There are a lot of SQL queries in our website, ranging from simple, to complex join queries. To show the main functionality we will describe the more complex queries done.

**Check\_for\_match.php:**

In our checking for match function, we must join two difference city tuples to a trip and then to a request:

*SELECT Dep.postalCode AS depPC, Dest.postalCode AS destPC, trip.departureDate,*

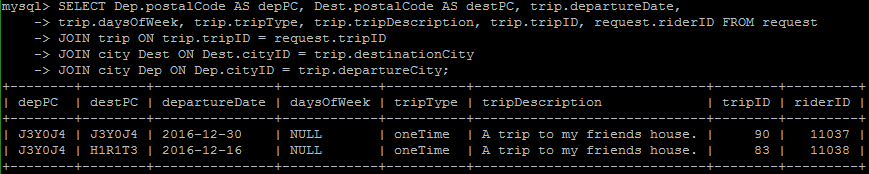
*trip.daysOfWeek, trip.tripType, trip.tripDescription, trip.tripID, request.riderID*

*FROM request*

*JOIN trip ON trip.tripID = request.tripID*

*JOIN city Dest ON Dest.cityID = trip.destinationCity*

*JOIN city Dep ON Dep.cityID = trip.departureCity*

The resulting sql statement on the mysql server looks like:

The query allows us to distinctly select the two difference postal codes from the departure city and destination city, and shows us the rest of the request and trip info.

After this is completed we do another sql statement looking for the info from offer

For a given request trip type, and riderID:

*SELECT \* FROM offer*

*JOIN trip ON trip.tripType = '$requestTripType' AND trip.tripID = offer.tripID*

*AND offer.driverID <> $riderID*

*JOIN city ON city.cityID = trip.departureCity*

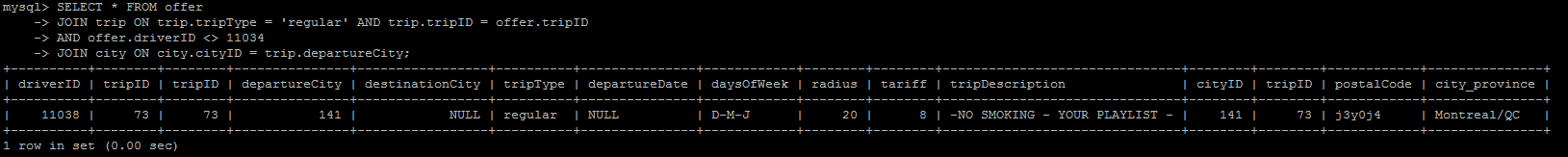
This sql statement will join an offer to its trip, where the driver is not equal to the rider found in the first sql statement and the trip type is the same as the first sql statement.

For a trip type of ‘regular’, and where the driverID is not 11034:

*SELECT \* FROM offer*

*JOIN trip ON trip.tripType = 'regular' AND trip.tripID = offer.tripID*

*AND offer.driverID <> 11034*

*JOIN city ON city.cityID = trip.departureCity;*

**Admin.php**

The admin page gets almost all the information of the database. We will show a query as an example of this:

Showing all members, and their drivers licence status and insurance coverage.

For all members exluding the admin that is viewing the page:

*SELECT \* FROM member JOIN driver ON member.memberID = driver.memberID AND member.memberID <> $userID*

For all members:

*SELECT \* FROM member JOIN driver ON member.memberID = driver.memberID*

mysql> SELECT member.memberID, name, address, email, accountBalance, status, privilege, DOB, lastActivity, driversLisence, insuranceStatus, rating FROM member JOIN driver ON member.memberID = driver.memberID;

+----------+-------------------+---------------------+--------------------+----------------+--------+-----------+------------+---------------------+----------------+-----------------+--------+

| memberID | name | address | email | accountBalance | status | privilege | DOB | lastActivity | driversLisence | insuranceStatus | rating |

+----------+-------------------+---------------------+--------------------+----------------+--------+-----------+------------+---------------------+----------------+-----------------+--------+

| 11034 | Michael Scalera | 578 rue Jean Rivard | michael@test.com | 150.704 | 1 | 1 | 1992-01-06 | 2016-12-06 17:55:46 | 1 | 1 | 4 |

| 11035 | Christian Kellner | 555 sherbrooke | chris@test.com | 400 | 1 | 0 | 1994-05-05 | 2016-12-06 17:45:45 | 1 | 1 | 5 |

| 11036 | Adam Rais | 150 st catherine | adam@test.com | 250 | 0 | 0 | 1996-03-15 | 2016-12-06 13:22:16 | 1 | 0 | 5 |

| 11037 | Pierre Saouma | 593 Bishop | pierre@hotmail.com | 306.808 | 1 | 1 | 1980-06-14 | 2016-12-06 17:46:43 | 1 | 1 | 4 |

| 11038 | Subhi el-Farram | 405 Laval | subhi@gmail.com | 351.463 | 1 | 0 | 1990-03-16 | 2016-12-06 17:46:31 | 1 | 1 | 5 |

| 11040 | Aarij | Montreal/qc | aarij@gmail.com | 178.206 | 1 | 0 | 1990-12-03 | 2016-12-06 17:05:55 | 1 | 1 | 5 |

| 11041 | Arshad Saidoo | 100 cote vertu | arshad@test.com | 400 | 1 | 0 | 1994-12-13 | 2016-12-06 16:04:00 | 1 | 1 | 5 |

+----------+-------------------+---------------------+--------------------+----------------+--------+-----------+------------+---------------------+----------------+-----------------+--------+

7 rows in set (0.00 sec)

There are also a lot of inserts and updates done in the website, we will show the code that we use to insert an offer into the system, an example result is viewable from the user manual:

*//Insert Departure City*

*$sql = "INSERT INTO city (postalCode, city\_province)*

*VALUES ('$postalCode', '$city\_province')";*

*$conn->query($sql);*

*//SaveID of Departure city*

*$sql = "SET @departureCityID = LAST\_INSERT\_ID()";*

*$conn->query($sql);*

*if(isset($\_POST['loginReg']))*

*{*

*//Insert Into Trip*

*$sql = "INSERT INTO trip (departureCity, destinationCity, tripType, departureDate, daysOfWeek, radius, tariff, tripDescription)*

*VALUES(@departureCityID, null, '$tripType', null,'$daysOfWeek', $radius , $rate,'$tripDescription')";*

*}*

*else*

*{*

*$sql = "INSERT INTO trip (departureCity, destinationCity, tripType, departureDate, daysOfWeek, radius, tariff, tripDescription)*

*VALUES(@departureCityID, null, '$tripType', '$departureDate' ,null, $radius,$rate,'$tripDescription')";*

*}*

*$conn->query($sql);*

*$sql = "SET @tripID = LAST\_INSERT\_ID()";*

*$conn->query($sql);*

*$sql = "UPDATE city SET tripID = @tripID WHERE cityID = @departureCityID";*

*$conn->query($sql);*

*$sql = "INSERT INTO offer VALUES ($userID, @tripID)";*

*$conn->query($sql);*

Because each id of offer/request/city/trip increment automatically, we use the LAST\_INSERT\_ID() function to store those ids and update the tables that require them as foreign key contraints.