CS4347 Final Project Deliverable II

<CS4347.0U1 Final Project Deliverable I>

**Deliverable II starts on page 14**

1. **Title of our database system**

Movies/TV Series Production Database

1. **Name of the participants and delegation of tasks**

Alex Armstrong: Chose our database and structured EER/relational diagram

Kevin Nguyen: Implementation of design to create database, tables, and populate data

Jaemin Lee: Creation of diagrams and putting information into the word document

Steven Nall: Comparison of our database with existing similar databases

Matthew Chalcraft: Comparison of our database with existing similar databases

Adhiraj Sen: Helped find research material for database comparison

1. **An introduction of our database including why we have chosen to design this particular database**

Our database helps model a real online service when it comes to keeping movies and TV shows information and customer/purchase information stored in an organized way that can adapt to any changes in structure. It also helps solve common ecommerce problems in keeping track of large data and scalability.

1. **Comparison of our work with existing similar databases**

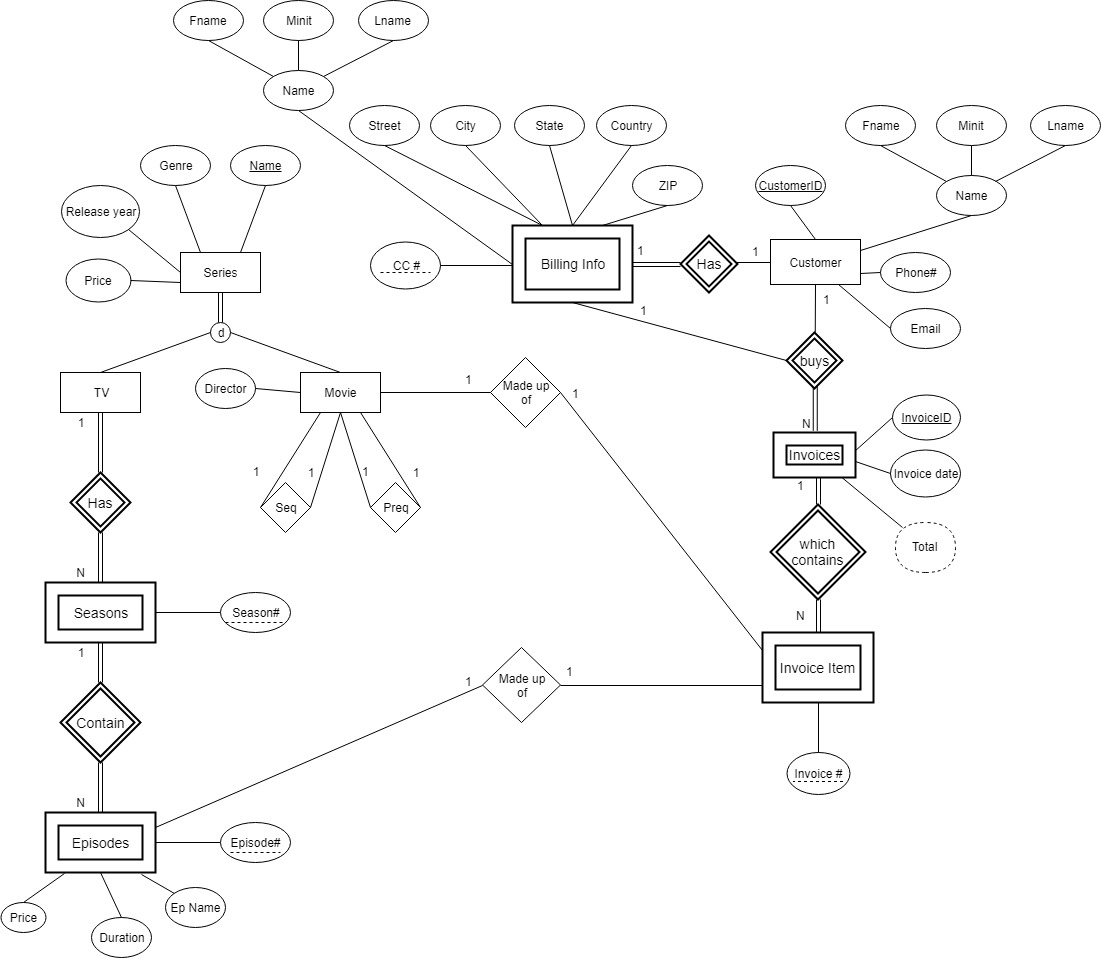
Currently our service is a Movie/TV series rental and purchasing database, where the user can go and search through the database and purchase some form of entertainment for themselves. This service is like that of Apples Itunes, Amazon Prime video and Netflix all of which are media rental platforms, although with some additional functionality. Netflix and APM both work using the AWS databases and Netflix is one of Amazons biggest, and most advanced, customers when it comes to their webservice [1]. Netflix is somewhat different from what we will be offering but has a comparable database with the amount of data that will incorporated, however they run on a schemeless database and have implemented “NMDB as a “schema-on-write” system” [2]. Amazon has similar system where you can watch some movie and TV series with a paid subscription however you can also purchase or rent movies, Tv episodes, and Tv seasons of specific series as the user wants. Finally, with Itunes there is no paid subscription for movies on Itunes however you can rent or purchase movies and tv series as you wish and it is saved to the users account. Again, this is similar to what we are hoping to produce, a database where you user can explore various movies and Tv series to then purchase or rent them for their own entertainment.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature | Our Service | Amazon Prime Video (APM) | Itunes | Netflix |
| Online Streaming | N | Y | N | Y |
| Paid Subscription | N | Y | N | Y |
| Purchases | Y | Y | Y | N |
| Rental | Y | Y | Y | N |

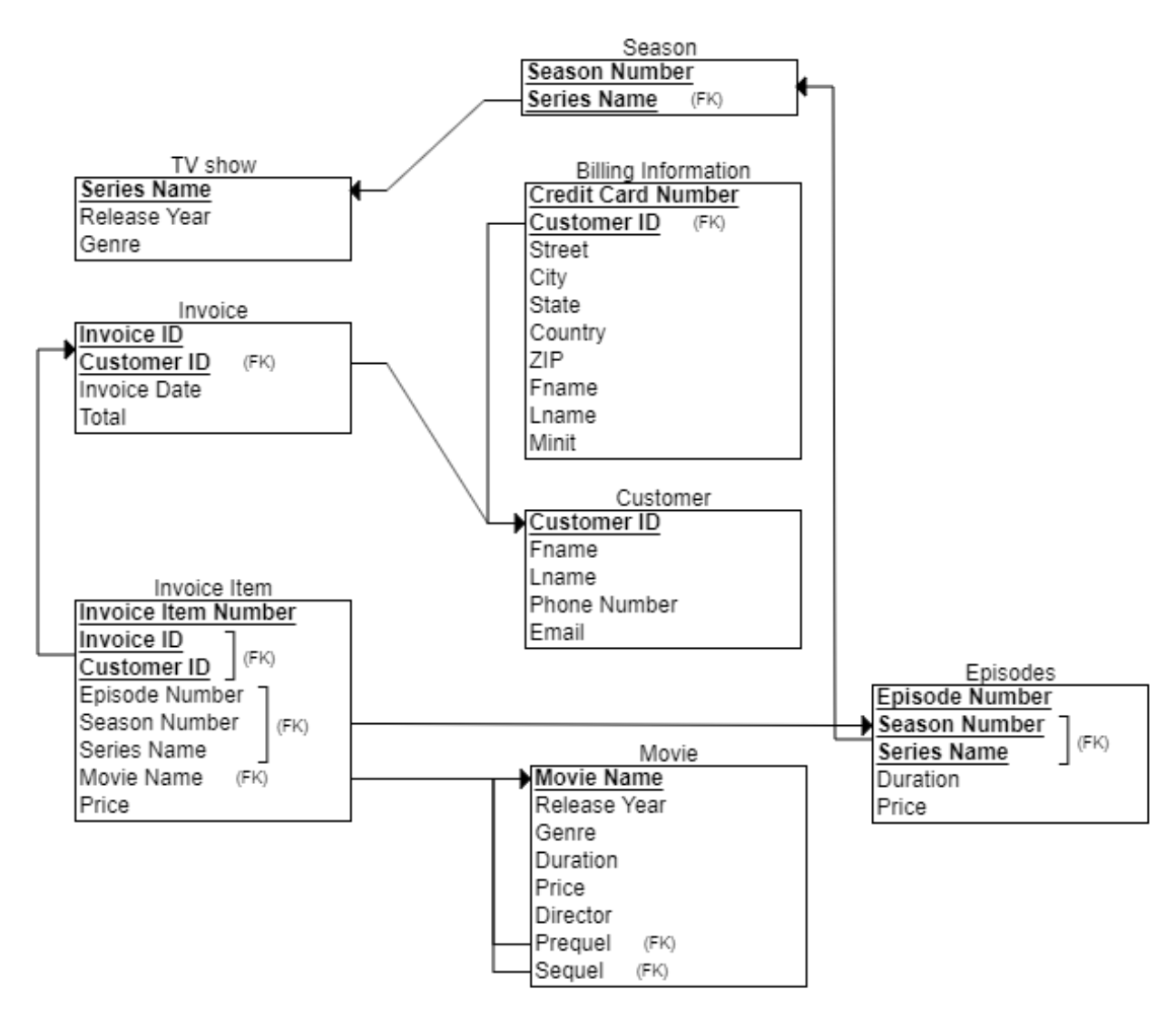
[1] Nair, M. (2019). *How Netflix works: the (hugely simplified) complex stuff that happens every time you hit Play*. [online] Medium. Available at: https://medium.com/refraction-tech-everything/how-netflix-works-the-hugely-simplified-complex-stuff-that-happens-every-time-you-hit-play-3a40c9be254b [Accessed 29 Jun. 2019].

[2] Tiwary, S. (2019). *Implementing the Netflix Media Database - Netflix TechBlog - Medium*. [online] Medium. Available at: https://medium.com/netflix-techblog/implementing-the-netflix-media-database-53b5a840b42a [Accessed 29 Jun. 2019].

1. **EER Diagram**

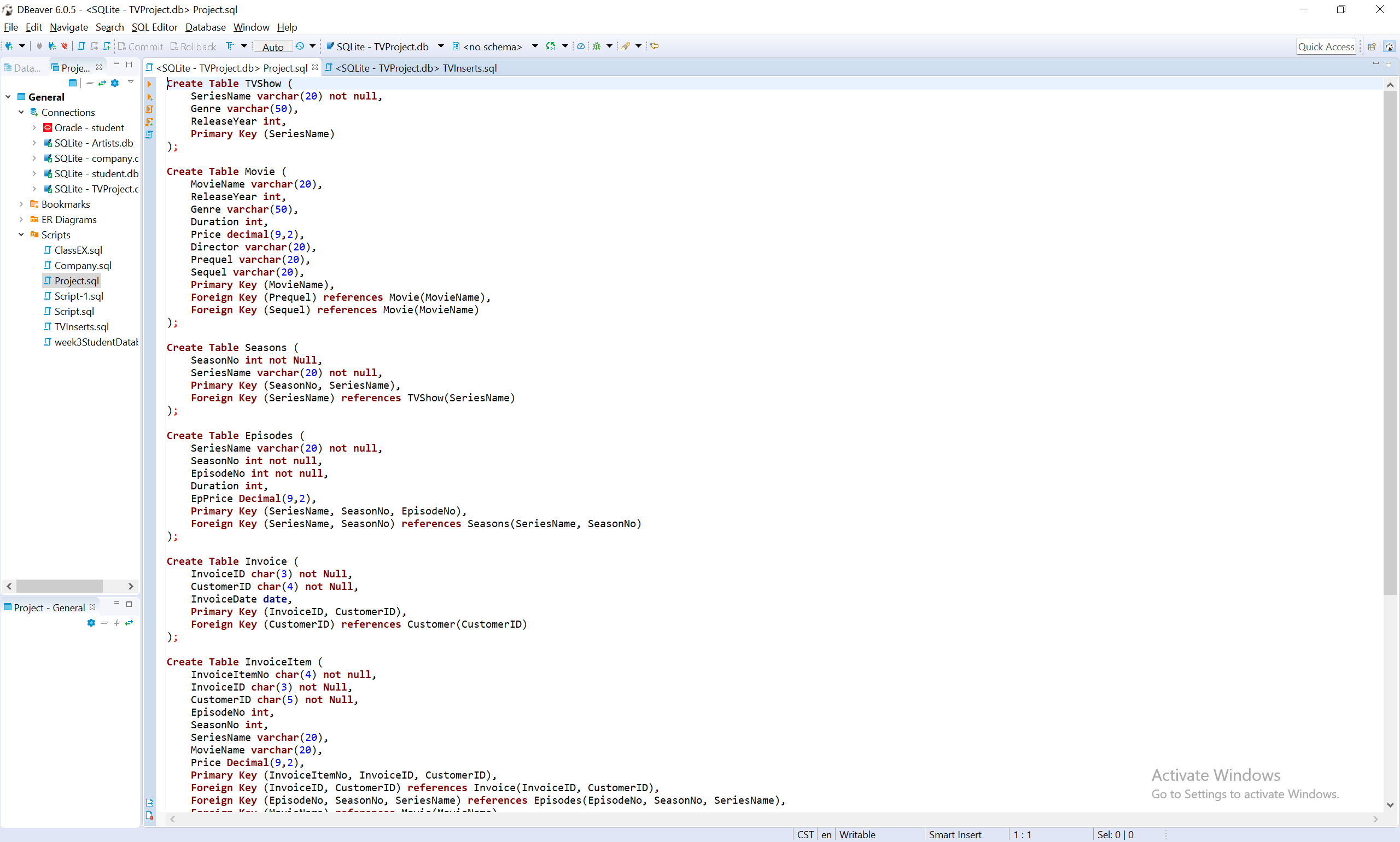
****

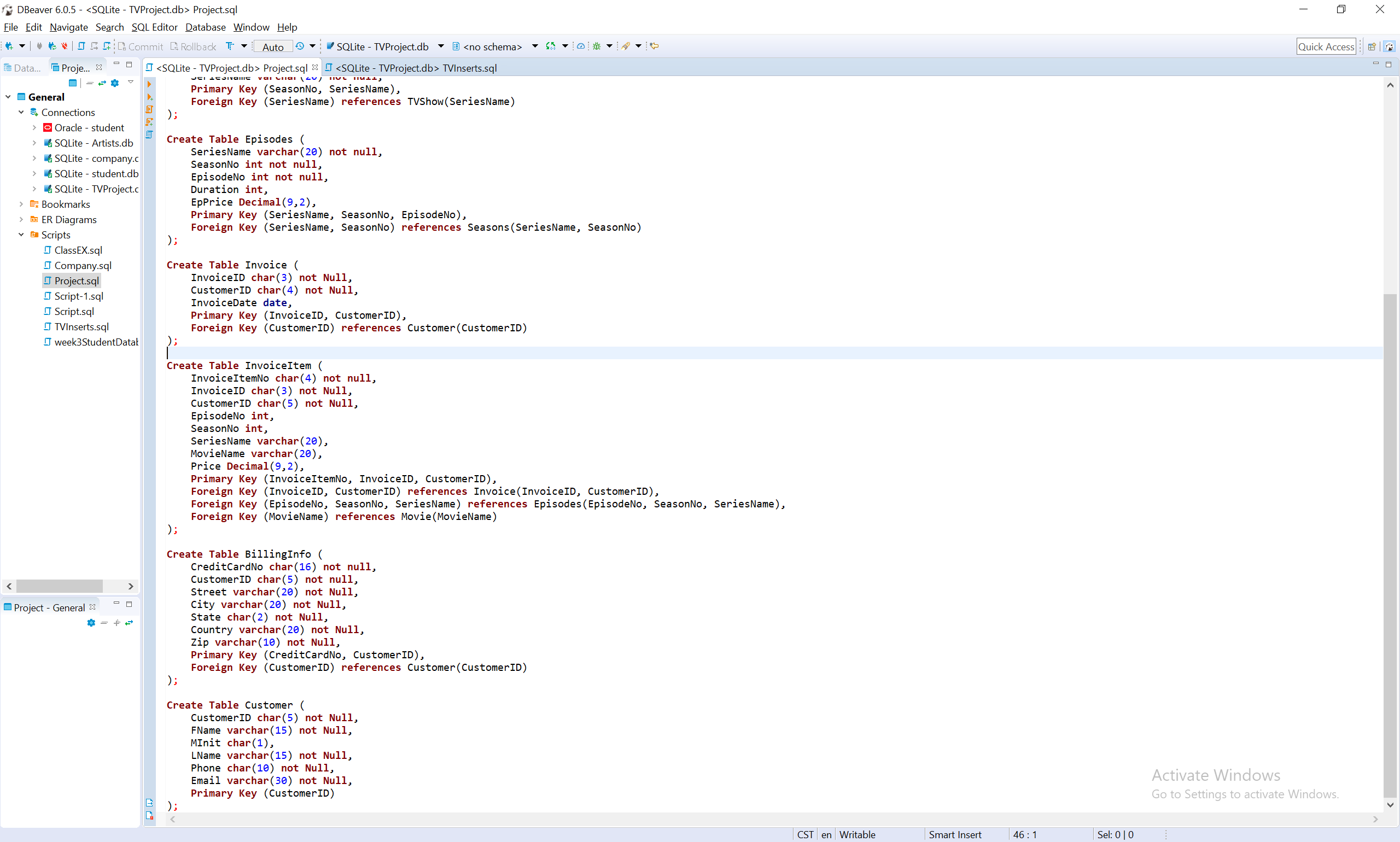
1. **EER to relational model mapping**

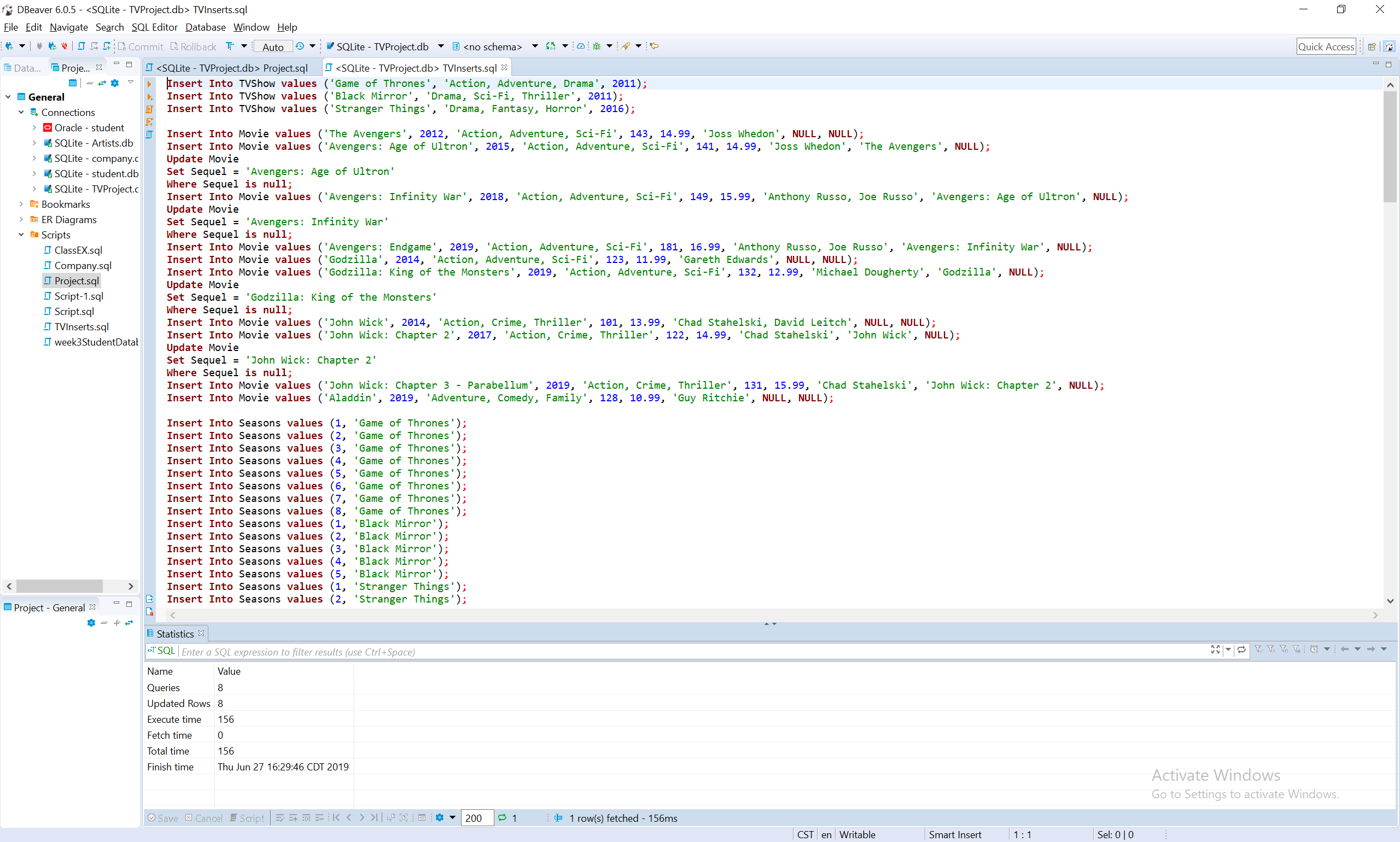


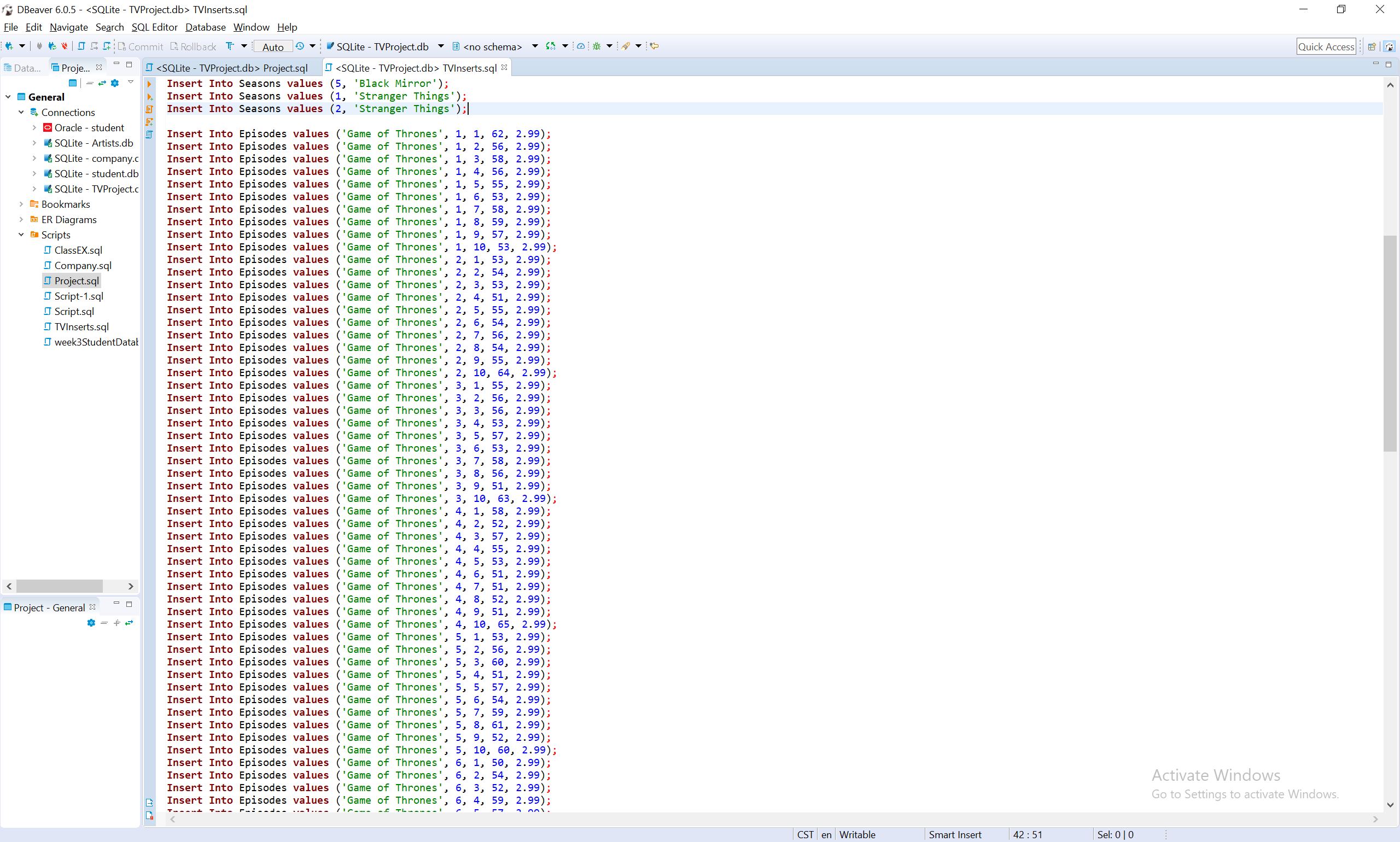
1. **Implementation of our design to create the database, tables, and also populate data**

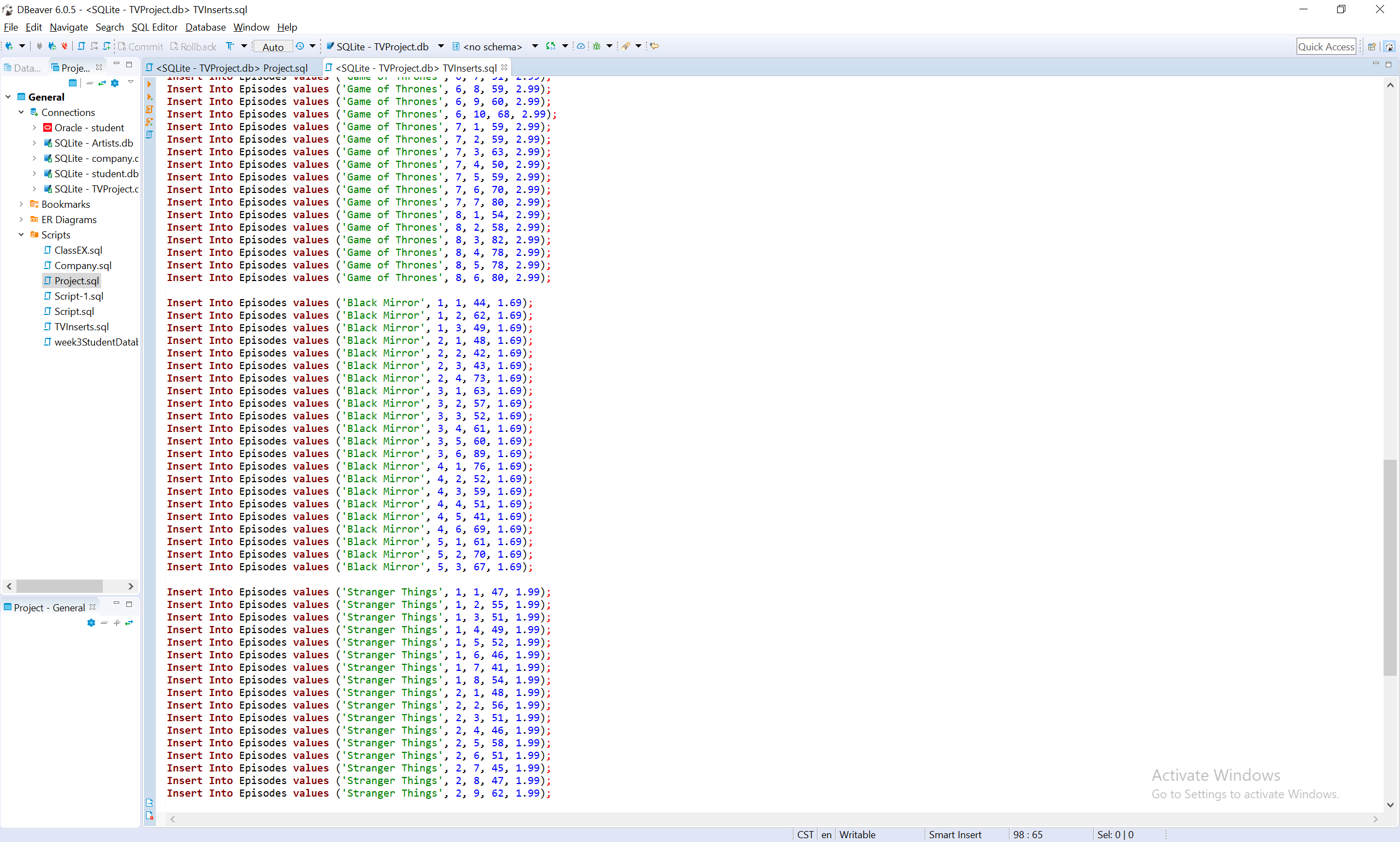
Also attached the SQL files (Project.sql and TVInserts.sql)

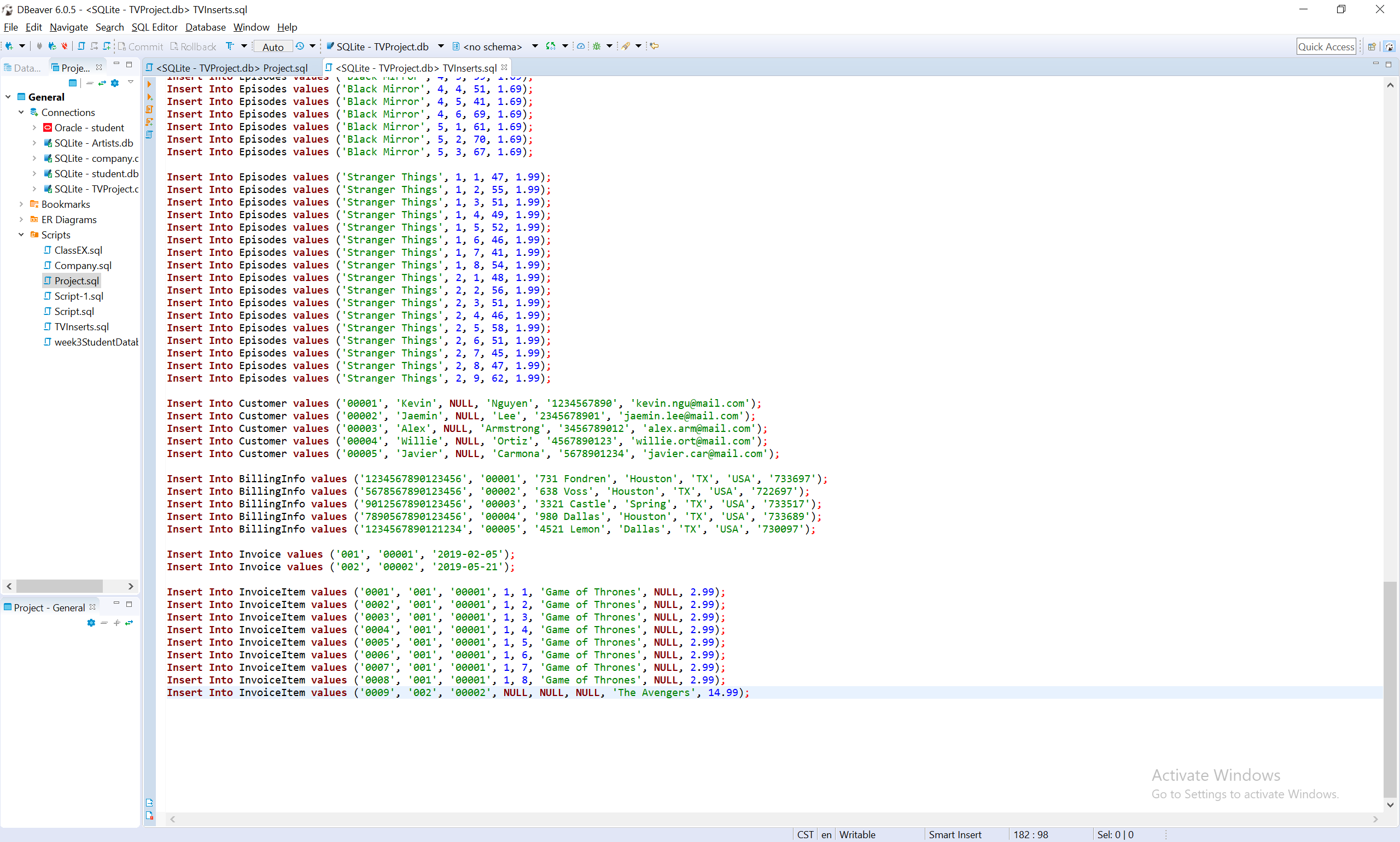


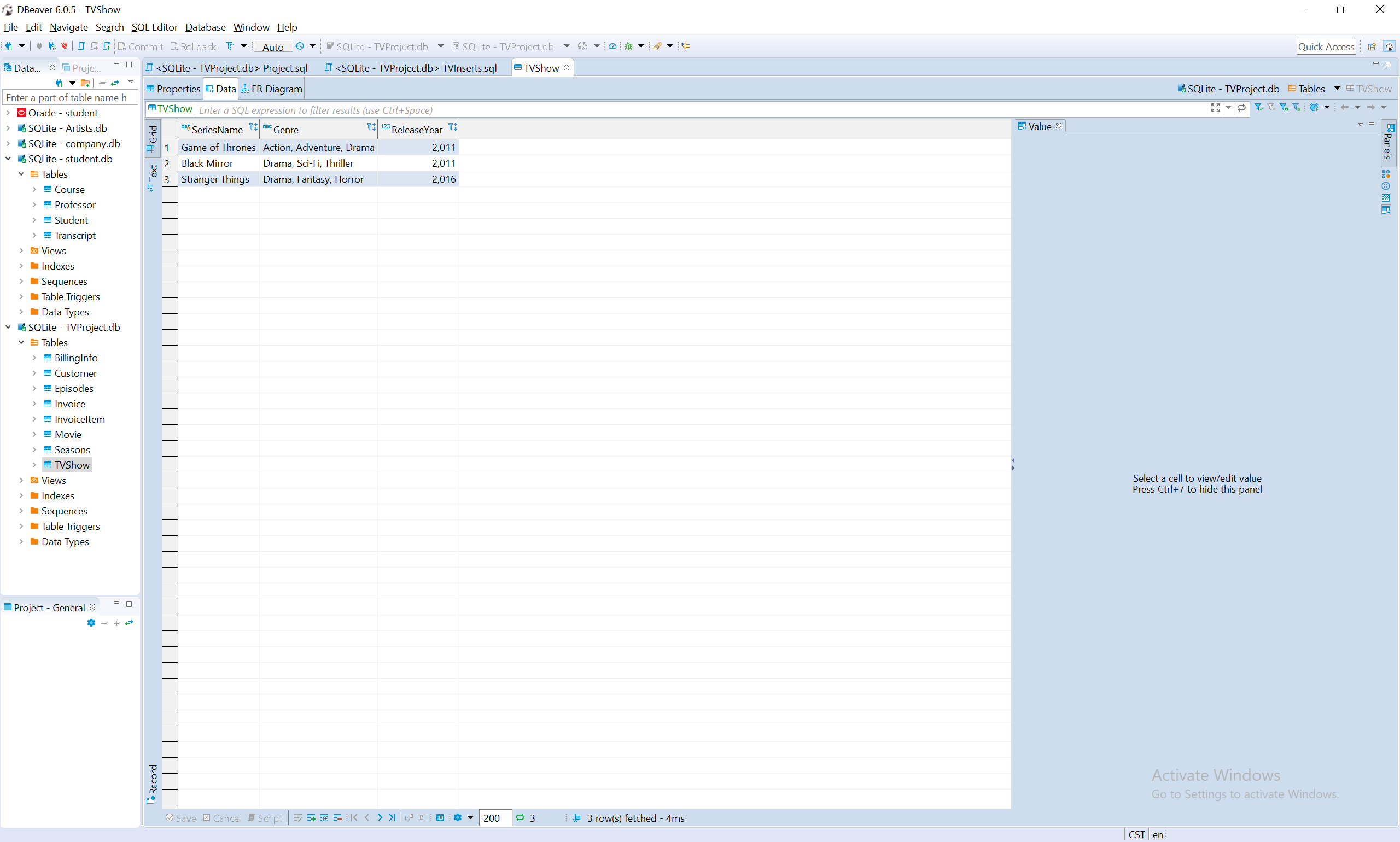
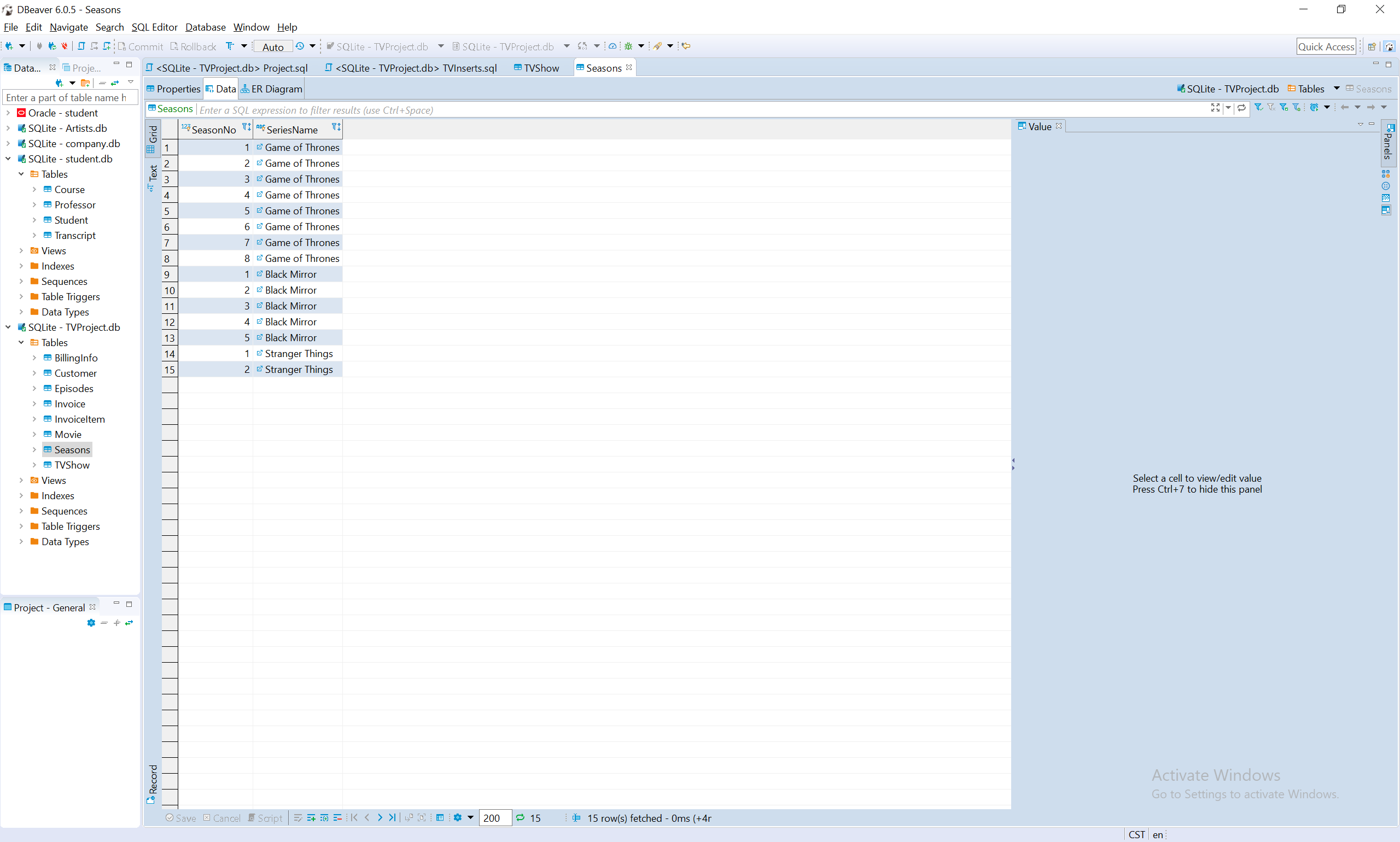


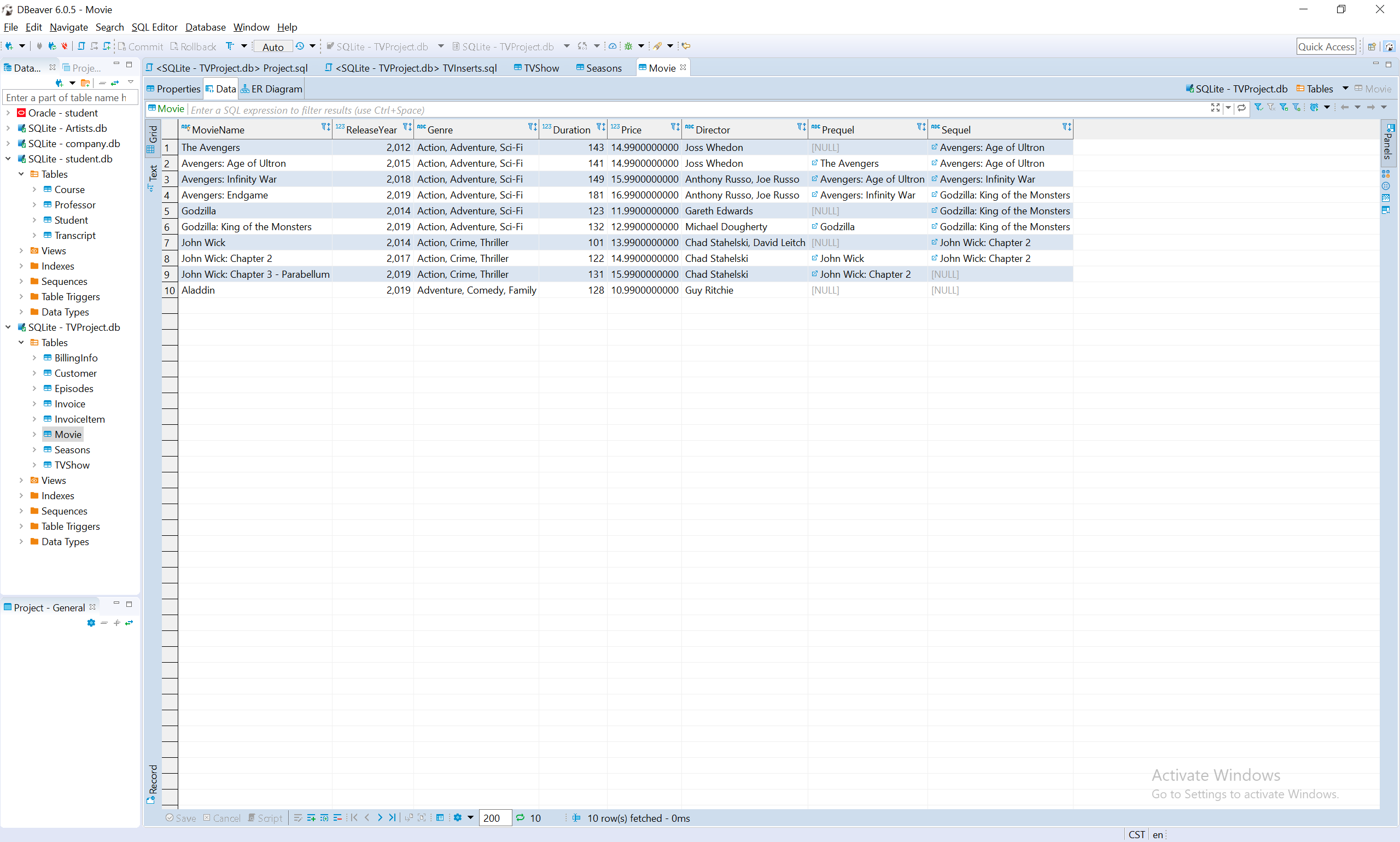


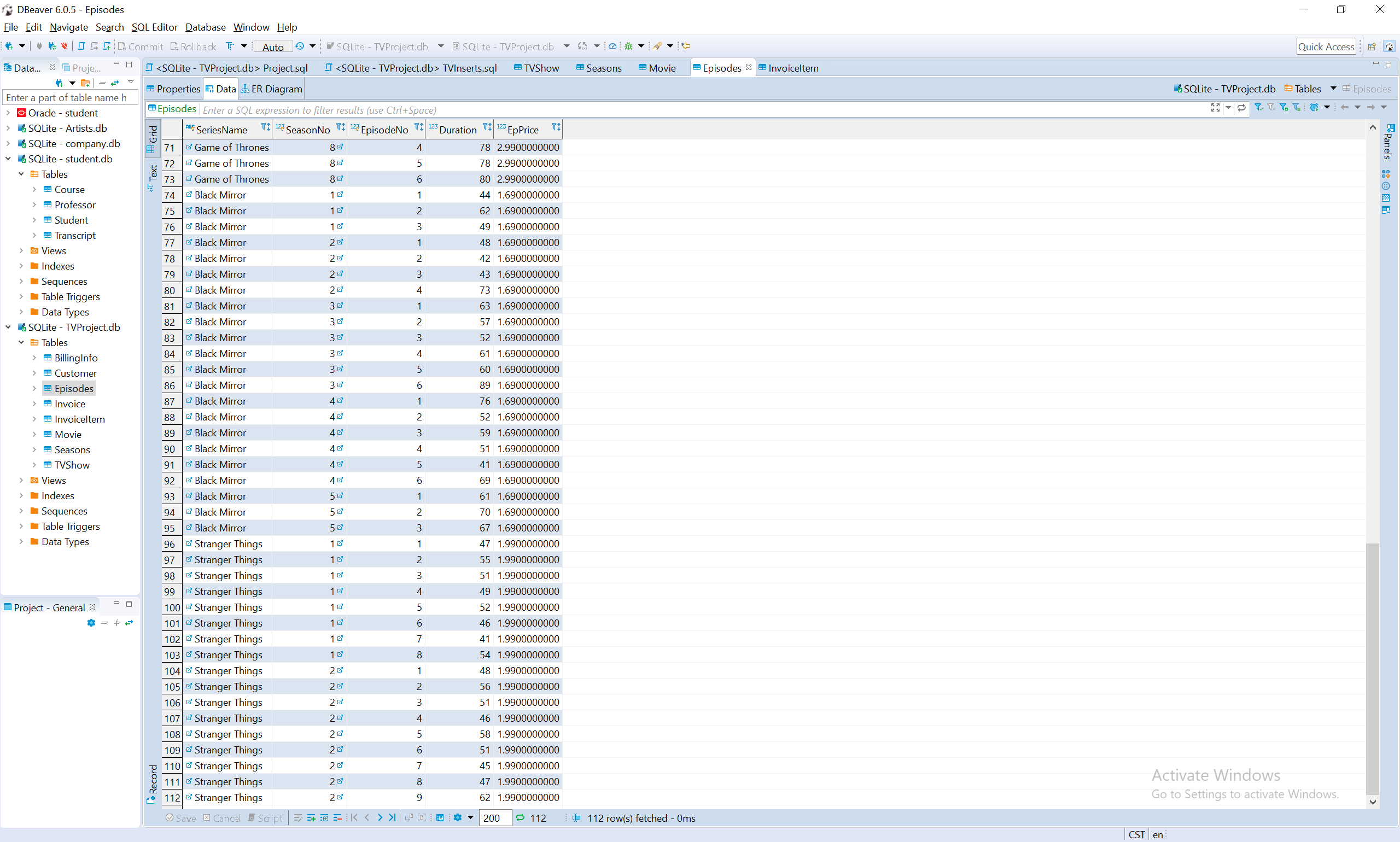
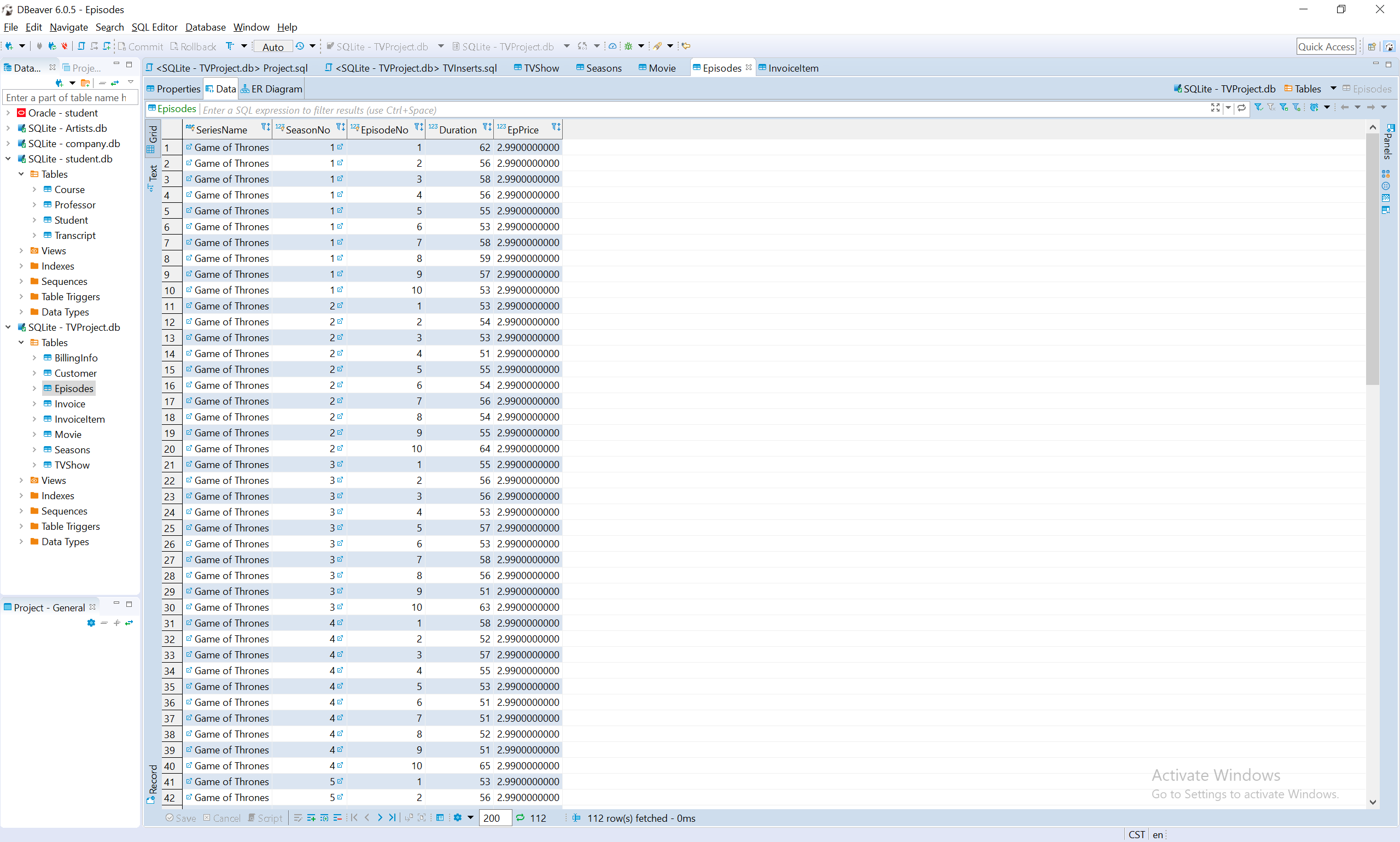


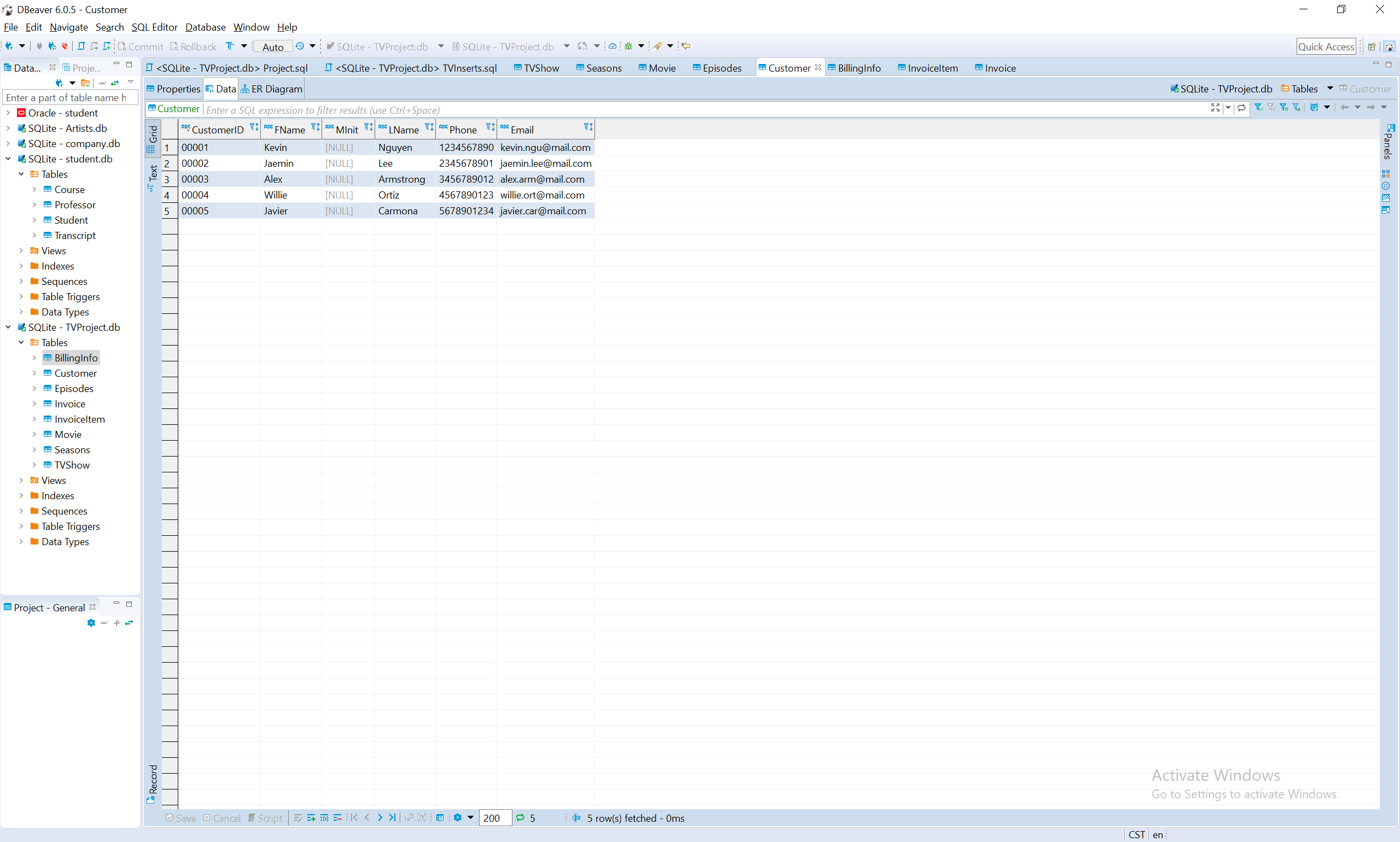


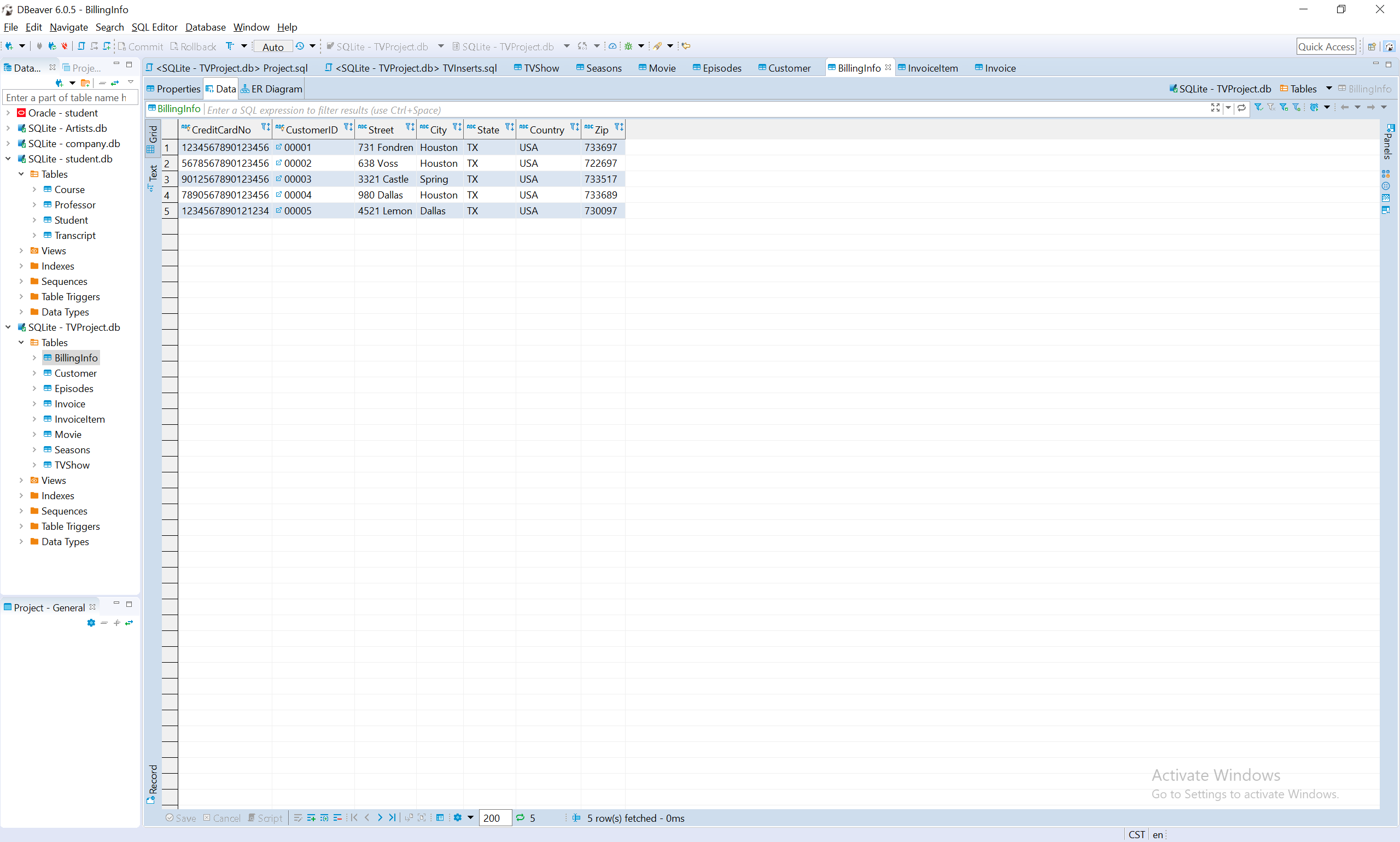


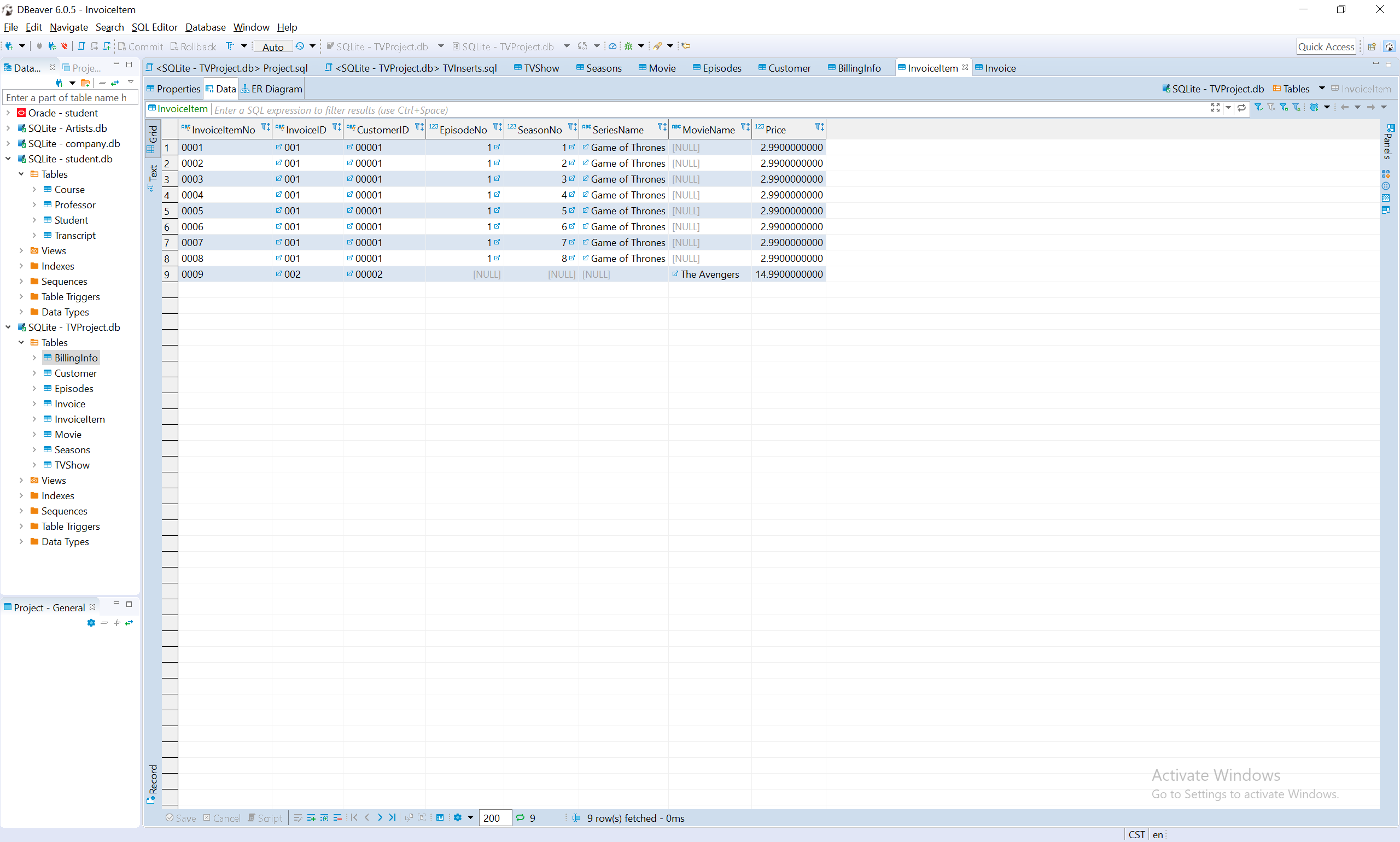
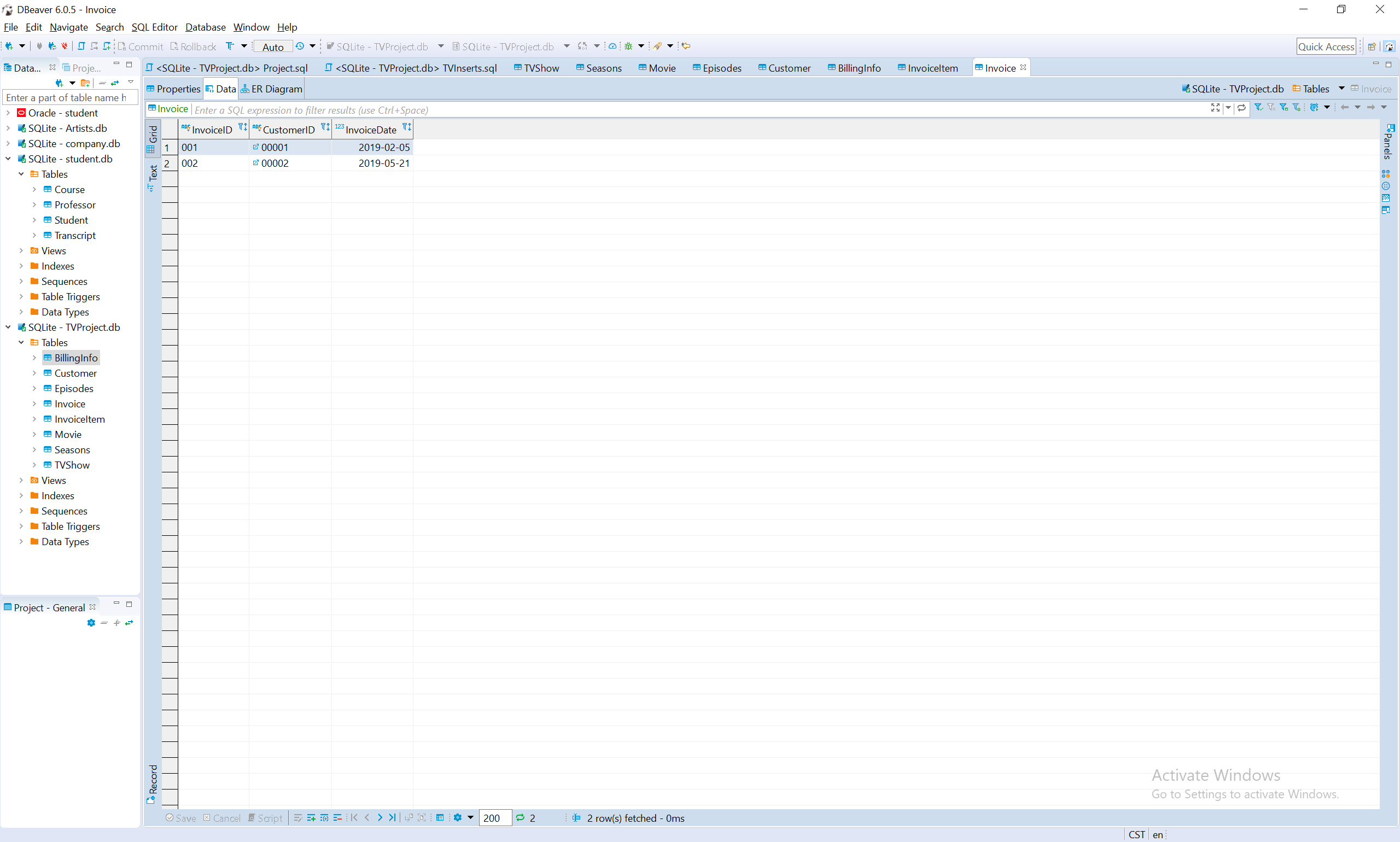




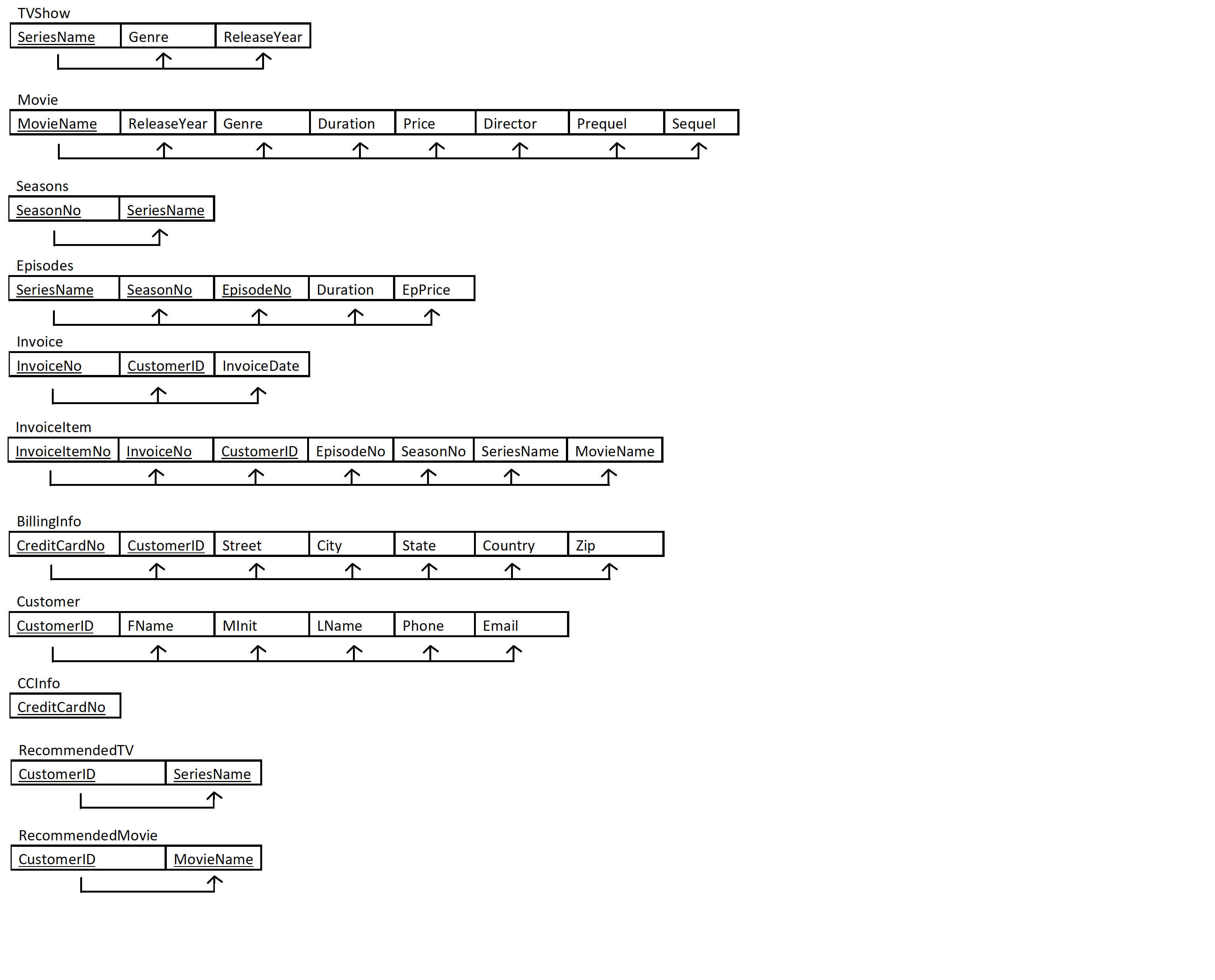
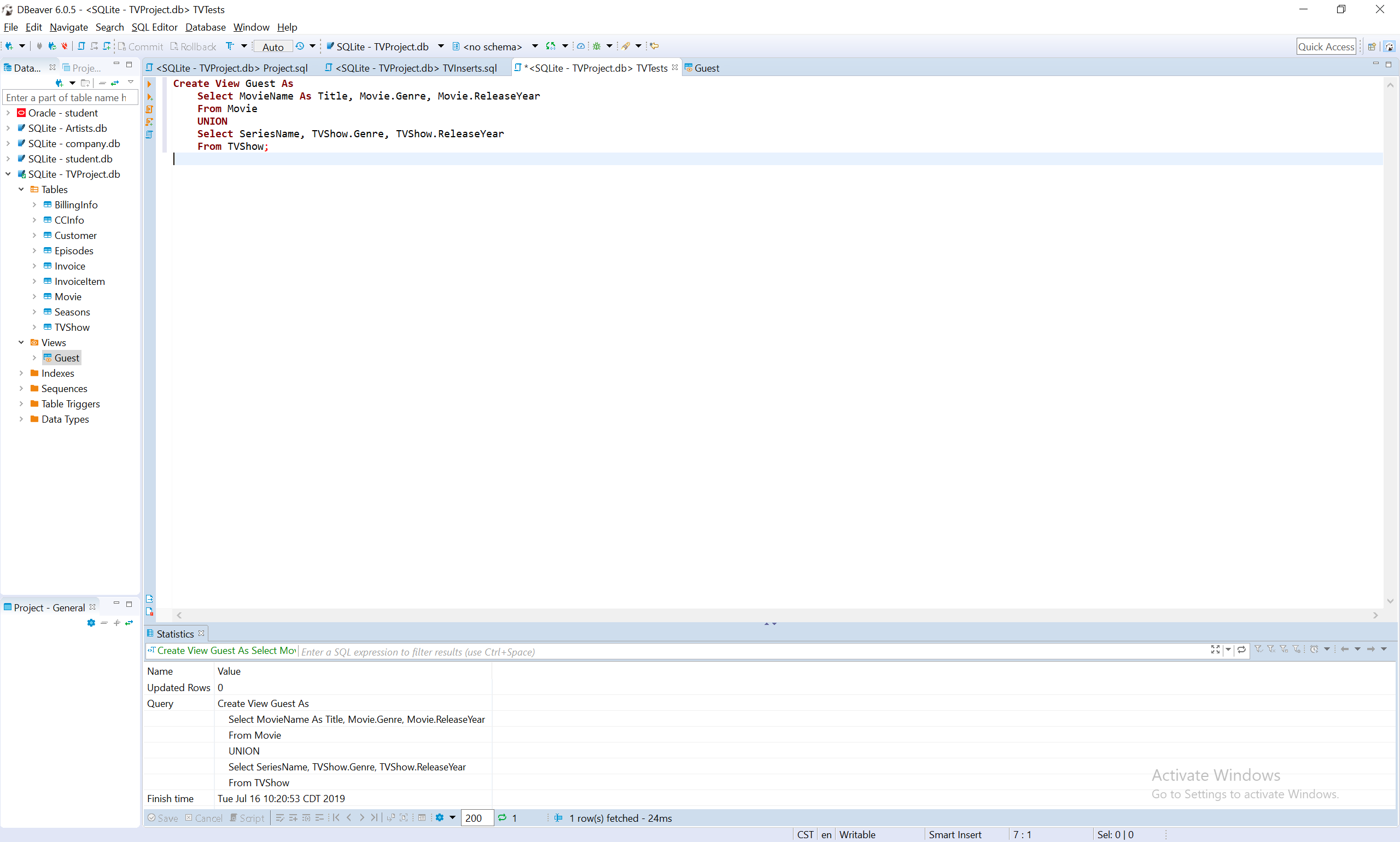


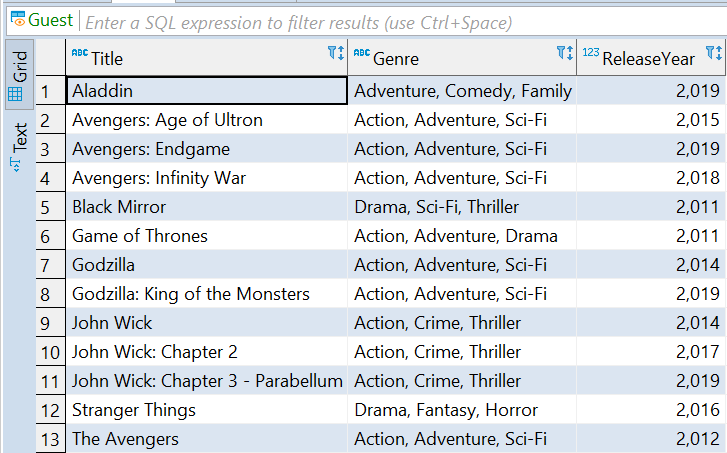






**Deliverable II starts here**

* 3NF Normalized relational diagram
* Dependency Diagram
* View Examples

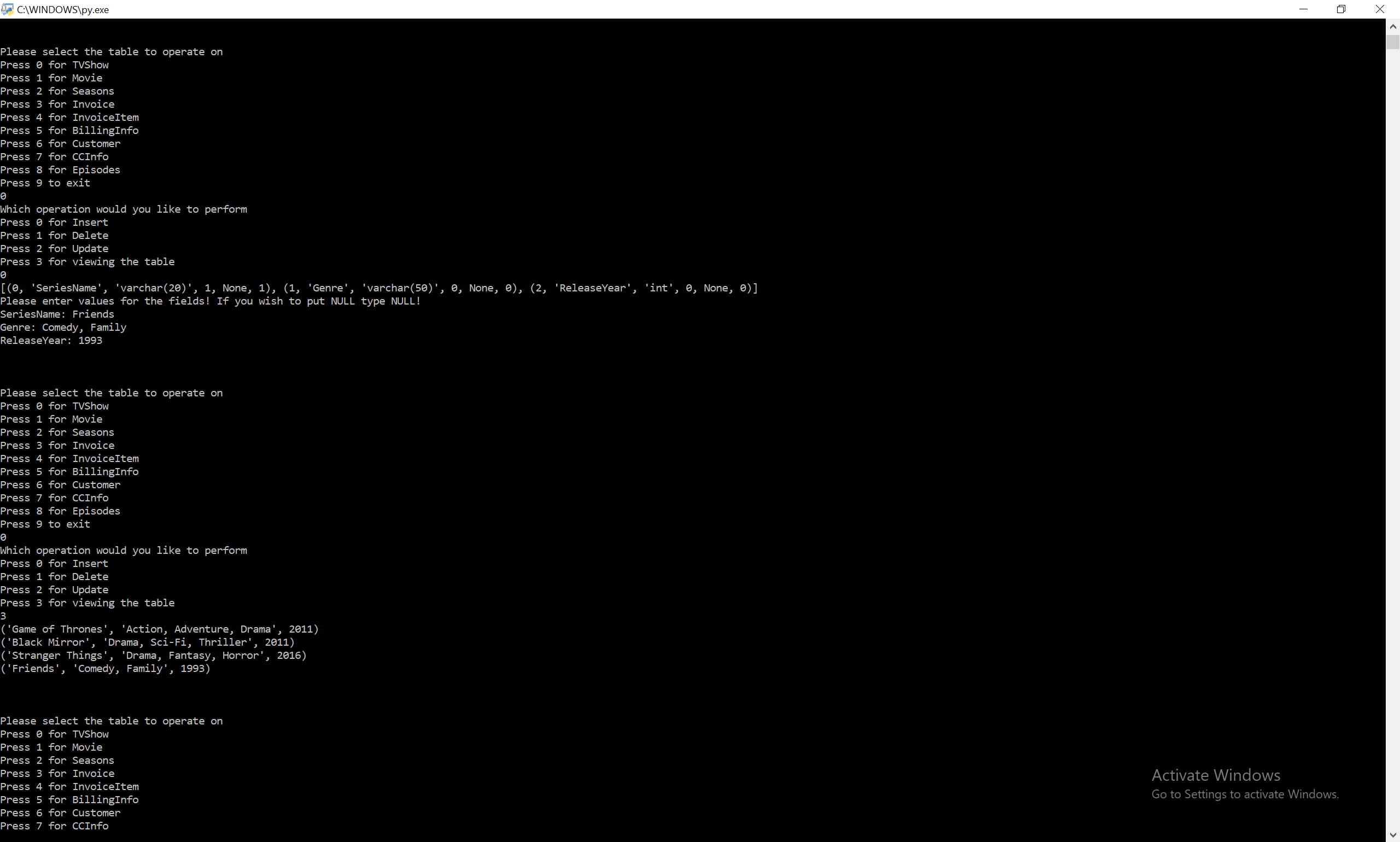


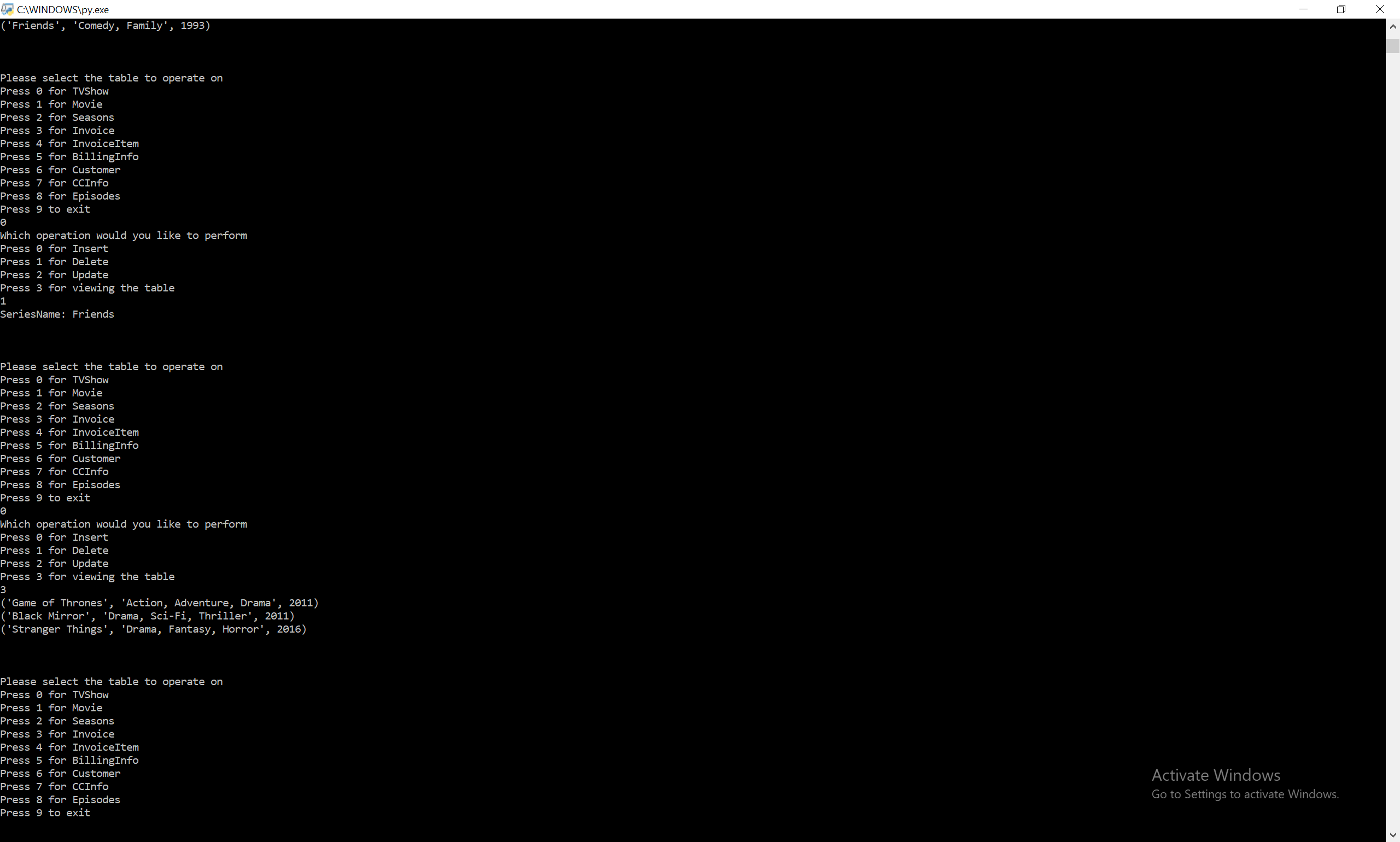
<This view is created for non-registered users to view TV shows and movies.>

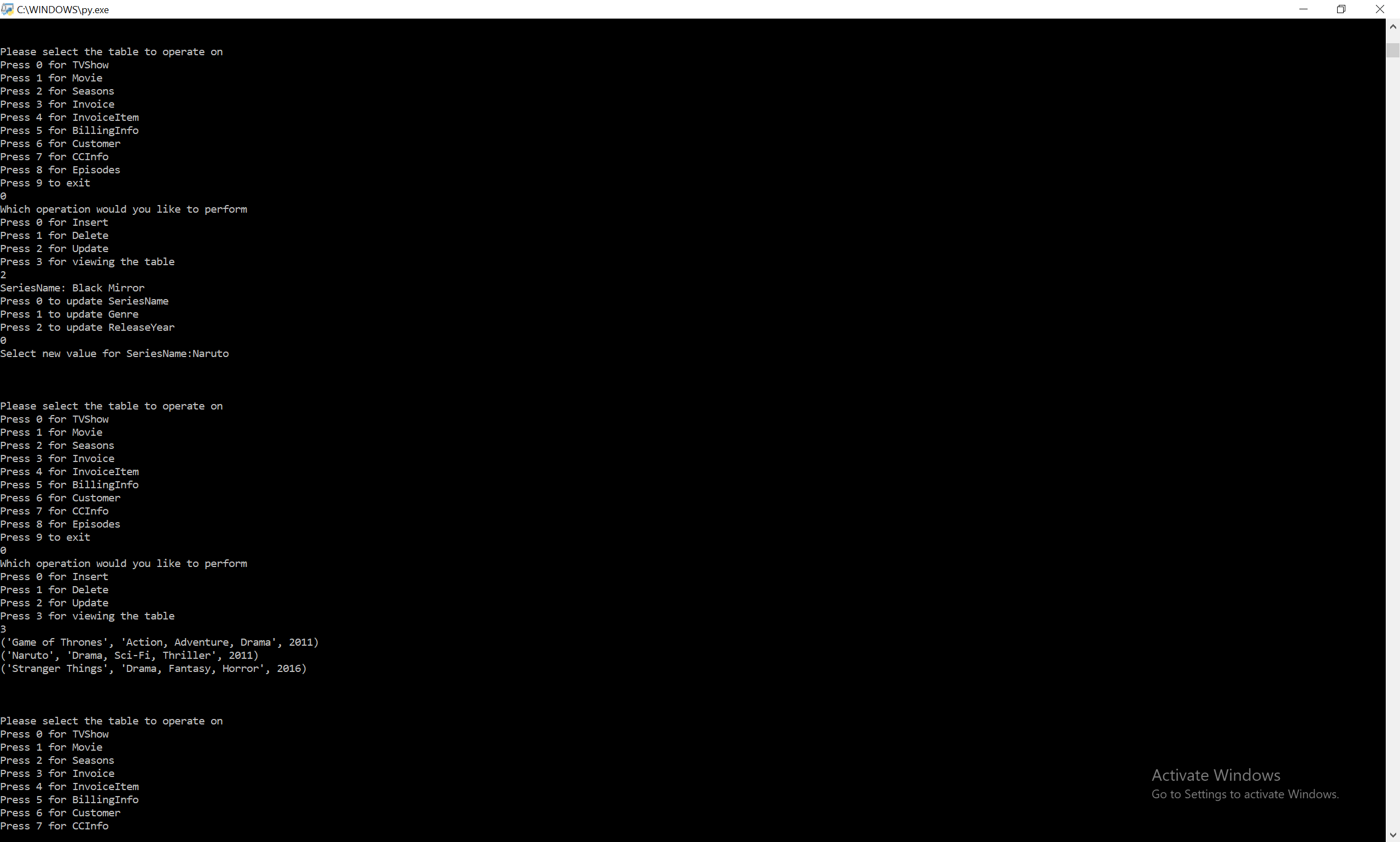
* View TVShow Example:

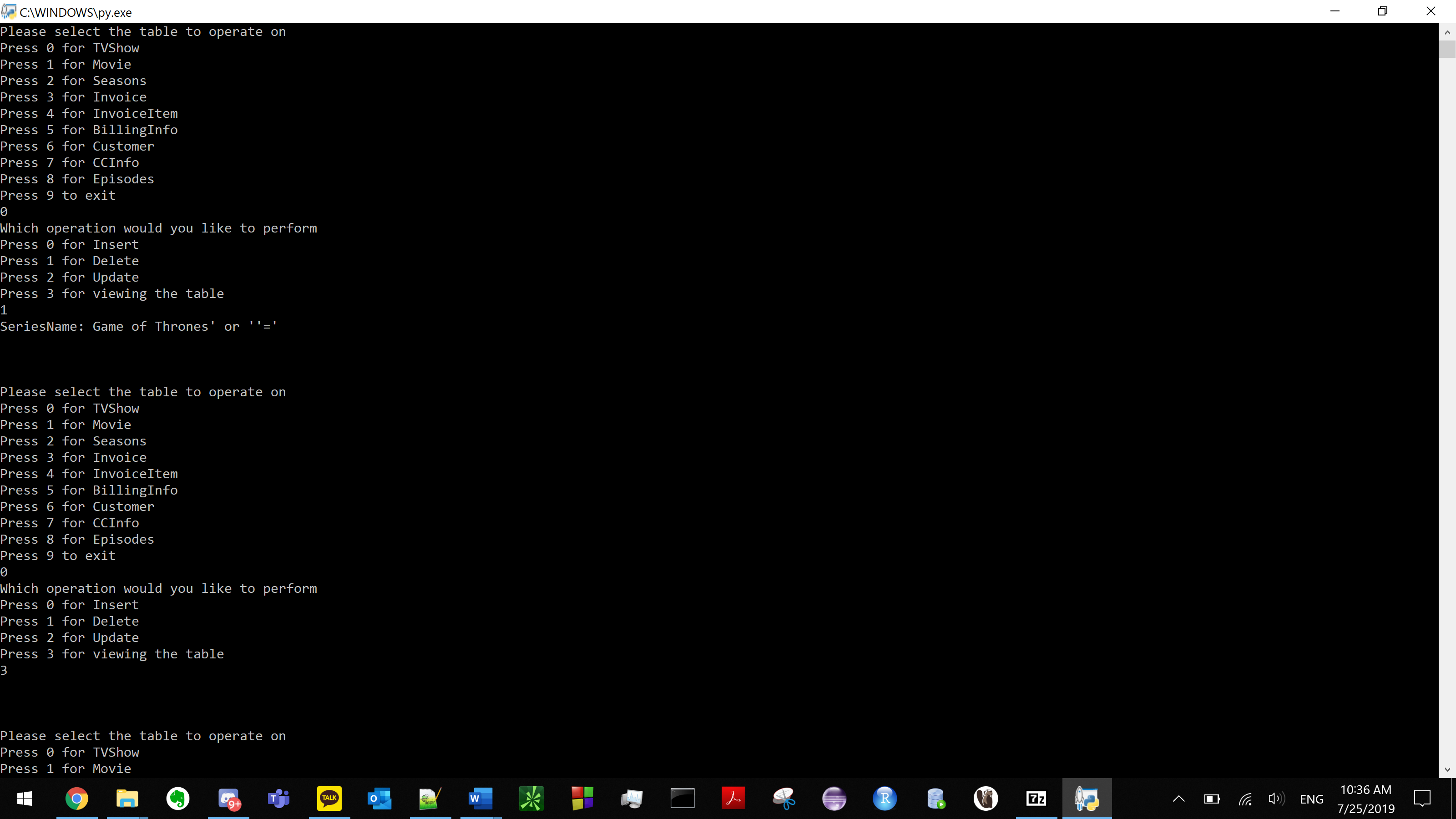


* Insert TVShow Example:



* Delete TVShow Example:
* Update TVShow Example:



* The unique part of our database/interface is that it’s very portable for e-commerce template. The customer invoice tables can easily be moved to multiple different types of databases for e-commerce. Also, our interface program was made intentionally vulnerable to sql injection attacks for learning purposes. For instance, **(MAKE SURE YOU BACKUP THE DB FILE)** after you choose 0 then 1, then enter Game of Thrones’ or ‘’=’ you can see that all the data has been deleted from database. (example below).

Since the empty string always equals the empty string, the deletion operation will apply to all entries in the table. Since deletion cascades all episodes and seasons along with every invoice and recommendation which will destroy the database.

* In order to run our program, it is required to use Python 3.6 or above.
* Our python program is named “project.py” and is included in the zip file.
* We have also included ReadMe.txt and the .db file in the zip file.