# **Database Management**

Restaurant Database

Team Members: Varsha Menon, Sai Charan Pappala, Abdoulaye Djire, James Vining

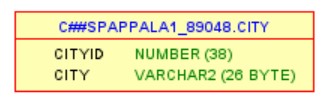
# Requirements

The client’s member management team wants to keep the current member order management practices and conduct analyses to better understand the members’ behaviors and preferences.

1. (45 points) Design the member order management database;
   1. Identify the entities, attributes, and relationships from the sample data.
   2. Draw an entity-relationship (E-R) diagram using the Crow’s feet or Chen notation.
   3. Transform the E-R diagram into a relational database design that includes all the keys, data types, and constraints (e.g., null or not-null values). Show how the transformation rules are applied.
2. Entities and Attributes:
   1. **Entity:** City, **Attributes:** CityID, City, **Relationship:** City has one-to-many relationship with Member and Restaurant
   2. **Entity:** Item\_Type, **Attributes:** ItemTypeID, ItemType, **Relationship:** Item\_Type has a one-to-many relationship with Meal
   3. **Entity:** Meal, **Attributes:** MealID, HotCold, MealName, Price, **Relationship:** Meal has a one-to-one relationship with Item\_Type and Meal\_Type
   4. **Entity:** Meal\_Type, **Attribute:** MealTypeID, MealType, **Relationship:** Meal\_Type has a one-to-many relationship with Meal
   5. **Entity:** Member, **Attribute:** MemberID, FirstName, LastName, Gender, Email, Monthly Budget, **Relationship:** Member has a one-to-many relationship with Order and Member\_Monthly\_Total and a one-to-one relationship with City
   6. **Entity:** Member\_Monthly\_Total, **Attribute:** MemberID, FirstName, LastName, Gender, Email, City, Year, Month, OrderCount, MealCount, **Relationship:** Member\_Monthly\_Total has a one-to-one relationship with Member
   7. **Entity:** Order, **Attribute:** OrderID, Date, Time, TotalOrder **Relationship:** Order has a one-to-one relationship with Restaurant and Member and one-to-many relationship with Order\_Detail
   8. **Entity:** Order\_Detail, **Attribute:** OrderDetailID, **Relationship:** Order\_Detail has a one-to-one relationship with both Meal and Order
   9. **Entity:** Restaurant, **Attribute:** RestaurantID, RestaurantName, RestaurantTypeID, IncomePercentage, **Relationship:** Restaurant has a one-to-many relationship with Order and Meal and one-to-one relationship with City
3. ERD:

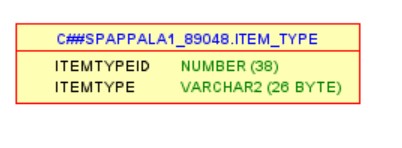
Diagram

Description automatically generated

1. Relational model:
   1. City (CityID, City,)  
      

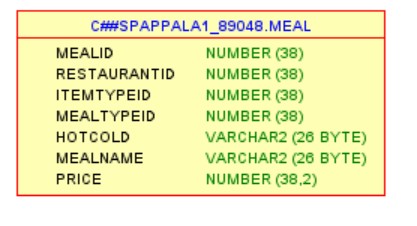
Primary Key: CityID

* 1. Item\_Type (ItemTypeID, ItemType)



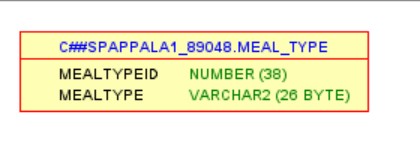
Primary Key: ItemTypeID

* 1. Meal (MealID, RestaurantID, ItemTypeID, MealTypeID, HotCold, MealName, Price)



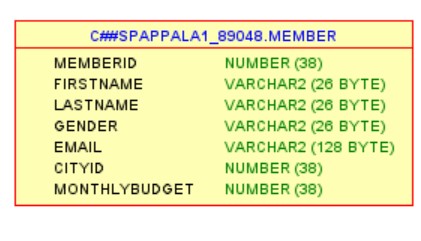
Primary Key: MealID, Foreign Key: RestaurantID, ItemTypeID, MealTypeID

* 1. Meal\_Type (MealTypeID, MealType)



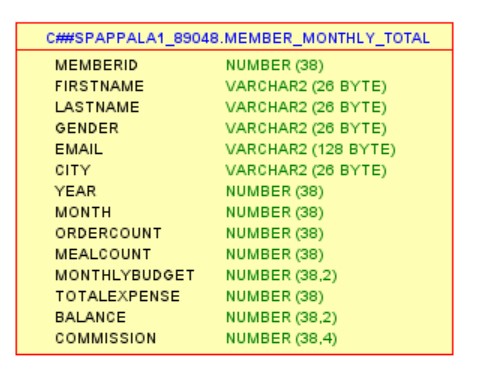
Primary Key: MealTypeID

* 1. Member (MemberID, FirstName, LastName, Gender, Email, CityID, Monthly Budget)



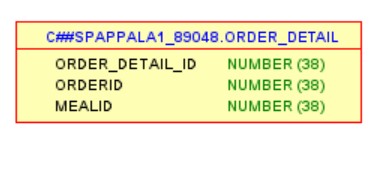
Primary Key: MemberID, Foreign Key: CityID

* 1. Member\_Monthly\_Total (MemberID, FirstName, LastName, Gender, Email, City, Year, Month, OrderCount, MealCount)



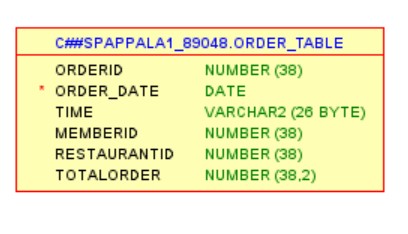
PrimaryKey: MemberID, Month, Year, Foreign Key: FirstName, LastName

* 1. Order (OrderID, Date, Time, MemberID, RestaurantID, TotalOrder)



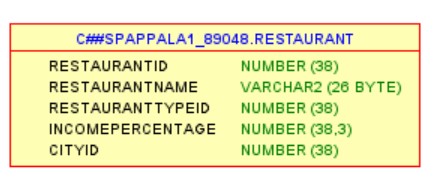
PrimaryKey: OrderID, ForeignKey: MemberID, RestaurantID

* 1. Order\_Detail (OrderDetailID, OrderID, MealID)



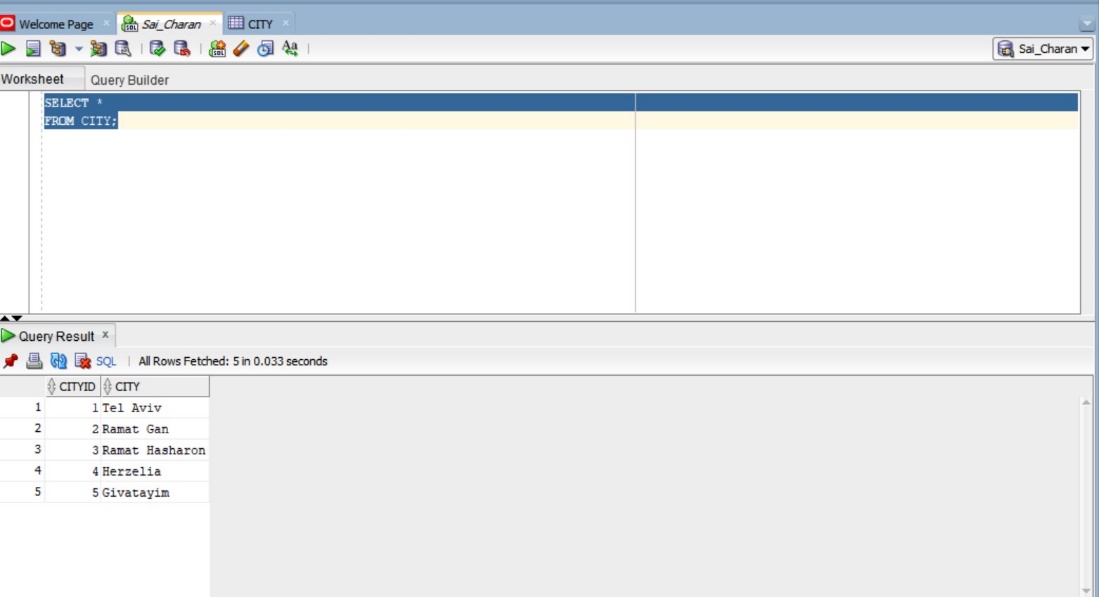
PrimaryKey: OrderDetailID, ForeignKey: OrderID, MealID

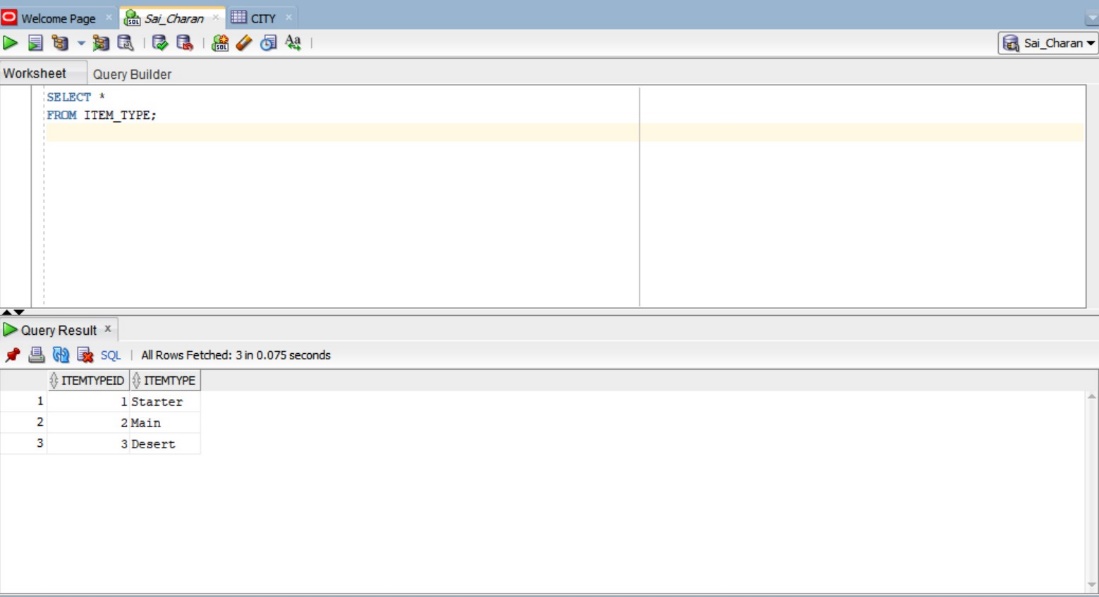
* 1. Restaurant (RestaurantID, RestaurantName, RestaurantTypeID, IncomePercentage, CityID)

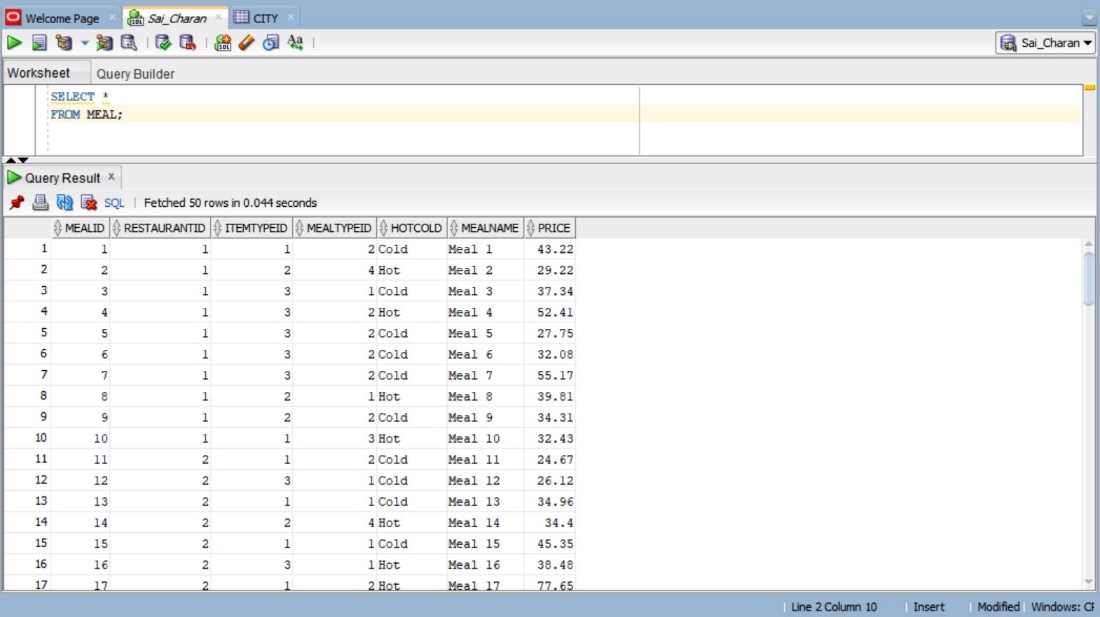


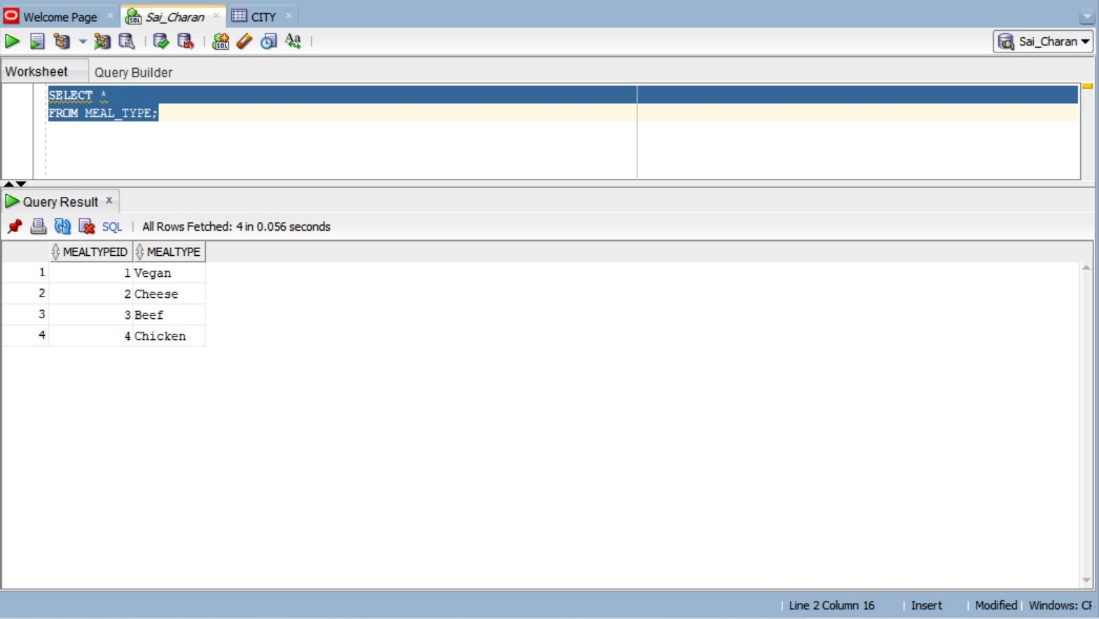
PrimaryKey: RestaurantID, ForeignKey: CityID

The transformation rules indicate that each entity becomes a relation with the same key. The second rule is that each relationship represented by foreign key or separate relation depending on min-max cardinality. In Meal, the foreign keys are RestaurantID, ItemTypeID and MealTypeID because of a 1:N relationship with Meal\_Type, Item\_Type and Restaurant. Member has a foreign key CityID because of a 1:N relationship with City. Order has foreign keys MemberID and RestaurantID because of 1:N relationship with Restaurant and Member. Order\_Detail has foreign keys MealID and OrderID because of 1:N relationship with Meal and Order. Restaurant has a foreign key CityID because of 1:N relationship with City

1. (15 points) Implement the database. Use Oracle, which is provided by GSU. Use a portion of the given data to populate the database; e.g., 15-25 entries for each entity relation for which the data exists. For each relation, show the data you use by using the Select \*; command.  
     
   a. CITY Table   
    
2. ITEM\_TYPE Table



1. MEAL Table   
   
2. MEAL\_TYPE Table



1. MEMBER Table

Graphical user interface, table

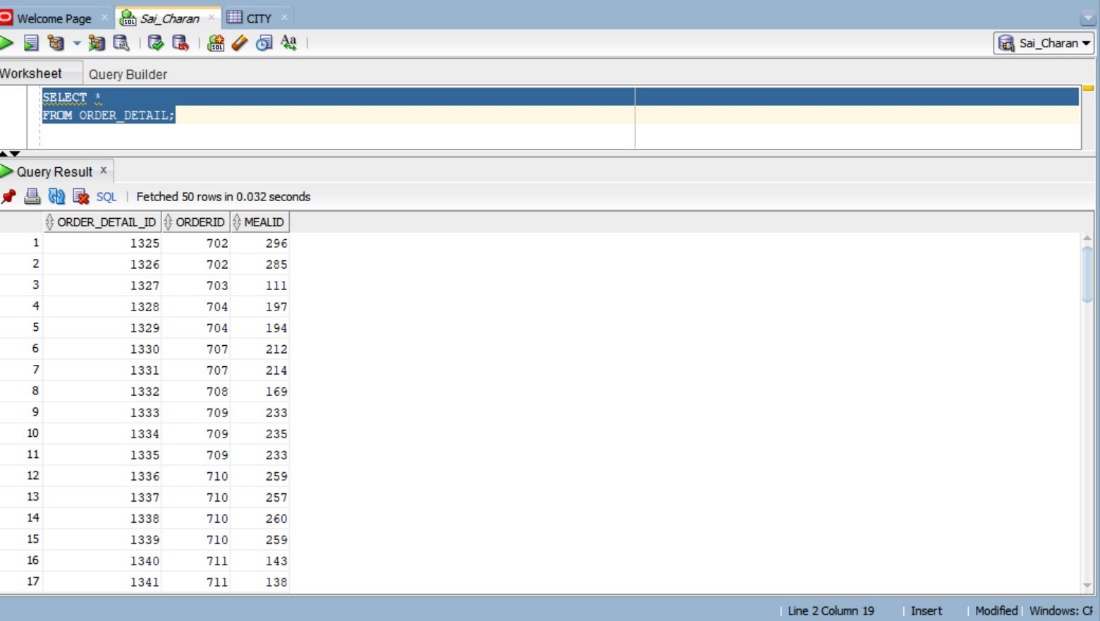
Description automatically generated

1. MEMBER\_MONTHLY\_TOTAL Table

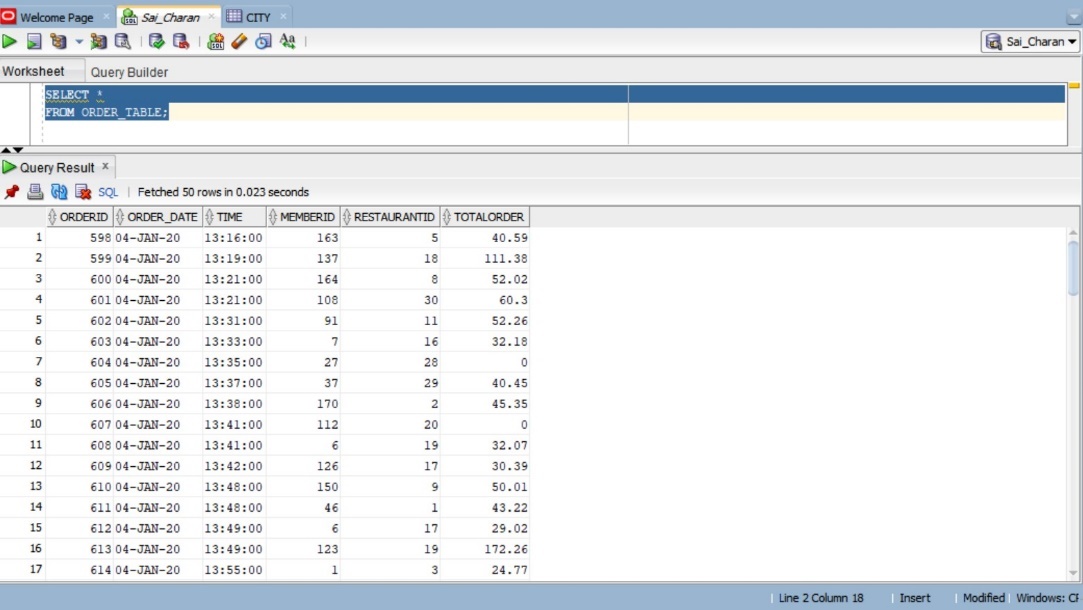
Graphical user interface, text, application, email

Description automatically generated

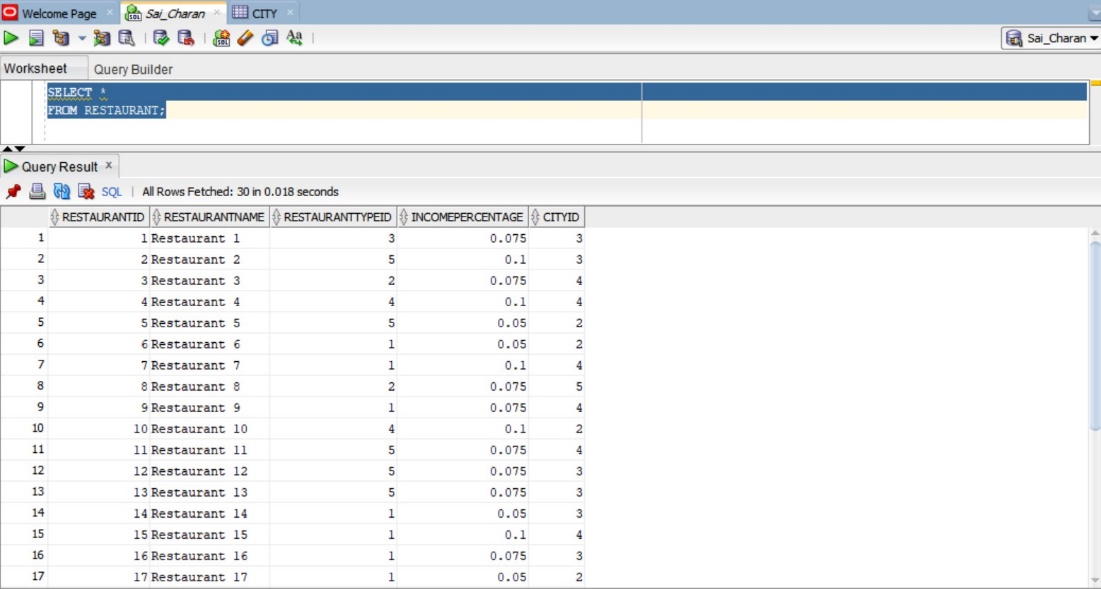
1. ORDER\_DETAIL Table



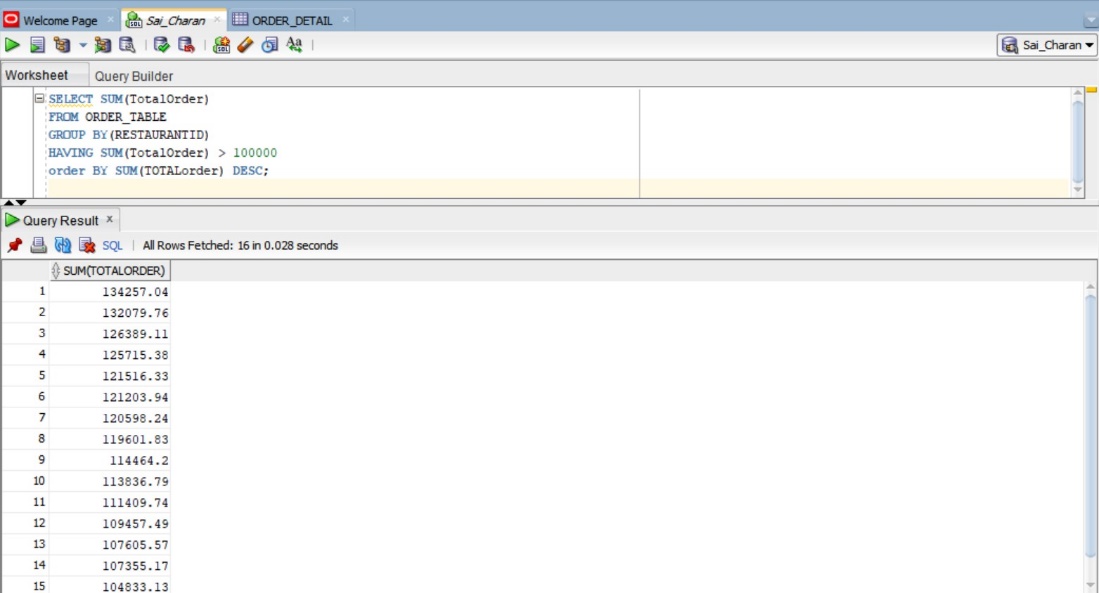
1. ORDER\_TABLE Table



1. RESTAURANT table



1. (25 points) Identify 10 important, non-trivial queries for this database. Write the queries in English and state the importance of each query. Write the queries in SQL and run them against the database. Show the SQL commands. Provide screen shots of the results obtained. Trivial queries are, for example, ‘retrieve the names of the customers.’ At least 7 of the queries should require a join operation.   
     
     
   Query#1   
    SELECT SUM(TotalOrder)  
   FROM ORDER\_TABLE  
   GROUP BY(RESTAURANTID) HAVING SUM(TotalOrder) > 100000   
   order BY SUM(TOTALorder) DESC;

**finding out how many restaurants crossing 100K total order volume ( 16) out of 30**.  
  
  
Query#2   
SELECT restaurant.restaurantid,city.city,sum(order\_table.totalorder)

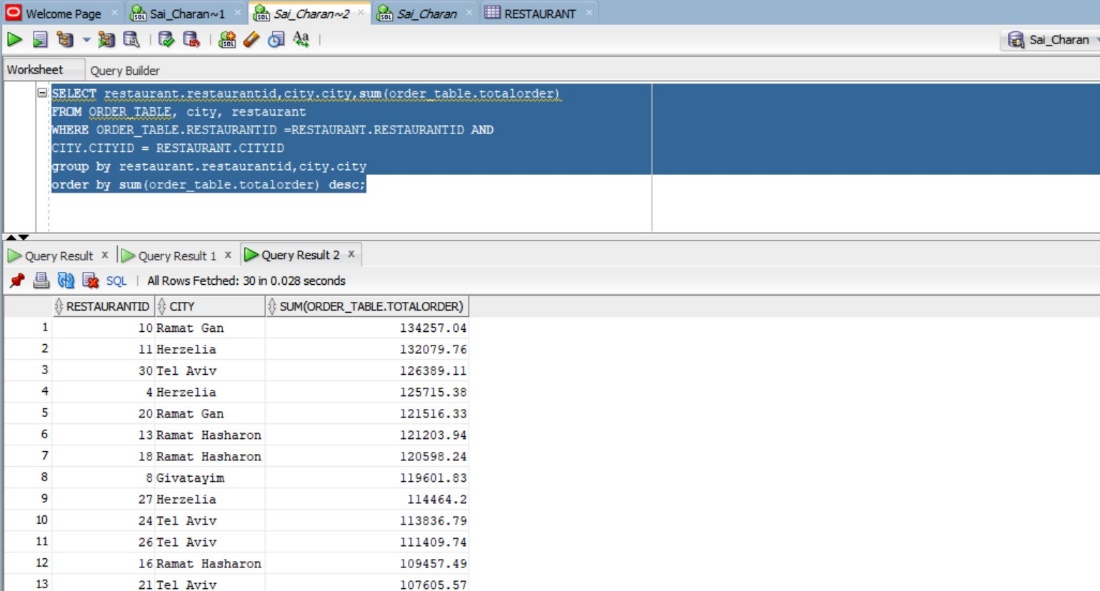
FROM ORDER\_TABLE, city, restaurant

WHERE ORDER\_TABLE.RESTAURANTID =RESTAURANT.RESTAURANTID AND

CITY.CITYID = RESTAURANT.CITYID

group by restaurant.restaurantid,city.city

order by sum(order\_table.totalorder) desc;



**From this query we understood that top 5 producing restaurants, we have ramat gan city (2), Herzelia(2) and Tel aviv(1).**   
**While maintaining the stock and inventory of the restaurants, these cities take huge consumption**.

**Query 3**

SELECT city.city,sum(order\_table.totalorder)

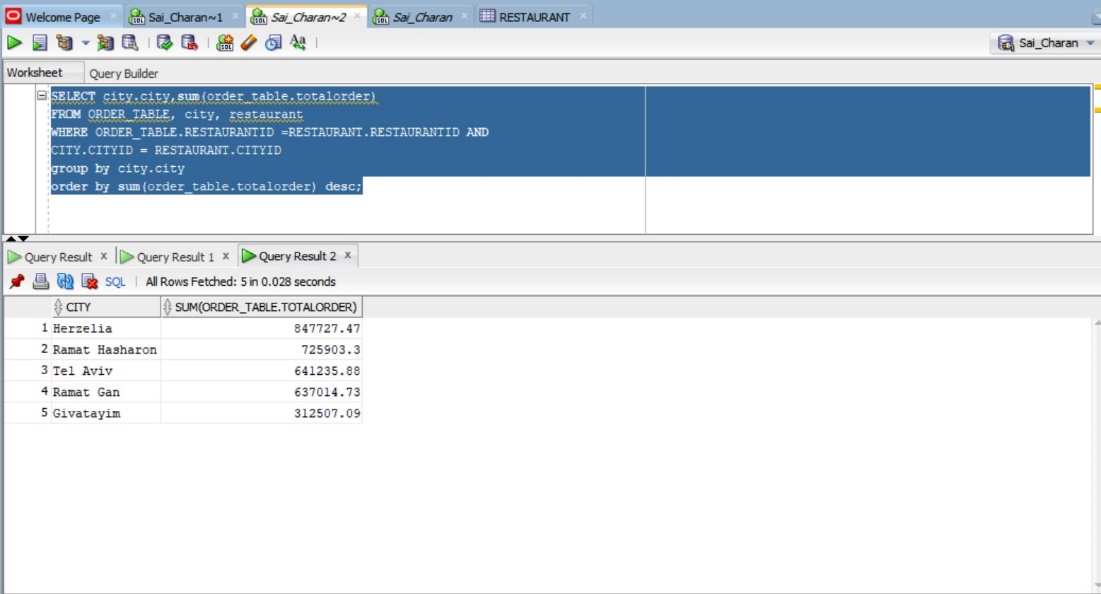
FROM ORDER\_TABLE, city, restaurant

WHERE ORDER\_TABLE.RESTAURANTID =RESTAURANT.RESTAURANTID AND

CITY.CITYID = RESTAURANT.CITYID

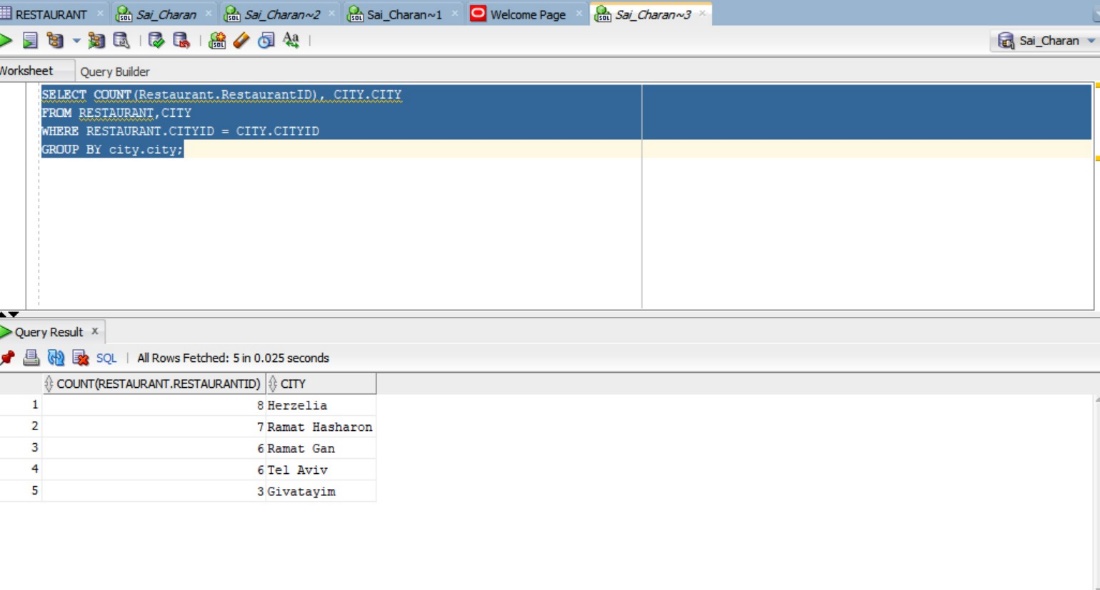
group by city.city

order by sum(order\_table.totalorder) desc;

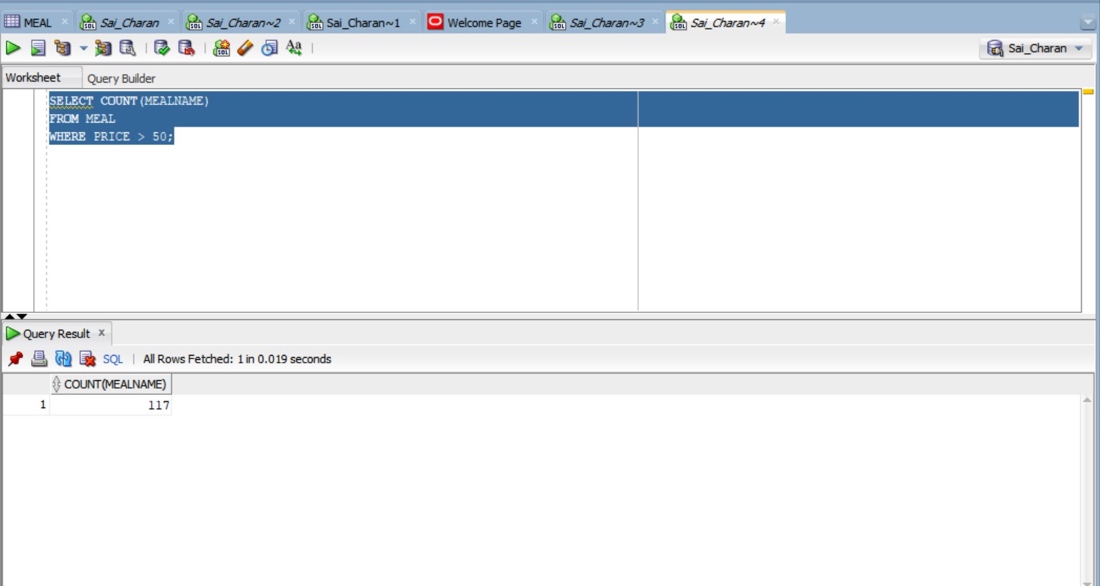
  
  
**Herzelia city produced the top turnover 847727.47 of sales while Givatayim Producing 312507.09 USD. It helps how much inventory and workforce must be divided.**   
  
**Query#4**   
  
  
**SELECT COUNT(Restaurant.RestaurantID), CITY.CITY**

**FROM RESTAURANT,CITY**

**WHERE RESTAURANT.CITYID = CITY.CITYID**

**GROUP BY city.city;**  
  
  
**Herzelia City has 8 restaurants followed by ramat hashron has 7 restaurants and finally givatim .**   
  
  
  
**Query#5**   
  
**SELECT COUNT(MEALNAME)**

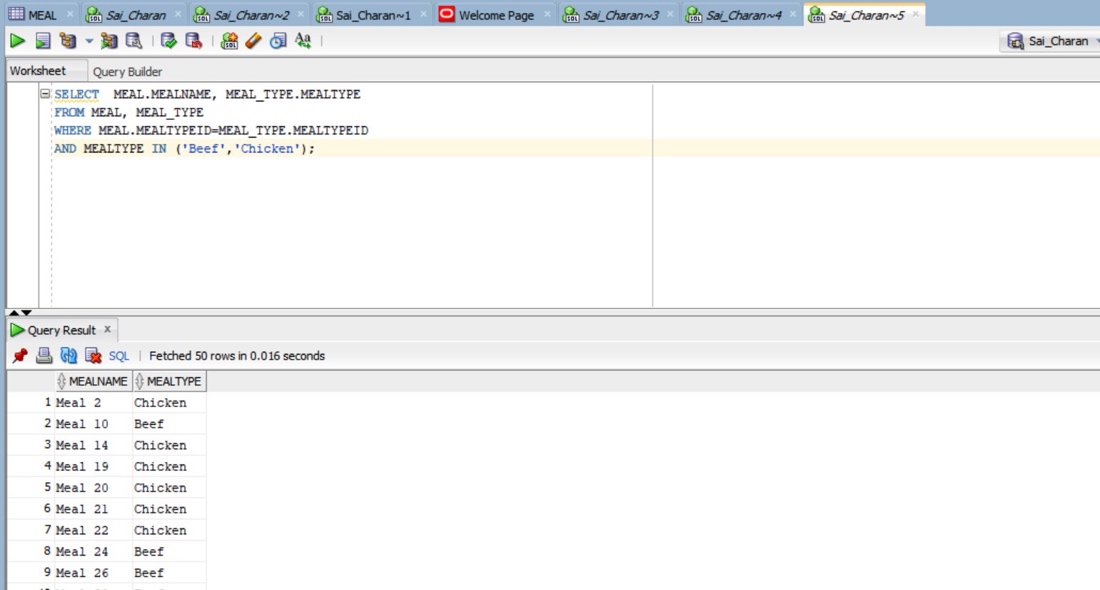
**FROM MEAL**

**WHERE PRICE > 50;**  
  
  
  
**117 meals out of 350 meals costs more than 50 dollars per meal.**

Query 6   
  
SELECT MEAL.MEALNAME

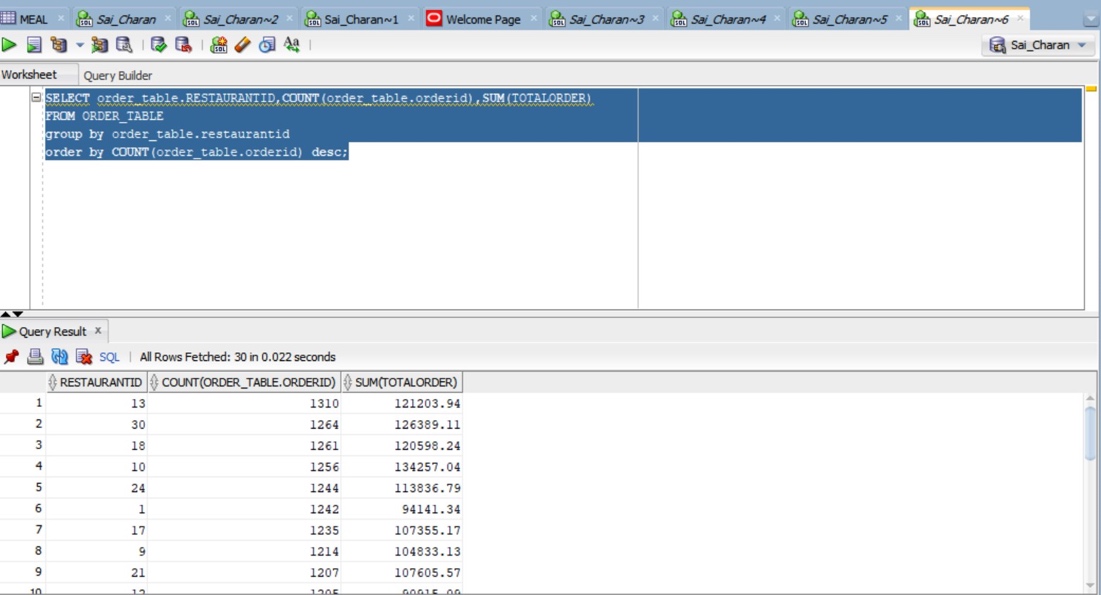
FROM MEAL, MEAL\_TYPE

WHERE MEAL.MEALTYPEID=MEAL\_TYPE.MEALTYPEID

AND MEALTYPE IN ('Beef','Chicken');  
  
  
  
**From this query we can know what type of meals beef and chicken and their respective meal name are and mealtype. It was quite interesting to know that 103 meals are meat based out of 350 different meals offered by restaurants**.  
  
QUERY 7   
  
SELECT order\_table.RESTAURANTID,COUNT(order\_table.orderid),SUM(TOTALORDER)

FROM ORDER\_TABLE

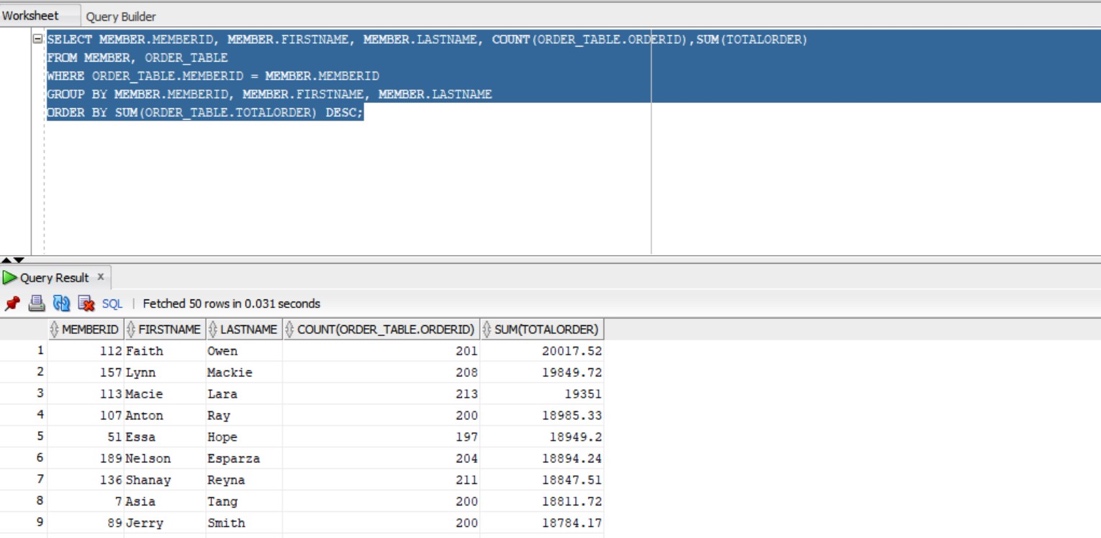
group by order\_table.restaurantid

order by COUNT(order\_table.orderid) desc;  
  
  
  
From this query we can understand from which restaurant orders are highest.   
Restaurant #13 -1310 Orders and restaurant #7 - 1129 have lowest orders.   
  
Query 8   
  
SELECT MEMBER.MEMBERID, MEMBER.FIRSTNAME, MEMBER.LASTNAME, COUNT(ORDER\_TABLE.ORDERID),SUM(TOTALORDER)

FROM MEMBER, ORDER\_TABLE

WHERE ORDER\_TABLE.MEMBERID = MEMBER.MEMBERID

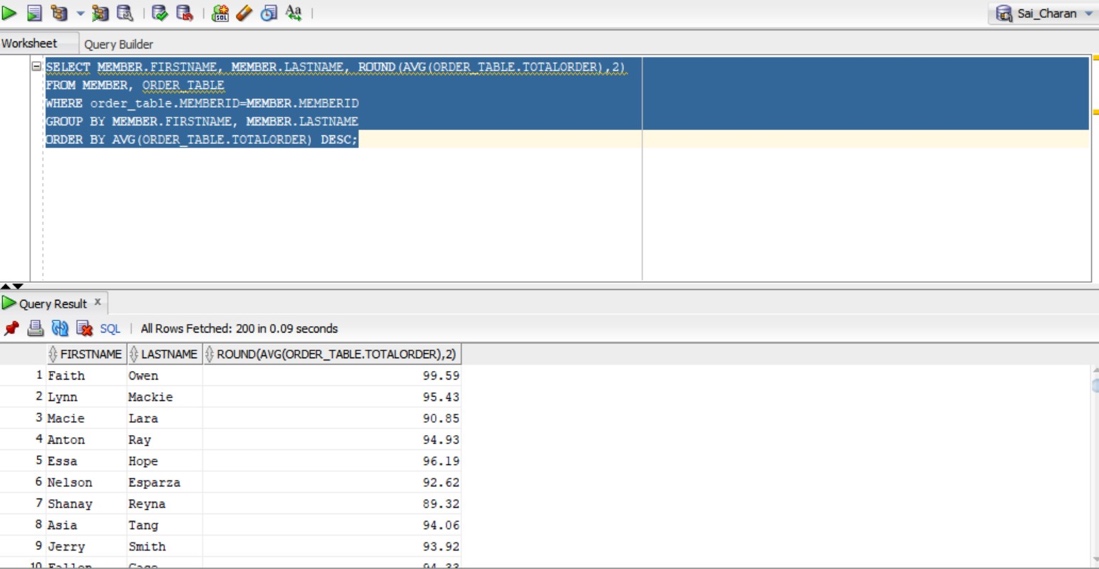
GROUP BY MEMBER.MEMBERID, MEMBER.FIRSTNAME, MEMBER.LASTNAME

ORDER BY SUM(ORDER\_TABLE.TOTALORDER) DESC;  
  
Which member made highest total order from the members and his first name and last name. When he visits a restaurant, we can avail best services for highest paid customers.   
  
Query 9   
  
SELECT MEMBER.FIRSTNAME, MEMBER.LASTNAME, ROUND(AVG(ORDER\_TABLE.TOTALORDER),2)

FROM MEMBER, ORDER\_TABLE

WHERE order\_table.MEMBERID=MEMBER.MEMBERID

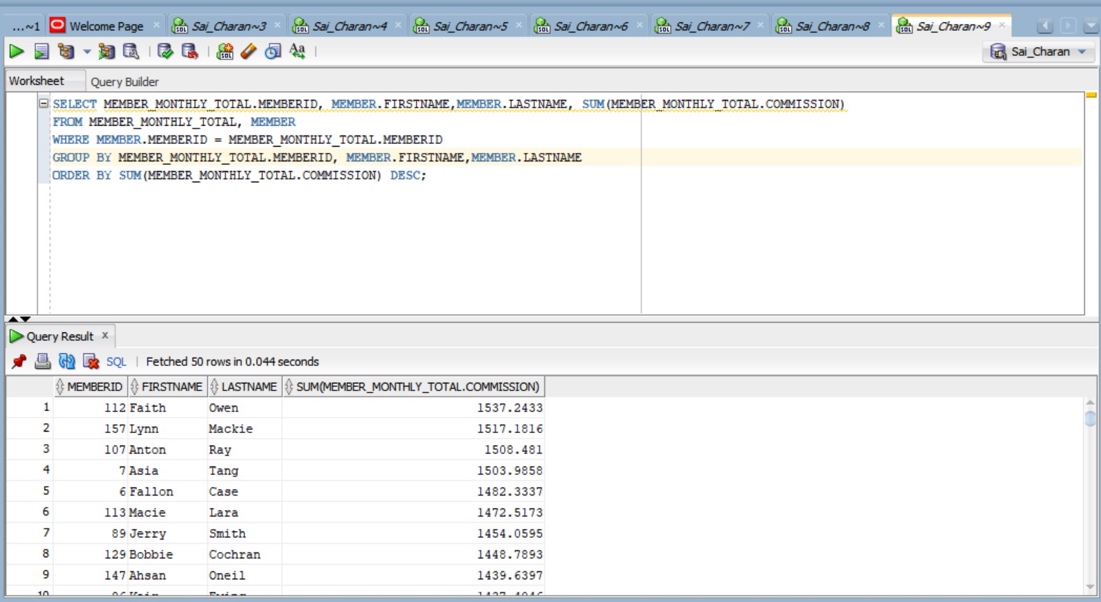
GROUP BY MEMBER.FIRSTNAME, MEMBER.LASTNAME

ORDER BY AVG(ORDER\_TABLE.TOTALORDER) DESC;  
  
  
From this query we can learn how much average a member spending , it ranges between 99.59 usd to 74.78 usd.   
  
Query 10   
  
SELECT MEMBER\_MONTHLY\_TOTAL.MEMBERID, MEMBER.FIRSTNAME,MEMBER.LASTNAME, SUM(MEMBER\_MONTHLY\_TOTAL.COMMISSION)

FROM MEMBER\_MONTHLY\_TOTAL, MEMBER

WHERE MEMBER.MEMBERID = MEMBER\_MONTHLY\_TOTAL.MEMBERID

GROUP BY MEMBER\_MONTHLY\_TOTAL.MEMBERID, MEMBER.FIRSTNAME,MEMBER.LASTNAME

ORDER BY SUM(MEMBER\_MONTHLY\_TOTAL.COMMISSION) DESC;  
  


From this query we can understand which member gets the highest commission and the list displayed descending wise.

1. (15 points) Discuss the real-world significance of this database. You will need to research and understand the purposes of customer relationship management systems. Then, based on your general knowledge of the application domain, and your research into customer relationship management systems, identify what additional constructs should be included in the database and state how they would be useful to the client. This question requires well-conceived and well-written arguments that reflect the role of data and its analysis in a company.

Customer relationship management systems are extremely important because they manage the company's relationships and interactions with current and potential customers. The system creates organization, efficiency, better time management, and satisfied customers. A business cannot be well run without knowing the results of their leads and what strategies they are doing that are progressing the company or hurting the company. Customer relationship management systems dive deeper than a spreadsheet by being able to track emails and contact has been opened, what pages were visited on your site, and purchase dates.

Additional constructs that should be included in the database to better aid the company would be the geolocation of where many of the customers purchases are. This could help the company by identifying which restaurants the customer will most likely order from and provide discounts to the customer for them to spend even more money at the restaurant the company has a partnership with. Another helpful tool is identifying what types of restaurants their customers are ordering from. From instance, if a customer frequently purchases vegan meals from a restaurant, they may be inclined to purchase vegan meals from another restaurant as well. Customer relationship management systems can play a pivotal role in turning a lead to a customer, keeping a customer, or knowing when further efforts will not work to turn a particular lead to a customer. Due to close competition between similar companies these systems can mean the difference in a company gaining market share.

Another additional construct is human resource management. If there are one or more restaurants to manage, a database helps manage information centrally such as personnel, salaries, budget, etc. It also helps with other personnel related information like shift information. Training and incentive programs for hard workers. It helps build the team and makes them feel important, even as the company keeps growing. A database helps scale the business and grow it, while keeping track of all the data efficiently.