Receiver\_Mob | Amt | Payment\_Method

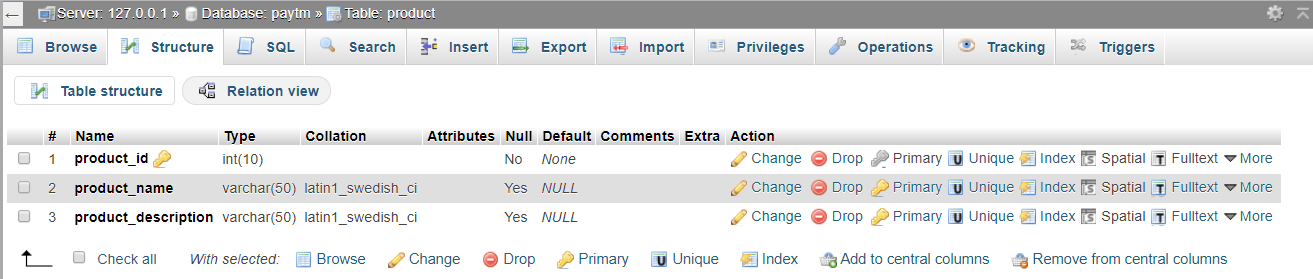


**Table Name:** Product

This class catalogues the products that are being sold by the sellers listed in the PayTM application. A store can have many products, but a product can only be associated to one store.

**Primary Key:** product\_id

**Attributes:** Product\_name | Product\_description

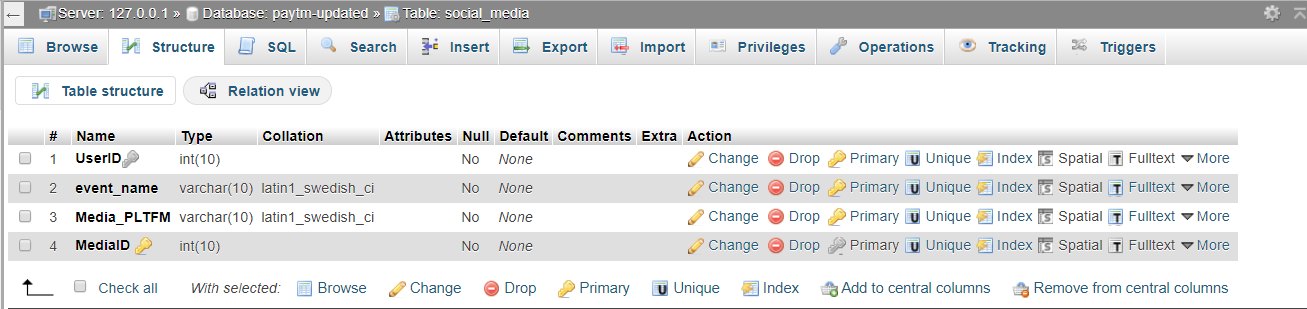


**Table Name:** Social Media

This class keeps a track of the users and the events that they have shared on social media.

**Primary Key:** MediaID

**Attributes:** event\_name |Media\_PPPLTFM



**SCENARIO 1**

Which user has the max shipping cost and for what product item?

SELECT users.userID,users.First\_Name,users.Last\_Name,product.product\_id,product.product\_description,package\_track.ShippingCost

FROM package\_track,transaction,offering,users,product

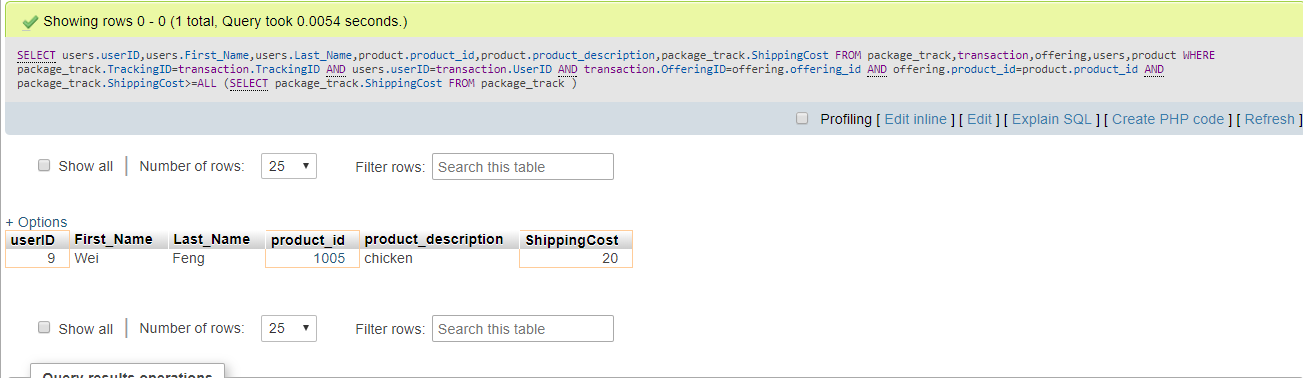
WHERE package\_track.TrackingID=transaction.TrackingID AND users.userID=transaction.UserID AND transaction.OfferingID=offering.offering\_id AND offering.product\_id=product.product\_id

AND package\_track.ShippingCost>=ALL

(SELECT package\_track.ShippingCost

FROM package\_track

)



**SCENARIO 2**

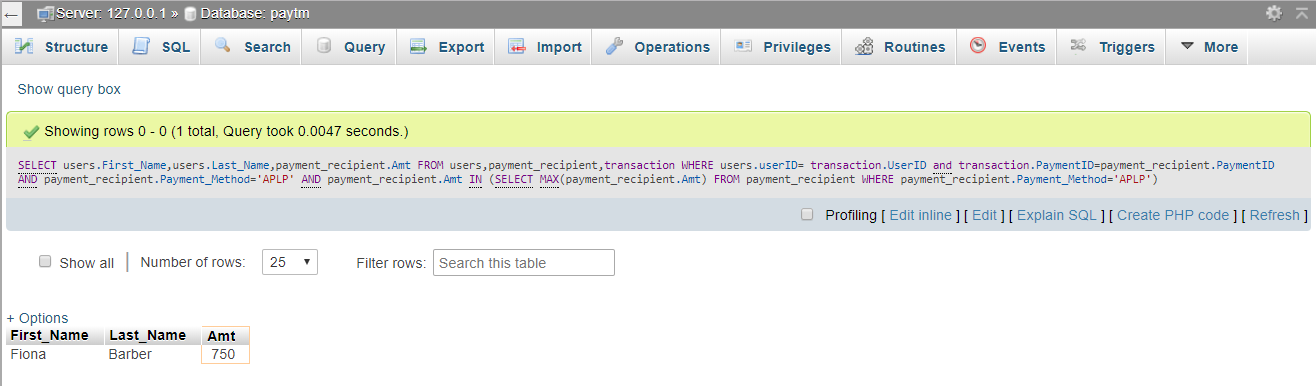
Name of users and the maximum amount that used Apple Pay to send money to other user?

SELECT users.First\_Name,users.Last\_Name,payment\_recipient.Amt FROM users,payment\_recipient,transaction WHERE users.userID= transaction.UserID and transaction.PaymentID=payment\_recipient.PaymentID AND payment\_recipient.Payment\_Method='APLP' AND payment\_recipient.Amt IN

(SELECT MAX(payment\_recipient.Amt)

FROM payment\_recipient

WHERE payment\_recipient.Payment\_Method='APLP')



**SCENARIO 3**

Which stores sell the most products?

SELECT \* FROM(

SELECT store.Store\_Name,store.StoreID,COUNT(transaction.TransactionID) AS sale

FROM transaction,offering,store

WHERE transaction.OfferingID=offering.offering\_id AND offering.store\_id=store.StoreID

GROUP BY store.Store\_Name,store.StoreID) AS t1

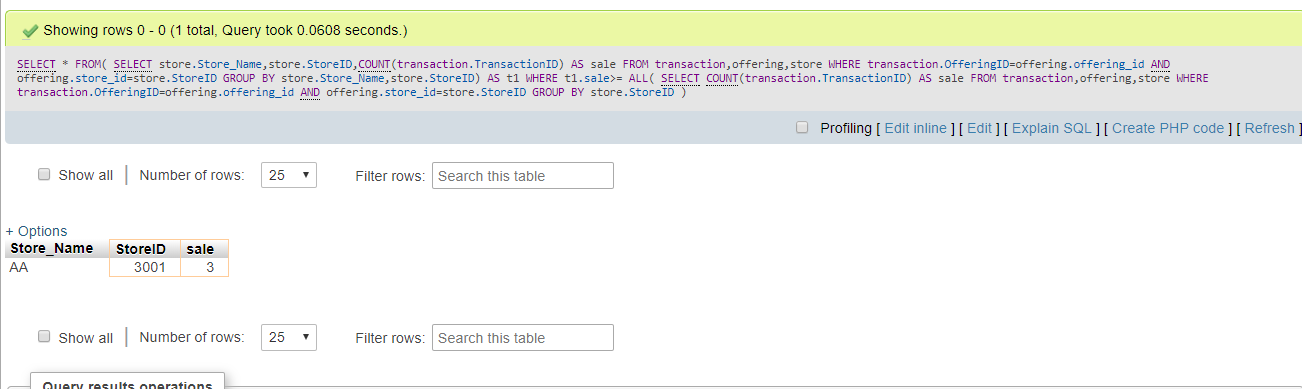
WHERE t1.sale>= ALL(

SELECT COUNT(transaction.TransactionID) AS sale

FROM transaction,offering,store

WHERE transaction.OfferingID=offering.offering\_id AND offering.store\_id=store.StoreID

GROUP BY store.StoreID

)

**SCENARIO 4**

Which user used flight to travel?

SELECT users.First\_Name,users.Last\_Name,travel.Travel\_Method

FROM users,passenger,travel

WHERE users.userID=passenger.userID AND passenger.TravelID=travel.Travel\_ID AND travel.Travel\_Method='Flight'



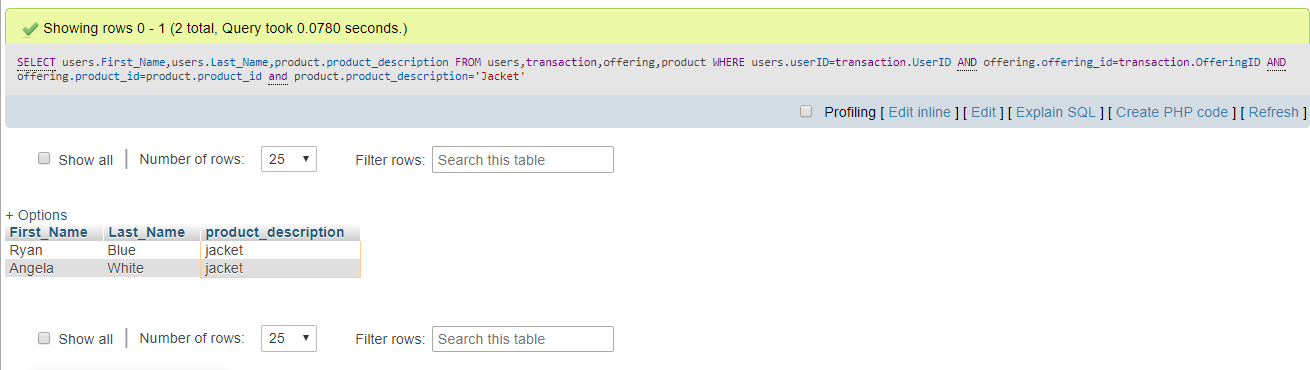
**SCENARIO 5**

Which user bought Jackets?

SELECT users.First\_Name,users.Last\_Name,product.product\_description

FROM users,transaction,offering,product

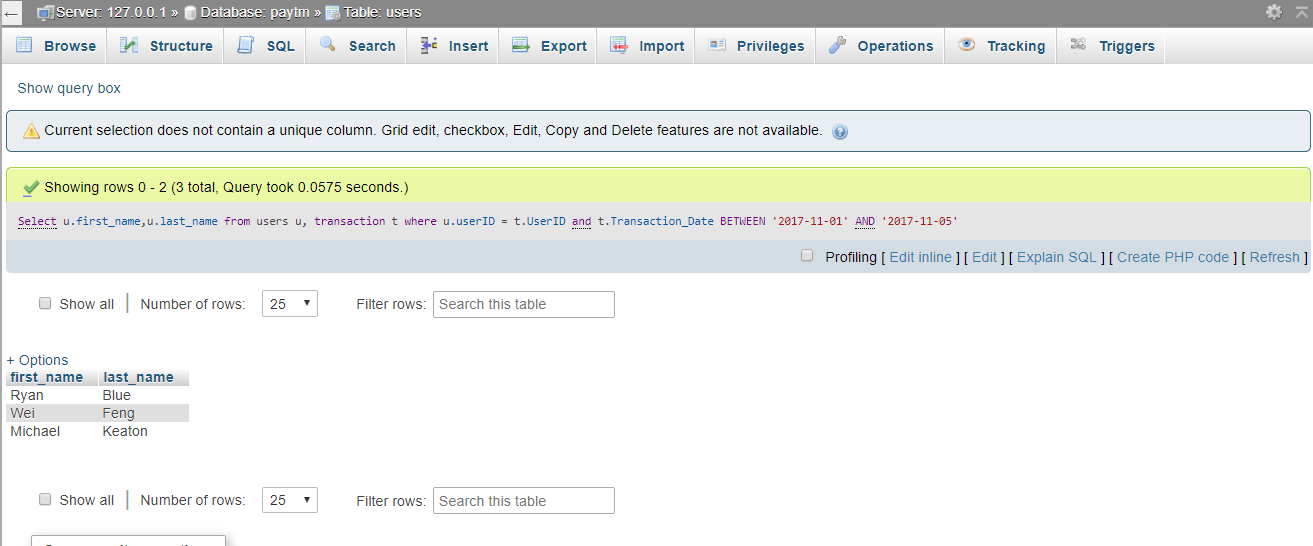
WHERE users.userID=transaction.UserID AND offering.offering\_id=transaction.OfferingID AND offering.product\_id=product.product\_id and product.product\_description='Jacket'



**SCENARIO 6**

Name of the users who made transactions between a specific date?

Select u.first\_name,u.last\_name from users u, transaction t where u.userID = t.UserID and t.Transaction\_Date BETWEEN '2017-11-01' AND '2017-11-05'



**SCENARIO 7**

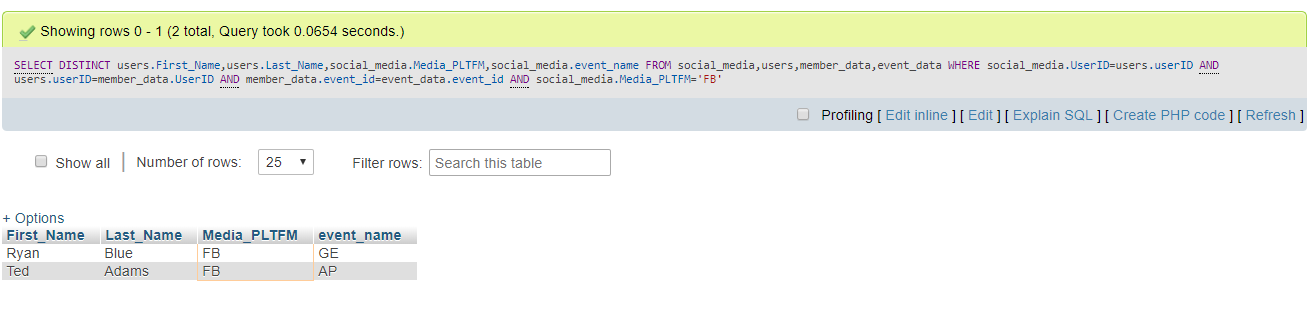
Name of the users who used FB to broadcast their events?

SELECT DISTINCT users.First\_Name,users.Last\_Name,social\_media.Media\_PLTFM,social\_media.event\_name

FROM social\_media,users,member\_data,event\_data

WHERE social\_media.UserID=users.userID AND users.userID=member\_data.UserID AND member\_data.event\_id=event\_data.event\_id

AND social\_media.Media\_PLTFM='FB'



**SCENARIO 8**

Max Number of members associated to a user going to a specific event?

Select u.userID, u.first\_name, u.last\_name, m.event\_id, e.event\_type, count(m.member\_id) AS NUMBER\_OF\_MEMBERS from users u, member\_data m, event\_data e where u.userID = m.UserID and m.event\_id = e.event\_id group by m.UserID limit 1

