Hai Dang

1/28/17

CIS 3308: Lab 2 Populate and Extract Data from Your Database

## **Functionality:**

The proposed functionality for my web app is to provide customers the ability to buy spaceships from my site. Customers will be able to register, login, log off the site. Also they will be able to insert/delete records for spaceships along with transactions. Users can list spaceships for sale by inserting into the table after logging in. Users will ultimately be able to buy spaceships as well, after registration and choose the type of spaceship they’d like and where it would be built.

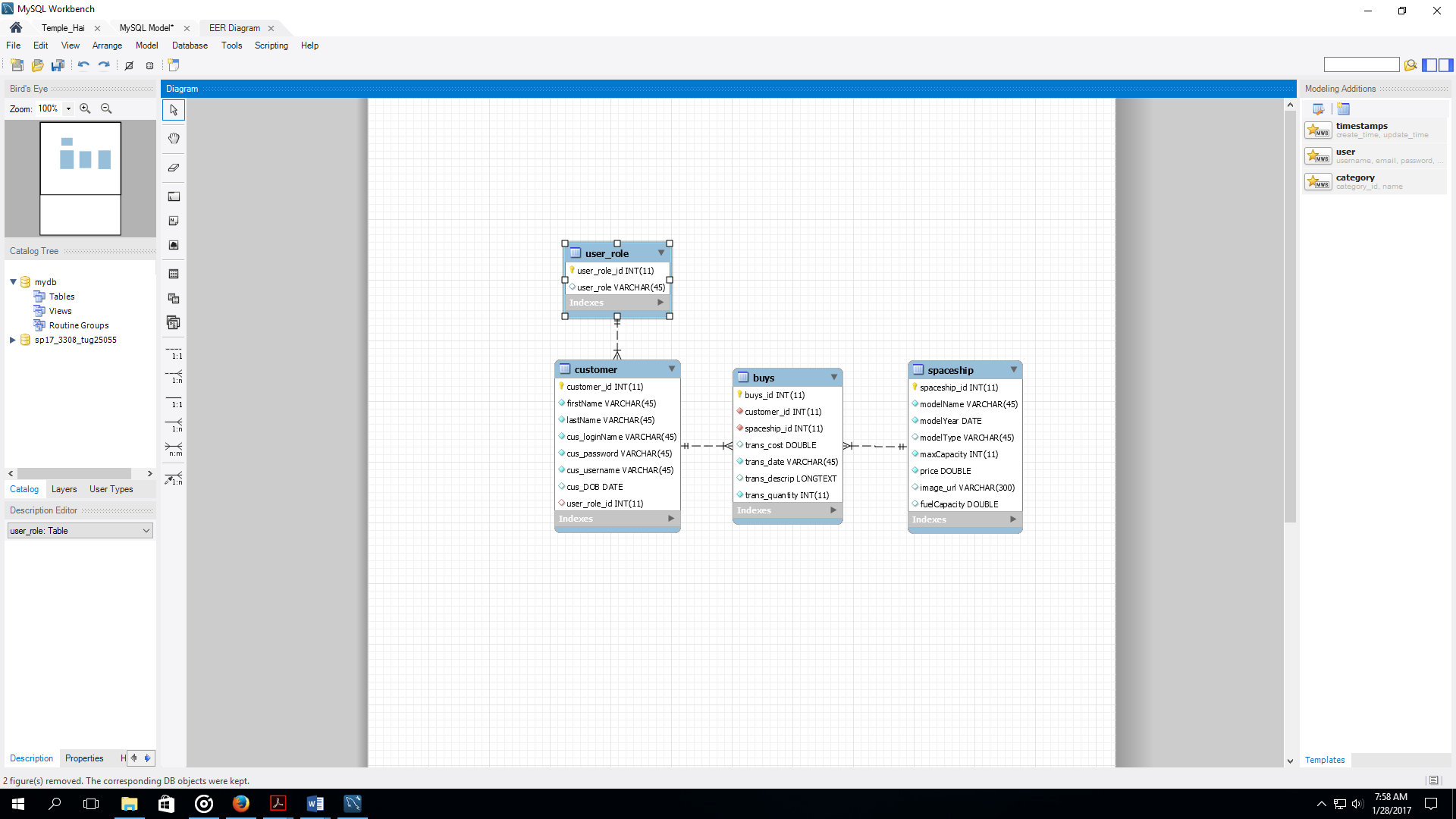
## **Marketing Material:**

The year is 2217 and humankind have spread out through the stars. Here at Named Space, we are building the future, the future you have in mind. If you’ve ever wanted to find that home away from home then why not on settle on another planet, or a comet, or a satellite, or a star? From the bustling metropolis of Mars to the newly-built outposts in Proxima Centauri b, the universe is your limit. In our various interstellar spaceships, we outperform the competition in price, speed, and comfortability. Cruise through the depths of space in our ships and explore the unknown. What undreamt voyages will you pursue? What new wonders await your discovery? The frontier is everywhere.

Notes:

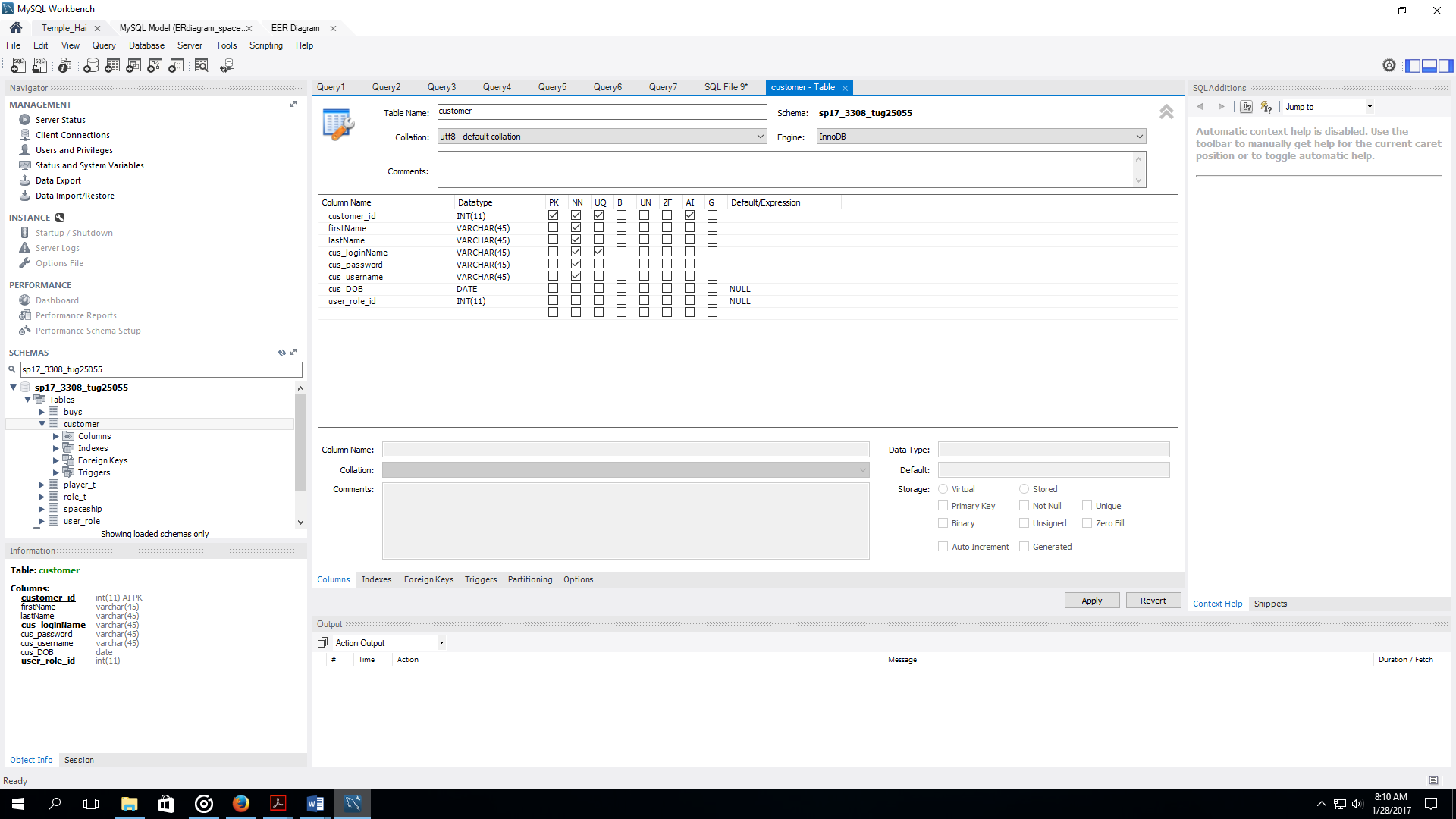
* Date range is from January 1, 1753, through December 31, 9999 in SQL so this year is feasible.
* In this scenario, all spaceships are not unique but are like products/commodities such as cars.

## **Data Model:**

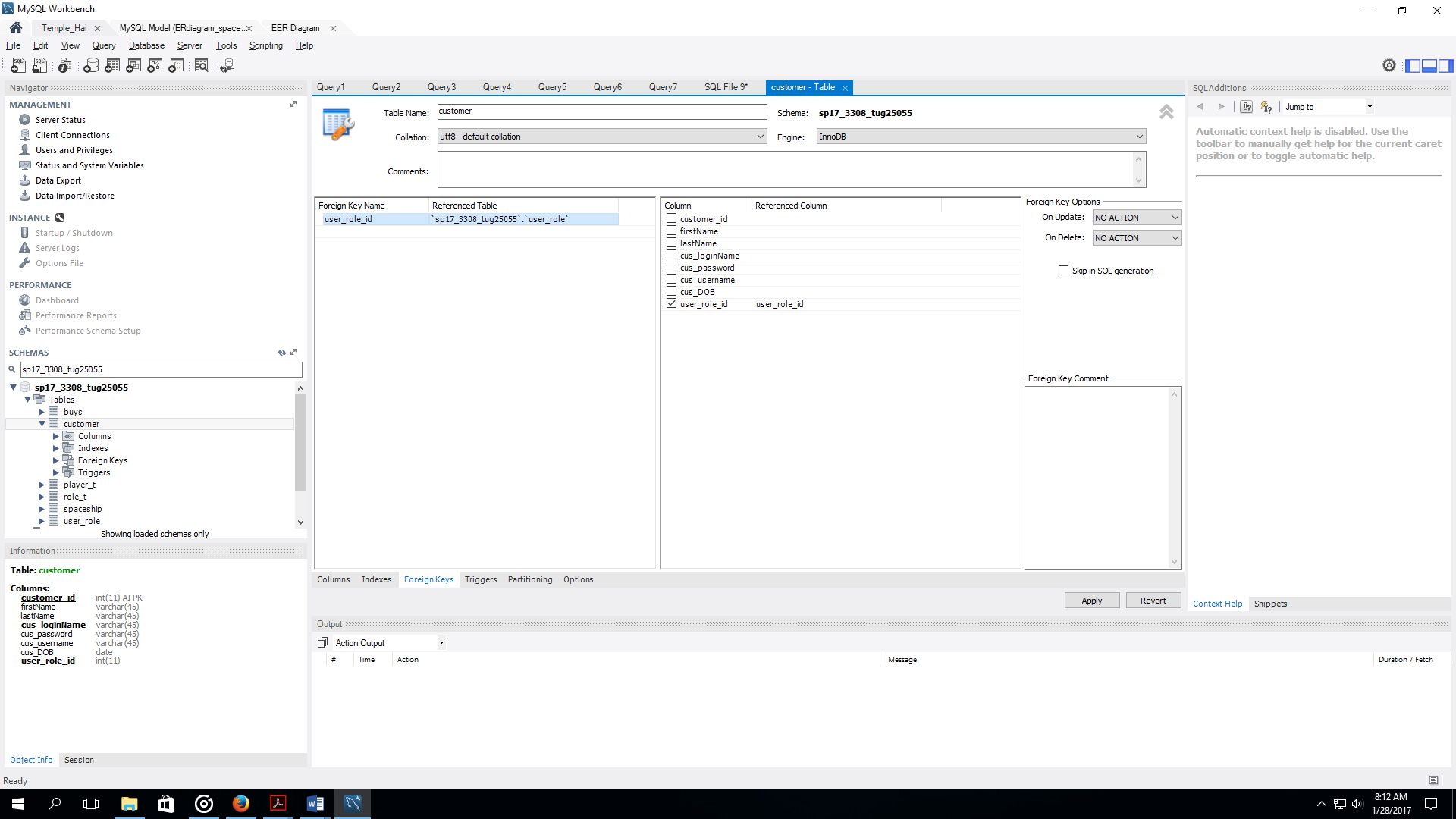


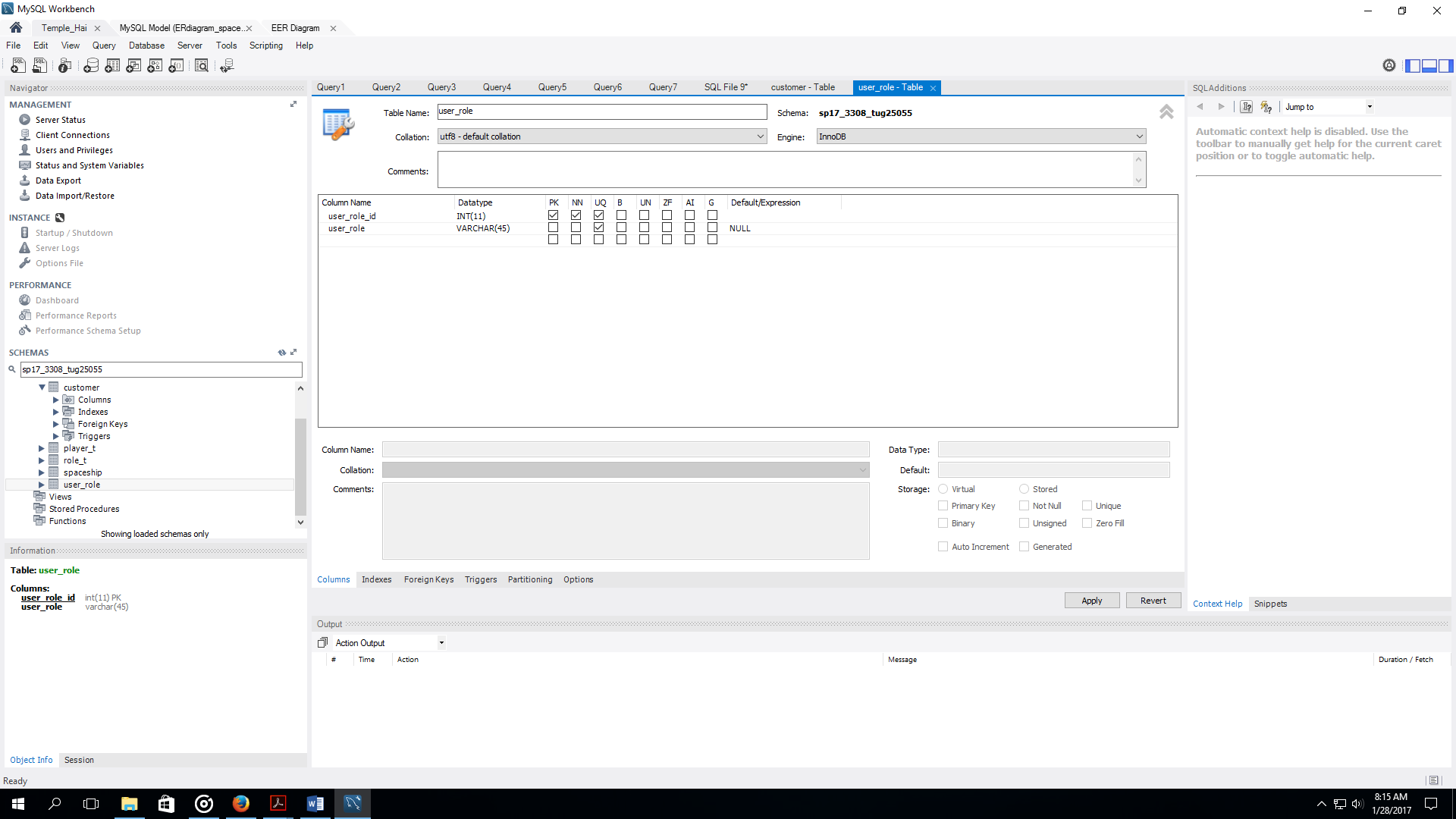
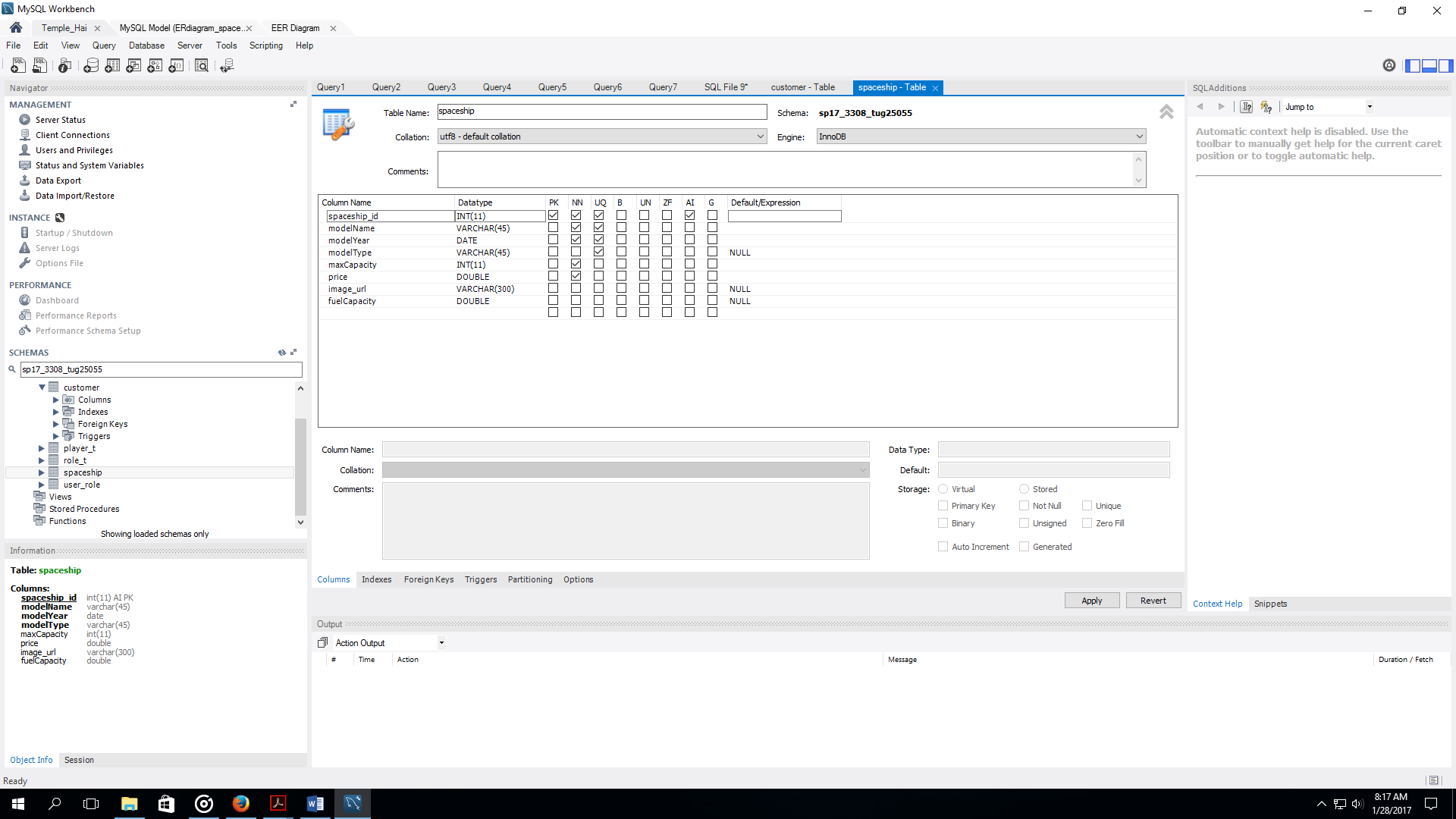
## **Table Designs:**

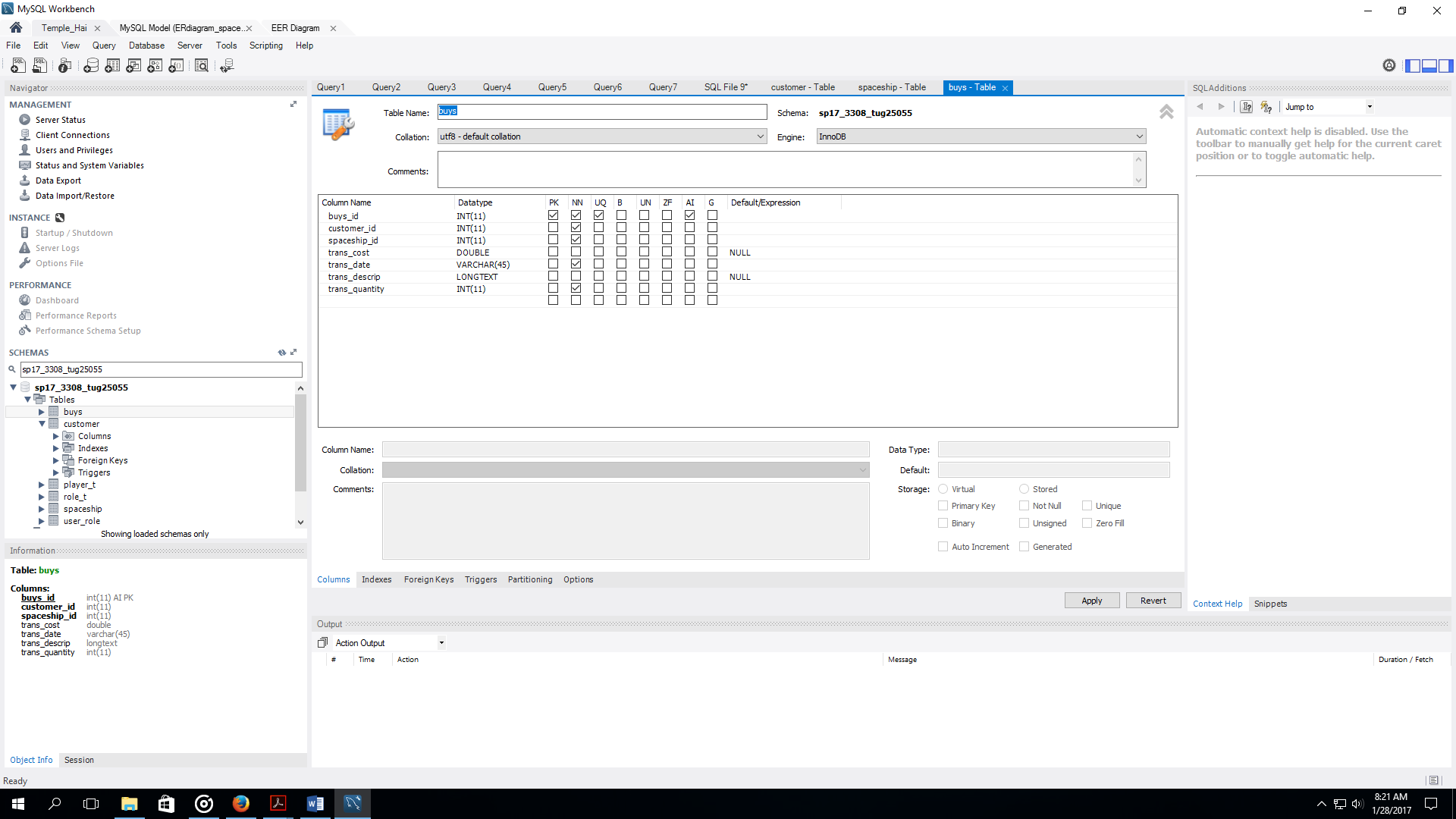
1. Customer Table



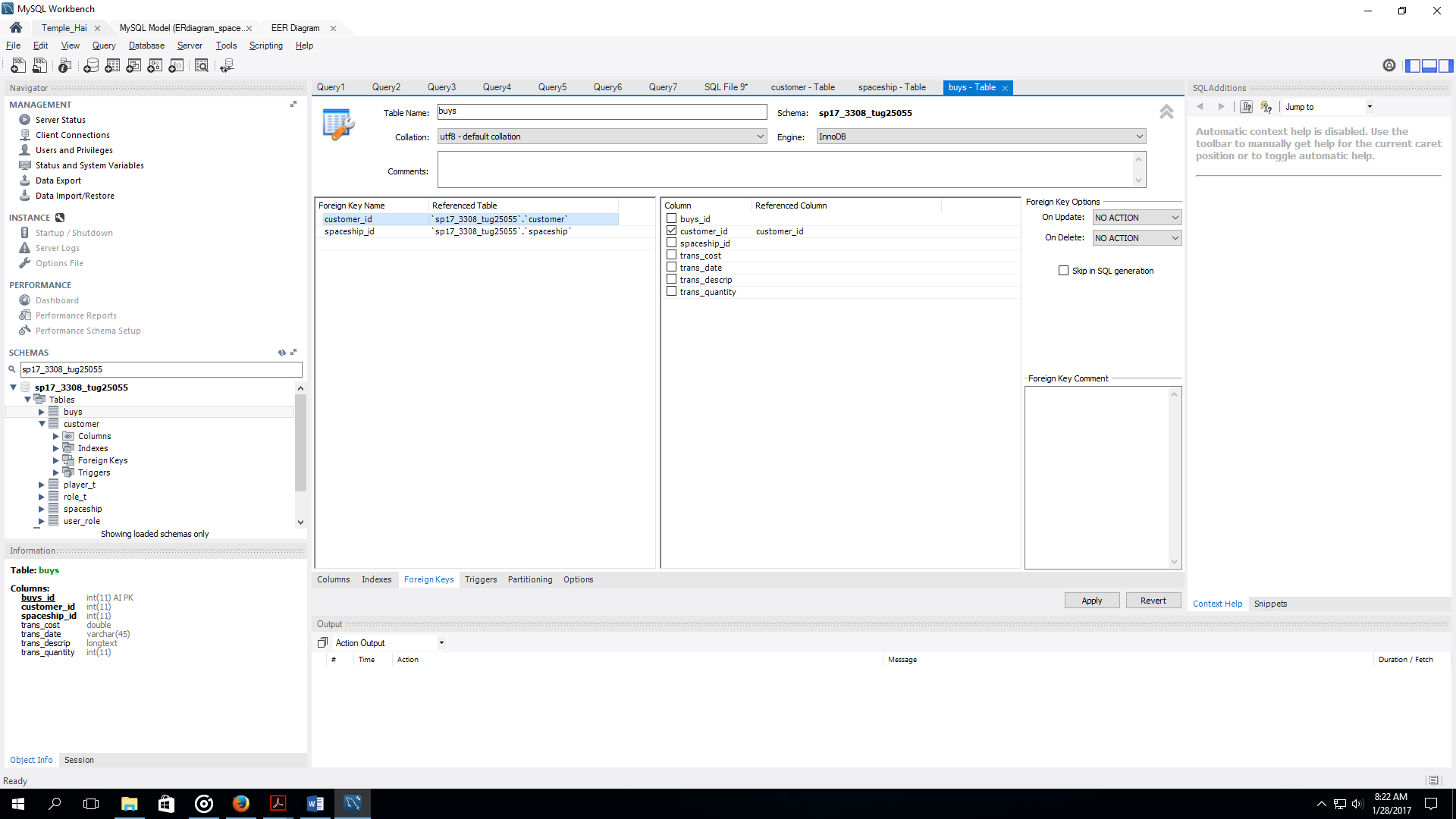
* + Foreign Keys



1. User\_Role
2. Spaceship
3. Buys



* + Foreign Keys:



## **Feedback From Classmates:**

Dev Maru

* + For Spaceship entity, add a nullable field for a non-char.
  + Password is a sql keyword. (We discussed this earlier and looked at the wrong version of sql 5.0.0, which I later rechecked)

James Ryan

* + None

## **My Feedback:**

Dev Maru

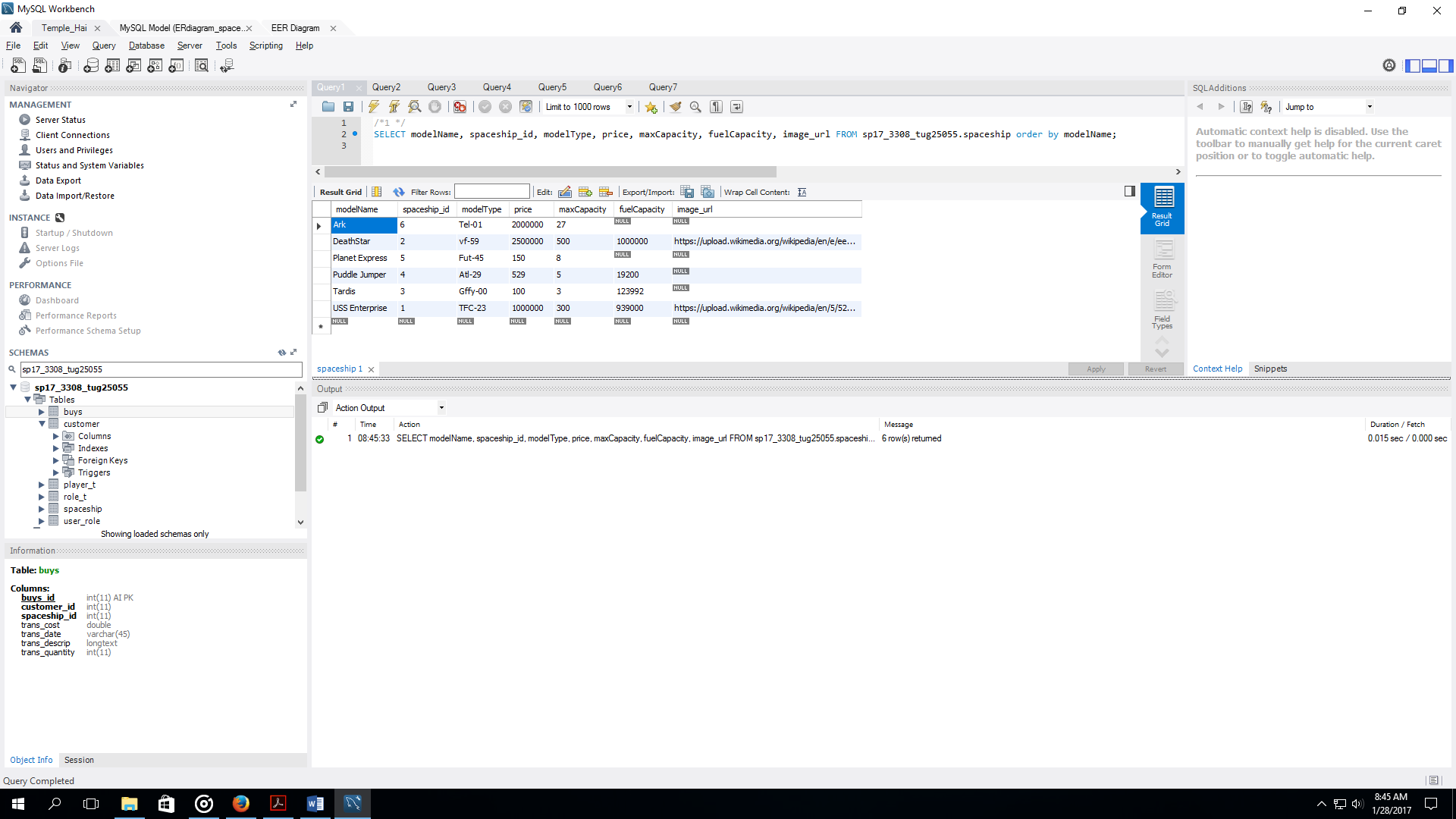
* + Change website name
  + Add a bit more in the marketing material to make it more alluring.
  + Scope of teams (which exactly)?

James Ryan

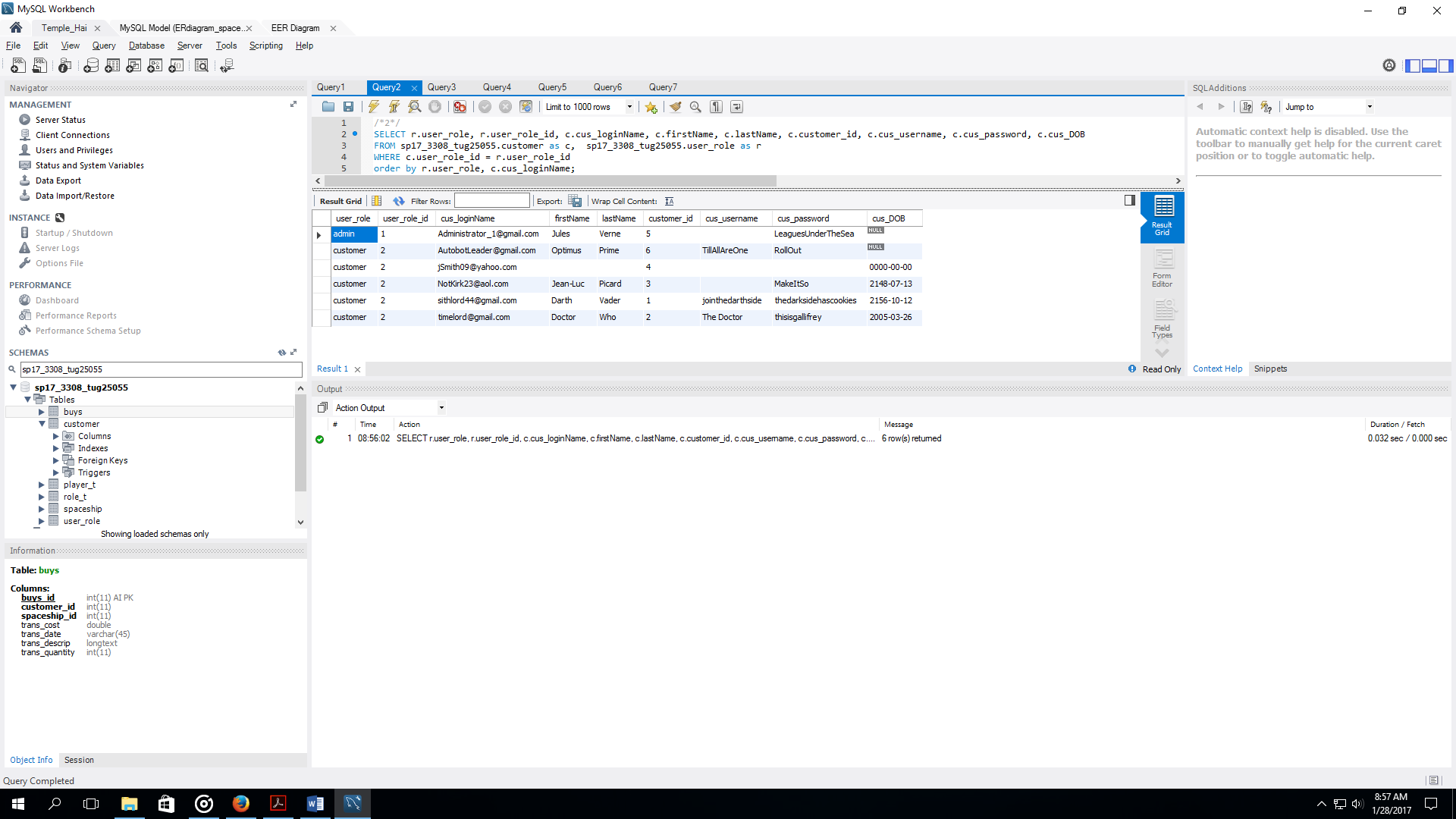
* In Venue entity, add a nullable field per project instructions
* For lab proposal, the marketing material is overall good, but could be a bit better to address maybe how it differs from other websites.

## **SELECT Statements:**

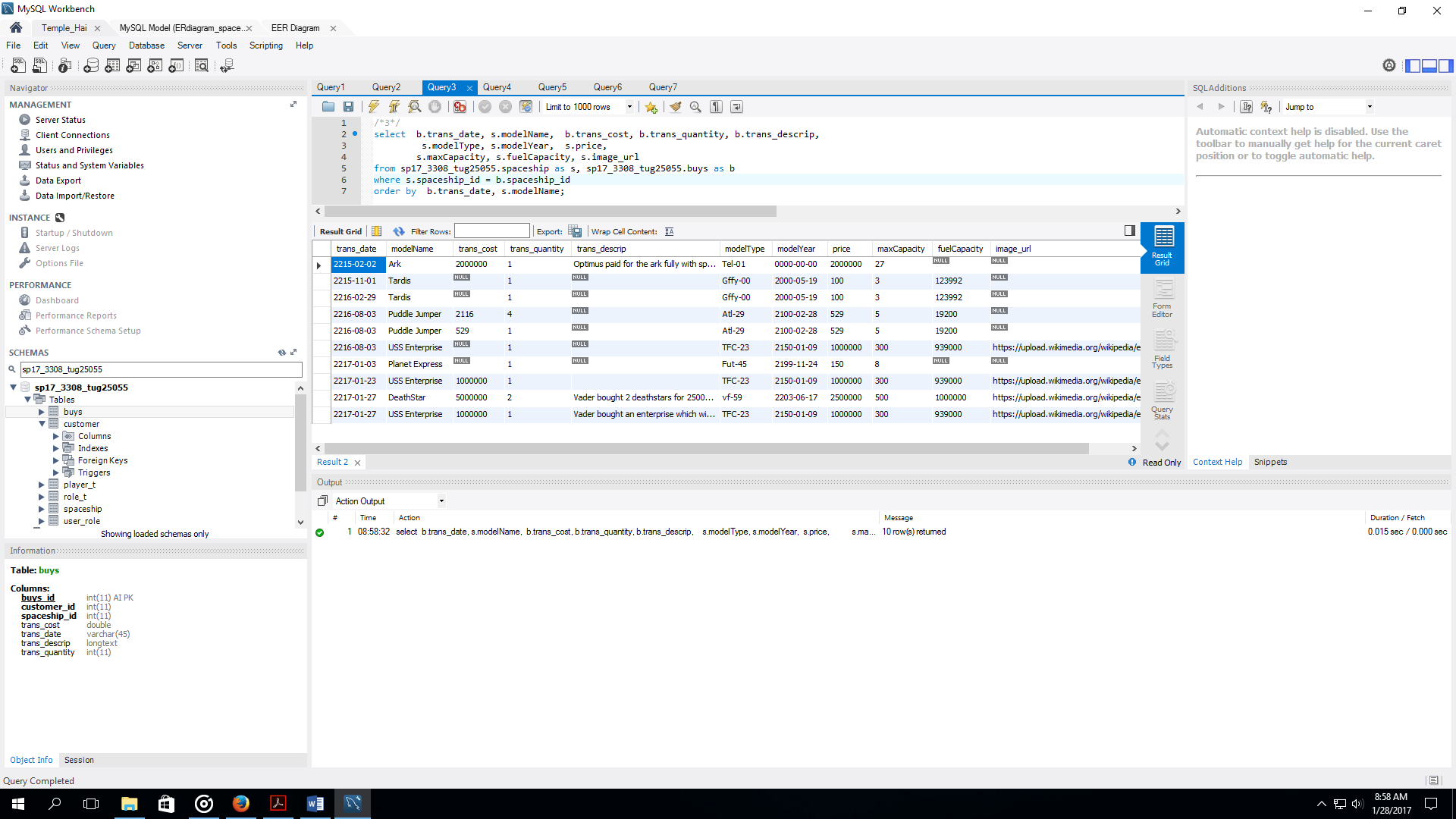
1. List all of the columns of your “other” table, ordering the columns in a way that you think users would like to view the data (don't use "select \*"). Sort the data by the first column. Include all records of the "other" table.



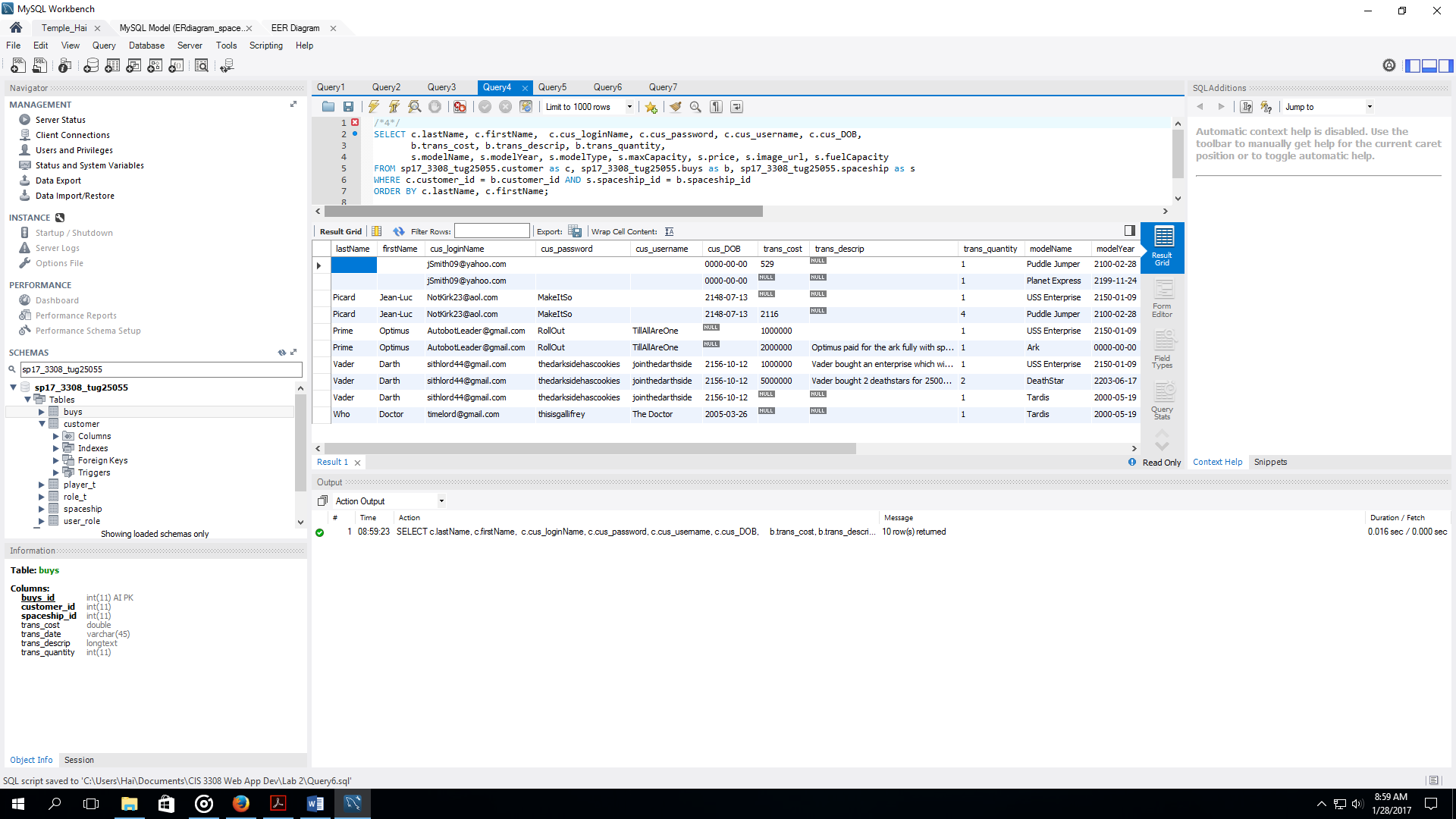
1. Show all the records from your user table joined with your role table. Show the role name first, then the role id, then the email address, followed by all the rest of the columns of your user table (ordered in a way you think users would like to view the data – show the role id once, not twice.) Order the result set by role name, then email address (as a secondary sort). Include all records.



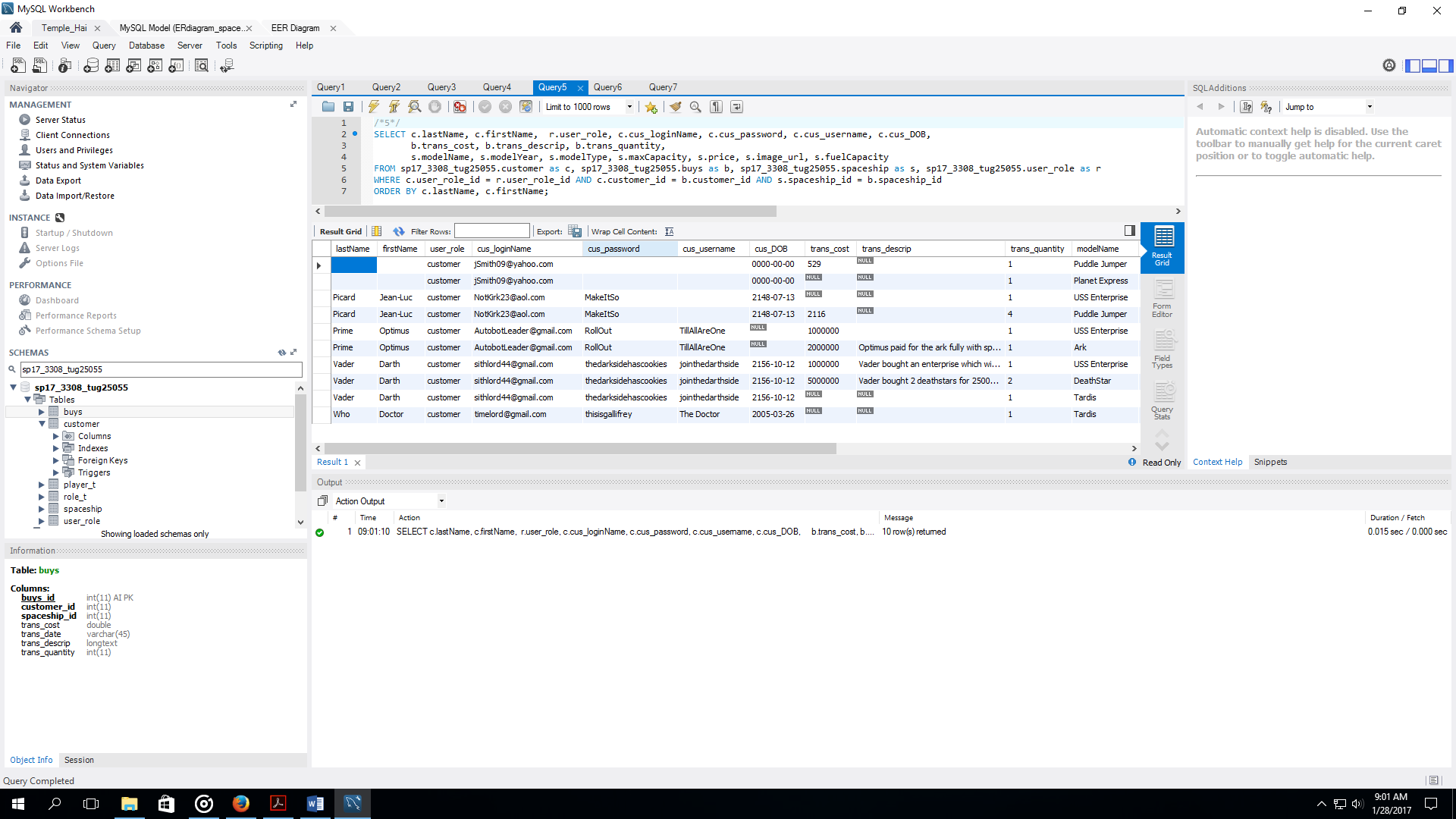
1. Join your associative table with your “other” table. Include all the columns except PK and FK (id) columns. Order the columns in a way you think users would like to see them (not "select \*"). Order the result set by the first two or three columns.



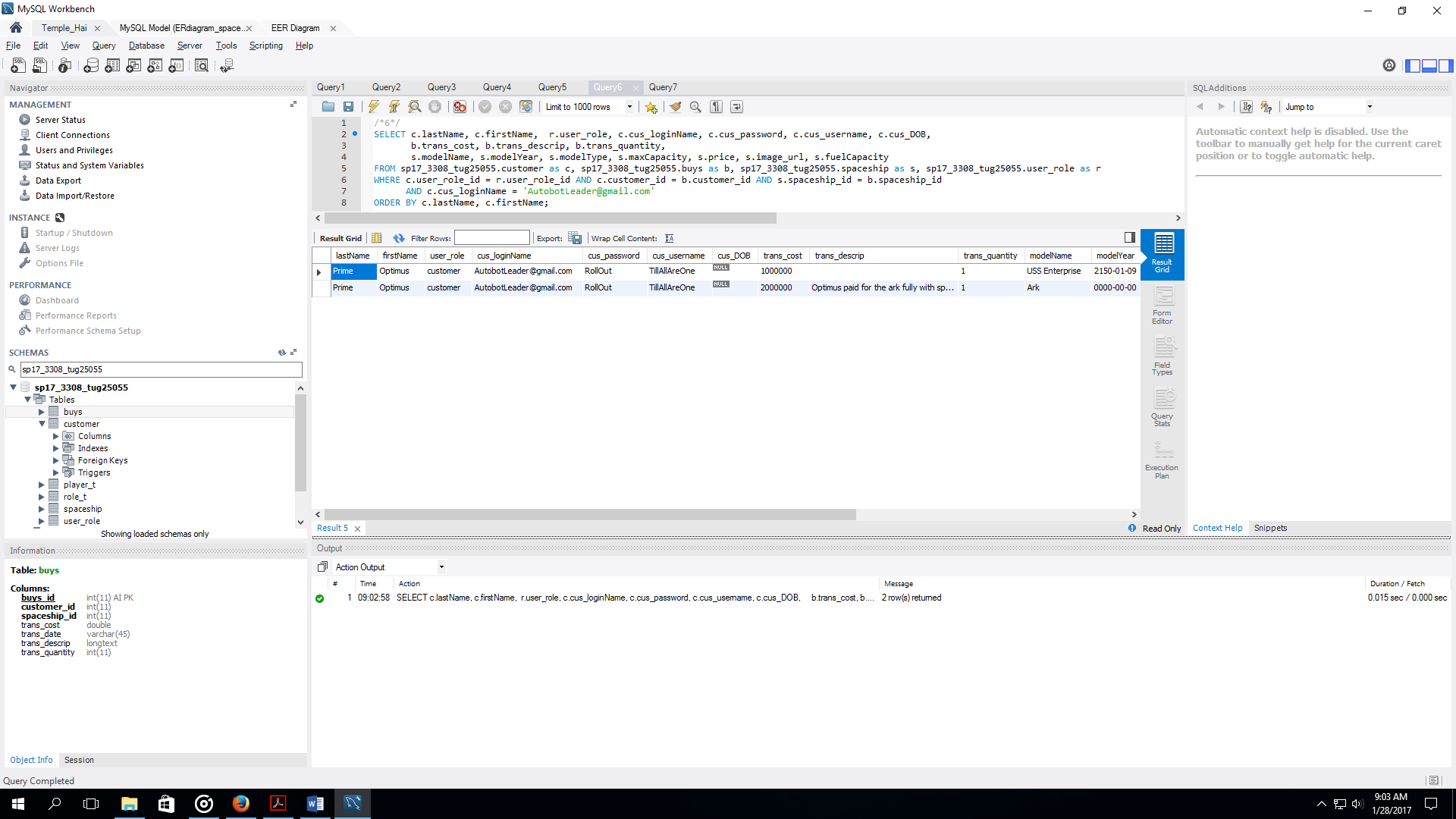
1. Join your associative table with your “other” table and with your user table. Include all the columns except PK and FK (id) columns. Order the columns in a way you think users would like to see them. Order the result set by the first two or three columns.



1. Join your associative table with your “other” table and with your user table and with your role table. Include all the columns except PK and FK (id) columns. Order the columns in a way you think users would like to see them. Order the result set by the first two or three columns.



1. Modify your SELECT STATEMENT from item “e”, adding a condition to your WHERE so that you see only the data from one user (selected by user.user\_id). Select a user such that your result set has at least 2 records in it.



1. Modify your SELECT STATEMENT from item “e”, so that it selects one user id, two or more "other" records (using LIKE keyword and % wildcard match), and some condition testing a field from the associative table. Come up with conditions such that your result set has at least 2 records in it.

