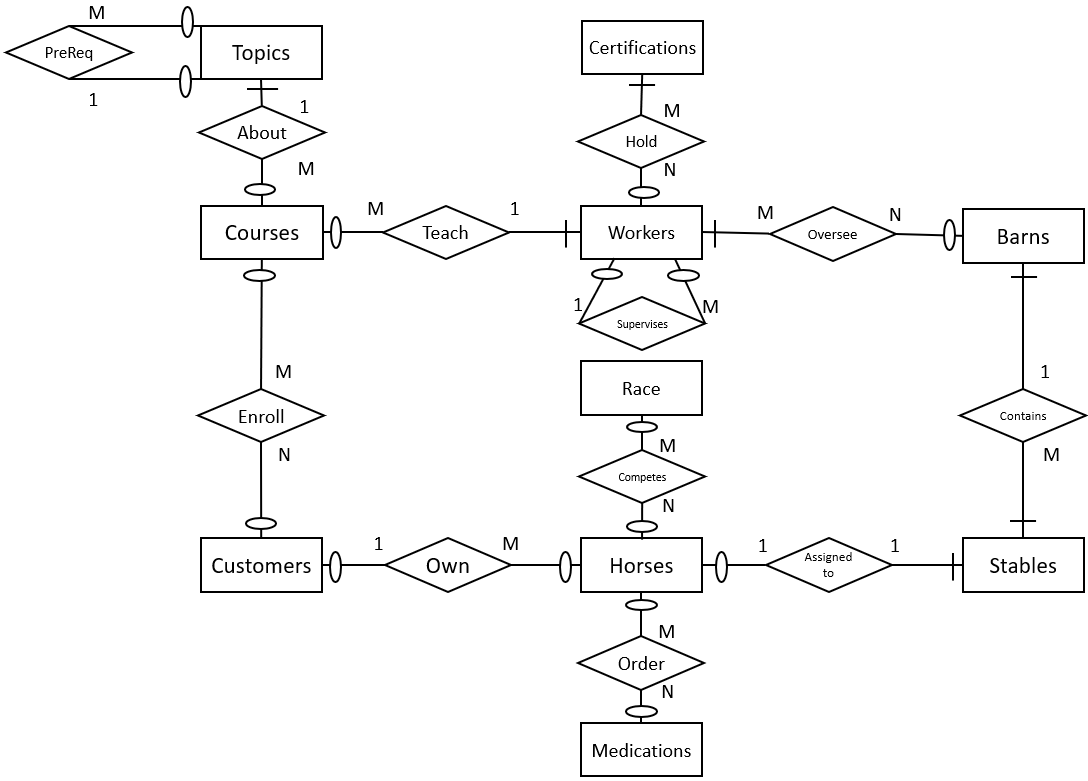
**Spring 2024 SQL Project**

In this project you will be creating a database to manage the operations of MG Horse Habitat (MGHH), then answering several question using your SQL skills. While the business case presented below does not describe the operations in the depth some of our other cases have, the high level concept is:

Customers of MGHH may own horses, and may choose to enroll in courses that cover a variety of topics. Horses are assigned to stables (which are part of a barn), may be ordered medications, and may compete in races. Workers at MGHH may be assigned to oversee barns and may also teach courses. There are a variety of certifications that workers might hold.

Details about the attributes can be inferred by studying the provided DDL and data.



The number of instances of each entity are as follows:

|  |  |  |
| --- | --- | --- |
| * **Horses: 163** * **Customers: 55** * **Medications: 17** * **Stables: 205** | * **Barns: 6** * **Workers: 20** * **Certifications: 8** | * **Topics: 9** * **Courses: 17** * **Races: 15** |

The M:N relationships are decomposed by these gerunds:

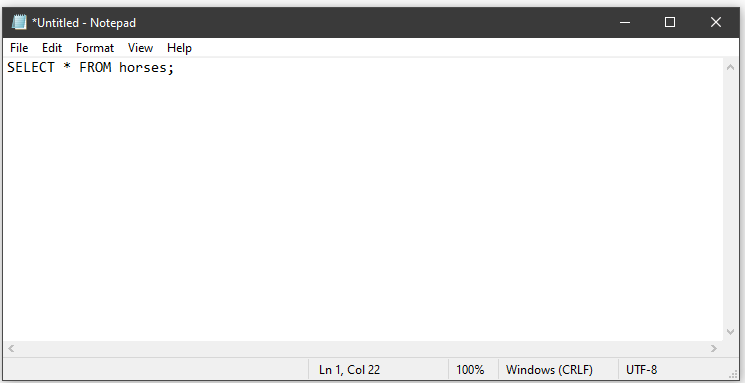
|  |  |  |
| --- | --- | --- |
| * Enroll: CourseCustomer * Hold: WorkerCert | * Oversee: WorkerBarn * Competes: HorseRace | * Order: Orders |

**Step 1: Draft SQL query that results in the data needed**

I **STRONGLY** suggest writing all your queries in a PLAIN TEXT editor like Notepad (Windows) or TextEdit (Mac). Do not use Microsoft word, as it will mess up your quotation marks (and other things) and cause error.

**SAVE A COPY OF YOUR SQL QUERIES IN THIS PLAIN TEXT DOCUMENT!** This will make it easier to edit and work on your queries over time. Also in case there is a problem with the submission system you will have a backup of your work.

Think about the question being asked and write the query in your text editor. Refer to the ERD, DDL and data in the provided data files.



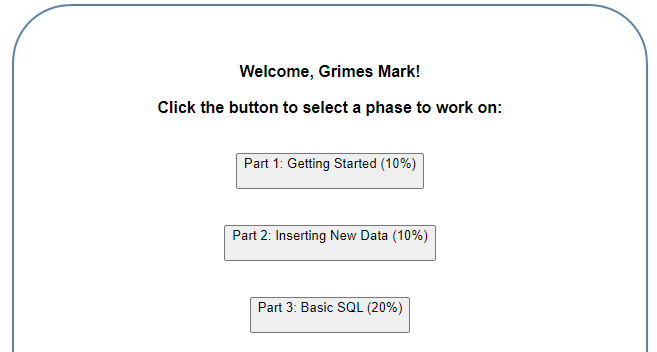
**Step 2: Test/debug in Oracle/DBeaver**

Copy/paste your query from notepad into your SQL client (probably DBeaver) and try running it. Did you get the results you expected? Does it match the screenshot I provided?

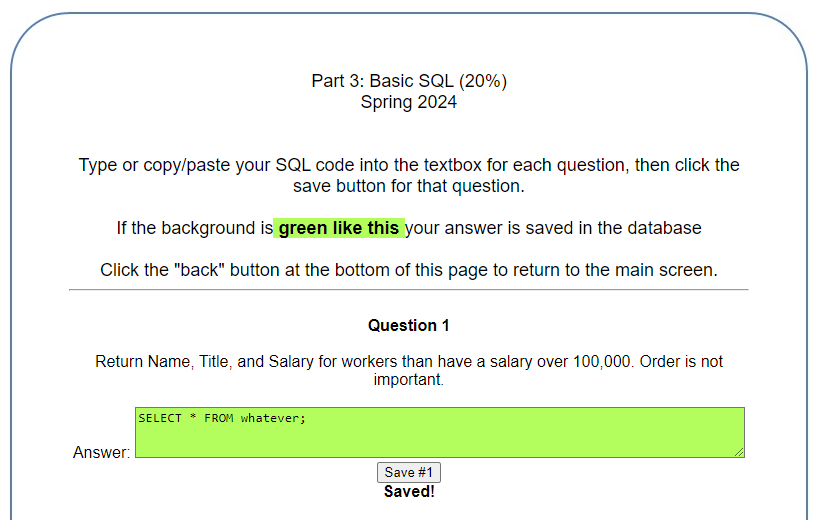
If you make changes to the query in your SQL client software, remember to copy/paste the modified query back into your notepad file.

**Step 3: Copy and paste your answers into the SQL Project submission system.**

* URL: <https://research.gmgrimes.com/SQLProject>
* Your username is the same username you use to log in to Oracle
* Your password is initially the same as what I sent you for Oracle – the first time you log in you will be required to change your password (you can change it to the same thing I sent you if you want)



**Make sure you click Save for EVERY query!  
The background will turn green to indicate you answer has been saved**



**If you are idle for too long (10 minutes or so) the server may “time out” and not save your answers. I suggest closing your browser and logging back in to make sure your answers were saved! If the system times out, your answers might not be saved…but that’s OK because you still have them in the PLAIN TEXT file I mentioned before, right??**

**Note: If you count up the percentages for each section you will notice it adds up to 122% including the bonus. Since this assignment is worth 10% of you grade, you can add an EXTRA 2.2% to your FINAL grade!**

**A few general notes:**

* **For all queries, unless the question specifically asks for the rows to be returned in a specific order, do not worry about the order.**
* **When using ORDER BY, provide the NAME of the attribute you are ordering by – do not do something like “ORDER BY 2”, which some internet examples would lead you to do. This is not a robust way of writing SQL.**

# Part 1 – Getting Started (5%)

Run the code provided in the files MGHH\_DDL.txt and MGHH\_INSERTS.txt.

Note that the DROP TABLE statements at the start of the DDL file should only be executed if you want to clear everything out and start with a blank database.

**Assessment:**

* I will be checking each student’s database that all 15 tables have been created and populated with data.
* In the SQL Submission system, just answer “Yes” to the question for this part

# Part 2 – INSERTING some data of your own (5%)

1. INSERT a new Customer with your first name for the value of “fname”, your last name for the value of “lname”, and your PSID for the value of “Username”. You can put any reasonable value you want for the value of phone number.
2. Update the Horses table so that the new customer you created in the previous step (you) is the owner for the horse name “Flash”.
3. Update the CourseCustomer table so that the new customer you created (you) is taking the following courses:
   1. Introduction to Racing in Summer of 2024
   2. Introduction to Polo in Fall of 2024

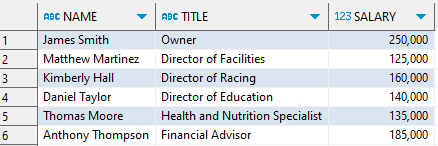
(*Note: you will need to find the course names in the “Topics” table, then find the CourseID for the particular term and year in the “Courses” table*)

**Assessments:**

* The name and Username of the customer you create matches your name and PSID
* The horse “Flash” is owned by the customer that matches your name
* The customer you created is enrolled in the two specified courses
* In the SQL submission system, include the INSERT and UPDATE queries you used here

# Part 3 – Basic Single Table SQL (10%)

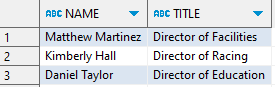
**1. Return Name, Title, and Salary for workers than have a salary over 100,000. Order is not important.**

****

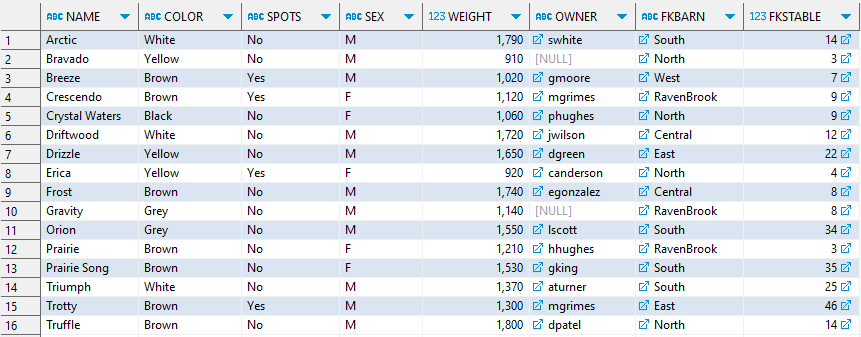
**2. Return all attributes for horses that have a weight greater than or equal to 1800 and are Female.**

****

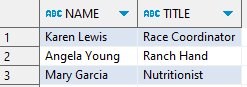
**3. Return the name and title of workers whose title starts with the word “Director”**



**4. Return all attributes for horses that have “r” as the second letter of their name**

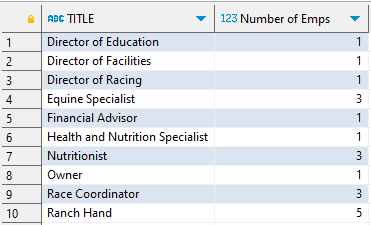
****

**5. Return the name and title for all workers that are no longer working for MGHH (i.e., they have a value for the attribute TerminationDate).**

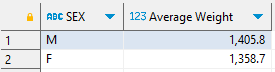


# Part 4 - SQL with Aggregate Functions (30%)

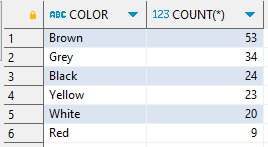
**1. Return the number of workers for each job title ordered alphabetically by job title.**



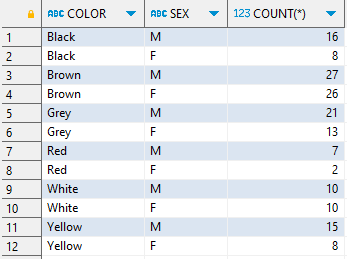
**2. Find the average weight of horses for each value of Sex. Round the result to one decimal place.**



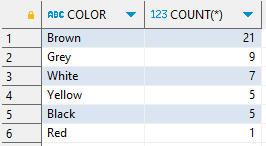
**3. Find the number of horses of each color. Return the results so that the color with the most horses is first, and the color with the fewest horses is last.**



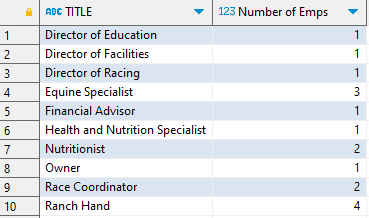
**4. Return the number of horses for each combination of color and sex. Return the results alphabetically by color, then sex within color.**



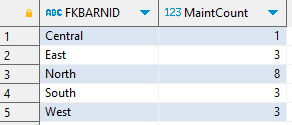
**5. Find the number of horses of each color for horses that are female and do not have spots. Return the results so that the color with the most horses is first, and the color with the fewest horses is last.**



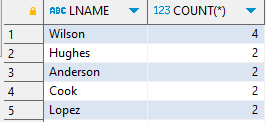
**6. Return the number of workers for each job title ordered alphabetically by job title. Do not include workers than have been fired (i.e., those that have a value for the attribute “TerminationDate”)**



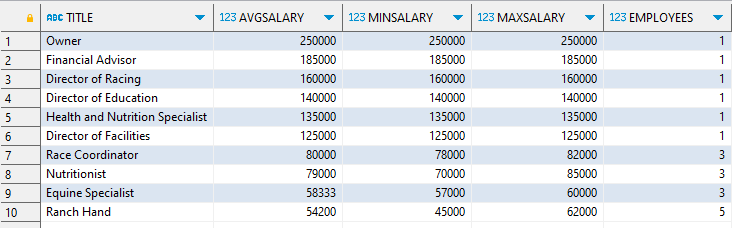
**7. Return the number of stables for each barn that currently have a status of “Maintenance”**



**8. Return last names that are shared by two or more customers. Order the results so the most popular last name is first.**



**9. List each job title and the average salary, rounded to zero decimal places, minimum salary, maximum salary, and number of employees for each job title. Order the results with the highest salary at the top.**

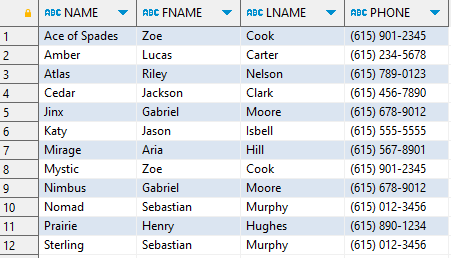


**10. Return the total number of horses in the database, as well as the number of horses that are owned by someone. You may only use the data in the “horses” table (do not do a join to the customers table).**

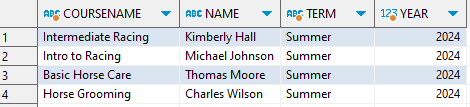
****

# Part 5 – Binary Queries (21%)

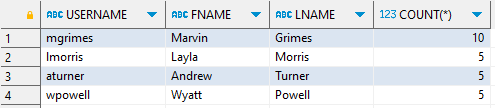
**1. Return the name of the horse and first name, last name, and phone number of the owner for all Female horses whose owner’s phone number is in the (615) area code. Order by the horse’s name.**

****

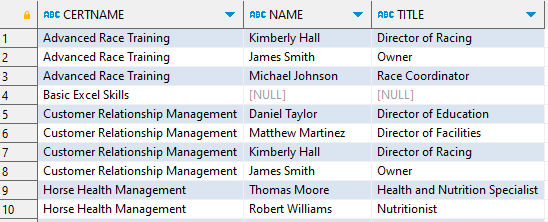
**2. Return the course name and name of the instructor for all courses offered in Summer of 2024**



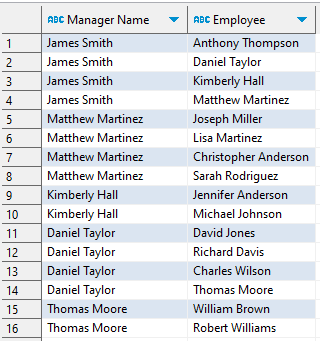
**3. Return the number of horses owned by each customer than owns more than 4 horses. Order by county of horses in descending order.**



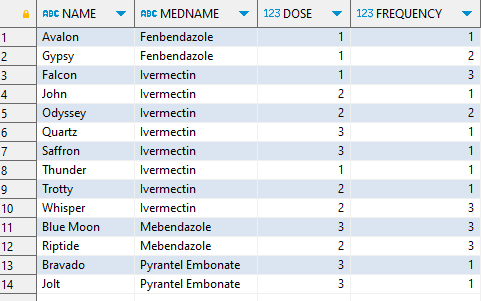
**4. For each certificate, list all the workers that hold that certificate. List all certificates even if no workers have them (Note: there are 54 rows returned, only the first 10 are shown here)**



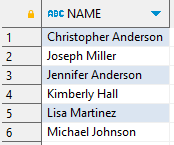
**5. Using a recursive join, generate this org chart. Only include employees that are currently working for MGHH.**



**6. Return horse name, medication name, dose, and frequency for all horses that have been ordered to take a “Dewormer” medication. Order alphabetically by medication name, then horse name.**

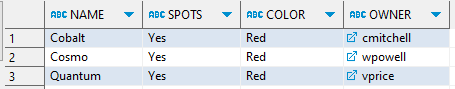
****

**7. Return the name of all workers that teach a class and are also assigned to oversee one or more barns.**

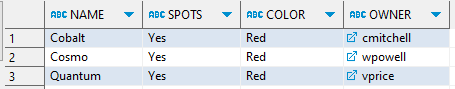
****

# Part 6 – Using Multiple Approaches (24%)

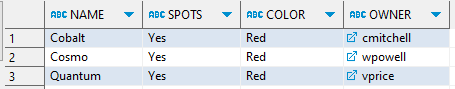
**1. Return the Name, spots, color, and username of the owner for horses that are the color Red and have spots using an “AND” operator in your WHERE statement.**



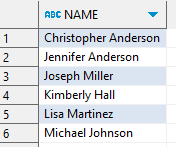
**2. Return the Name, spots, color, and username of the owner for horses that are the color Red and have spots using a set operation.**



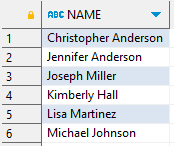
**3. Return the Name, spots, color, and username of the owner for horses that are the color Red and have spots using a subquery.**



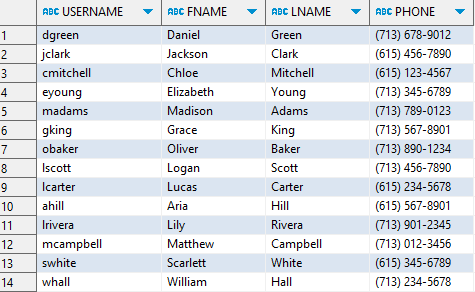
**4. Find all workers who are teaching a class but do not have the teaching certification. Use a subquery.**

****

**5. Find all workers who are teaching a class but do not have the teaching certification. Use the MINUS set operation.**

****

**6. Return all customers than have not enrolled in a course using a subquery**

****

**7. Return all customers than have not enrolled in a course using an outer join**

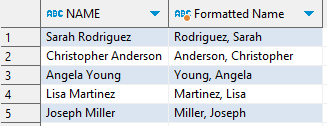
**<<Same result as question above>>**

**8. Return the username of all customers than have not enrolled in a course using a set operation**

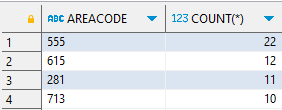
****

# Part 7 – String Manipulation (15%)

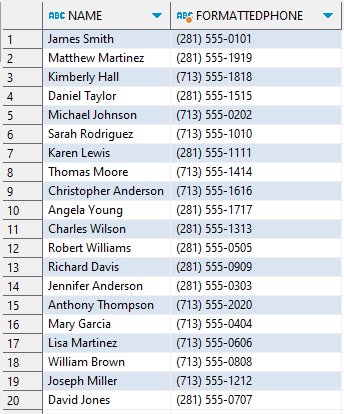
**1. Return each worker’s name as it is stored in the database, and the name formatted as “Lastname, Firstname” for workers that have a title “Ranch Hand”**



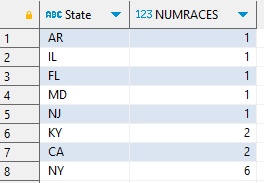
**2. Return the number of customers for each area code (note: the area code is the first part of the phone number found in parentheses). Return the results with the most popular area code first.**



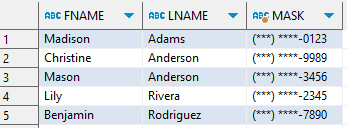
**3. Return the names of all worker along with their phone number formatted as (xxx) xxx-xxxx.,**



**4. Return the number of races held in each state**

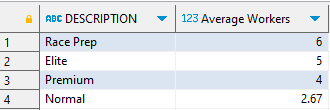


**5. Return the first and last name and a “masked” version of their phone number than shows only the last four digits, like this: (\*\*\*) \*\*\*-1234 for customers with a last name that starts with A or R.**

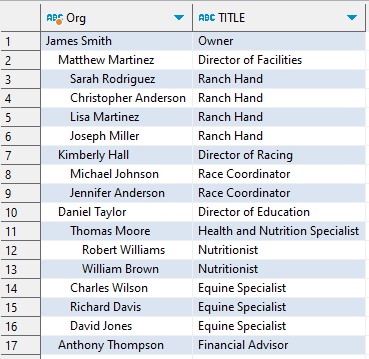
****

# Part 8 – Bonus (12% extra)

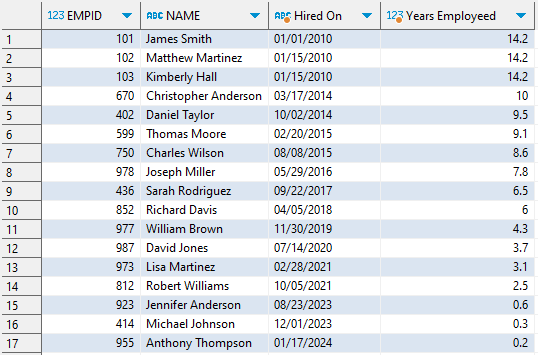
**1. Find the average number of workers assigned to each type of barn. Order by the average number of workers in descending order. Round to two decimal places.**



**2. Using a hierarchical query, generate this org chart (complete with indentions) using the data in the workers table. Only include employees that are currently working for MGHH.**



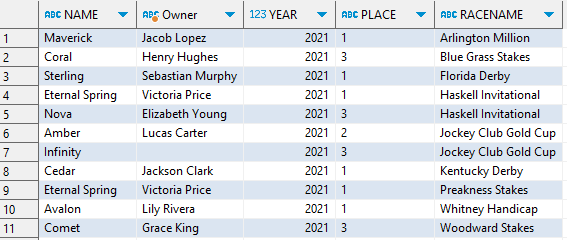
**3. Return each employee’s empid, name, hire date, and number of years employed for workers that are still working for MGHH. Be sure to format the hired date appropriately. Note that your results may vary SLIGHTLY depending on when you run the query.**

****

**4.** **Generate this report of courses being offered, the instructor and their job title, the current enrollment, and the calculated percentage full.**

****

**5. Return the name of the horse, the owner’s first and last name (concatenated, with an attribute name of “Owner”), the year, place, and racename for all horses that placed 1, 2, or 3 in any race in 2001. Order alphabetically by racename, and numerically by place within racename. Be sure to include horses even if they do not have an owner.**



**6. Return the total number of top-10 finishes each customer has had across all the horses they own**

