TABLE OF CONTENTS

1. OVERVIEW

1.1. About Food Delivery service

According to Euromonitor International, food delivery via app is expected to reach 38 million USD, growing 11% on average in next 5 years. Kantar TNS estimates that the revenue of the online food delivery market could reach 449 million USD by 2023. These days, the F&B industry in general and the restaurant industry no longer focus on "fighting" too much based on product quality, they began to shift direction and gradually compete fiercely through the quality of customer experience they provide. Concretely, in this case, it is the experience of using Food Delivery service for diners.

Using Food Delivery service is more and more popular compared to traditional business. According to statistics of Q&Me, 75% of the total people surveyed in Vietnam use the restaurant's Food Delivery service. The more positive the delivery experience, the higher the percentage of shoppers who will choose the restaurant again. According to Delivering Consumer Choice by MetaPack, up to 96% of consumers return home if they feel that delivery experience is positive.

Currently, people's use of Food Delivery is increasingly driven by the convenience and rise of many platforms such as GrabFood, Go-Food, LoShip, Baemin, etc. Nowadays, busy lifestyles become more popular. As a common point of many people, Food Delivery has obviously become a basic service that every restaurant should have.

1.2. About GrabFood

a) Introduction

GrabFood is a service that connects local food businesses to people. Grab has always brought people to things that matter to them, and they are now committed to bringing things that matter closer to people. When customers place an order with GrabFood, they receive it, send it off to the restaurant and assign a GrabFood delivery partner to pick up the order to bring it to them. By the way, Grab's advantage is that it already has a force of GrabBike drivers, then the service should be expanded. GrabFood is a more convenient segment in the Grab ecosystem, which both

helps increase income for drivers and helps users become more attached to the Grab application besides hitchhiking.

Thus, with the view of that point, we determine to choose topic “Food Delivery – GrabFood” and build up demo database of this system.

b) How does GrabFood work?

With the opening of more servicethe Grab application will have an additional Grab Food section. Food and beverage stores associated with Grab will be displayed on the application with the registered menu. Customers can type in the search box in the Grab app to find the food they want and choose to purchase. In addition to the fee for the dish paid by the Grab driver in advance, the user will bear additional shipping costs.

c) Basic function of GrabFood

In this project, our database would simulate 2 basic functions of GrabFood.

- *Order food*: To order food on GrabFood, customers need to follow these steps:
  - Access the Grab application and select the Food service.
  - Search and choose a restaurant, otherwise customers can directly choose the name of the dish they want to eat.
  - Order
  - View the food basket, if do not want to change, just switch to the payment section.
  - Enter Promo code (if any)
  - When customers have a specific request for the dish, they can add it to the note for the driver.
  - When receiving the order, the driver will go to the restaurant to pick up the food according to the order and deliver it.
- *Promotion:* GrabFood encourage customers to make more orders by voucher code, coupon, support the discount campaign of restaurants, etc.

2. BUSINESS RULES

2.1. As a User

- Users make account on Grab mobile app.
- Users can decide whether to be a driver (Grab's partner) or a customer.

2.2. As a Customer

- Customers can book an order via the app and choose whatever they want to eat. But they can order foods in only one restaurant at a time.
- They will decide which payment methods they are going to use. Except for COD, every other payment method usually come with a discount voucher which can lower the total price.
- If their order is in stage 1 (which is ORDERED), they can withdraw it and get 100% refunded.
- Sometimes, lucky customers might get free vouchers up to 70%.
- After receiving the food, customers can rate both the food and the driver.
- If customers want to order some foods for someone, they only need to type in an appointed location, pay for the order and they are good to go.

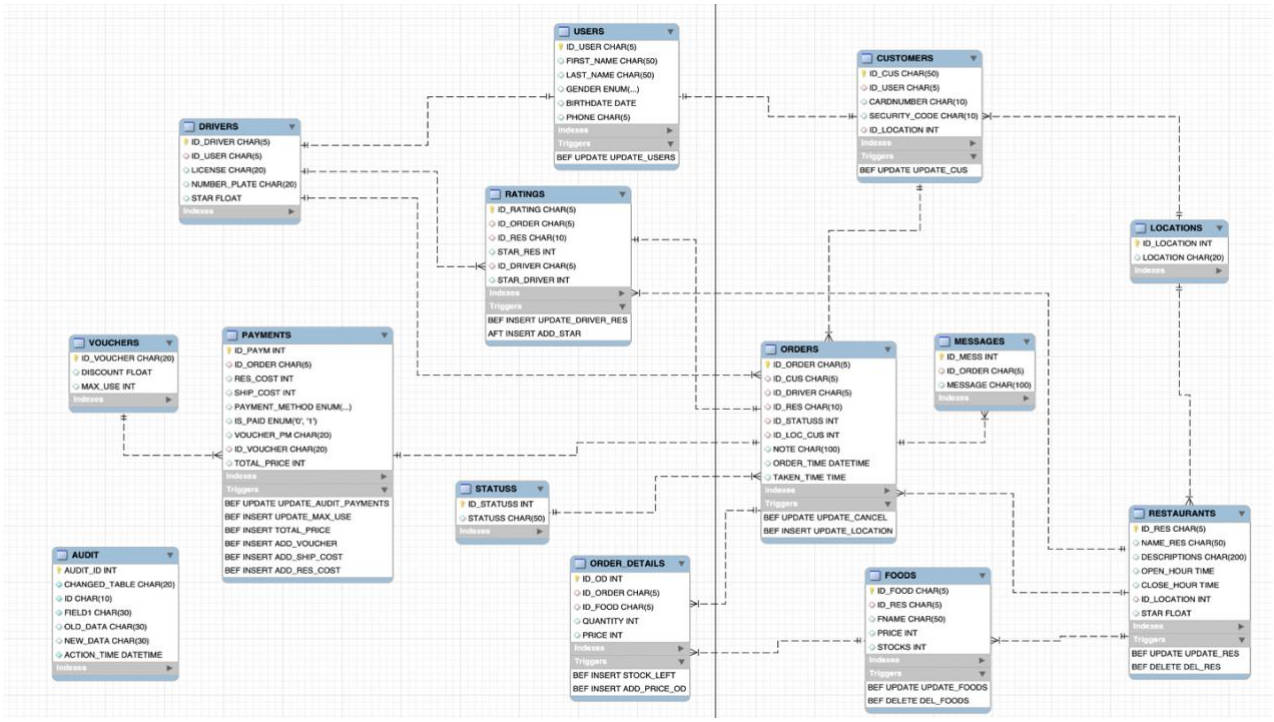
2.3. As a Driver

- Drivers will take the order and go pick up the food from the restaurant. They will be paid depends on the distance from the customer's location to the restaurant's address.
- After each order, drivers will receive stars from customers' ratings. The more stars the driver gets, the more reliable he becomes.

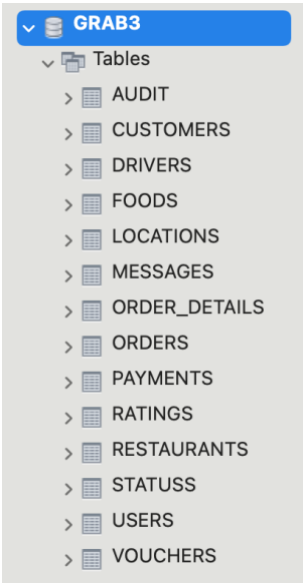
2.4. As a Restaurant

Restaurant will take the order and cook the food. But they have limited stock for each dish, so they are unable to cook unless customers order less than the remaining stocks.

3. ENTITY RELATIONSHIP DIAGRAM (ERD)



4. LIST OF TABLES



Our database was created by 14 tables: AUDIT, CUSTOMERS, DRIVERS, FOODS, LOCATIONS, MESSAGES, ORDER_DETAILS, ORDERS, PAYMENTS, RATINGS, RESTAURANTS, STATUSS, USERS, VOUCHERS. The remain of this chapter would describe the data fields in each table in detail. The script of creating database and tables that consists of data are in GRAB_DATABASE_FINAL.sql file.

4.1. Table AUDIT

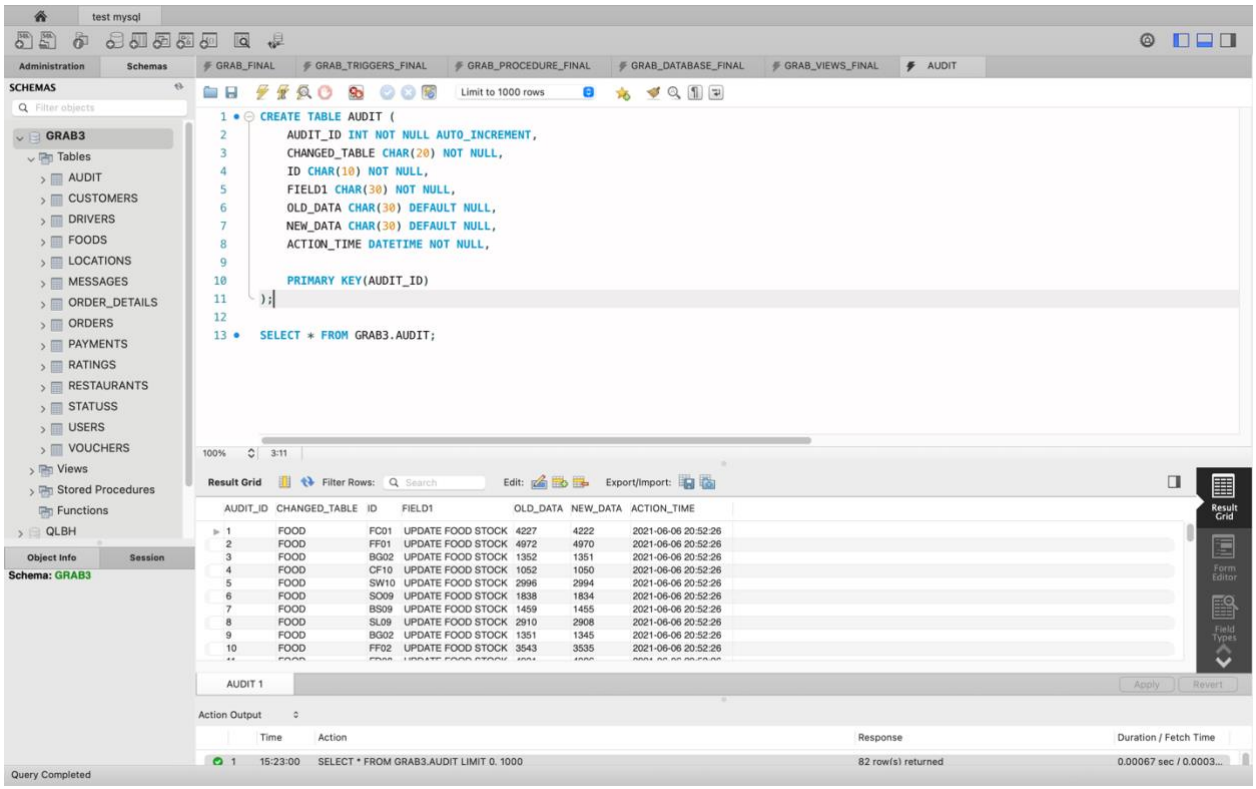
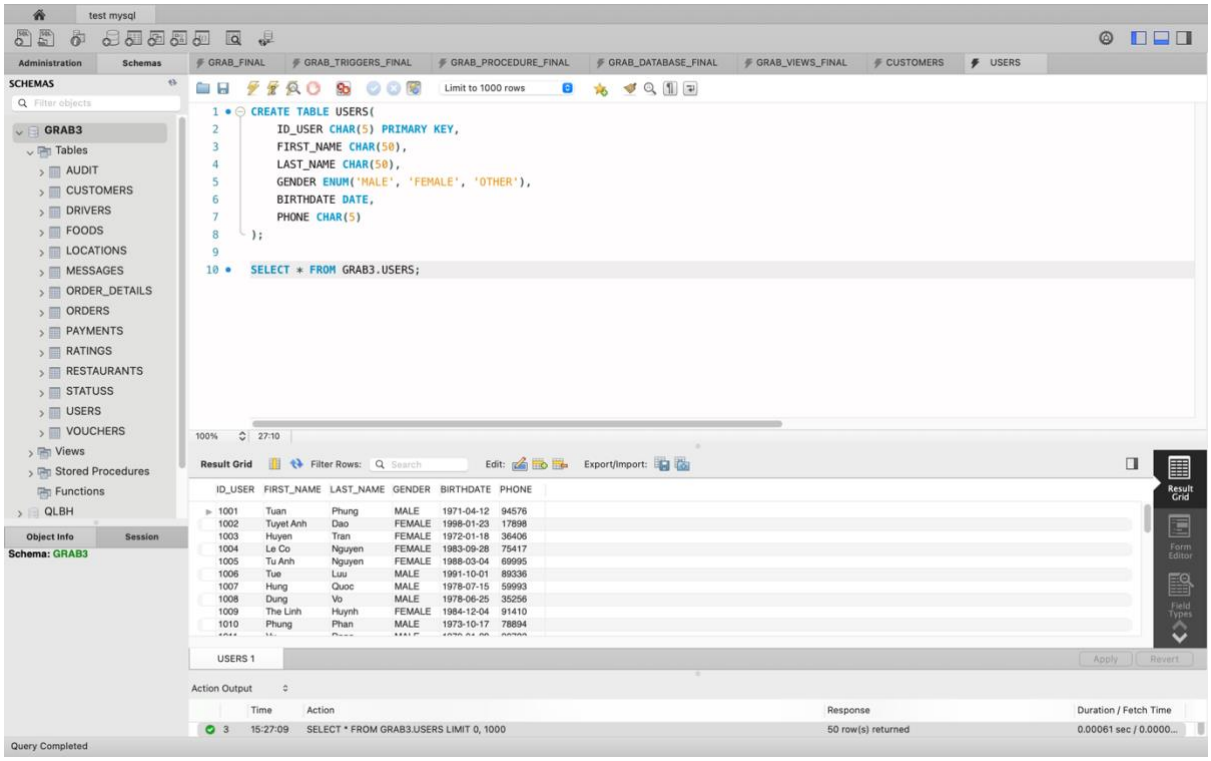


Table AUDIT will store every change in other tables in the database:

- AUDIT_ID: Primary Key
- CHANGED_TABLE: name of the table that is changed
- ID: ID of the object that received the change
- FIELD1: accomplised action
- OLD_DATA: data before changing
- NEW_DATA: data after changing
- ACTION_TIME: time when applying changes

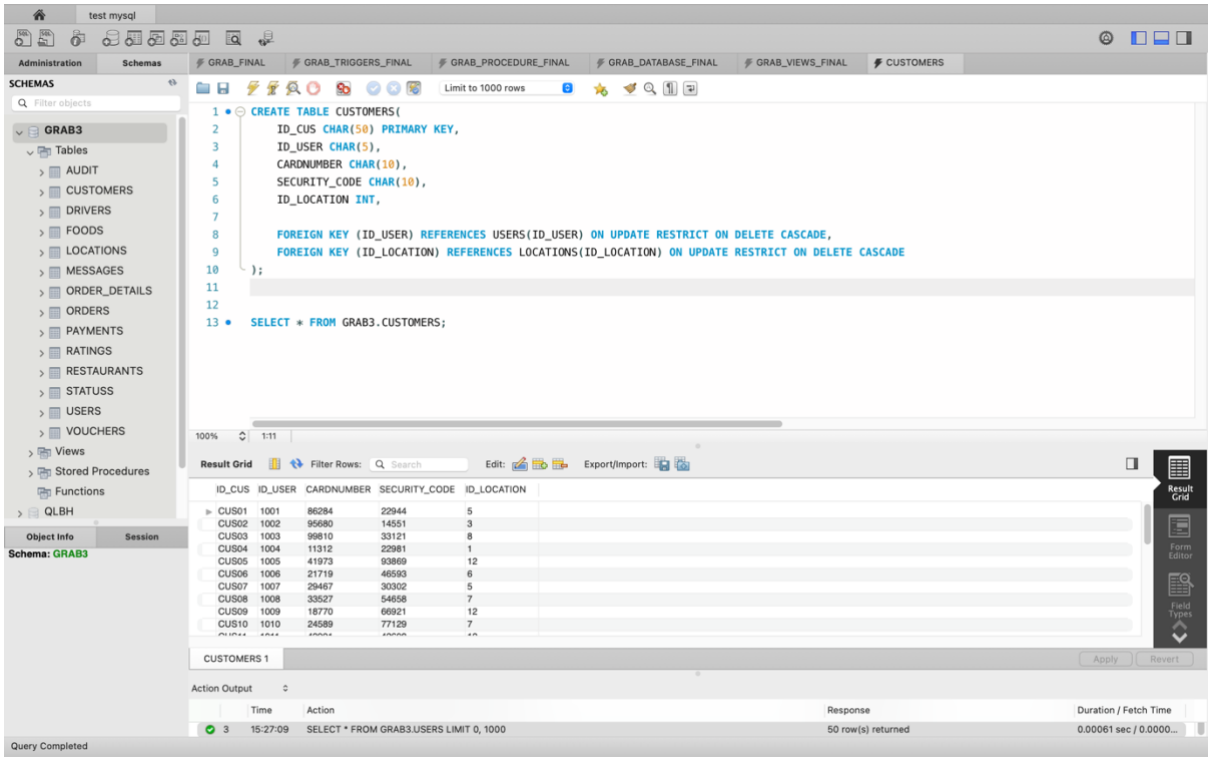
4.2. Table USERS



- ID_USER: ID of each user (Primary Key)
- FIRST_NAME: first name of the user
- LAST_NAME: last name of the user
- GENDER: gender of the user
- BIRTHDATE: birthdate of the user
- PHONE: telephone number

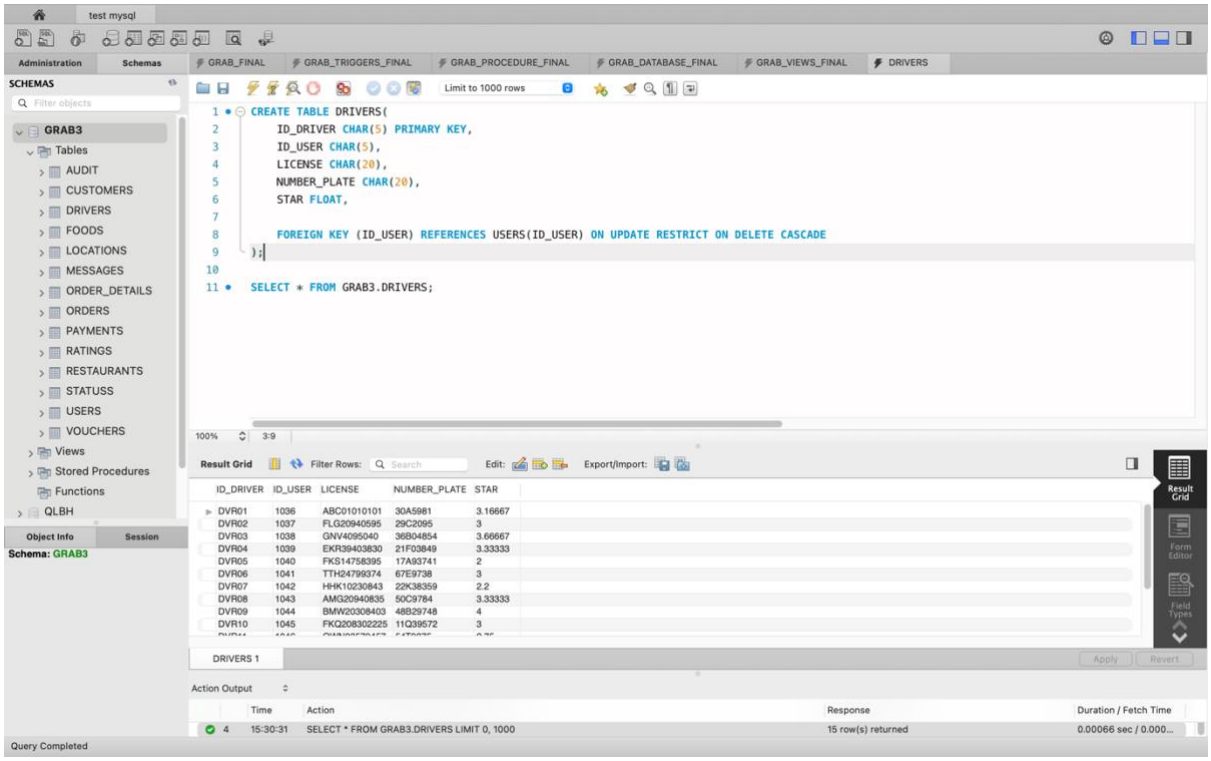
Concretely, in USERS, we divide them into two main groups: CUSTOMERS and DRIVERS. Since not only they have some private information that couldn't be shared to anyone else but also, they have different purposes, one is served, and the others serves.

4.3. Table CUSTOMERS



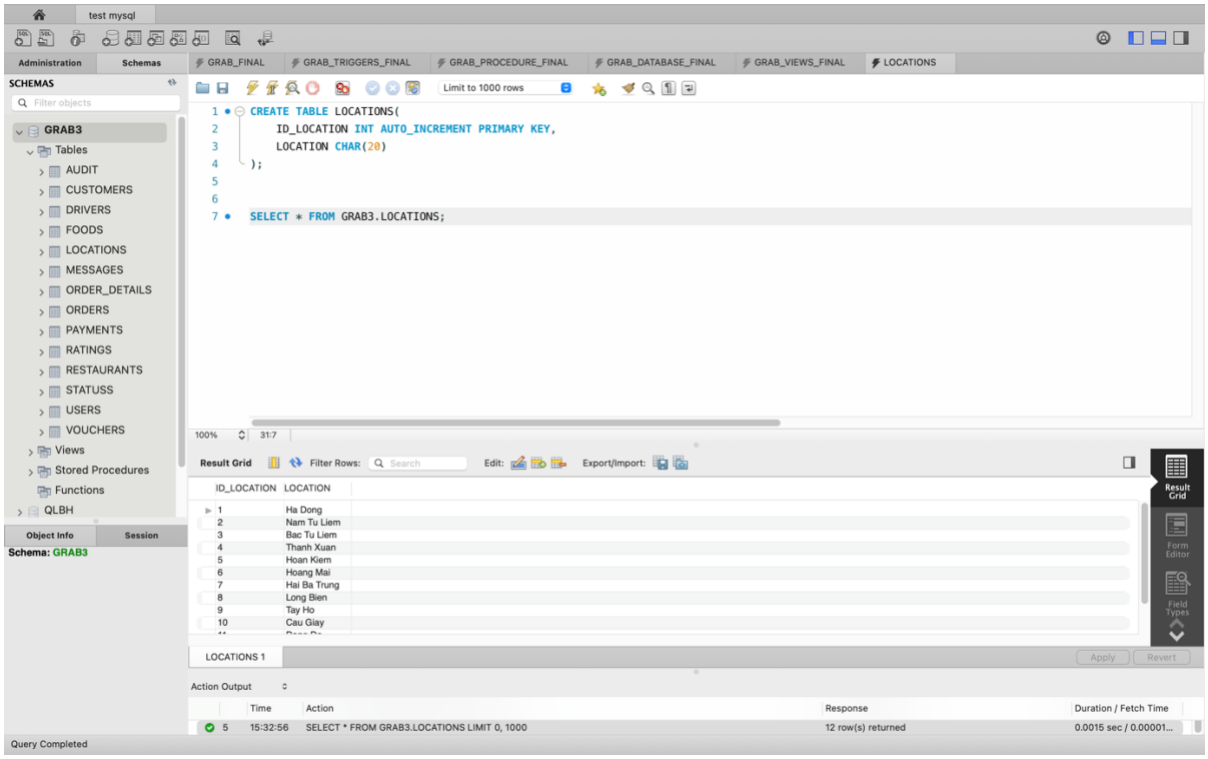
- ID_CUS: ID of each customer (Primary Key)
- ID_USER: ID of each user (Foreign Key from USERS)
- CARDNUMBER: credit card number
- SECURITY_CODE: identity card number
- ID_LOCATION: location of the customer (Foreign Key from LOCATIONS)

4.4. Table DRIVERS



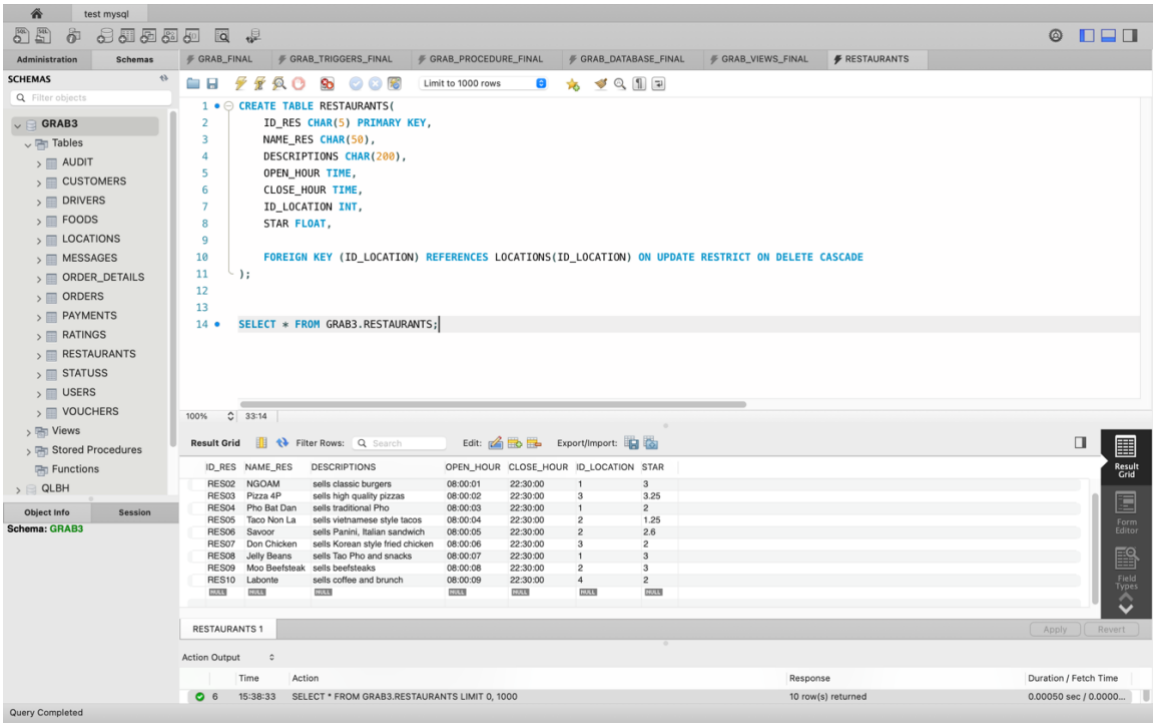
- ID_DRIVER: ID of each driver (Primary Key)
- ID_USER: ID of each (Foreign Key from USERS)
- NUMBER_PLATE: the number plate of the motorbike that driver used to deliver.
- LICENSE: verify driver
- STAR: evaluate delivery quality of driver. Concretely, it will be added after every succeed order through a trigger called ADD_STAR. It will calculate the average star in the RATINGS table that has ID_DRIVER.

4.5. Table LOCATIONS



- ID_LOCATION: ID of each location (Primary Key)
- LOCATION: name of the location

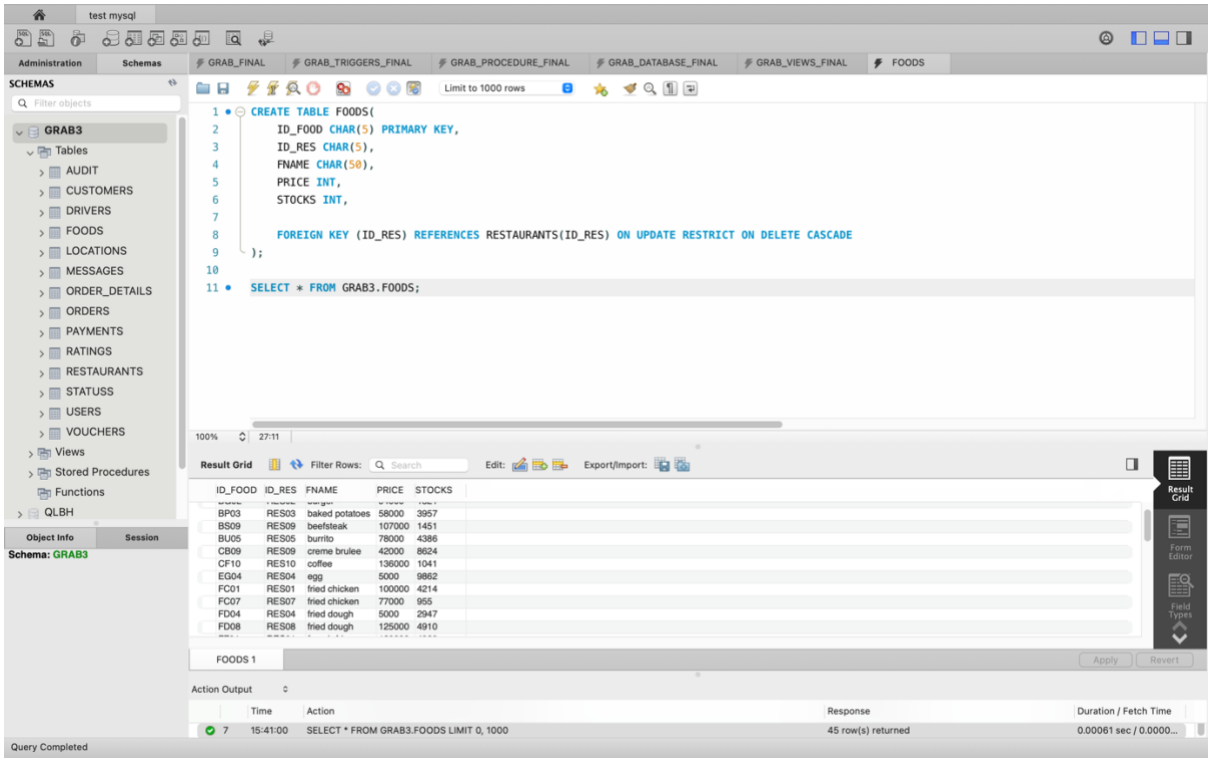
4.6. Table RESTAURANTS



- ID_RES: ID of each restaurant (Primary Key)
- DESCRIPTION: describes overall about the restaurant

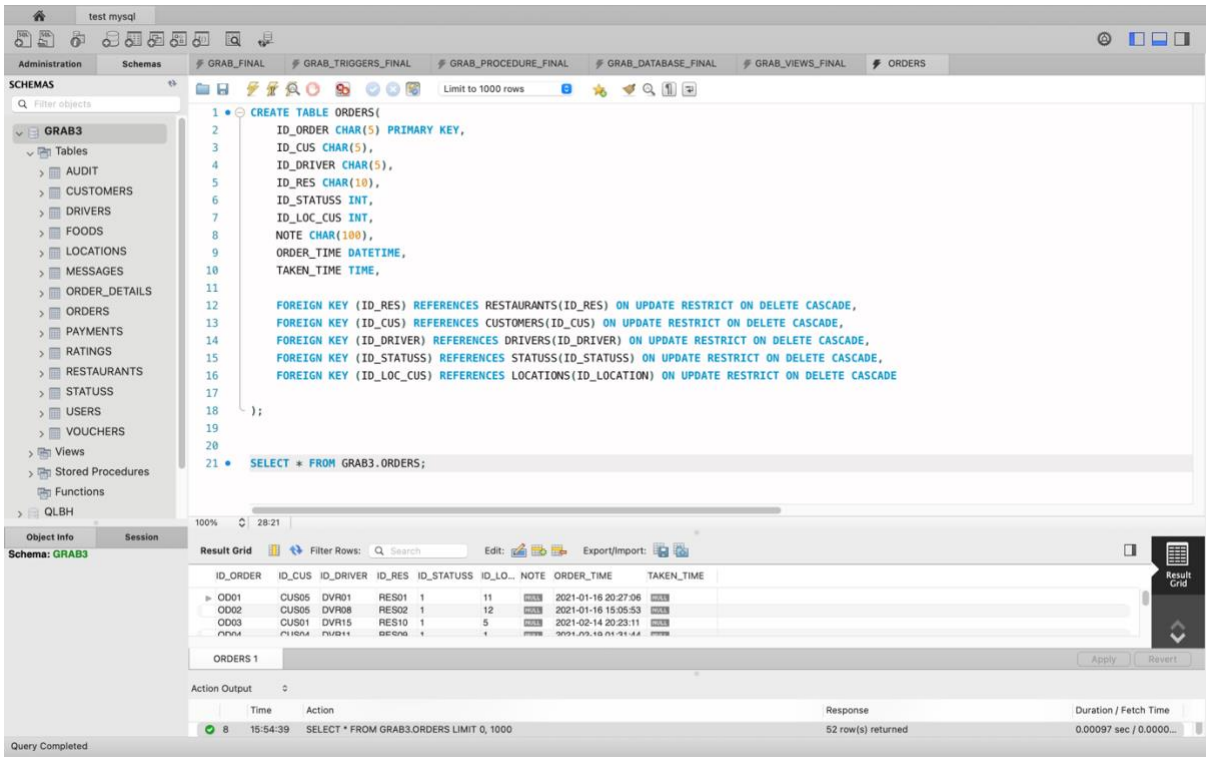
- OPEN_HOUR: the time when restaurant opens
- CLOSE_HOUR: the time when the restaurant closes
- ID_LOCATION: stores the restaurant's district number in order to calculate the shipping cost by the procedure GET_SHIP_COST (Foreign Key from LOCATIONS)
- STAR: rating of customer about the restaurant (by the number of star). Average star received from customers which will be updated by trigger ADD_STAR_RES

4.7. Table FOODS



- ID_FOOD: ID of each food, each ID_FOOD belongs to only one restaurant (Primary Key)
- ID_RES: ID of restaurant (Foreign Key from RESTAURANTS)
- FNAME: the name of the food.
- PRICE: food's price per unit.
- STOCKS: the food's quantity left. After each order, food's stocks will be deducted by the amount in the customer's order. If the amount ordered is larger than the stocks, system will not let they order the food and pop out a message says "Only {stocks} {fname} left"

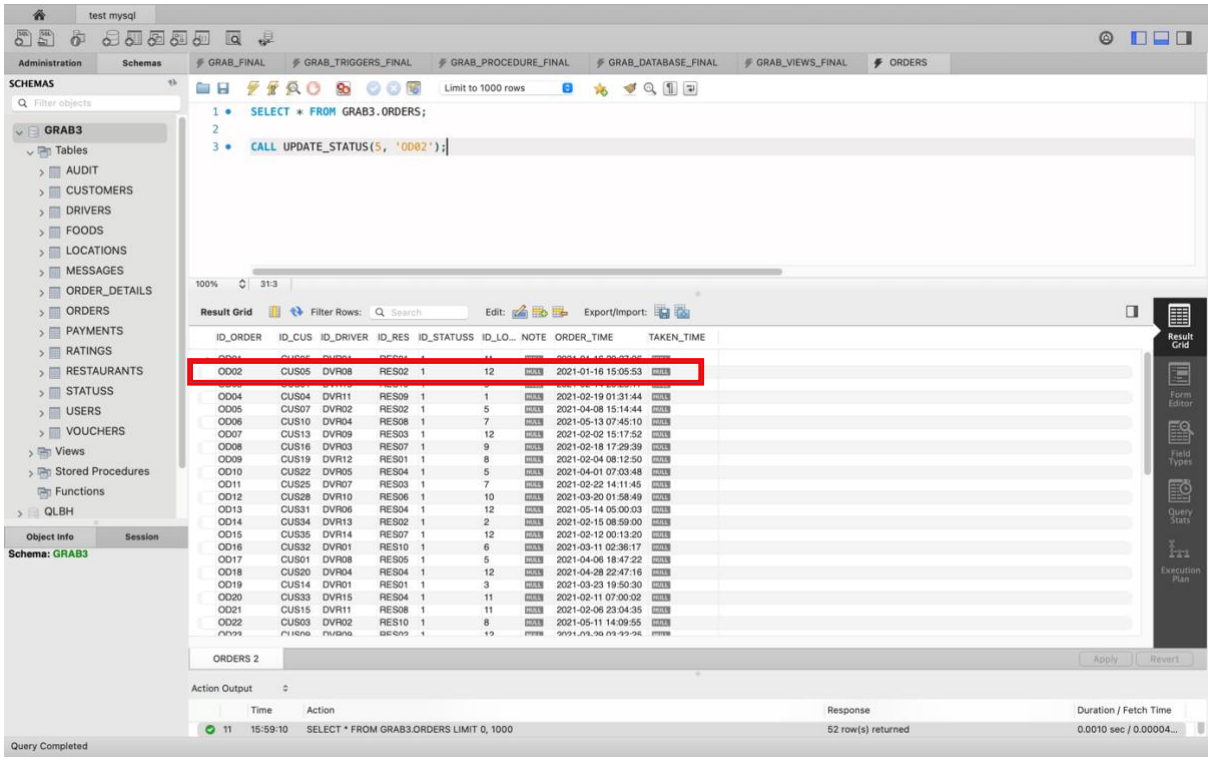
4.8. Table ORDERS

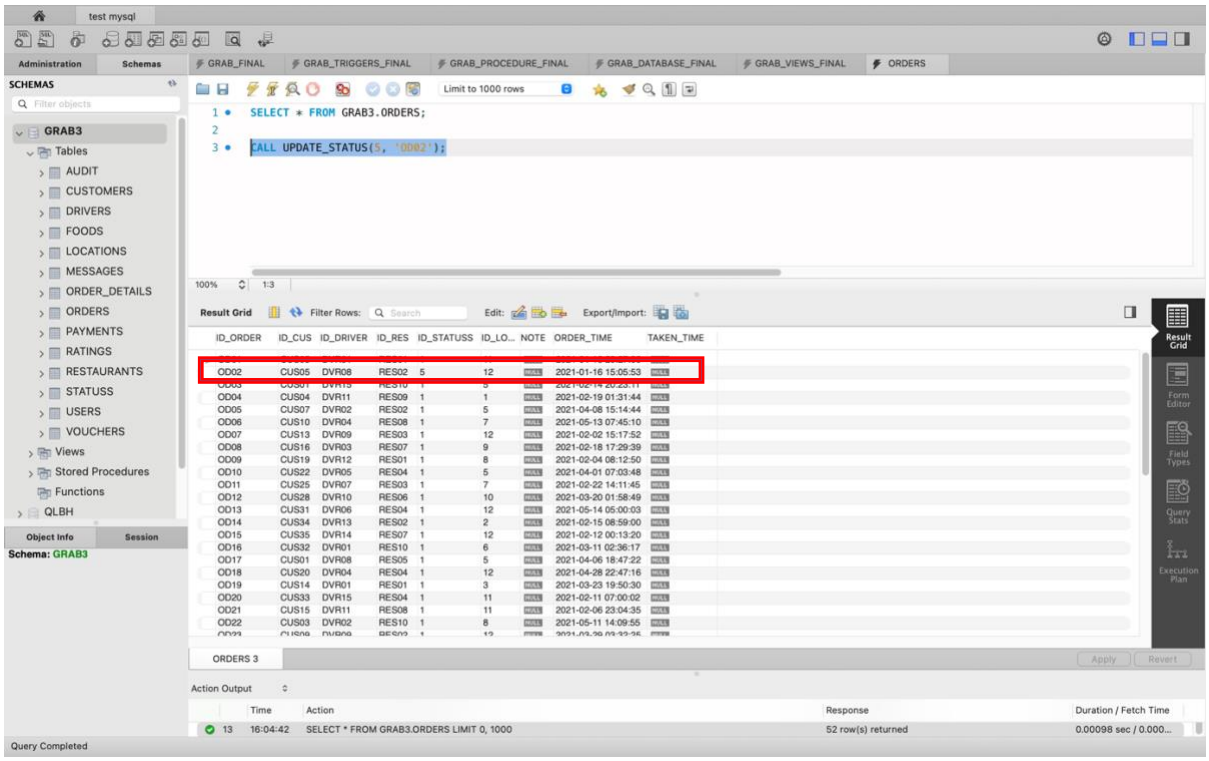


This table store data of *one* order:

- ID_ORDER: ID of each order (Primary Key)
- ID_CUS: ID of the customer who made this order (Foreign Key from CUSTOMERS)
- ID_DRIVER: ID of the driver who delivery this order (Foreign Key from DRIVERS)
- ID_RES: ID of restaurant that provided the food for this order (Foreign Key from RESTAURANTS)
- ID_STATUSS: the status of the order which would tell the customers the condition of their orders (Foreign Key from STATUSS). It is also used to determine whether an order can be canceled or not. Based on that, we could decide the amount of money must be returned to the customer. Each order would have 4 stages which will be described in detail later in the table STATUSS. Besides, ID_STATUSS will be updated through a procedure called UPDATE_STATUS.
- An order might be able to cancel if ID_STATUS is 1 (ORDERED) and the ID_STATUSS will be set to 5 (CANCELED), other ID_STATUSS couldn't

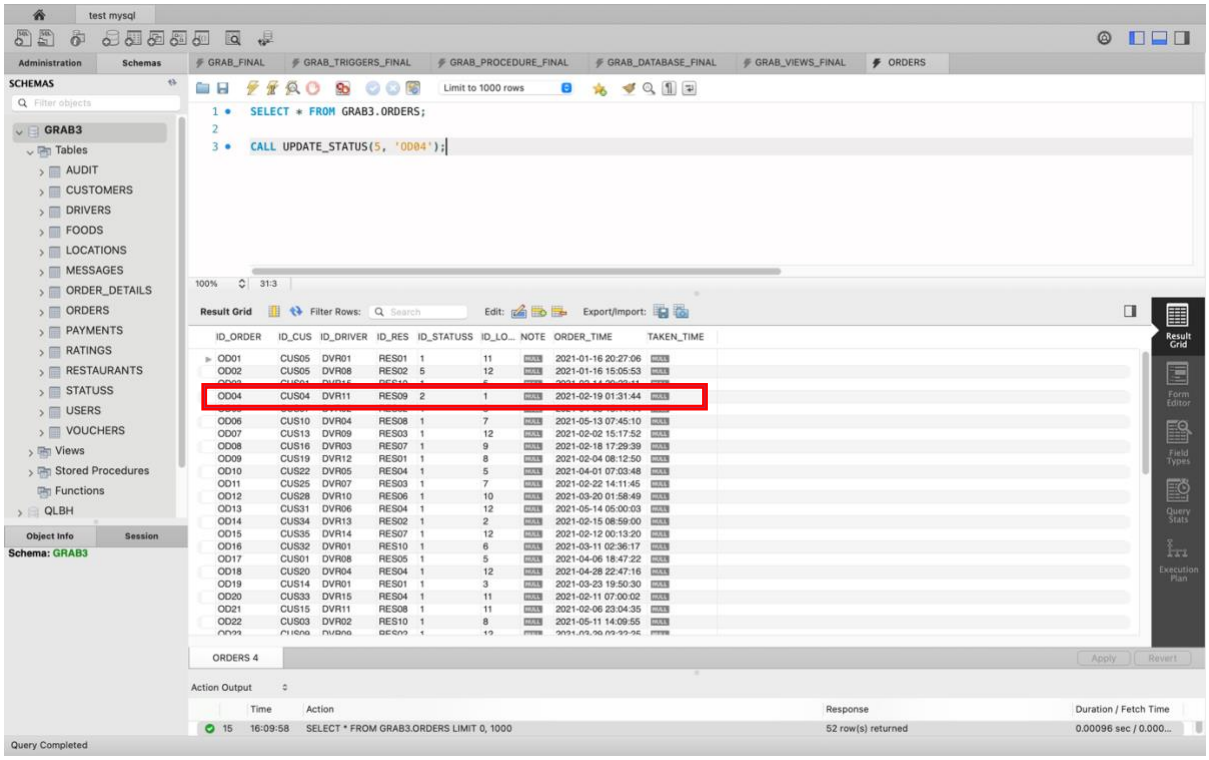
be withdrawn. The UPDATE_STATUS will check it for the customers. Here is what a customer must do if he wants to cancel an order. Let's take OD02 as an example.

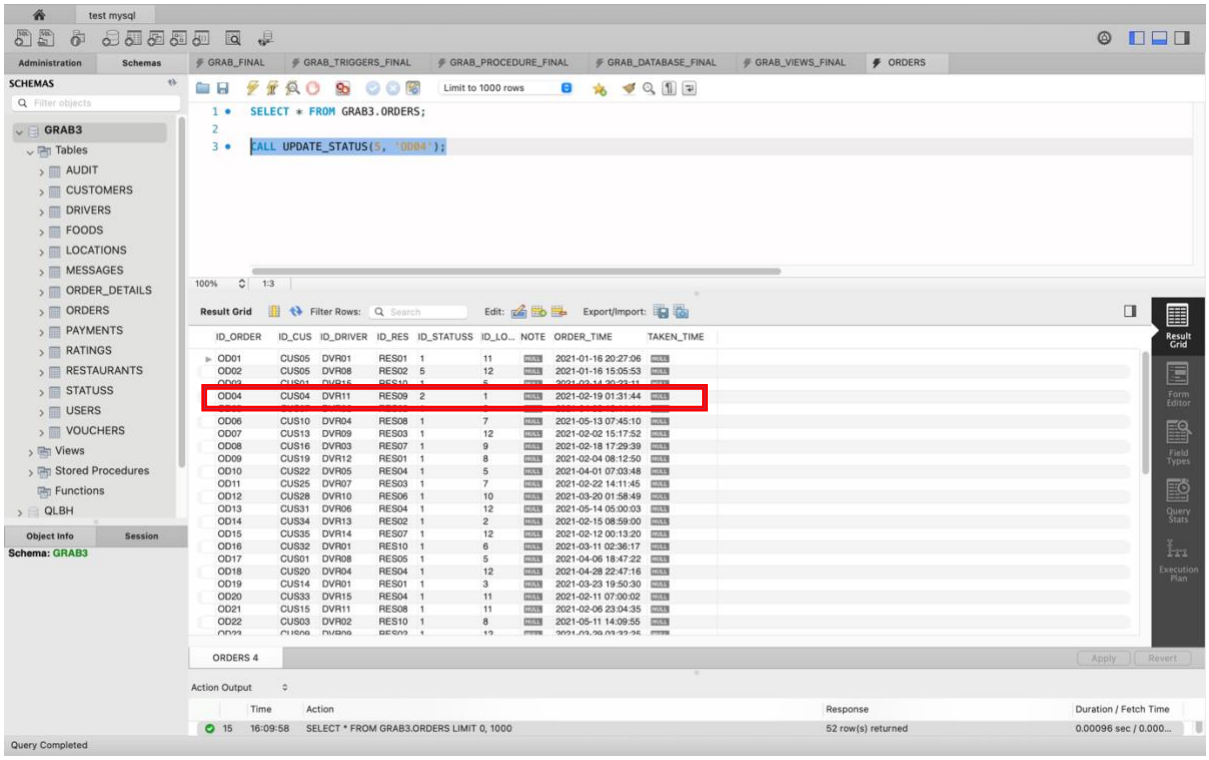




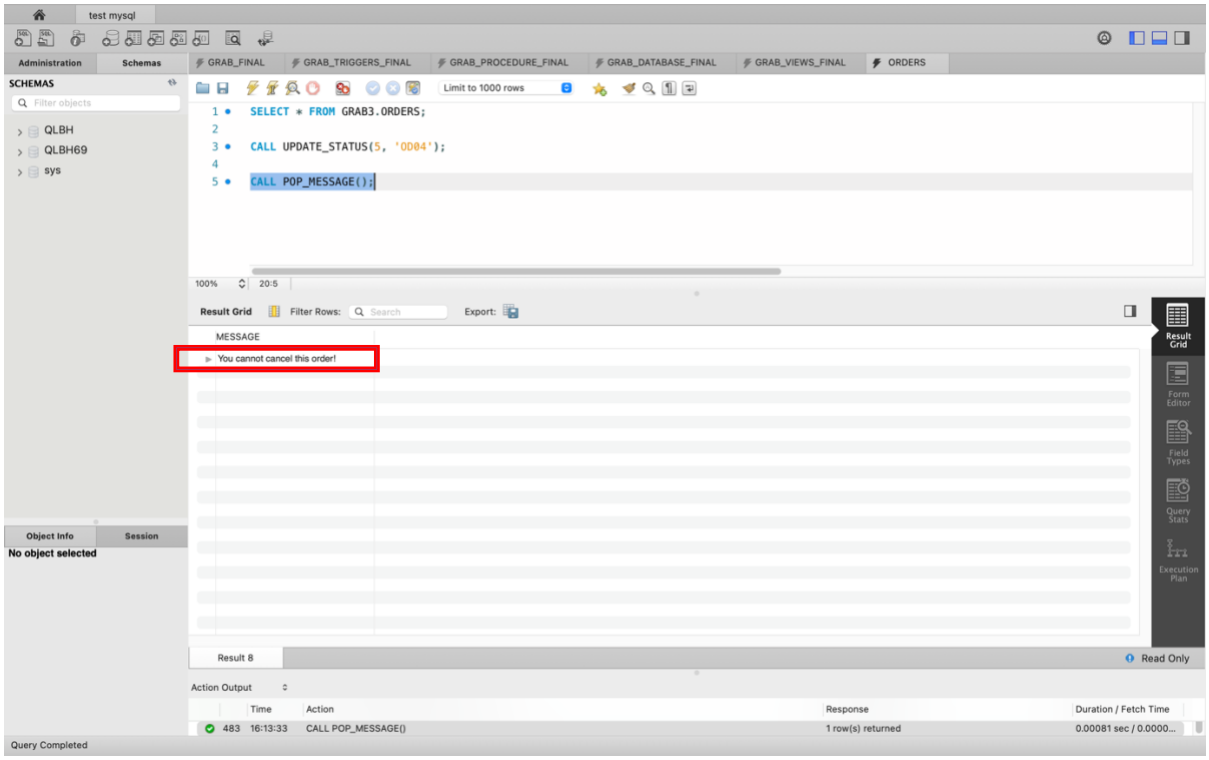
When the ID_STATUSS changes into 5, his order is successfully canceled.

- Let’s take another example, the “OD04” order has ID_STATUSS = 2.

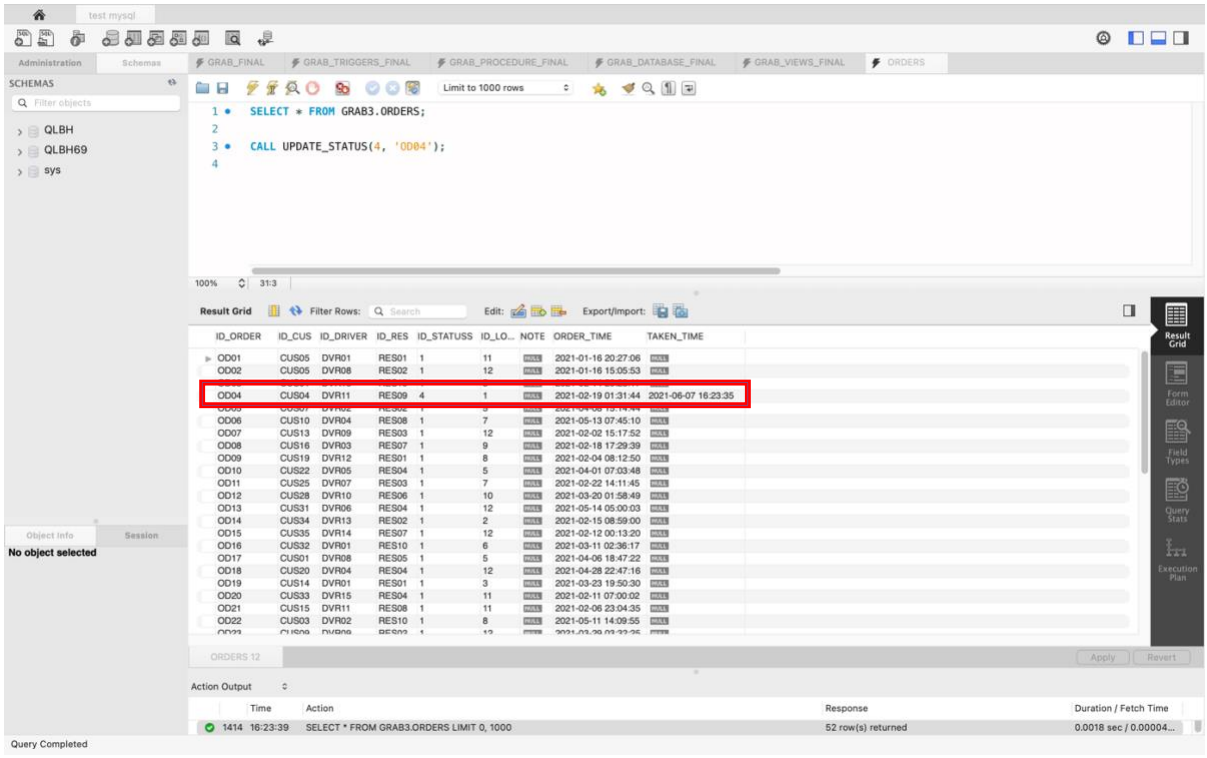




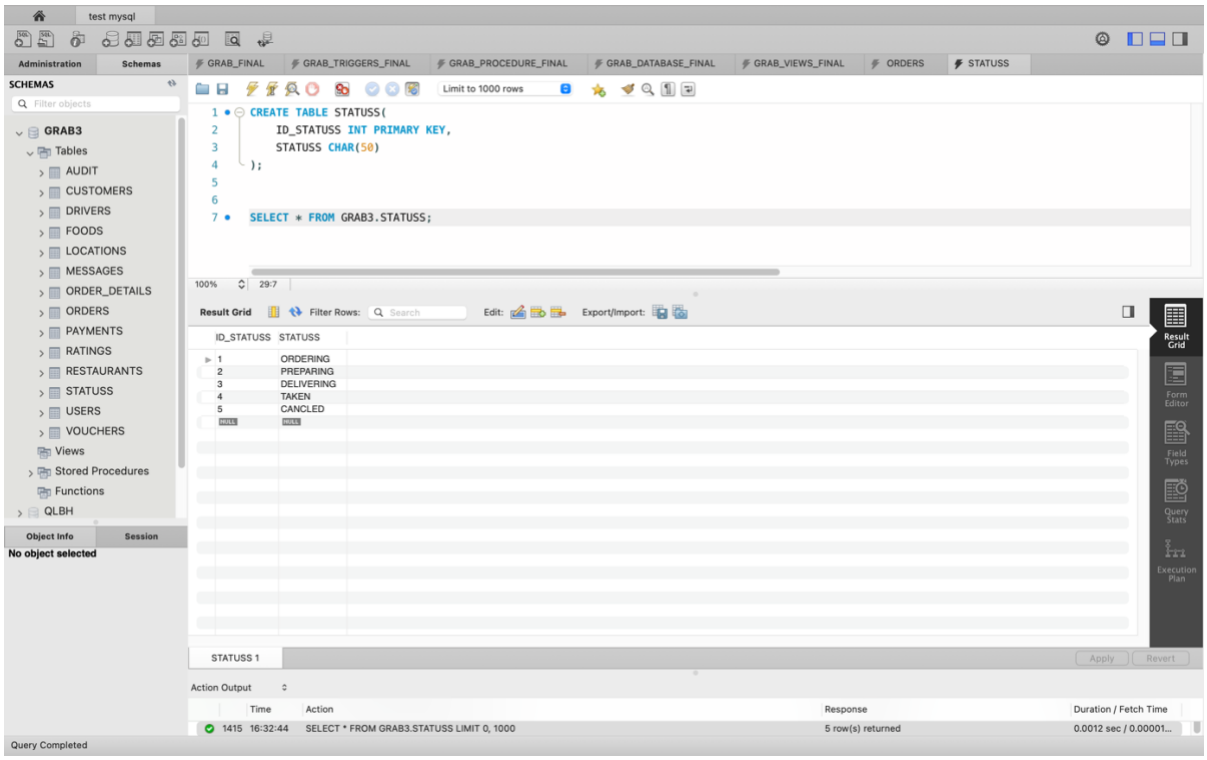
After running the UPDATE_STATUS procedure, ID_STATUSS of OD04 still has not changed.



- ID_LOCATION is the address that the order is delivered to. If the customers want to order some foods for someone, they just need to insert in a location. Else, system will automatically set the customers' address as default. (Foreign key from LOCATIONS)
- ORDER_TIME: represents the time when the driver receives the order
- TAKEN_TIME: represents the time when the customer receives the order.
- TAKEN_TIME will be updated whenever ID_STATUSS is 4.

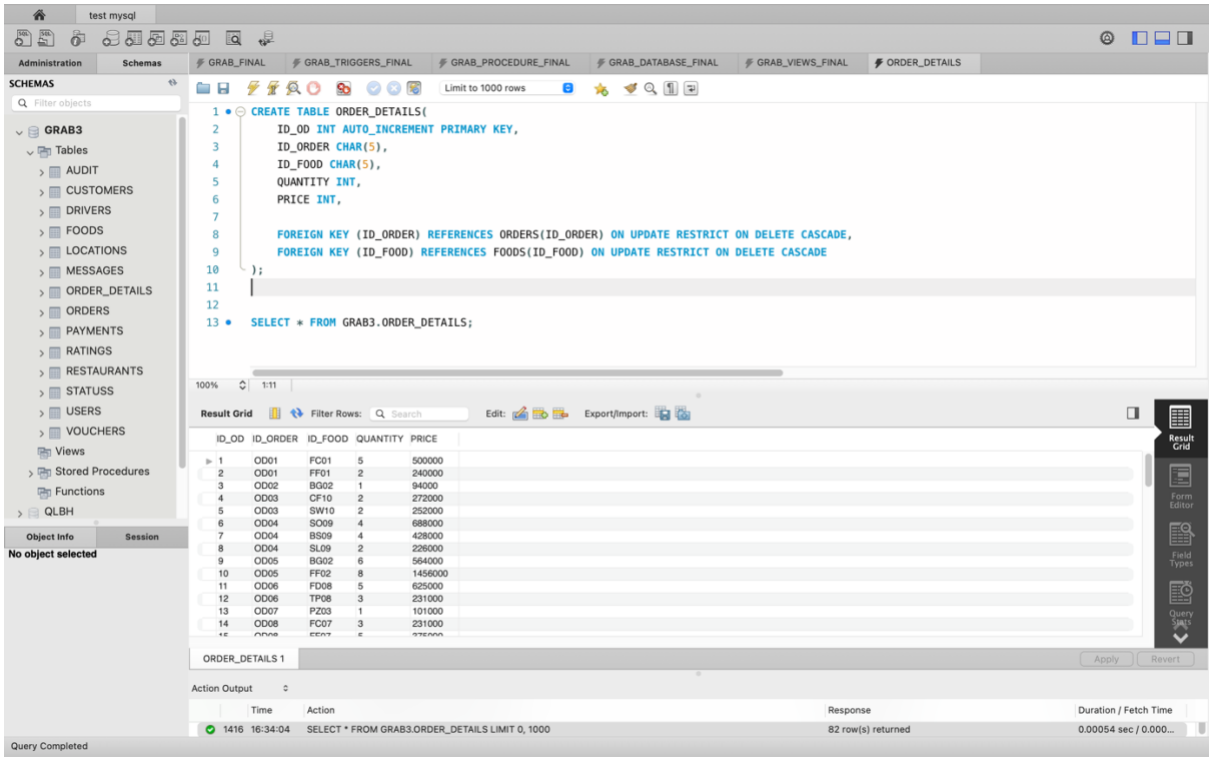


4.9. Table STATUSS



- ID_STATUSS: ID of each status (Primary Key)
- STATUSS: store the status of each order

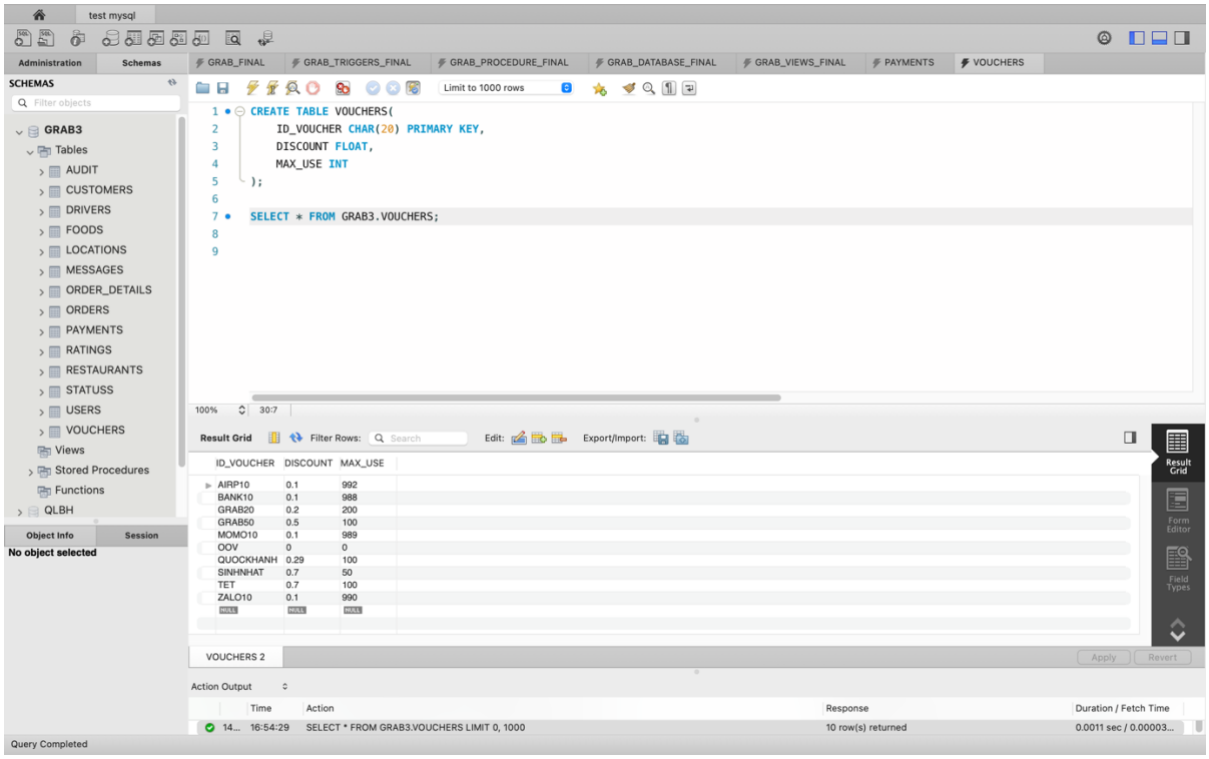
4.10. Table ORDER_DETAILS



This table store every food that customer ordered orderly:

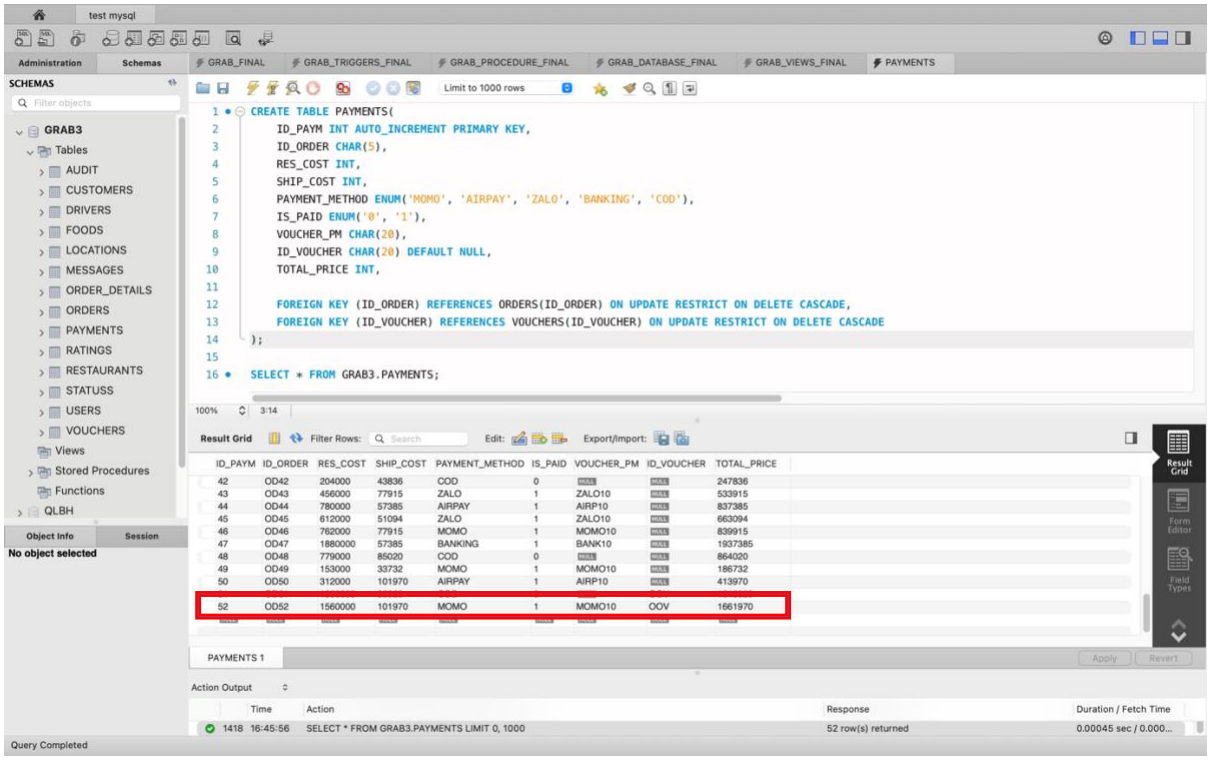
- ID_OD: ID of each detail order line (Primary Key)
- ID_ORDER: ID of the order (Foreign Key from ORDERS)
- ID_FOOD: ID of the food (Foreign Key from FOODS)
- QUANTITY: quantity of each food
- PRICE: price of the food per unit times quantity

4.11. Table VOUCHERS

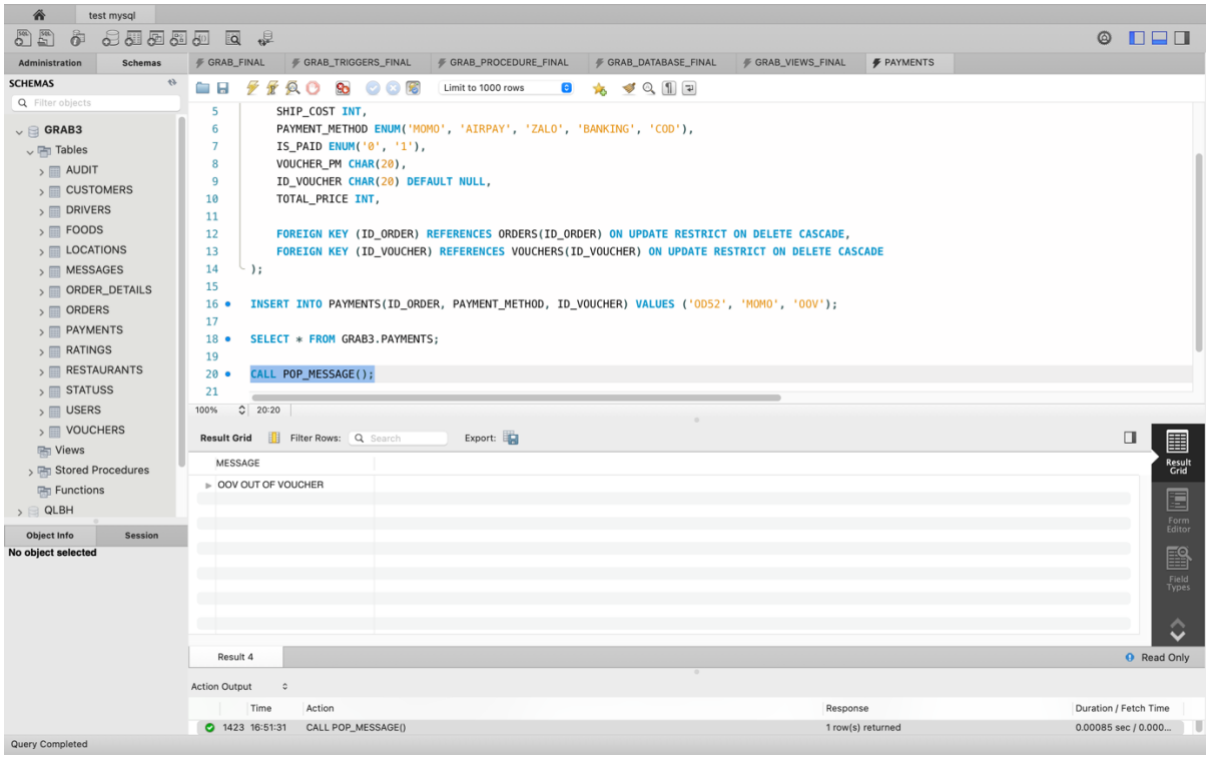


- ID_VOUCHER: ID of each voucher (Primary Key)
- DISCOUNT: a deduction percentage of the price towards each order
- MAX_USE: maximum time of the voucher be used. It will automatically be deducted after being inserted in PAYMENTS.

4.12. Table PAYMENTS

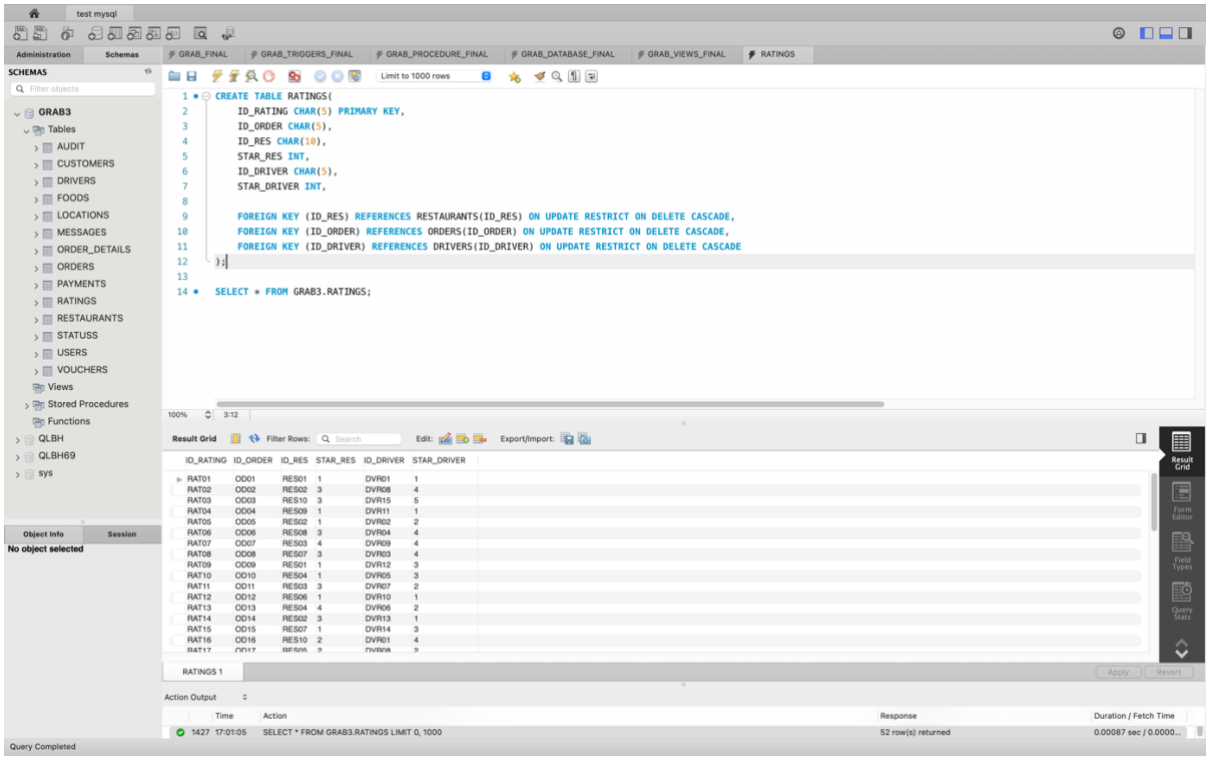


- ID_PAYM: ID of each payment transaction (Primary Key)
- ID_ORDER: ID of each order (Foreign Key from ORDERS)
- RES_COST: the cost for the food ordered by customer calculated by multiply food price with quantity in each order
- SHIP_COST: the cost for the delivery from the restaurant to the appointed location. It will be calculated by the GET_SHIP_COST procedure.
- PAYMENT_METHOD: customers can choose which payment method they would like to use. Except for COD, other payment method comes with a voucher that will automatically added by ADD_VOUCHER trigger. It will directly reduce the total price.
- IS_PAID: is added automatically based on the payment method above. It also let the driver knows whether the order has been paid or not.
- VOUCHER_PM: contains vouchers which come with the payment methods.
- ID_VOUCHER: ID of voucher that might be released by GrabFood (Foreign Key from VOUCHERS)
- If the MAX_USE of any voucher is 0 (Example: ID_PAYM = 52, MAX_USE of OOV = 0)



- TOTAL_PRICE: total price customers must pay after discounting which will be added by TOTAL_PRICE Trigger.

4.13. Table RATINGS



- ID_RATING: ID of each rating (Primary Key)
- ID_ORDER: ID of each order (Foreign Key from ORDERS)

- ID_RES: ID of restaurant will automatically be updated by trigger UPDATE_DRIVER_RES (Foreign Key from RESTAURANTS)
- STAR_RES: stores the number of stars rated by customers for the restaurant
- ID_DRIVER: ID of each driver (Foreign Key from DRIVERS)
- STAR_DRIVER: stores the number of stars rated by customers for the driver.

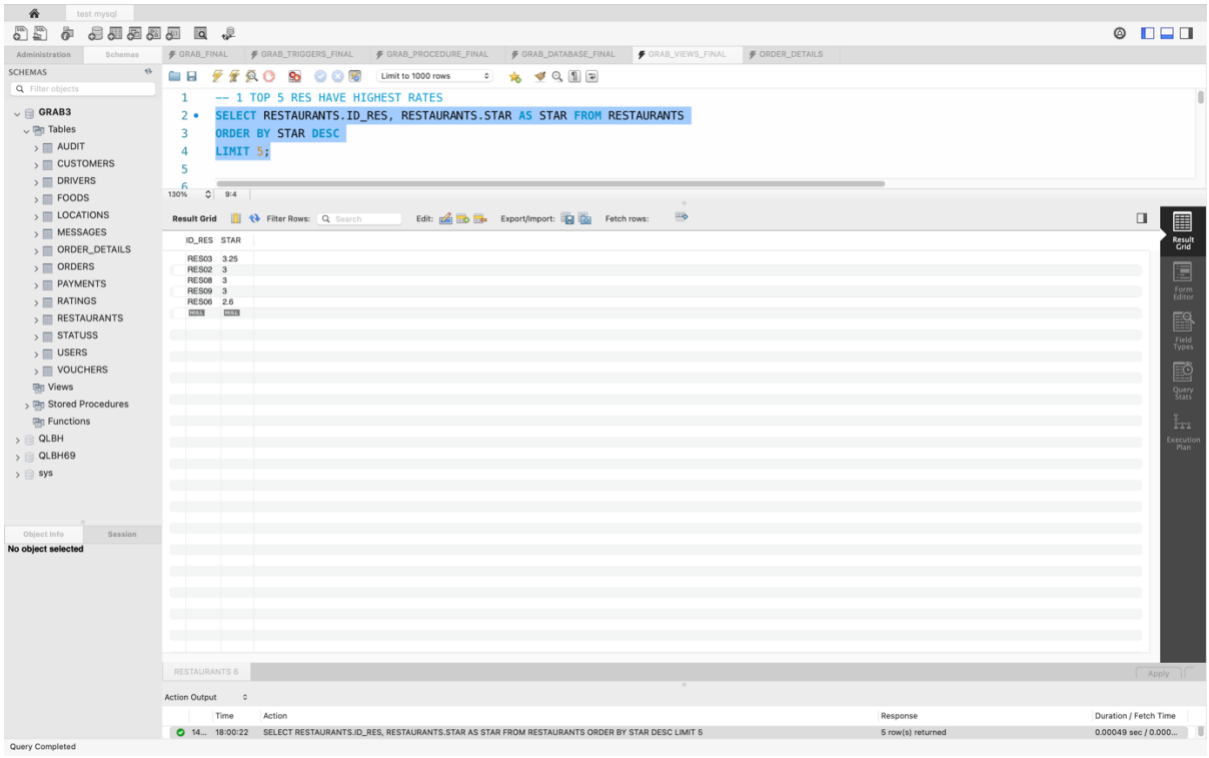
5. QUERY

In order to explore insights from the database, we implement some demo queries. The script and results are in GRAB_VIEWS_FINAL.sql file.

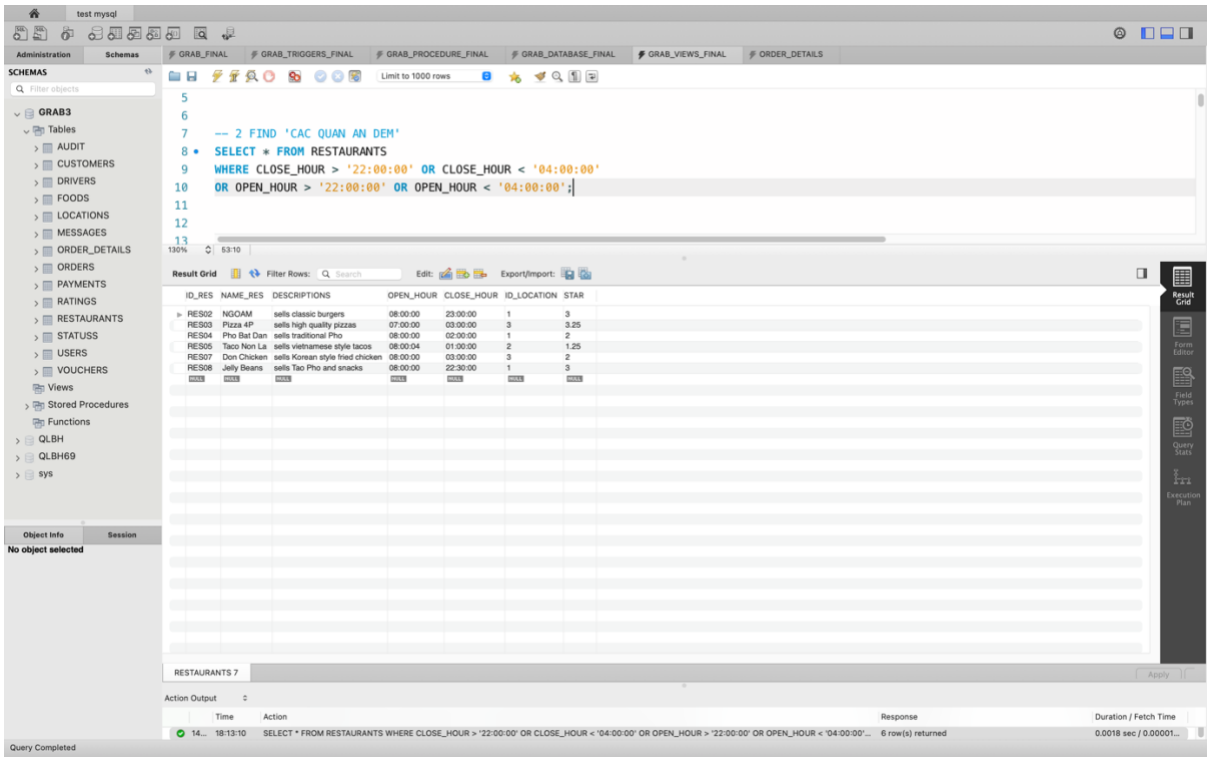
5.1. SELECT

The 10 queries below provide some basic information about our database. Results after implemented could be used by a variety of audiences such as restaurants or market researcher company.

- Top 5 restaurants have the highest rating. Since customers tend to order from restaurants with high ratings, GrabFood can organize a campaign in order to promote restaurants that are highly rated and help the others with poor ratings.

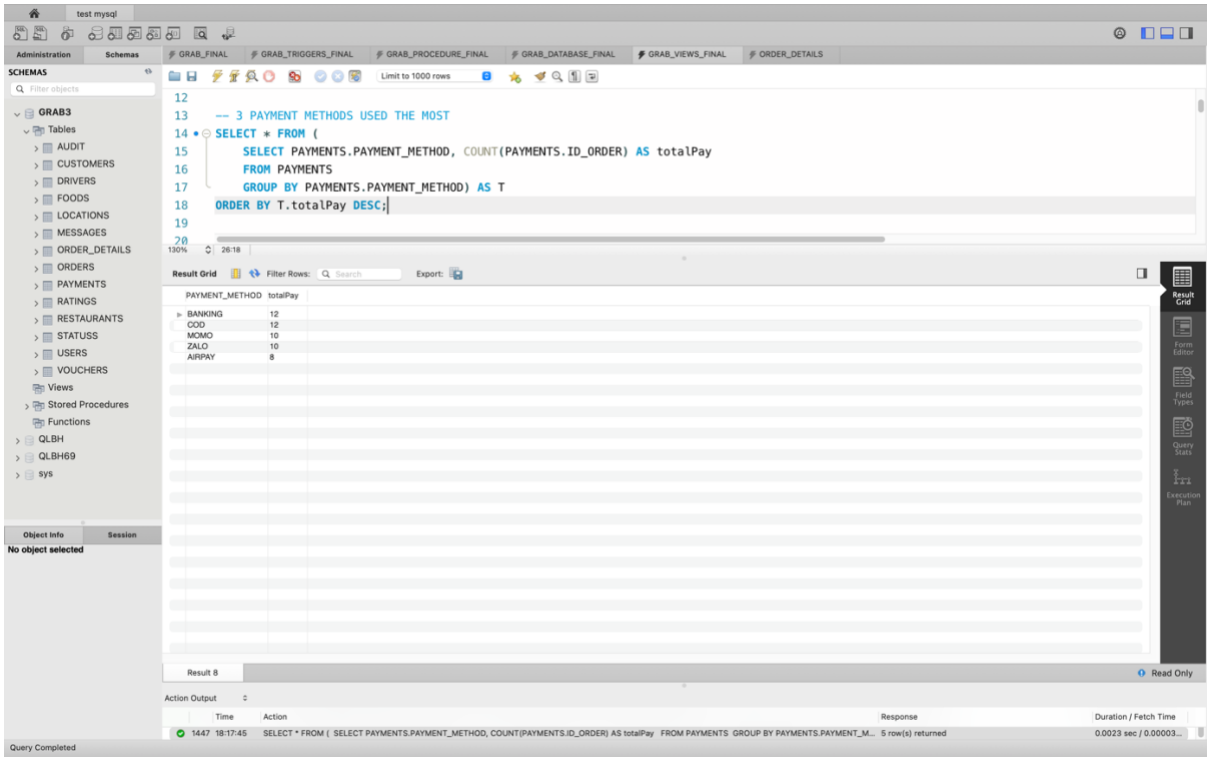


- Restaurants still open at night. Some of GrabFood's customers mostly work at night so creating a list of restaurants that still open late will help them ordering some foods.

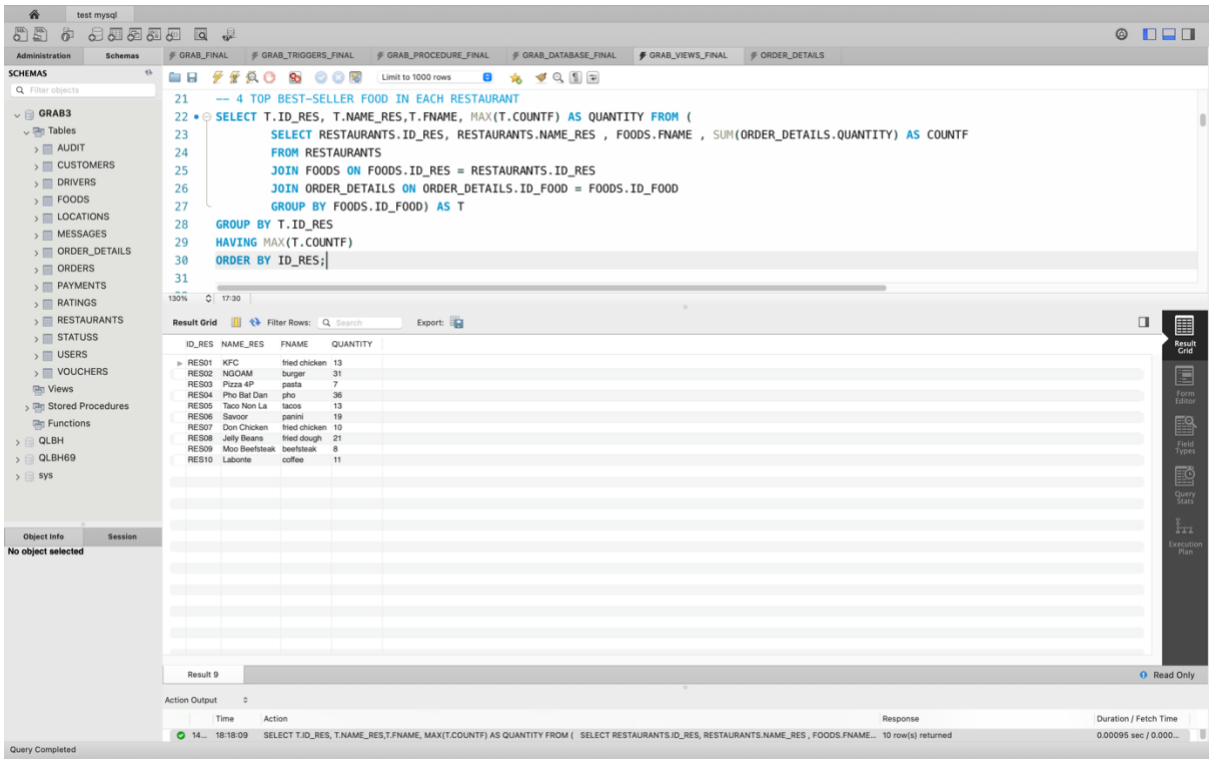


- Payment methods are used mostly. This will help GrabFood visualizing the growing trend in payment methods (PM) so that they can associate with

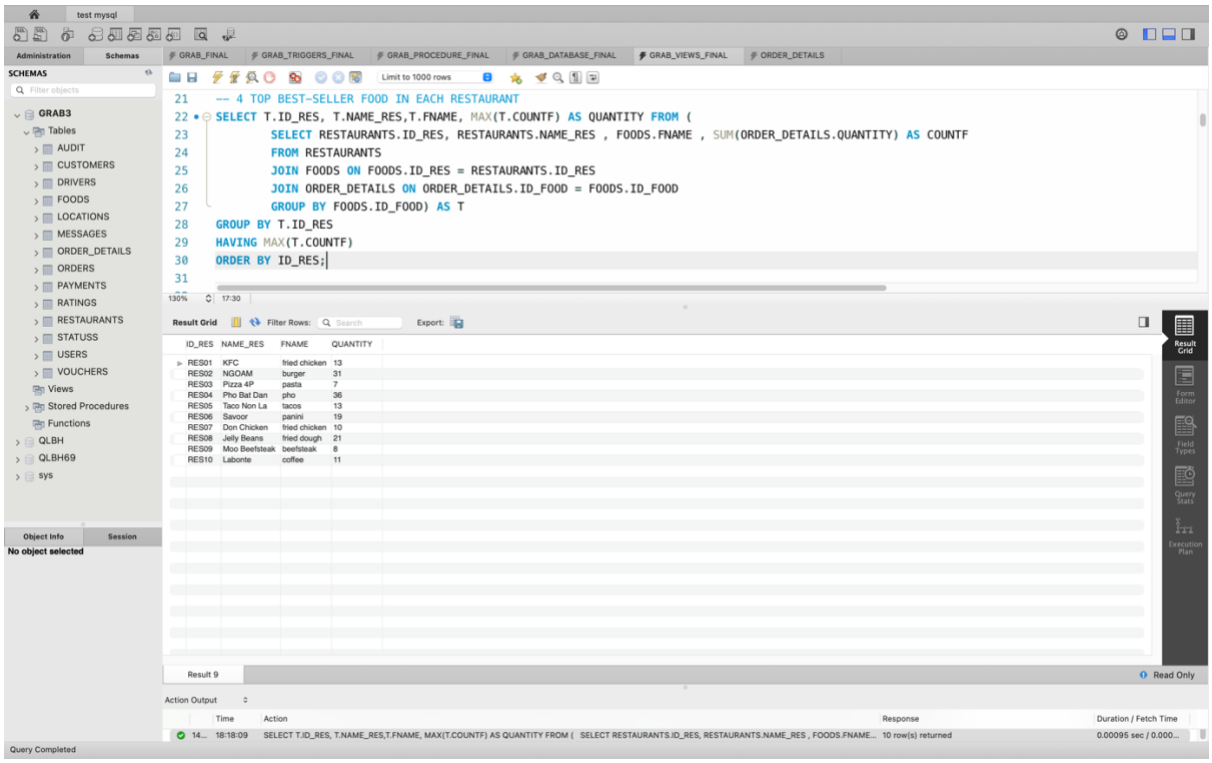
the PM's owner to release more vouchers, marketing campaigns in order to a attract more customers.



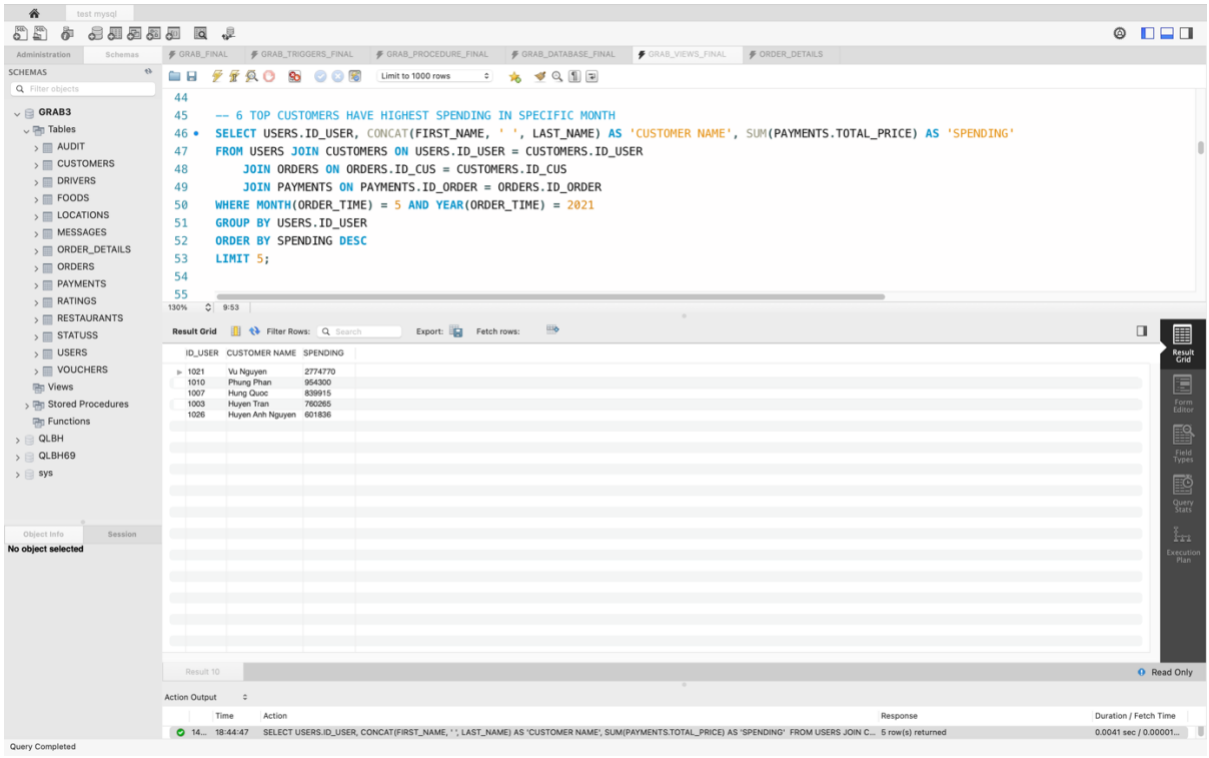
- Top best-seller food in each restaurant. New customers tend to order the best-seller in the restaurant that they are about to order. So this table is going to give them some recommendations.



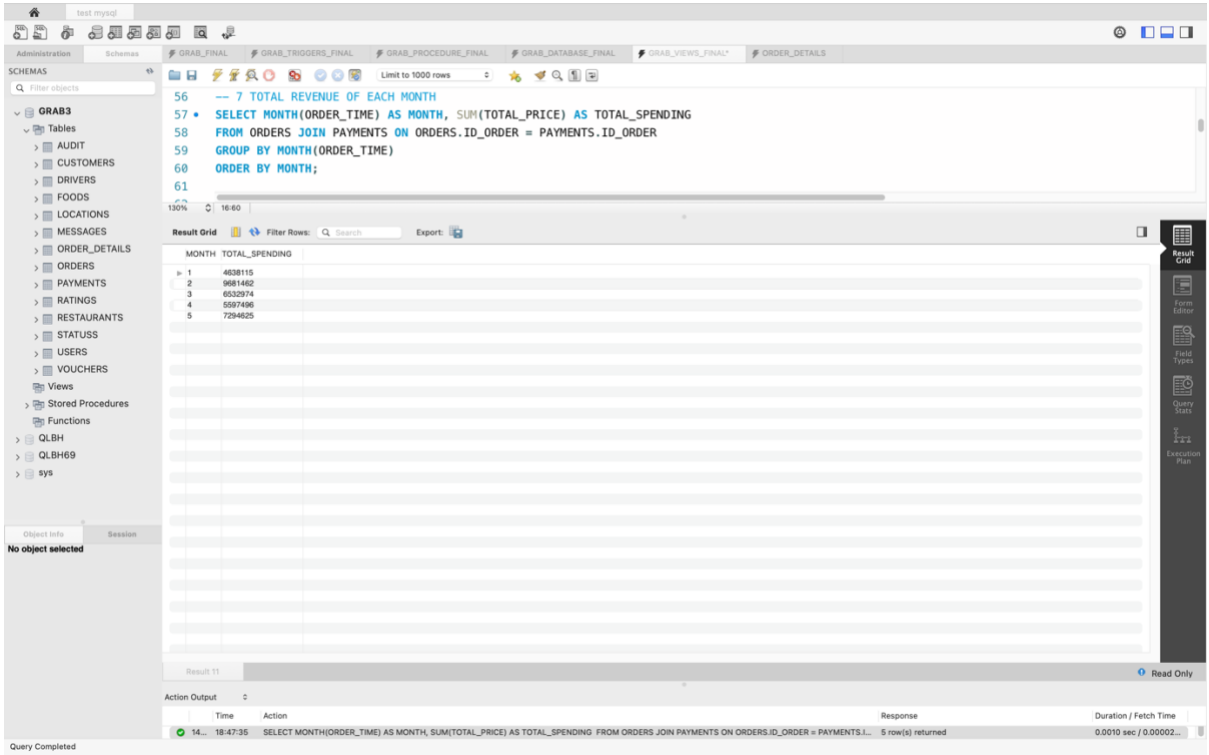
- Top foods ordered by customers under 25 years old. This will not only help GrabFood visualize the trend in a specific market share but also help the restaurants to know which food that increase their revenue so that they can either improve the dish's quality or increase the stocks.



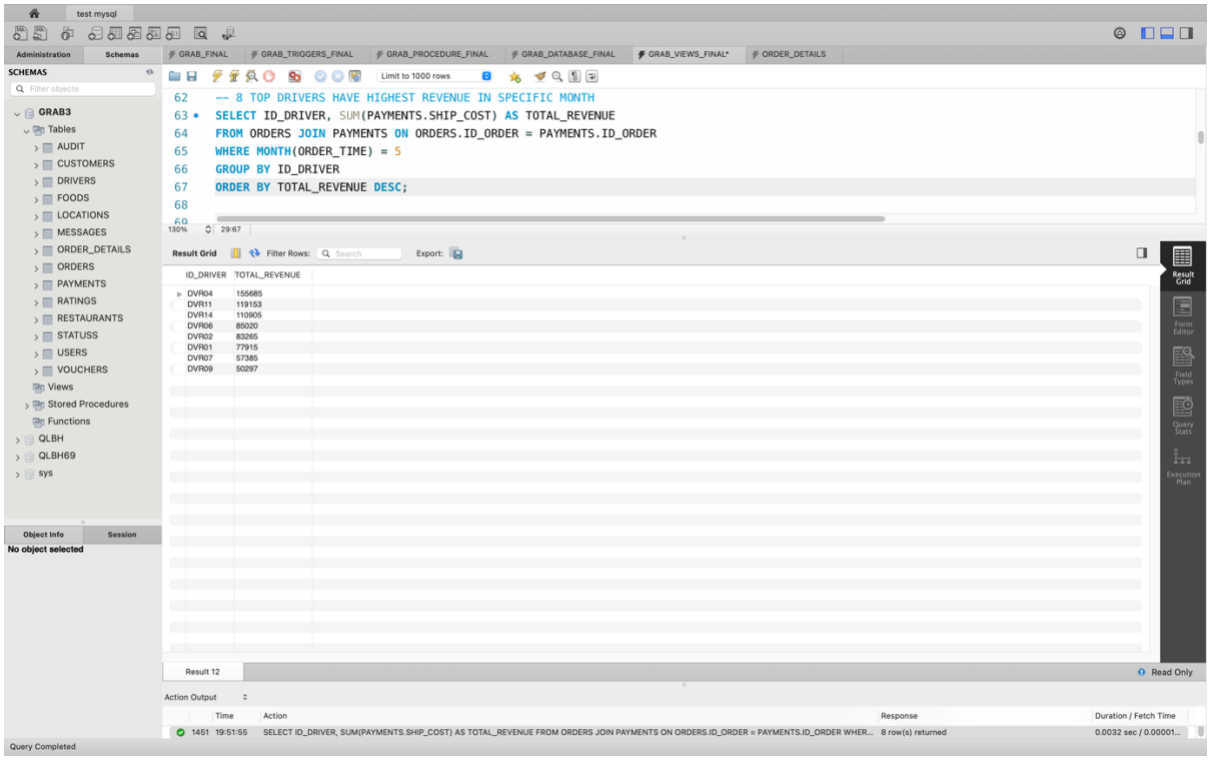
- Top customers have highest spending in specific month in order to discover whether there is a special occasion in that month which made customers spend more than usual.



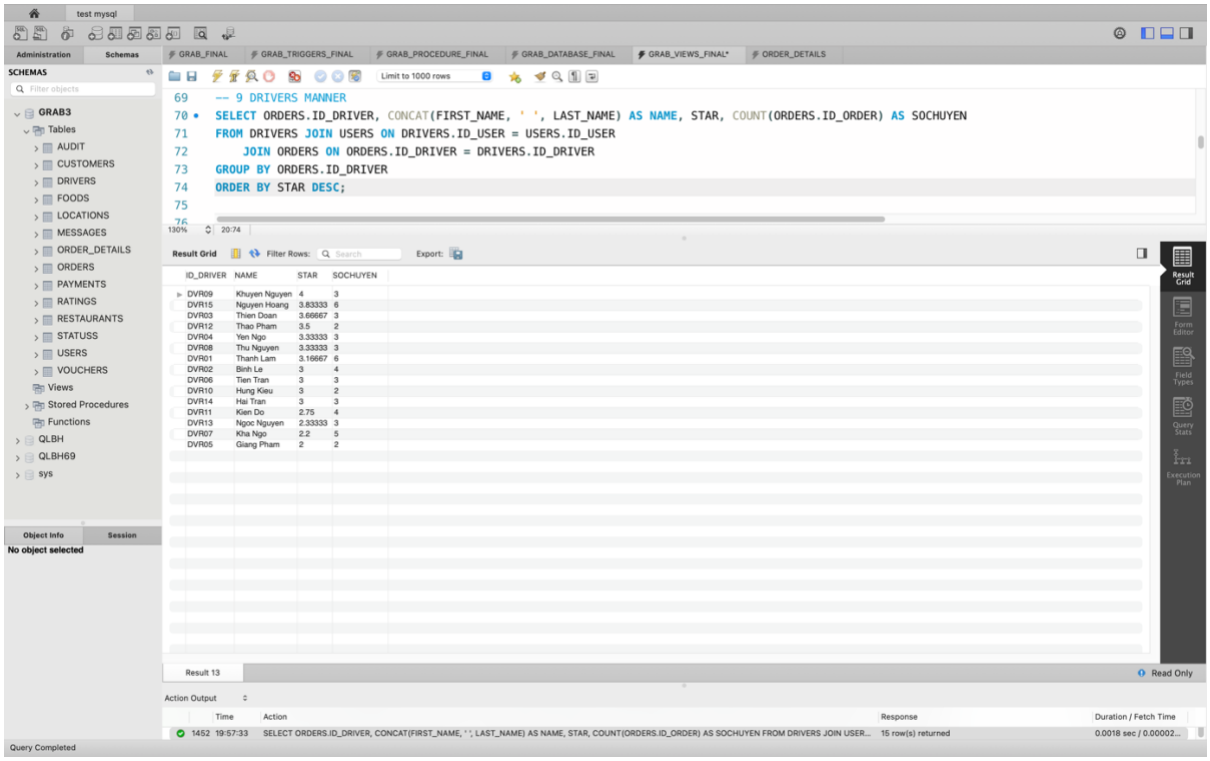
- Total GrabFood’s revenue in each month



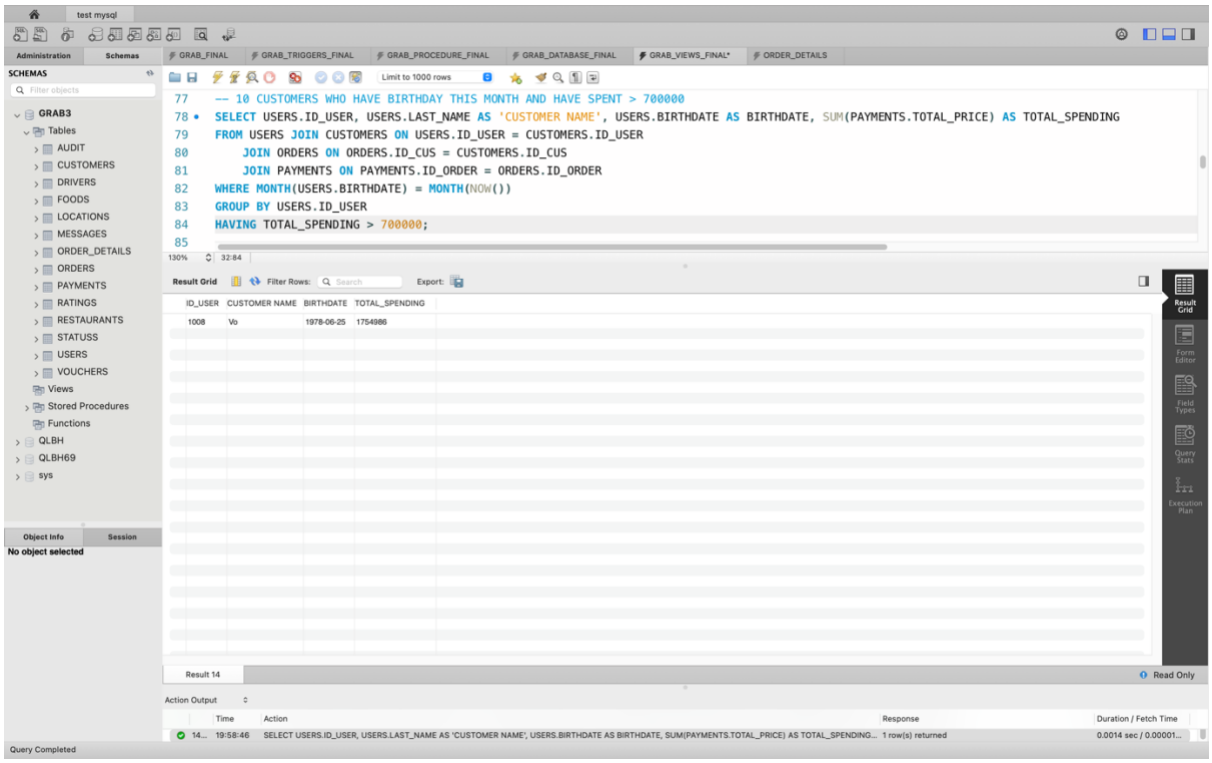
- Top drivers have highest revenue in May. These are the people who bring the most profit for GrabFood so that Grab might reward these drivers.



- Top 5 drivers have the highest ratings. This table points out drivers might not earn the most money, but they do provide excellent services which satisfy the customers.

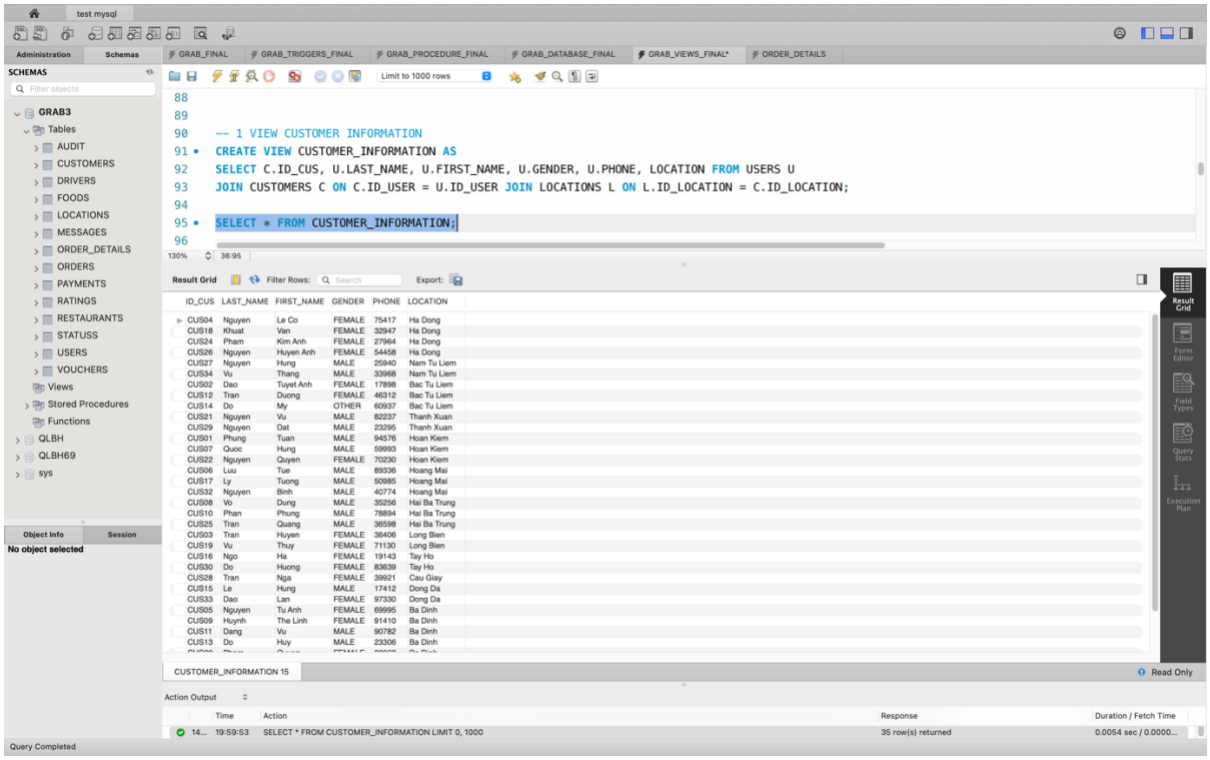


- Customers who have the birthday in this month and spent more than 700.000 VNĐ to help restaurants to offer suitable promotions to customers who have the birthday in this month and have spent a certain amount of money for food delivery service.



5.2. VIEW

- **Customer information**: View CUSTOMER_INFORMATION displays some basic information of customers, such as name, gender, phone, location. Drivers could have this information about customer to deliver food easier. Some private information such as credit card number or security code of customers must have been secured.

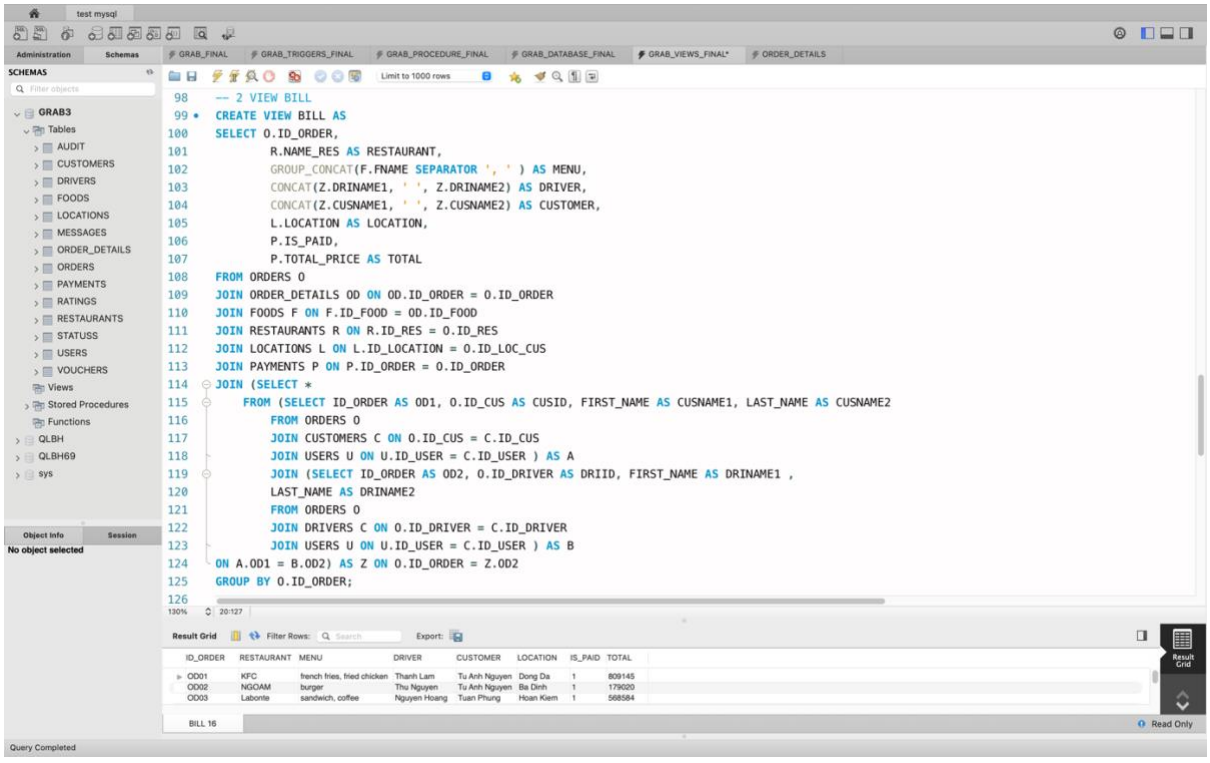


```
-- 1 VIEW CUSTOMER INFORMATION
CREATE VIEW CUSTOMER_INFORMATION AS
SELECT C.ID_CUS, U.LAST_NAME, U.FIRST_NAME, U.GENDER, U.PHONE, LOCATION FROM USERS U
JOIN CUSTOMERS C ON C.ID_USER = U.ID_USER JOIN LOCATIONS L ON L.ID_LOCATION = C.ID_LOCATION;

SELECT * FROM CUSTOMER_INFORMATION;
```

ID_CUS	LAST_NAME	FIRST_NAME	GENDER	PHONE	LOCATION
CUS04	Nguyen	Le Co	FEMALE	75417	Ha Dong
CUS18	Khuat	Van	FEMALE	32947	Ha Dong
CUS24	Pham	Kim Anh	FEMALE	27964	Ha Dong
CUS26	Nguyen	Huyen Anh	FEMALE	54458	Ha Dong
CUS27	Nguyen	Hung	MALE	25940	Nam Tu Liem
CUS34	Vu	Thang	MALE	33968	Nam Tu Liem
CUS02	Dao	Tuyet Anh	FEMALE	17898	Bac Tu Liem
CUS12	Tran	Duong	FEMALE	46312	Bac Tu Liem
CUS14	Do	My	OTHER	60937	Bac Tu Liem
CUS21	Nguyen	Vu	MALE	82237	Thanh Xuan
CUS29	Nguyen	Dat	MALE	23295	Thanh Xuan
CUS01	Phung	Tuan	MALE	94576	Hoan Kiem
CUS07	Quoc	Hung	MALE	59993	Hoan Kiem
CUS22	Nguyen	Quyen	FEMALE	70230	Hoan Kiem
CUS06	Luu	Tue	MALE	89336	Hoang Mai
CUS17	Ly	Tuong	MALE	50865	Hoang Mai
CUS32	Nguyen	Binh	MALE	40774	Hoang Mai
CUS08	Vo	Dung	MALE	35256	Hai Ba Trung
CUS10	Phan	Phung	MALE	78894	Hai Ba Trung
CUS25	Tran	Quang	MALE	36598	Hai Ba Trung
CUS03	Tran	Huyen	FEMALE	36406	Long Bien
CUS19	Vu	Thuy	FEMALE	71130	Long Bien
CUS16	Ngo	Ha	FEMALE	19143	Tay Ho
CUS30	Do	Huong	FEMALE	83639	Tay Ho
CUS28	Tran	Nga	FEMALE	39921	Cau Giay
CUS15	Le	Hung	MALE	17412	Dong Da
CUS33	Dao	Lan	FEMALE	97330	Dong Da
CUS05	Nguyen	Tu Anh	FEMALE	69995	Ba Dinh
CUS09	Huynh	The Linh	FEMALE	91410	Ba Dinh
CUS11	Dang	Vu	MALE	90782	Ba Dinh
CUS13	Do	Huy	MALE	23306	Ba Dinh

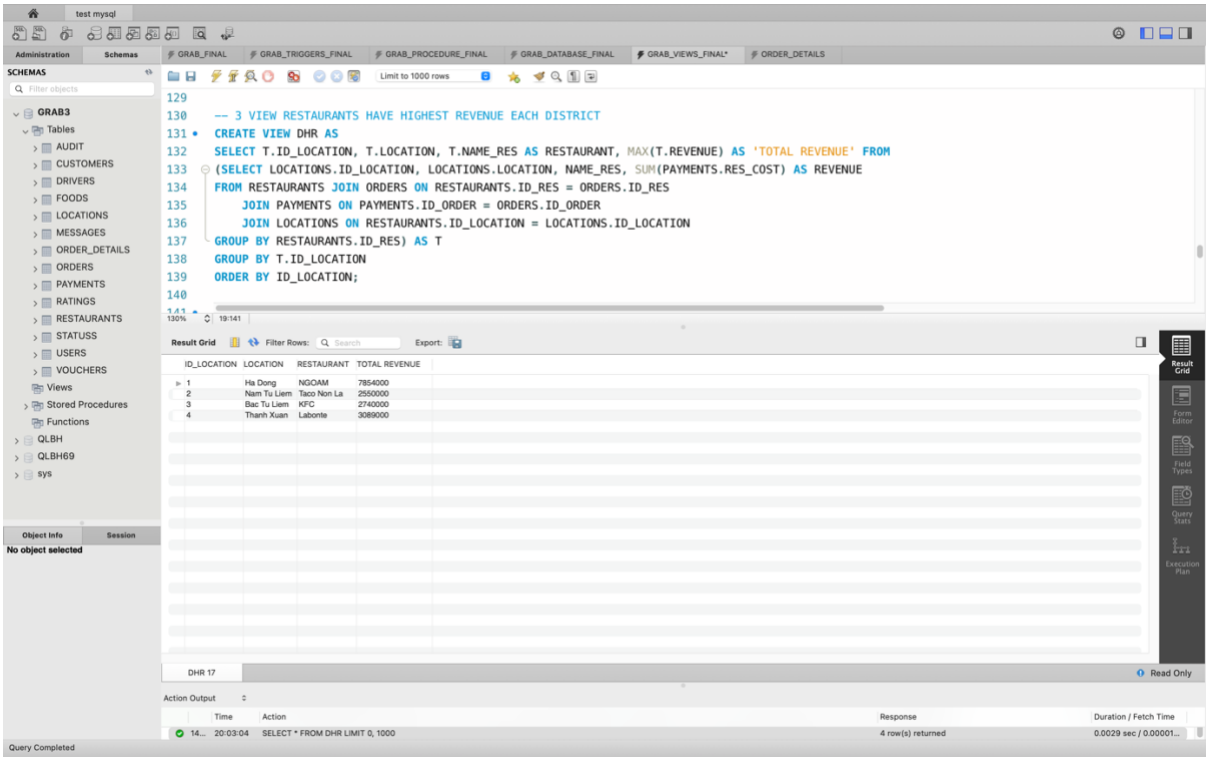
- **Bill**: View BILL print out the bill of accomplised order (after customers ordering the food and the chosen restaurant has finished cooking). Informations which were printed on the bill consist of restaurant's name, customer's name, driver's name, food, location, payments in details.



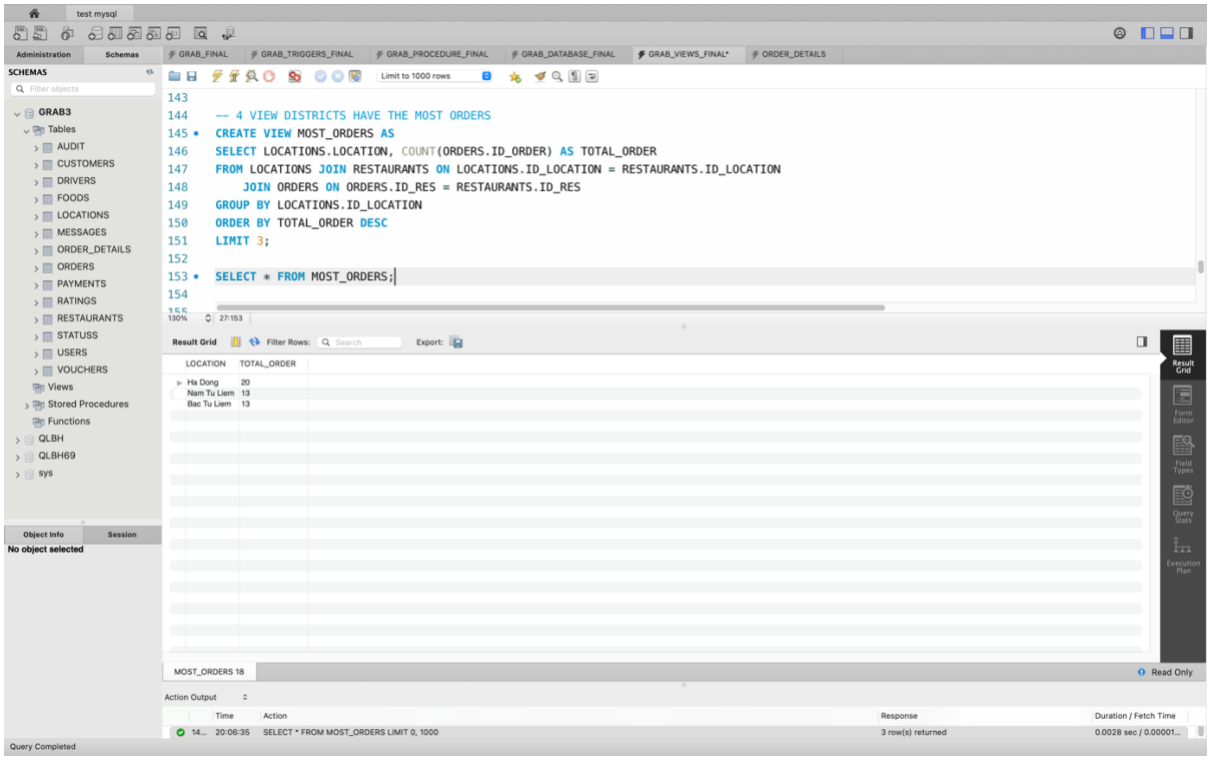
```
-- 2 VIEW BILL
CREATE VIEW BILL AS
SELECT O.ID_ORDER,
       R.NAME_RES AS RESTAURANT,
       GROUP_CONCAT(F.FNAME SEPARATOR ', ' ) AS MENU,
       CONCAT(Z.DRINAME1, ' ', Z.DRINAME2) AS DRIVER,
       CONCAT(Z.CUSNAME1, ' ', Z.CUSNAME2) AS CUSTOMER,
       L.LOCATION AS LOCATION,
       P.IS_PAID,
       P.TOTAL_PRICE AS TOTAL
FROM ORDERS O
JOIN ORDER_DETAILS OD ON OD.ID_ORDER = O.ID_ORDER
JOIN FOODS F ON F.ID_FOOD = OD.ID_FOOD
JOIN RESTAURANTS R ON R.ID_RES = O.ID_RES
JOIN LOCATIONS L ON L.ID_LOCATION = O.ID_LOC_CUS
JOIN PAYMENTS P ON P.ID_ORDER = O.ID_ORDER
JOIN (SELECT *
    FROM (SELECT ID_ORDER AS OD1, O.ID_CUS AS CUSID, FIRST_NAME AS CUSNAME1, LAST_NAME AS CUSNAME2
        FROM ORDERS O
        JOIN CUSTOMERS C ON O.ID_CUS = C.ID_CUS
        JOIN USERS U ON U.ID_USER = C.ID_USER ) AS A
        JOIN (SELECT ID_ORDER AS OD2, O.ID_DRIVER AS DRIID, FIRST_NAME AS DRINAME1 ,
        LAST_NAME AS DRINAME2
        FROM ORDERS O
        JOIN DRIVERS C ON O.ID_DRIVER = C.ID_DRIVER
        JOIN USERS U ON U.ID_USER = C.ID_USER ) AS B
ON A.OD1 = B.OD2) AS Z ON O.ID_ORDER = Z.OD2
GROUP BY O.ID_ORDER;
```

ID_ORDER	RESTAURANT	MENU	DRIVER	CUSTOMER	LOCATION	IS_PAID	TOTAL
OD01	KFC	french fries, fried chicken	Thanh Lam	Tu Anh Nguyen	Dong Da	1	809145
OD02	NGOAM	burger	Thu Nguyen	Tu Anh Nguyen	Ba Dinh	1	179020
OD03	Labonte	sandwich, coffee	Nguyen Hoang	Tuan Phung	Hoan Kiem	1	568584

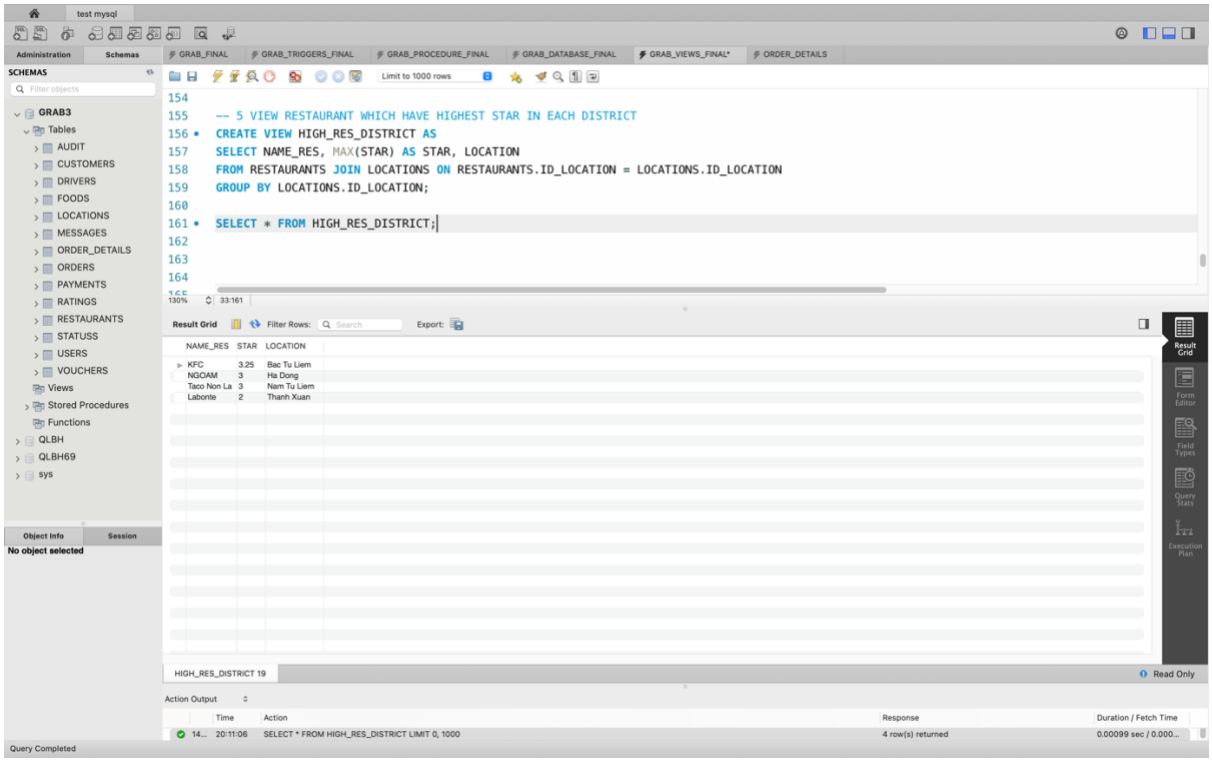
- **Restaurants that have highest revenue in each district**: View DHR displays the list of restaurants that have the highest revenue from districts by descending order so that they can promote the marketing strategy for that place, and also to catch the trend of the market.



- **Districts have the most orders:** View MOST_ORDERS displays the list of orders of from districts (through restaurants in each district) by descending order. Then it shows the district with the most orders so that they can send more drivers to that place to meet the customer's need.



- **Restaurants which have highest star in each district:** It is helpful for customers who come to one district for the first time and want to find out which one is the best restaurants there.



6. TRIGGERS

Trigger is a procedure that is executed on the server side when an event such as Insert, Delete, or Update occurs. Trigger is a special type of stored procedure (with no parameters) that is executed automatically when there is a data modification event. In our project, Triggers are stored and managed in the database, which is used in case we want to update and control changes to the data in the table. The script and results are in GRAB_TRIGGERS_FINAL.sql file. In details, we created 11 Triggers:

6.1. TRIGGER ADD_SHIP_COST

Trigger ADD_SHIP_COST will add the shipping cost into PAYMENTS. It will take the distance from ID_LOC_CUS in ORDERS to ID_LOCATION in RESTAURANTS. Meanwhile, the shipping cost will be calculated by Procedure GET_SHIP_COST.

6.2. TRIGGER ADD_RES_COST

It calculates the total food price in each order. Simultaneously, it will be added into PAYMENTS.

6.3. TRIGGER ADD_PRICE_OD

After customer inserts food's name and quantity, Trigger ADD_PRICE_OD will calculate the price for that food and add into ORDER_DETAILS.

6.4. TRIGGER ADD_VOUCHER

Trigger ADD_VOUCHER will add a voucher into VOUCHER_PM bases on the payment method customer has chosen. It also set IS_PAID to 0 if payment method is COD. With others, it will set to 1.

6.5. TRIGGER UPDATE_LOCATION

When customers insert into ORDERS, if they do not insert another address, UPDATE_LOCATION will set the ID_LOC_CUS to customers who made that order as default.

6.6. TRIGGER STOCK_LEFT

After every ORDER_DETAILS, it will update STOCKS by deducting the amount customers ordered. If the amount ordered is larger than the stocks, the system will not receive the food and displays a message “Only {stocks} {fname} left”.

6.7. TRIGGER UPDATE_MAX_USE

It is used to update MAX_USE in VOUCHERS by deducting by 1. If MAX_USE is less than or equal to 0, it will display a message “{name of voucher} out of voucher”. Also, if MAX_USE is equal to 0, the system will automatically update the discount (%) of that voucher to 0.

6.8. TRIGGER TOTAL_PRICE

This Trigger takes the sum of discounted food price after applying vouchers and shipping cost, then add into PAYMENTS.

6.9. TRIGGER UPDATE CANCEL

The objective of this Trigger is to check whether the order can be canceled or not. If the order could be canceled, the system will automatically set the RES_COST & SHIP_COST in table PAYMENTS to 0, which means that the restaurant and the driver do not gain any money from this order. Else if the order cannot be canceled, it will insert into table MESSAGES line “You cannot cancel this order!”.

6.10. TRIGGER UPDATE_DRIVER_RES

It will update ID_RES and ID_DRIVER into RATINGS after customers insert ID_ORDER.

6.11. TRIGGER ADD_STAR

Update STAR in DRIVER and STAR in RESTARANTS by calculating average of all the ratings.

7. PROCEDURES

Stored Procedure is compiled and stored in memory on initialization. That means it will execute faster than sending each SQL statement to MySQL. In this project, to avoid recompiling many times, which is very time consuming compared to precompiled, we use stored procedure to speed up code execution and can reuse code. The script and results are in GRAB_VIEWS_FINAL.sql file.

7.1. PROCEDURE UPDATE_STATUS

DRIVER will update the status of an order. If he or she updates the status of order to 4 (TAKEN), it will set the TAKEN_TIME in ORDERS to the moment.

7.2. PROCEDURE POP_MESSAGE

It will print out the latest messages.

7.3. PROCEDURE LAT_LO

This Procedure take an ID_LOCATION and set a pair of latitude and longitude to the location in order to simplify the calculating process.

7.4. PROCEDURE GET_SHIP_COST

After getting the latitude and longitude of the location, GET_SHIP_COST will calculate the shipping cost customers must pay by taking the distance times 8000(VNĐ/km).

7.5. PROCEDURE AGE_GROUP_FOOD

After inputting the age range, the procedure will print out foods that have been ordered the most by customers that are in the range.

7.6. PROCEDURE GROUP_AGE_F

It will print out all the customers in some age ranges' favorite food.

7.7. PROCEDURE AGE_GROUP_SPENDING

It will print out the total amount of money spent by customers in the given age range.

7.8. PROCEDURE GROUP_AGE_S

It will print out the total amount of money spent by customers in some age ranges.

THE END