Video Rental Portfolio

John Wensink

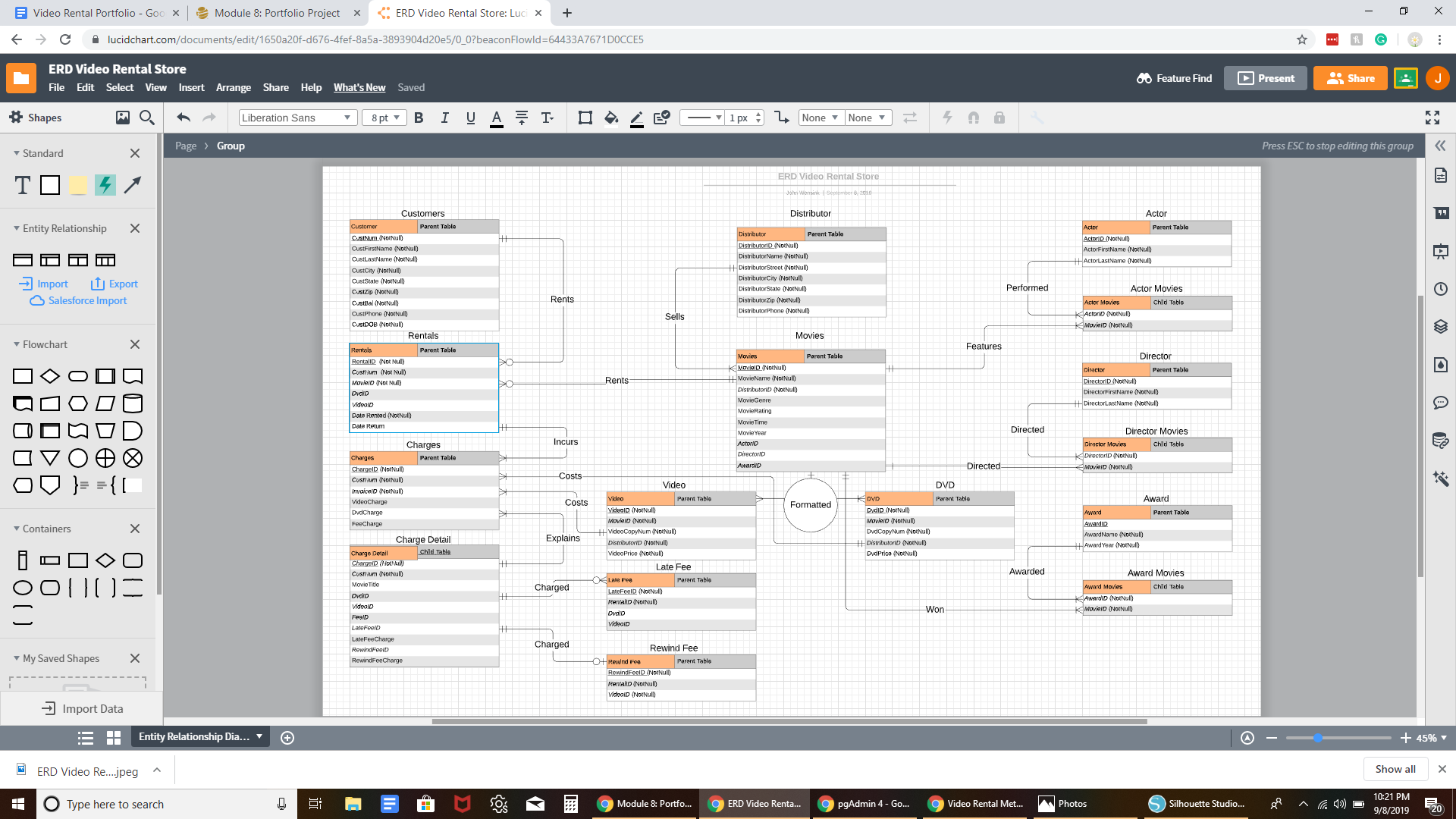
MIS407 – Database Concepts

Colorado State University-Global Campus

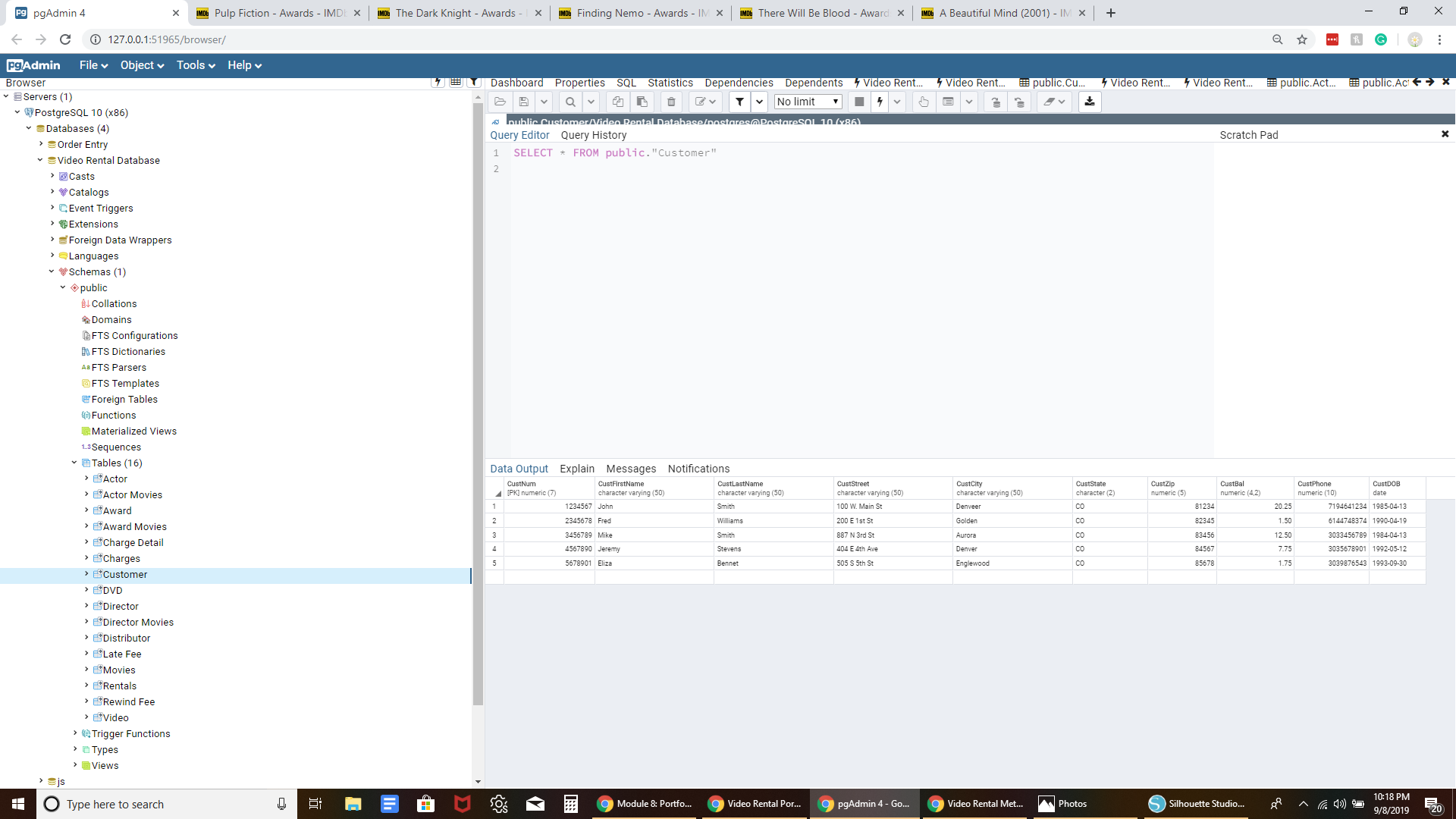
Professor Nathan Braun

September 8, 2019

Video Rental Portfolio

The database lifecycle of this project began with a bar napkin sketch with the entities as well as their cardinalities in order to gain a high-level understanding of the logical layout of the database system. From this rough draft, a rough Entity-Relationship Diagram (ERD) was created in LucidChart with entities and cardinalities only. The final ERD was then generated in LucidChart, this time adding rows of attributes as well as selecting primary and foreign keys that would be transferred into the physical model using Google Sheets. Constraints were added to the excel file to uniquely identify primary keys for different tables based on numeric length. For example, a CustomerID is seven digits long, a charge ID is nine digits long, and a movie ID has 10 digits. The physical database was created and the SQL statements below and were noted using Google Docs. 

**Creating the Physical Database**

****

1. The ERD table was created using LucidChart to develop a logical model of the video rental database
2. Metadata was stored in a Google Sheets file. The length of an entity’s primary key was selected to be at least 7 digits and each entity has a differing number of digits to serve as a unique primary key.
3. All tables were created by creating each table first before adding any attributes. Once all tables were created, attributes were added paying special attention to primary key constraints as well as NOT NULL requirements. Finally, foreign keys were added to the tables in order to make the database relational.
4. Data was then inserted into the tables paying special attention to constraints on what types of data were permissible for use in the database.

**The SQL INSERT statements used the following format:**

* 1. INSERT INTO MOVIE VALUES (MovieID,MovieName,DistributorID)

(‘123456’,’A BEAUTIFUL MIND’,’6543210))

(ElephantSQL, n.d.)

1. To show the contents of all tables I used simple SELECT statements for every table in the database.

**The SQL SELECT statements used the following format:**

SELECT \* from CUSTOMER;

SELECT \* from ACTOR;

SELECT \* from AWARD;

SELECT \* from MOVIE

SELECT\* from DIRECTOR

... (etc)

**To retrieve customer’s names, account number, and addresses sorted by account number the following format was used:**

SELECT customer\_FirstName,customer\_LastName,customer\_AccountNum,customer\_Street,customer\_Zip FROM customers

ORDER by customer\_AccountNum

(W3Schools, n.d.)

**To retrieve all of the DVD’s rented in the last 30 days and sort chronologically I used the following format:**

SELECT rentals\_DVDid,rentals\_DateRented ORDER by rentals\_DateRented WHERE rentals\_DateRented >= to\_char(trunc(sysdate) - 30, ‘YYYY-MM-DD)

(StackOverflow, 2014)

**To update a customer’s maiden name to their married name I used the following script:**

UPDATE Customers

SET LastName = Married

WHERE LastName = Maiden

(SQLTutorial, n.d.)

**To delete a customer from the database I ran the following script:**

DELETE FROM Customers

WHERE CustID = 1234567

(365DataScience, n.d.)

**Lessons Learned From This Project**

I thoroughly enjoyed working on this project. One thing I didn’t count on was the constraints I put on what kinds of data were permissible in the database would give me such a hard time when attempting to enter it into the tables. I learned that creating a working database is much harder than I expected it to be, and I would have practiced more throughout the course had I known the challenges I would face today. I learned a lot in this class about what relational databases are and what can be accomplished using a high-quality database. I learned that SQL is every bit as challenging as learning how to write procedural code. I also learned why people who work on databases make relatively high salaries. This is tricky material and attention to detail is mandatory. I work at an insurance company where our entire business is database driven. It is these high-quality databases that allow employees to work quickly and productively. When the databases are working as intended, most employees don’t give them a second thought. But as soon as there is trouble one begins to realize just how dependent the company is on the performance of these systems. I thoroughly enjoyed this class even if I was exceedingly frustrated in the final hours of the portfolio project. I look forward to future SQL classes and find online learning to be the perfect avenue to pick up these skills.

References:

365DataScience. (2019, August 28). SQL DELETE Statement. Retrieved from <https://365datascience.com/sql-delete-statement/>

DeMilia, B. (2014, July 4). Retrieve all videos rented in the last 30 days and sort in chronological rental date order. Retrieved from <https://stackoverflow.com/questions/24577342/retrieve-all-videos-rented-in-the-last-30-days-and-sort-in-chronological-rental>

ElephantSQL. (n.d.). Getting Started with PostgreSQL. Retrieved from <https://www.elephantsql.com/blog/2017-02-24-databases-for-beginners-what-is-sql-what-is-postgresql.html>

SQLTutorial. (n.d.). SQL UPDATE: Modify Existing Data in a Table By Examples. Retrieved from <http://www.sqltutorial.org/sql-update/>

TutorialsPoint. (n.d.). PostgreSQL - CREATE Table. Retrieved from <https://www.tutorialspoint.com/postgresql/postgresql_create_table.htm>

W3Schools. (n.d.). SQL Order By Keyword. Retrieved from <https://www.w3schools.com/sql/sql_orderby.asp>