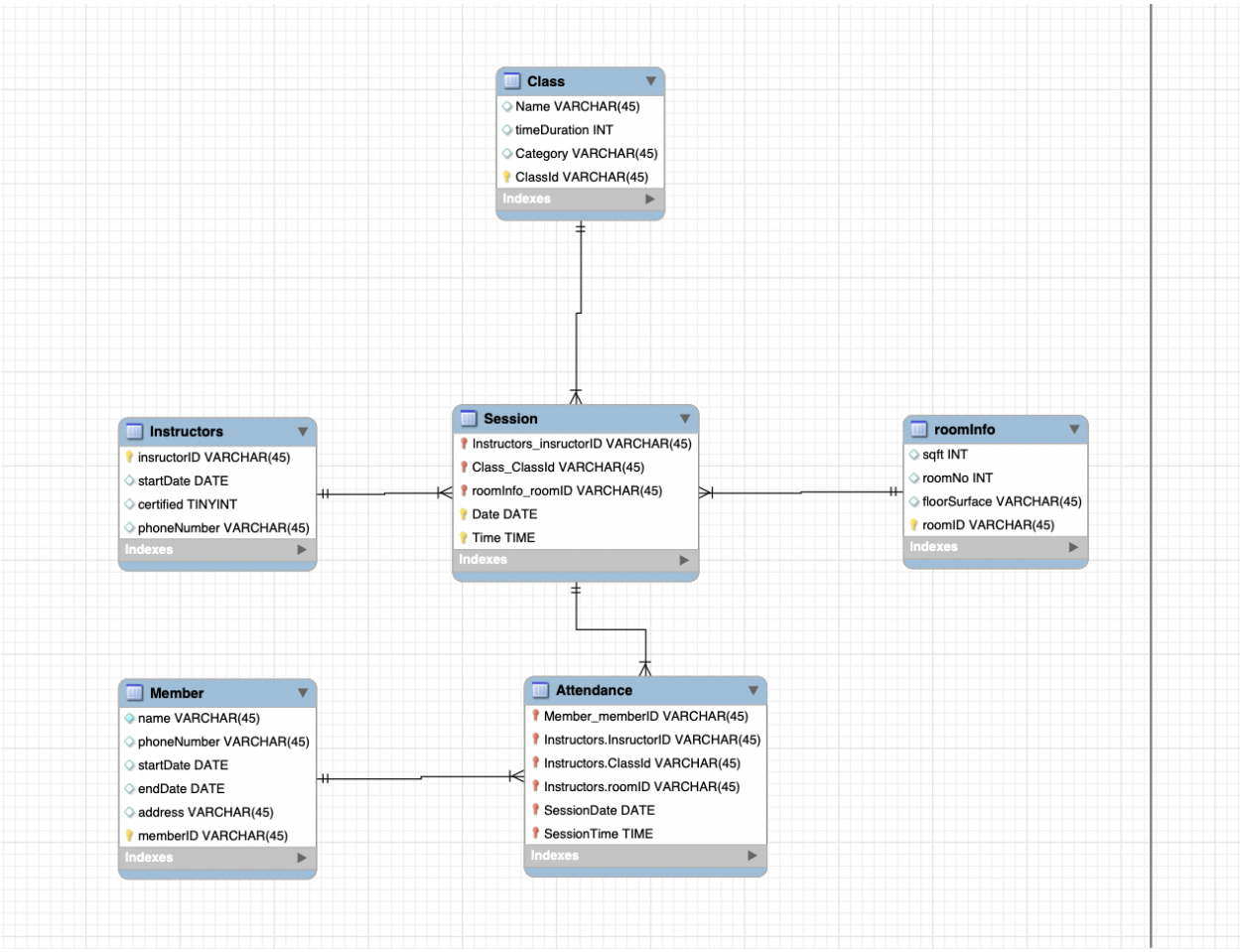
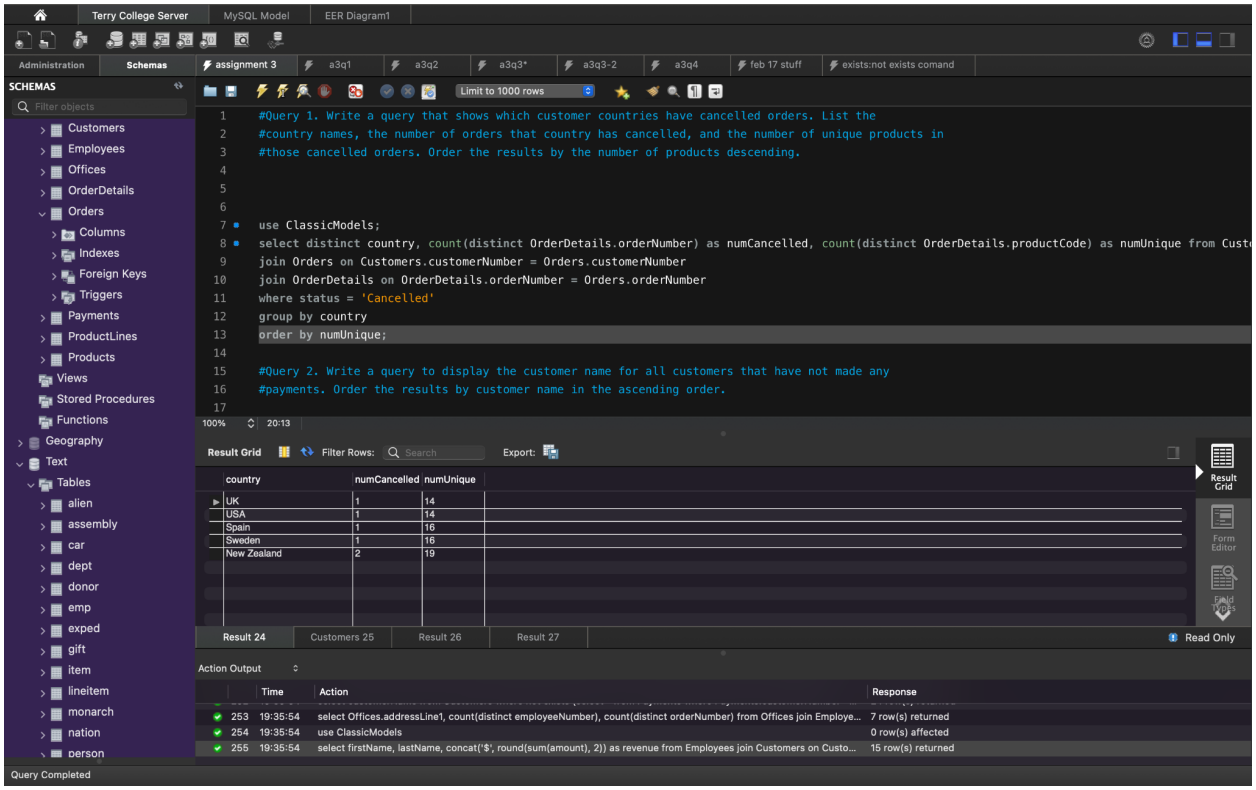


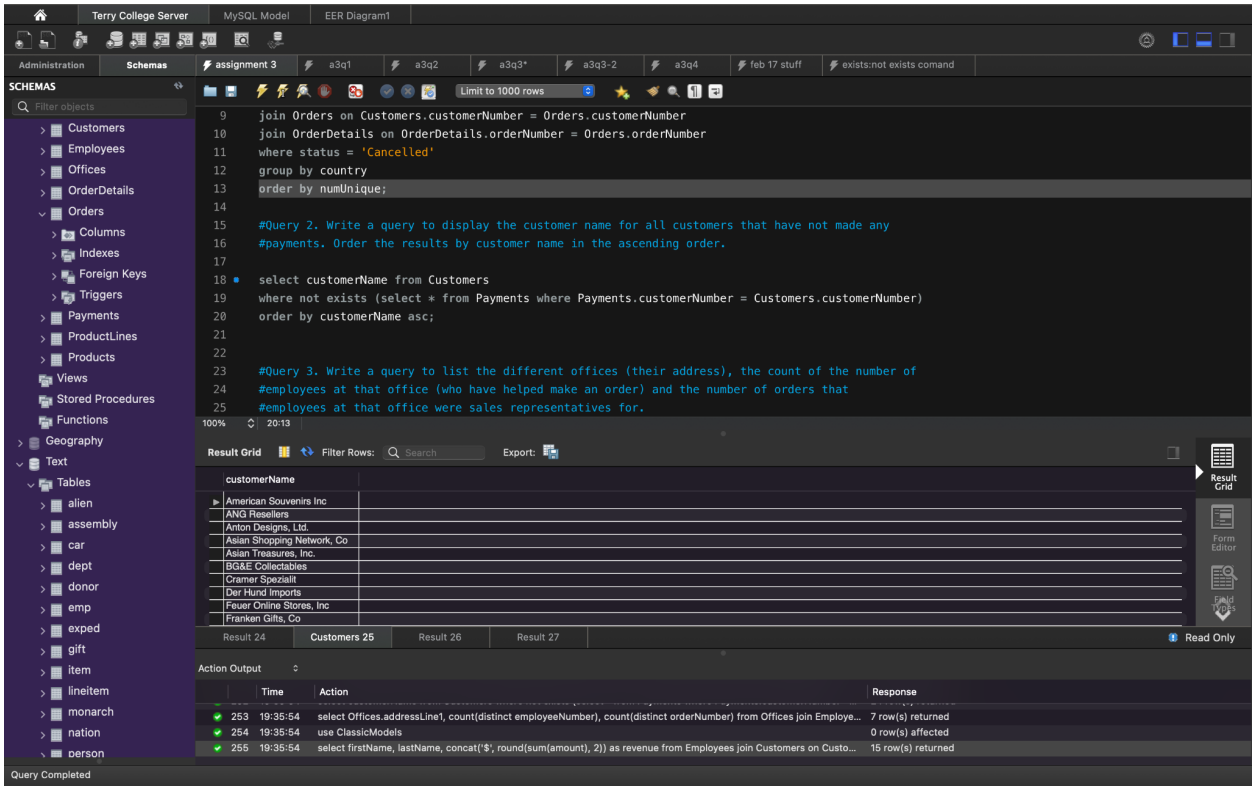
#Query 1. Write a query that shows which customer countries have cancelled orders. List the #country names, the number of orders that country has cancelled, and the number of unique products in
#those cancelled orders. Order the results by the number of products descending.

```
use ClassicModels;
select distinct country, count(distinct OrderDetails.orderNumber) as numCancelled,
count(distinct OrderDetails.productCode) as numUnique from Customers
join Orders on Customers.customerNumber = Orders.customerNumber
join OrderDetails on OrderDetails.orderNumber = Orders.orderNumber
where status = 'Cancelled'
group by country
order by numUnique;
```



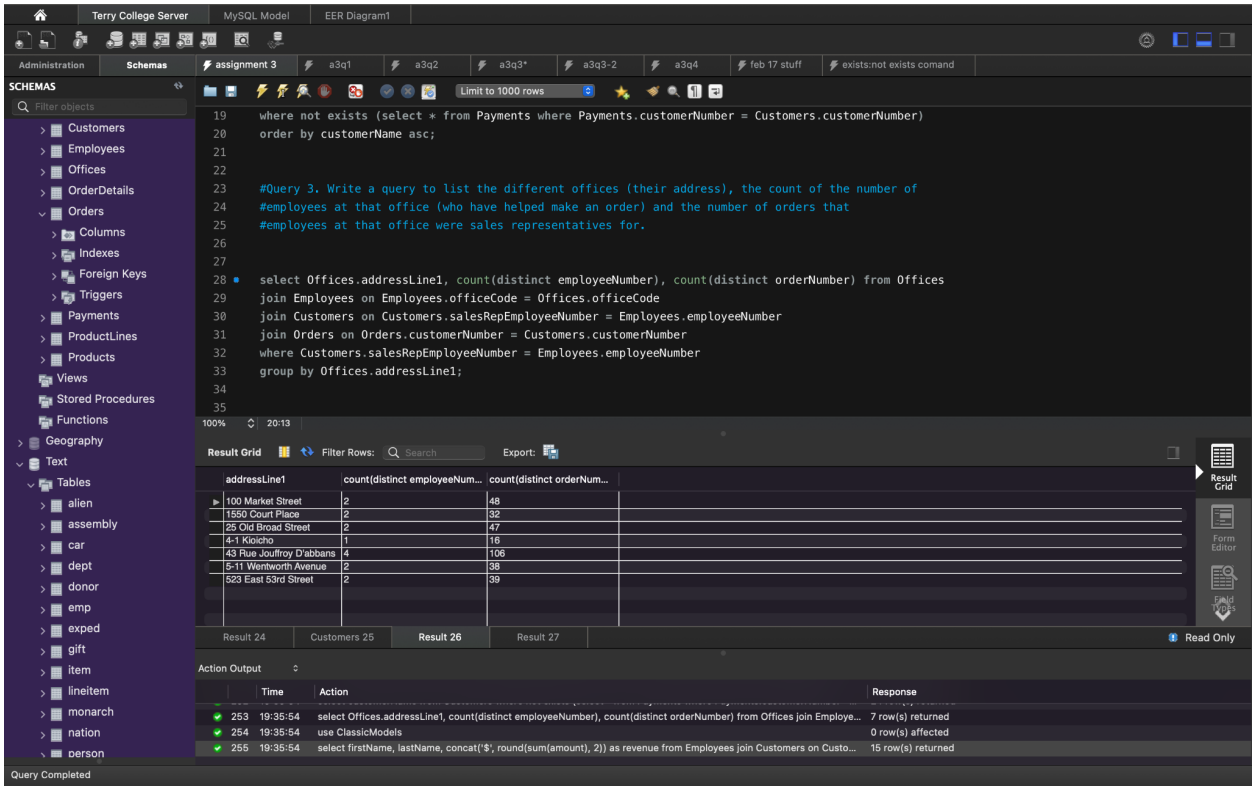
#Query 2. Write a query to display the customer name for all customers that have not made any #payments. Order the results by customer name in the ascending order.

```
select customerName from Customers
where not exists (select * from Payments where Payments.customerNumber =
Customers.customerNumber)
order by customerName asc;
```



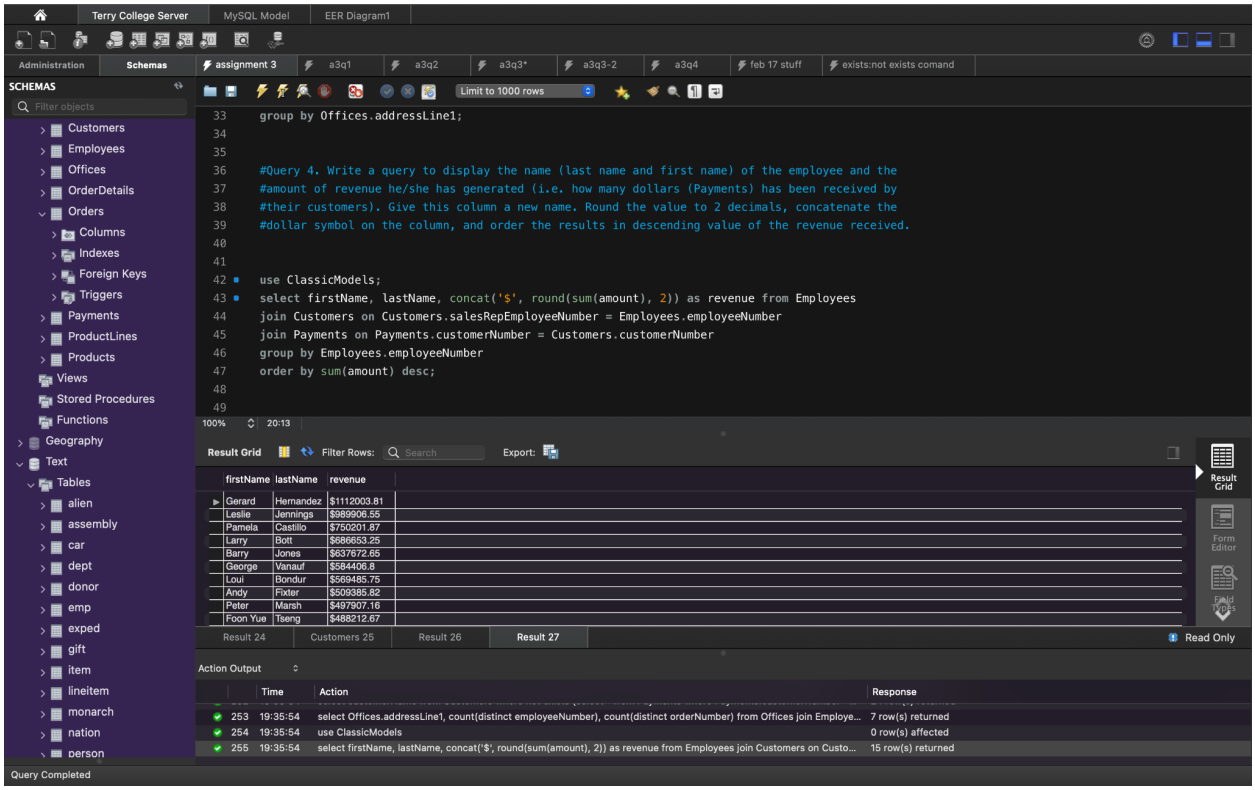
#Query 3. Write a query to list the different offices (their address), the count of the number of #employees at that office (who have helped make an order) and the number of orders that #employees at that office were sales representatives for.

```
select Offices.addressLine1, count(distinct employeeNumber), count(distinct orderNumber) from Offices
join Employees on Employees.officeCode = Offices.officeCode
join Customers on Customers.salesRepEmployeeNumber = Employees.employeeNumber
join Orders on Orders.customerNumber = Customers.customerNumber
where Customers.salesRepEmployeeNumber = Employees.employeeNumber
group by Offices.addressLine1;
```



#Query 4. Write a query to display the name (last name and first name) of the employee and the #amount of revenue he/she has generated (i.e. how many dollars (Payments) has been received by
#their customers). Give this column a new name. Round the value to 2 decimals, concatenate the
#dollar symbol on the column, and order the results in descending value of the revenue received.

```
use ClassicModels;
select firstName, lastName, concat('$', round(sum(amount), 2)) as revenue from Employees
join Customers on Customers.salesRepEmployeeNumber = Employees.employeeNumber
join Payments on Payments.customerNumber = Customers.customerNumber
group by Employees.employeeNumber
order by sum(amount) desc;
```


Terry College Server
MySQL Model
EER Diagram1
Administration
Schemas
assignment 3
a3q1
a3q2
a3q3*
a3q3-2
a3q4
feb 17 stuff
exists:not exists comand
SCHEMAS
Filter objects
Customers
Employees
Offices
OrderDetails
Orders
Columns
Indexes
Foreign Keys
Triggers
Payments
ProductLines
Products
Views
Stored Procedures
Functions
Geography
Text
Tables
alien
assembly
car
dept
donor
emp
exped
gift
item
lineitem
monarch
nation
person
Limit to 1000 rows
33 group by Offices.addressLine1;
34
35
36 #Query 4. Write a query to display the name (last name and first name) of the employee and the
37 #amount of revenue he/she has generated (i.e. how many dollars (Payments) has been received by
38 #their customers). Give this column a new name. Round the value to 2 decimals, concatenate the
39 #dollar symbol on the column, and order the results in descending value of the revenue received.
40
41
42 use ClassicModels;
43 select firstName, lastName, concat('$', round(sum(amount), 2)) as revenue from Employees
44 join Customers on Customers.salesRepEmployeeNumber = Employees.employeeNumber
45 join Payments on Payments.customerNumber = Customers.customerNumber
46 group by Employees.employeeNumber
47 order by sum(amount) desc;
48
49
100%
20:13
Result Grid
Filter Rows:
Search
Export:
firstName lastName revenue
Gerard Hernandez $1112003.81
Leslie Jennings $989906.55
Pamela Castillo $750201.87
Larry Bott $686653.25
Barry Jones $637672.65
George Vanauf $584406.8
Loui Bondur $569485.75
Andy Fixter $509385.82
Peter Marsh $497907.16
Foon Yue Tseng $488212.67
Result Grid
Form Editor
Field Types
Result 24
Customers 25
Result 26
Result 27
Read Only
Action Output
Time
Action
Response
253 19:35:54 select Offices.addressLine1, count(distinct employeeNumber), count(distinct orderNumber) from Offices join Employe... 7 row(s) returned
254 19:35:54 use ClassicModels 0 row(s) affected
255 19:35:54 select firstName, lastName, concat('$', round(sum(amount), 2)) as revenue from Employees join Customers on Custo... 15 row(s) returned
Query Completed