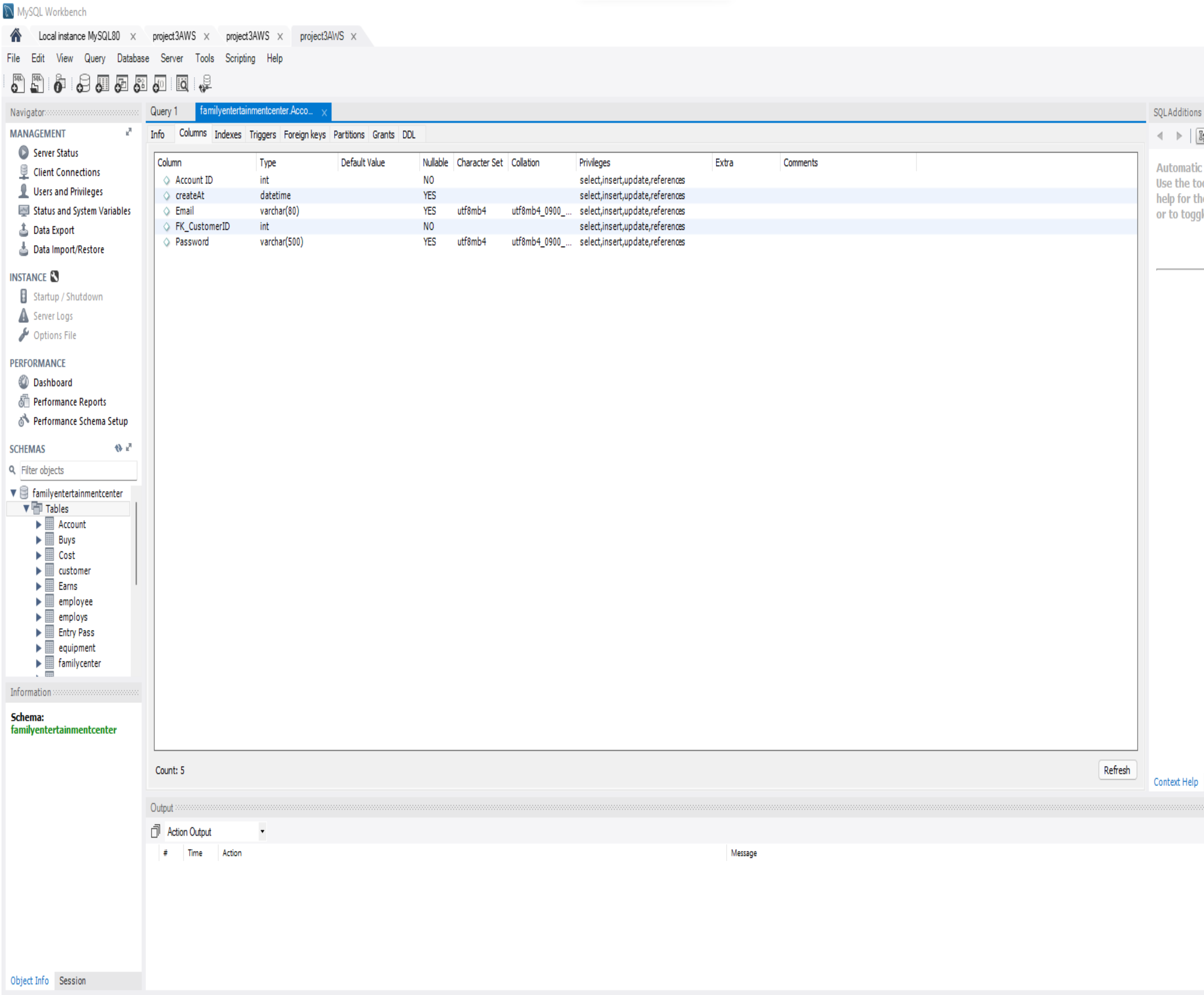
Software Discovery

For the creation of the database this time around we used MySQL community , downloaded from https://dev.mysql.com/downloads/installer/ . The date of download was March 26, 2022. The version of MySQL used was version 8.0.28.



Column	Type	Default Value	Nullable	Character Set	Collation	Privileges	Extra	Comments
Account ID	int		NO			select,insert,update,references		
createAt	datetime		YES			select,insert,update,references		
Email	varchar(80)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references		
FK_CustomerID	int		NO			select,insert,update,references		
Password	varchar(500)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references		

Count: 5

Schema:
familyentertainmentcenter

#	Time	Action	Message

Initially the software of choice some of the team were interested in moving forward with was pgAdmin 4 , as we believed the software would be very user friendly in the creation of the tables and insertion of data. PgAdmin 4 interface is also very easy to use for those beginning SQL as well and since some members of the team were new to the language it caught our interests. The icing on the cake for PgAdmin4 is that the interface has a very colorful theme and an overall elementary feel to it. One issue we did come across though was how we could work in PgAdmin as a group simultaneously or from our remote setting. Some of the members that had SQL experience in the past in a work environment were unaware how to make PgAdmin collaborative, but did know how to make an online server in MySQL. Although PgAdmin was very easy to use, we ultimately decided to make the switch to MySQL in the end, as a few members had trouble running pgAdmin 4 in a different OS (Mac). MySQL, similarly to PgAdmin, does make it easy to create the tables and insert data, but the sacrifice is the interesting interface. Although it did take some time to learn about shortcuts in creation of tables, primary/foreign keys, and connecting those keys to other tables. This shortcut that both softwares has to offer skips typing in the table manually, overall saving the cohort a lot of time. With MySQL's popularity, if we wanted to learn more about the software or had an issue, it is very easy to find a solution online as MySQL is one of the most widely used IDEs for SQL. MySQL did not seem difficult to the members of our cohort using SQL for the first time, as we did have many examples of MySQL in class. Additionally, using Amazon Web Service with MySQL for the whole group to use was very easy to do since one of the group members prior internship experience involved MySQL and AWS. For some of the cohort it was the first time using AWS, so the experienced member walked us through how to use MySQL with AWS. We could have gone with Google Cloud for the server as well, but none of the Cohort members had a history with Google cloud as they did with AWS.

Using CREATE is primarily done when the user wants to make a table of information. Create not only makes the table, but will also specify the columns one may want to insert information. If the user wanted to create a table for pets, containing an animals name, owner, species, sex, birthdate, and deathdate, it would would be structured as so:

```
CREATE TABLE pet (name VARCHAR(20), owner VARCHAR(20),
       species VARCHAR(20), sex CHAR(1), birth DATE, death DATE);
```

When this is executed, the table will have been made and now information such as the name, owner,species, sex, birthday and death day can be inserted. Here is a help page with more information regarding CREATE. https://dev.mysql.com/doc/refman/8.0/en/creating-tables.html

SELECT is used in order to receive information from one or multiple tables. Select is a form of explicit partition selection and uses the partition clause. Typically select is paired with the WHERE clause, which is used to specify the exact row one may want to extract data from. When paring Select with FROM we can specify the table we would want the information extracted from. Here is an example of SELECT in MySQL:

```
SELECT CONCAT(last_name,', ',first_name) full_name
  FROM mytable ORDER BY full_name;
```

When this is executed, last name and first name will be extracted by SELECT from mytable. Order by is use to refer columns selected for output. Here is a help page with more information on SELECT: https://dev.mysql.com/doc/refman/8.0/en/select.html

The INSERT statement is used to insert new rows into an existing table. With INSERT you may also specify the table name and can also begin providing the actual data you would like to insert as well. You can also insert multiple rows at once rather than one by one. Here is an example of an INSERT statement.

```
INSERT INTO tbl_name (a,b,c)
   VALUES(1,2,3), (4,5,6), (7,8,9);
```

The example above inserts the values (1,2,3) into tbl_name a, (4,5,6) in tbl_name b, and (7,8,9) in tbl_name c. Here is a help page with more information on INSERT.

https://dev.mysql.com/doc/refman/8.0/en/insert.html

The DELETE statement removes rows from a table. Delete, similar to other examples we have seen, can also be paired with FROM and the user can specify the exact row they may want to delete. Here is an example of the DELETE statement paired with an ORDER BY clause:

```
DELETE FROM somelog WHERE user = 'jcole'
ORDER BY timestamp_column LIMIT 1;
```

In this instance, ORDER BY is paired to delete rows in the order specified in the clause. Here is a help page with more information on DELETE:
https://dev.mysql.com/doc/refman/8.0/en/delete.html

The UPDATE statement is used to modify records that already exist in a table. Update can also update specific columns and can be paired with the where clause that will allow there to be some form of condition. Here is an example of the UPDATE statement.

```
UPDATE Customers
SET ContactName = 'Alfred Schmidt', City = 'Frankfurt'
WHERE CustomerID = 1;
```

Here UPDATE updates the first customer with the new name, “Alred Schmidt, ” and the city is now Frankfurt. Here is a link with more information on the UPDATE statement:
https://www.w3schools.com/mysql/mysql_update.asp

The DROP statement removes existing tables permanently from the database. You can also drop multiple tables and separate with tables you would like to delete by comma. The IF EXISTS option paired with delete drops a table only if the table exists. Here is a link with more information on DROP:
https://www.mysqltutorial.org/mysql-drop-table#:~:text=The%20DROP%20TABLE%20statement%20removes,to%20remove%20temporary%20tables%20only.