Jenna Hackett

000969721

Phase 1: Database Design  
Phase 2: Database Implementation  
Phase 3: SQL Queries and Reporting

Video on Demand

Database Design and Programming - CRPG250A

Table of Contents

[Phase One: Database Design 2](#_Toc206079071)

[1. Conceptual Model 2](#_Toc206079072)

[2. Logical Model 4](#_Toc206079073)

[3. Physical Model 8](#_Toc206079074)

[Phase 2: Database Implementation 9](#_Toc206079075)

[4. Create Tables with Constraints 9](#_Toc206079076)

[5. Populate Database Tables with Sample Data 10](#_Toc206079077)

[Phase 3: SQL Queries and Reporting 11](#_Toc206079078)

[6. Develop Reports Using SQL Queries 11](#_Toc206079079)

# Phase One: Database Design

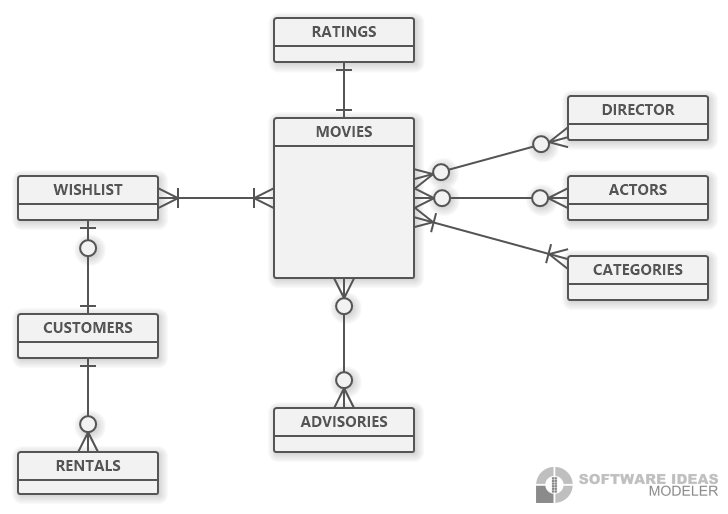
## Conceptual Model

#### List of Assumptions

* DB only tracks rentals which have occurred. The customer must have rented a movie to be in the DB.
* Email address for actor/actress/director will be ignored.
* Ignoring long advisory.
* A movie must have a rating.
* A movie must belong to at least one category but can be more than one.
* A movie may not have any advisories.
* A customer does not have to have a wish-list.
* Custom categories will be used due to Shaw VOD being unavailable.
  + Horror, Drama, Romance, Thriller, Comedy, Sci-Fi, New Releases, Most Popular, etc.
* There are no subcategories of movies.
* Duration (minutes) will always be above 0.
* Customer ratings range between 1 and 5.
* Rental expiry 24hrs after customer starts viewing.
* Rental expiry cannot be before the view date.
* Credit cards are numeric values only. Expiry and CVV are not saved.
* American Express, MasterCard and Visa are only accepted types of credit cards.
* Movie ratings
  + G, PG, 14A, 18A, R.
* Advisories
  + Course Language, Language may Offend, Violence, Frightening Scene, Brutal Violence, Gory Scene, Sexual Violence, Nudity, Sexually Suggestive Scenes, Sexual Content, Explicit Sexual Content.
* Only top 3 starring actors will be listed for each respective movie.
* Customer email address is mandatory.
* Postal Code à L9L9L9.
  + Canadian Addresses only.
* Phone number constraints (formatting).
* Wishlist will show date movie was added to it.
* Actor and director DOB will be ignored.
* Movies, Actors, Directors and Customers have an ID number.

#### Conceptual Model Diagram

Initial draft of the database, based off the list of assumptions from above.



## Logical Model

VOD (movie\_id, movie\_title, movie\_category, movie\_rating, movie\_advisory, movie\_duration, sd\_price, hd\_price, actor\_id, actor\_name, actor\_role, director\_id, director\_name, customer\_id, customer\_name, customer\_address, customer\_email, customer\_phone, customer\_rating, wishlist\_added\_date, creditcard\_number, creditcard\_type, rental\_id, rental\_start\_date, rental\_expiry\_date, rental\_watch\_date)

#### Steps For Normalization

**1Nf – Multi-part Attributes**

1. Customer\_name à customer\_first\_name, customer\_last\_name
2. actor\_name à actor\_first\_name, actor\_last\_name
3. director\_name à director\_first\_name, director\_last\_name
4. customer\_address à customer\_street, customer\_city, customer\_province, customer\_postal\_code

**1NF – Repeating Groups**

1. Top 3 cast leads as actor\_id1, actor\_id2, and actor\_id3, actor\_first\_name1, actor\_first\_name2, etc.
2. Multiple advisories per movie. movie\_advisory à advisory\_description

VOD (movie\_id, movie\_title, movie\_category, movie\_rating, movie\_advisory, movie\_duration, sd\_price, hd\_price, actor\_id, actor\_first\_name, actor\_last\_name, actor\_role, director\_id, director\_first\_name, director\_last\_name, customer\_id, customer\_first\_name, customer\_last\_name, customer\_street, customer\_city, customer\_province, customer\_postal\_code, customer\_email, customer\_phone, customer\_rating, wishlist\_added\_date, creditcard\_number, creditcard\_type, rental\_id, rental\_start\_date, rental\_expiry\_date, rental\_watch\_date)

movie\_id à (movie\_title, movie\_category, movie\_rating, movie\_advisory, movie\_duration, sd\_price, hd\_price, actor\_id, actor\_first\_name, actor\_last\_name, actor\_role, director\_id, director\_first\_name, director\_last\_name)

actor\_id à (actor\_first\_name, actor\_last\_name, actor\_role)

director\_id à (director\_first\_name, director\_last\_name)

customer\_id à (customer\_first\_name, customer\_last\_name, customer\_street, customer\_city, customer\_province, customer\_postal\_code, customer\_email, customer\_phone, wishlist\_added\_date, creditcard\_number, creditcard\_type)

rental\_id à (rental\_start\_date, rental\_expiry\_date, rental\_watch\_date, customer\_rating)

rental\_start\_dateà rental\_watch\_date

rental\_watch\_dateà rental\_expiry\_date

creditcard\_numberà creditcard\_type

Primary Key = composite  
(movie\_id, customer\_id, rental\_id) à movie\_title, movie\_category, movie\_rating, movie\_advisory, movie\_duration, sd\_price, hd\_price, actor\_first\_name, actor\_last\_name, actor\_role, director\_id, director\_first\_name, director\_last\_name, customer\_first\_name, customer\_last\_name, customer\_street, customer\_city, customer\_province, customer\_postal\_code, customer\_email, customer\_phone, customer\_rating, wishlist\_added\_date, creditcard\_number, creditcard\_type, rental\_start\_date, rental\_expiry\_date, rental\_watch\_date

**2NF – Partial Dependencies**

1. movie\_id à movie\_title, movie\_category, movie\_rating, movie\_advisory, movie\_duration, sd\_price, hd\_price, actor\_id, actor\_first\_name, actor\_last\_name, actor\_role, director\_id, director\_first\_name, director\_last\_name
2. rental\_id à movie\_id, rental\_start\_date, rental\_expiry\_date, rental\_watch\_date, customer\_rating
3. customer\_id à customer\_first\_name, customer\_last\_name, customer\_street, customer\_city, customer\_province, customer\_postal\_code, customer\_email, customer\_phone, wishlist\_added\_date, creditcard\_number, creditcard\_type

**2NF – Remove Partial Dependencies**

MOVIES (movie\_id, movie\_title, movie\_category, movie\_rating, movie\_advisory, movie\_duration, sd\_price, hd\_price, actor\_id, actor\_first\_name, actor\_last\_name, actor\_role, director\_id, director\_first\_name, director\_last\_name)

RENTALS (movie\_id, rental\_id, rental\_start\_date, rental\_expiry\_date, rental\_watch\_date, customer\_rating)

CUSTOMERS (movie\_id, rental\_id, customer\_id, customer\_first\_name, customer\_last\_name, customer\_street, customer\_city, customer\_province, customer\_postal\_code, customer\_email, customer\_phone, wishlist\_added\_date, creditcard\_number, creditcard\_type)

VOD (movie\_id, rental\_id, customer\_id) à rental\_start\_date, rental\_expiry\_date, rental\_watch\_date

**3NF – Transitive Dependencies**

1. movie\_advisory à movie\_rating
2. actor\_id à actor\_first\_name, actor\_last\_name
3. actor\_id, movie\_id à actor\_role
4. movie\_id, customer\_id à customer\_rating
5. director\_id à director\_first\_name, director\_last\_name
6. rental\_start\_dateà rental\_watch\_date
7. rental\_watch\_dateà rental\_expiry\_date
8. creditcard\_numberà creditcard\_type

**3NF – Removing Transitive Dependencies**

ACTORS (actor\_id PK, actor\_first\_name, actor\_last\_name)

DIRECTORS (director\_id PK, director\_first\_name, director\_last\_name)

CREDITCARDS (customer\_id PK, creditcard\_number, creditcard\_type)

MOVIES (movie\_id PK, movie\_title, movie\_category, movie\_rating, movie\_advisory, movie\_duration, sd\_price, hd\_price)

RENTALS (rental\_id PK, movie\_id FK, customer\_id FK, rental\_start\_date, rental\_expiry\_date, rental\_watch\_date, customer\_rating)

CUSTOMERS (customer\_id PK, , customer\_first\_name, customer\_last\_name, customer\_street, customer\_city, customer\_province, customer\_postal\_code, customer\_email, customer\_phone)

ADVISORIES (advisory\_id PK, advisory\_description)

CATEGORIES (category\_name PK)

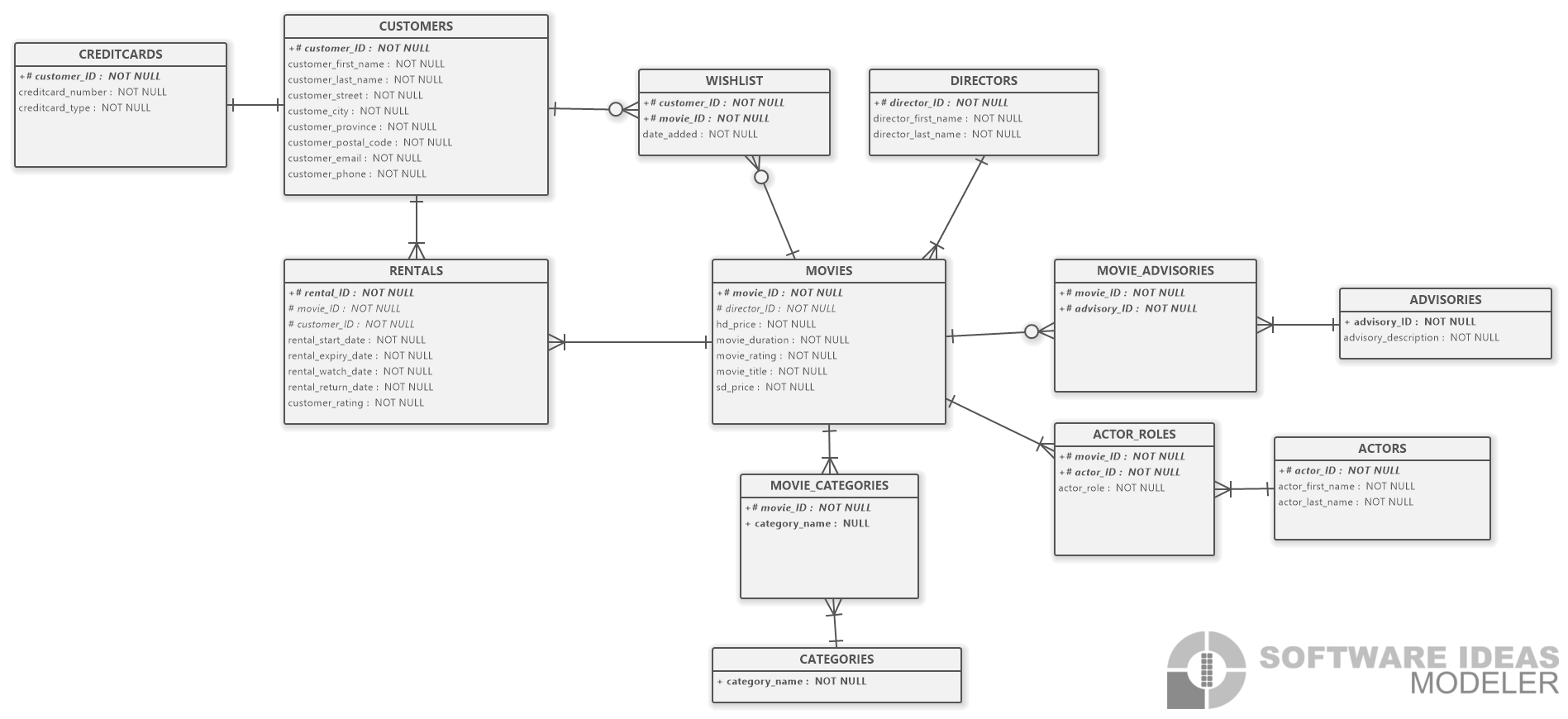
**3NF – Bridging tables to resolve Many to Many relationships:**

1. Movies can have many actors.
   * ACTOR\_ROLES (movie\_id PK, actor\_id PK, actor\_role)
2. Movies can have multiple advisories.
   * MOVIE\_ADVISORIES (movie\_id PK, advisory\_id)
3. Multiple movies can be on a Wishlist.
   * WISHLIST (customer\_id PK, movie\_id PK, wishlist\_added\_date)
4. A movie can appear in multiple categories.
   * MOVIE\_CATEGORIES (movie\_id PK, category\_name PK)

#### Logical Model Diagram

Expansion of the previous conceptual model, including entities, attributes, PKs and FKs. The previous many to many relationships are resolved.

+ = Primary Key (PK)  
# = Foreign Key (FK)  
+# = Composite PK

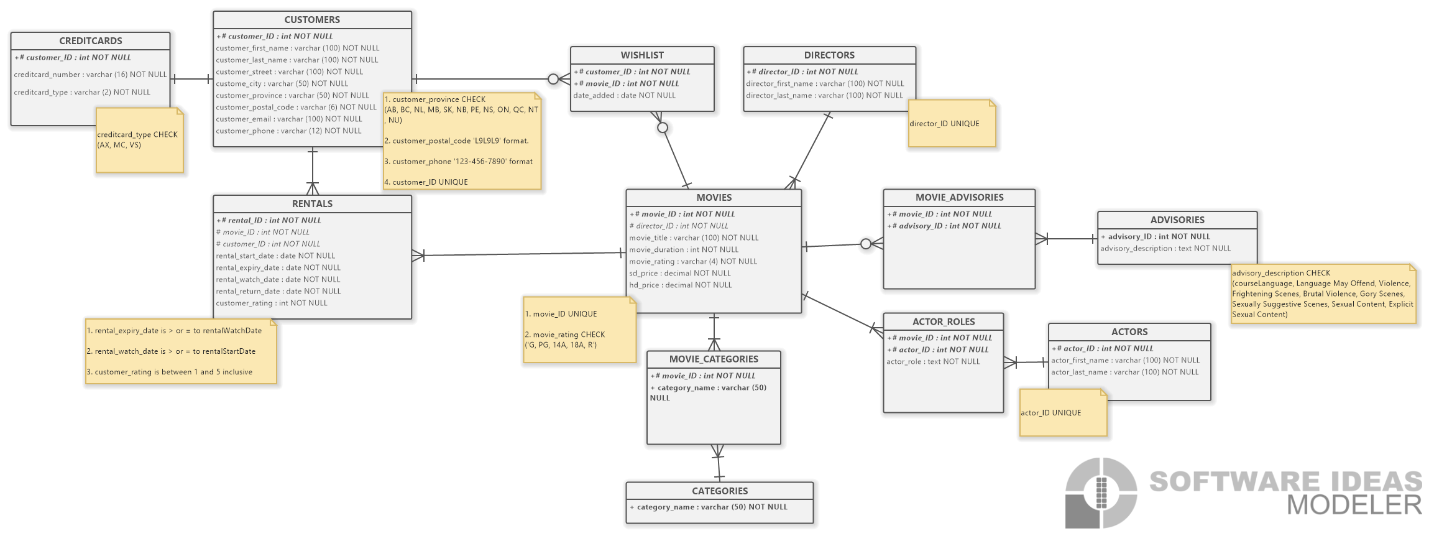


## Physical Model

An expansion on the logical model from above. The physical model includes datatypes and constraints, with check constraints commented within the diagram.

#### Physical Model Diagram

+ = Primary Key (PK)  
# = Foreign Key (FK)  
+# = Composite PK



# Phase 2: Database Implementation

## Create Tables with Constraints

A screenshot of a computer

AI-generated content may be incorrect.

Tables are implemented using best practices.  
DROP TABLE, CASCADE and CREATE TABLE. Parent tables created before child tables and there are table-level constraints for primary keys.

A screenshot of a computer

AI-generated content may be incorrect.

## Populate Database Tables with Sample Data

A screenshot of a computer

AI-generated content may be incorrect.  
A python file was used to populate the database tables with sample data. Most tables have 1000 rows, except for a few tables like advisories and categories, where 1000 rows do not really make sense. Rentals table has 5000 entries as there is likely more than 1000 rentals when there are 1000 customers.

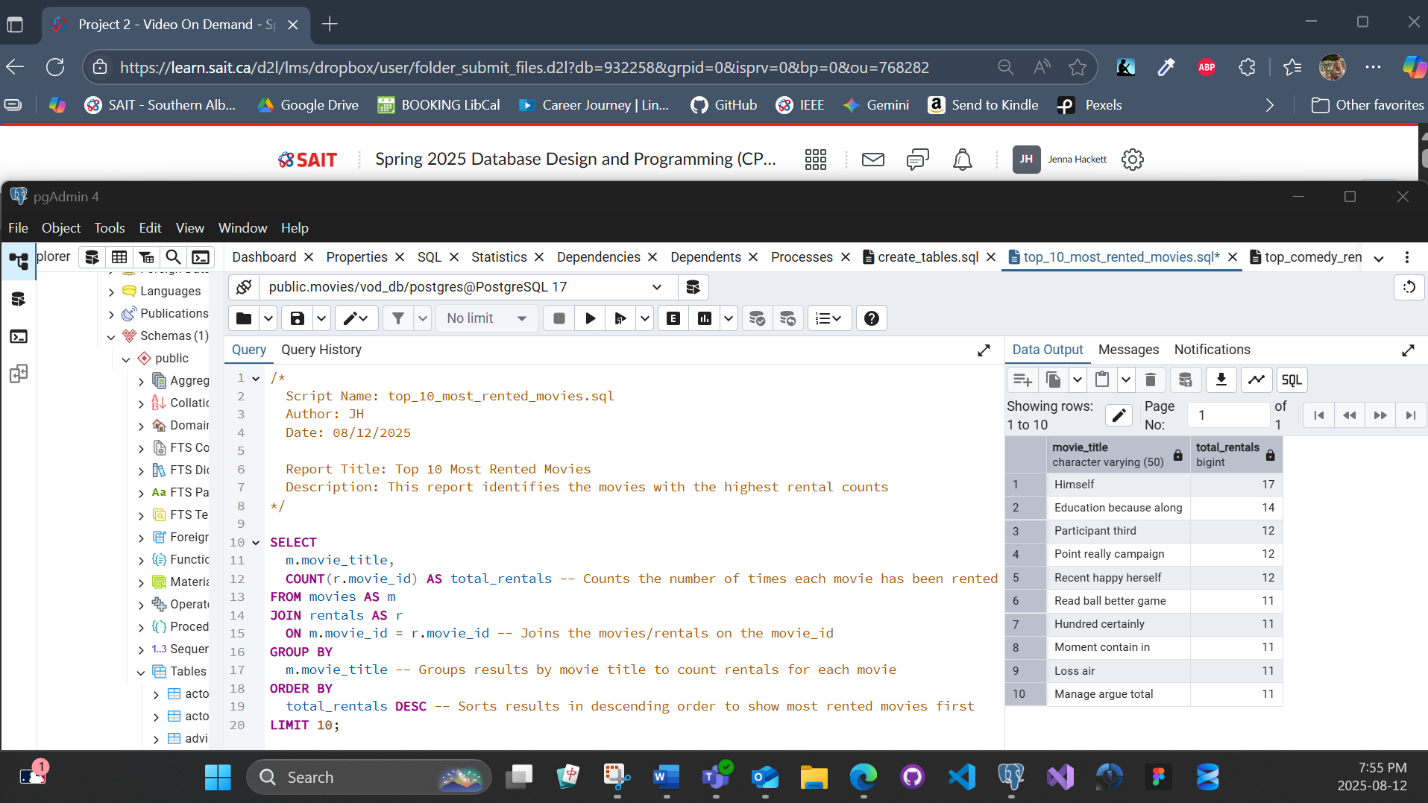
A screenshot of a computer

AI-generated content may be incorrect.

# Phase 3: SQL Queries and Reporting

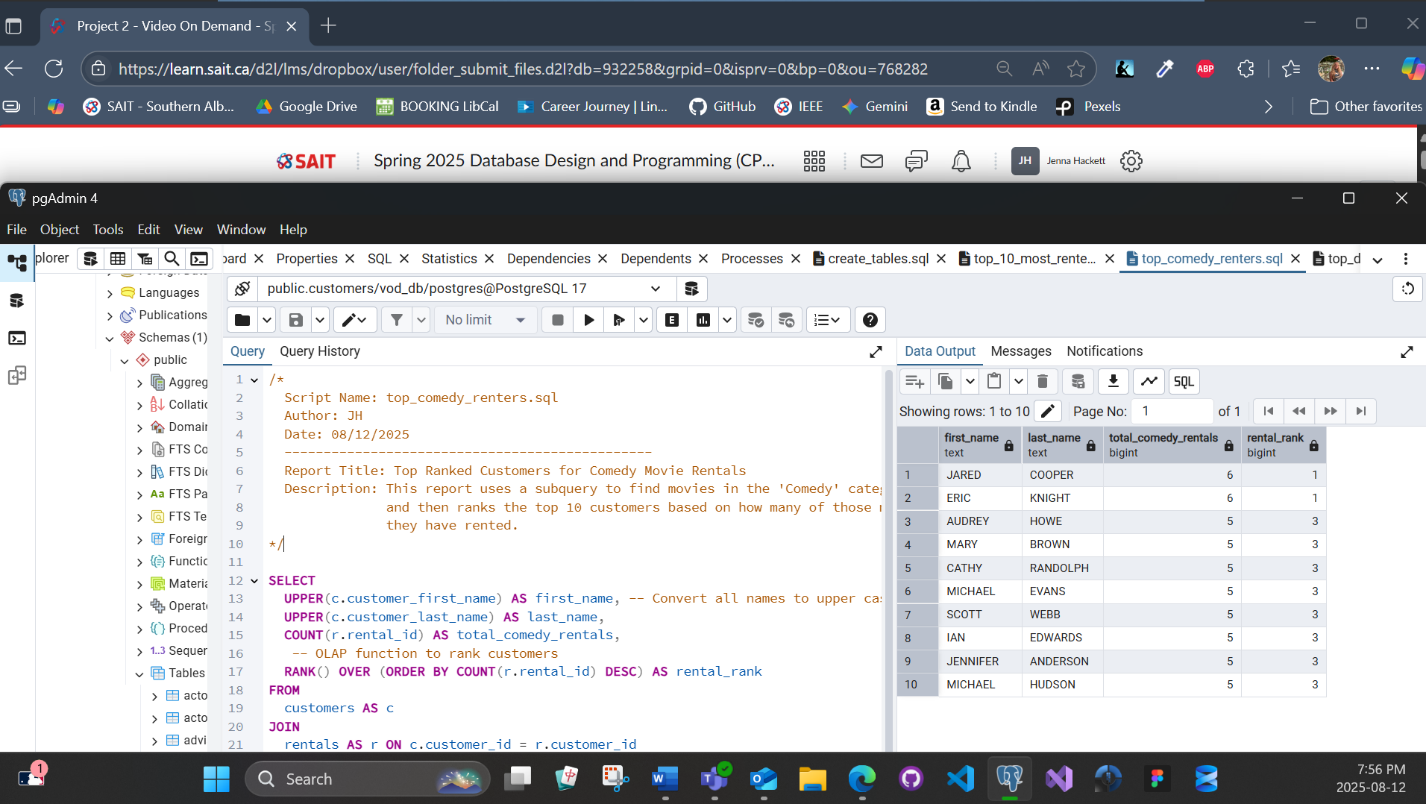
## Develop Reports Using SQL Queries

#### Top 10 Rented Movies



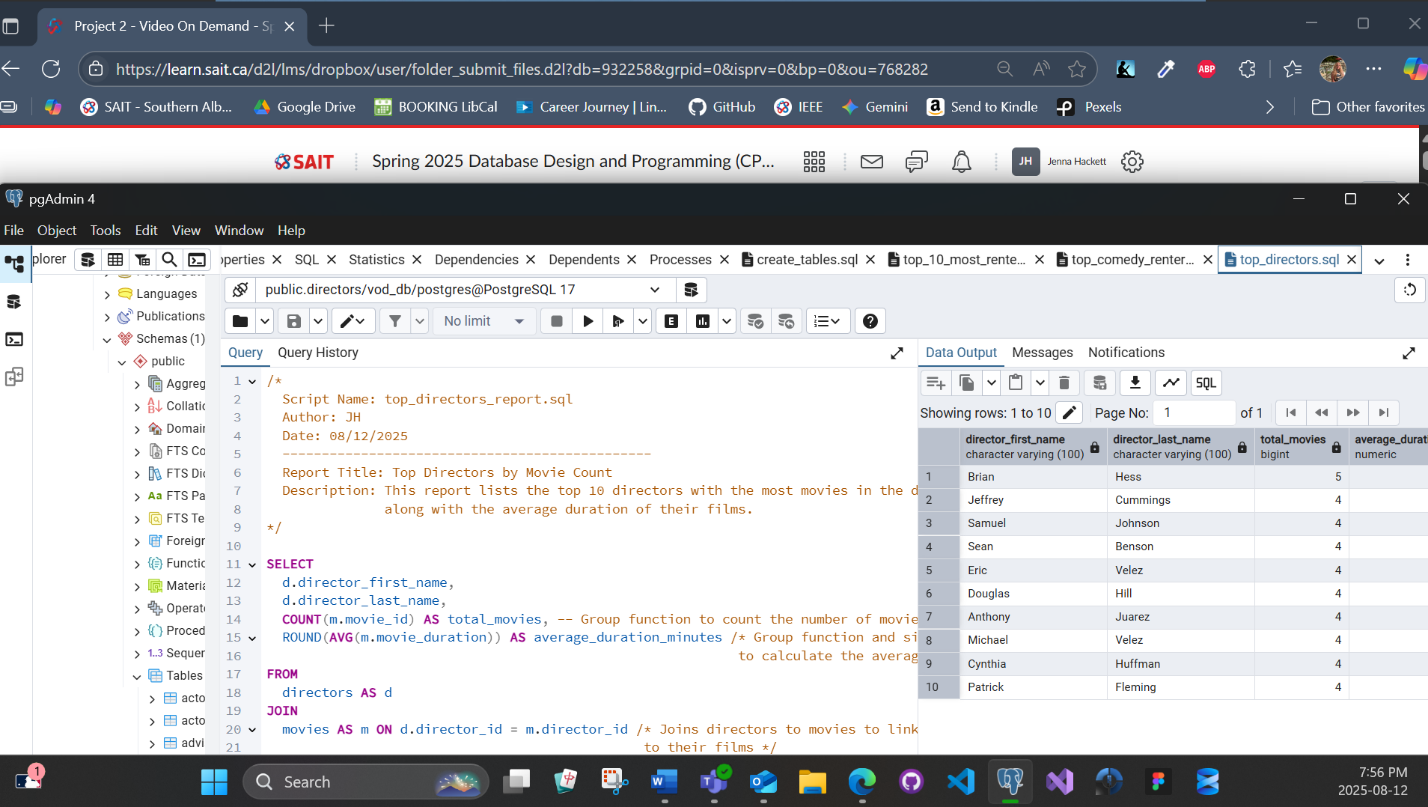
This report uses 3 SQL concepts. Joins, restricting/sorting data, and group functions. It uses JOIN and combines the movies and rentals tables together to merge the data.  
It uses ORDER BY to sort the movies by their rental count, and LIMIT to restrict the results to the top 10 most rented movies.  
It uses COUNT to aggregate the number of rentals for each movie.  
This report will identify the 10 most popular movie rentals at the time of the query.

#### Top Comedy Renters



This report uses all 6 SQL concepts. Single-row functions, subqueries, OLAP, joins, group functions and restricting/sorting data.  
It uses UPPER () to convert the customers first/last names into all upper-case letters.  
It uses the WHERE clause to filter the movies by the ‘comedy’ category.  
It uses RANK () to assign a ranking to each customer, based on their rental count.  
It uses JOIN to combine customers/rentals tables.  
It uses ORDER BY and LIMIT to get the top 10 results.  
This report finds the movies in the ‘comedy’ category and ranks/orders the customers based on who watched comedy the most.

#### Top Directors



This report shows 3 SQL concepts. Joins, group functions and restricting/sorting data.  
It uses JOIN to combine the directors/movies tables.  
It uses COUNT to count the number of movies for each director. It uses AVG to calculate the average movie duration of all movies, for each director.  
It uses ORDER BY to sort the director from greatest number of movies to the least number of movies. It uses LIMIT to set the limit at the top 10 directors.  
This report shows the top 10 directors with the most movies on the VOD streaming service, as well as the average duration of their films.