

Module Code: ISAD253SL

Module Name: Databases

Coursework Title: Coursework 2020 - 2021

Deadline Date: 04 January 2020

Member of staff responsible for coursework: Mr. Naji Saravanabavan

Programme: BSc(Hons) So4ware Engineering

Please note that University Academic RegulaVons are available under Rules and RegulaVons on the University

website

[k](http://www.plymouth.ac.uk/studenthandbook)

[www.plymouth.ac.uk/studenthandboo](http://www.plymouth.ac.uk/studenthandbook)

[.](http://www.plymouth.ac.uk/studenthandbook)

Group work: please list all names of all parVcipants formally associated with this work and state whether the work

was undertaken alone or as part of a team. Please note you may be required to idenVfy individual responsibility for

component parts.

P.K.Dilshan 10707182

R.M.S.M. Rathnayake 10707350

S.T.R. Fernando 10707200

S.M.D. Nawarathna 10707290

T.G.S. Avishka 18328 (NSBM ID)

***We confirm that we have read and understood the Plymouth University regulations relating to***

***Assessment Offences and that we are aware of the possible penalties for any breach of these***

***regulations. We confirm that this is the independent work of the group.***

Signed on behalf of the group: (P.K.Dilshan)

Individual assignment:

***I confirm that I have read and understood the Plymouth University regula7ons***

***rela7ng to Assessment Offences and that I am aware of the possible penal7es for any breach of these***

***regula7ons. I confirm that this is my own independent work.***

Signed :

Name: P.K.Dilshan

Student Reference Number: 10707182





**Databases – ISAD253SL**

**Coursework 2020–2021**

**PUBLIC LIBRARY SERVICE DBMS**

Group Name: Team 88

**Members**

ID

Name

Degree Program (SE/

CN/CS)

10707182

P.K.Dilshan

SE

10707350

R.M.S.M. Rathnayake

SE

10707200

S.T.R. Fernando

SE

10707290

S.M.D. Nawarathna

SE

18328(

NSBM ID

)

T.G.S. Avishka

SE

# DEFAULT SERVICE LOGINS

**Library Location** : Negombo Public Library

**6 Digit Key** : 123456

**Note** : More unlimited number of operators can be added via database : Login\_Details

**Registered students**

**Student ID**

**Student Password**

Colombo Public Library

906888

Negombo Public Library

123456

Kandy Public Library

980823

**INTRODUCTION TO PUBLIC LIBRARY SERVICE DBMS**

Public Library Service DBMS is a library service where only authorized library staff members can login, view and manage databases where all the Library, Employee, Book, Location, Copy , Borrower details and so much more are available in one place to review and also to manage if needed. This is a stand-alone Java Application that provides simple log in and view all the details of any library within the public library network.

Authorized library staff members like Librarians can login and manage database tables by viewing, updating and deleting, but the login privileges are bit under controlled to limit the access to other authorized members in the library network who works under Librarian designation like library assistant so on. They can only view any details available regarding all the libraries within the public library network.

This Public Library Service Database Management System is a user friendly Java application where you can simply interact with at a glance. This DBMS application provide one more feature other than viewing, updating and deleting records, that is generating full analytic reports which leads to take future decision whenever if you needed.

When deleting records from the database, it won’t be deleted permanently. At the time of record deletion, the deleted records will be added to another table. And also when inserting a new borrower to the database, any borrower who may not borrowed any books within 1 year period of time will be deleted from the database at the time of inserting a new borrower.

# C:\Users\Administrator\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Library Service - ER Diagram.pngENTITY RELATION DIAGRAM

**Note** : If the ER diagram is not clear, you can refer the PNG Image I have given in the

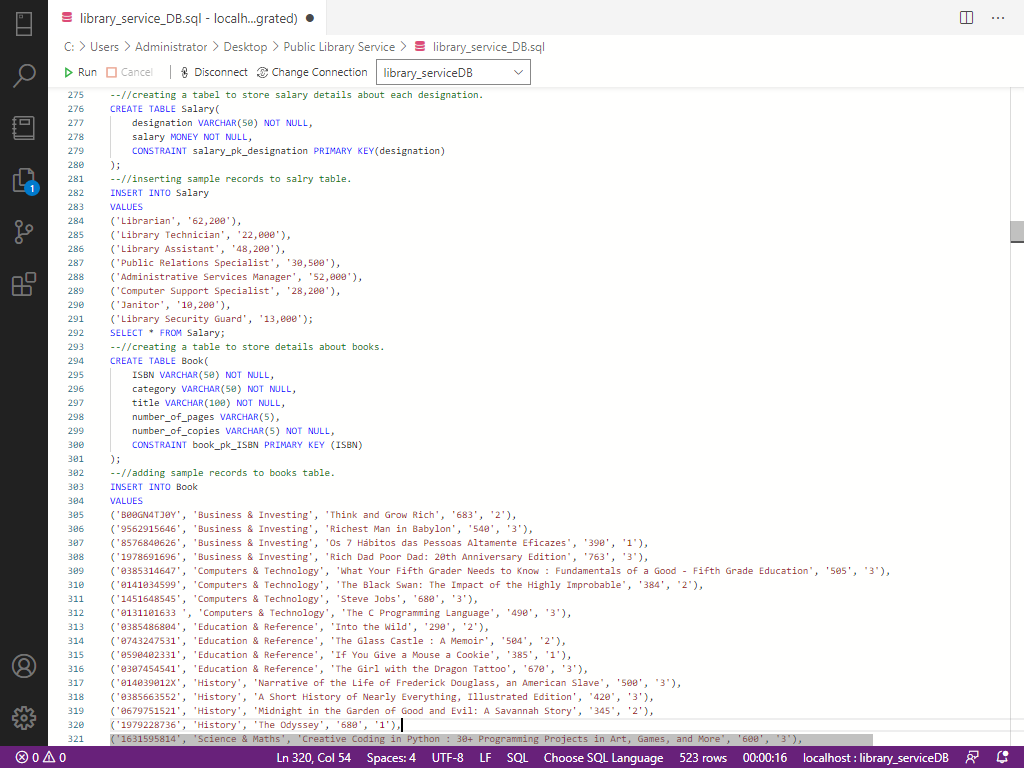
‘Report folder’. All the assumptions are listed in the image as well. Thank you.

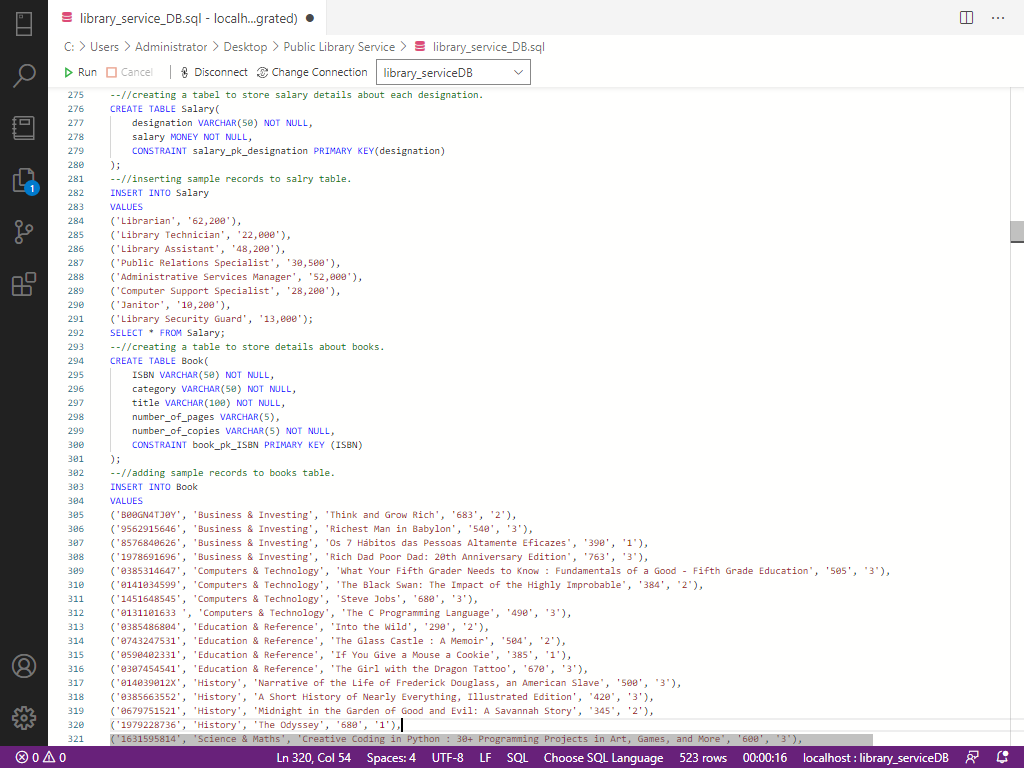
# RELATIONAL MAPPING

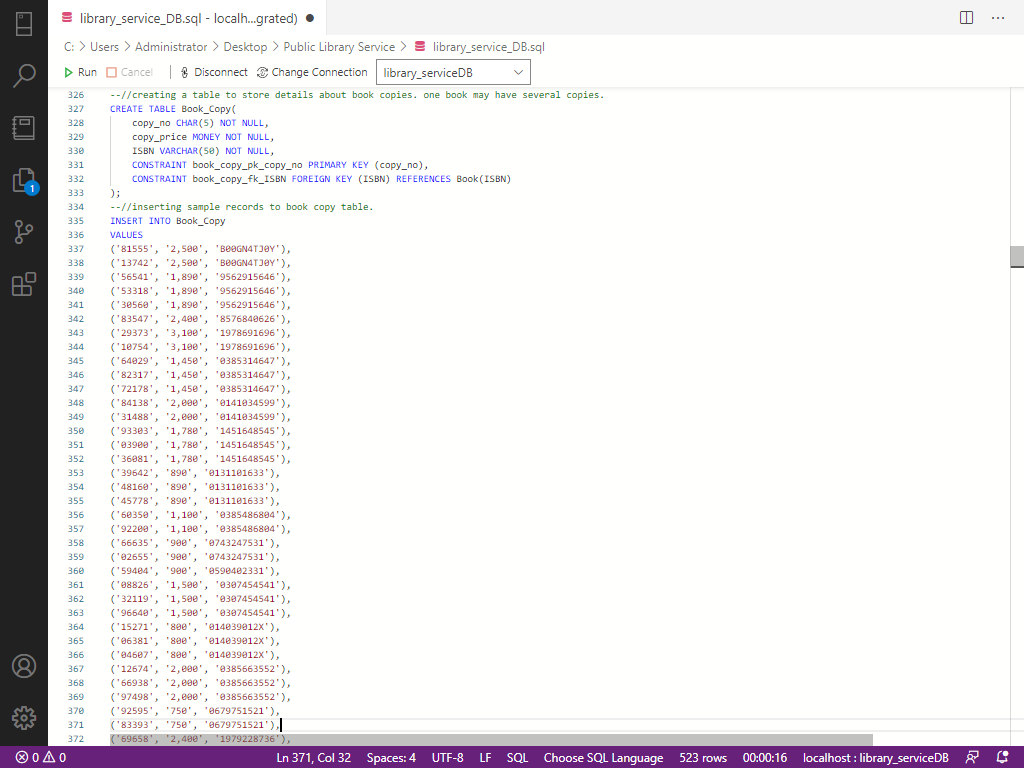


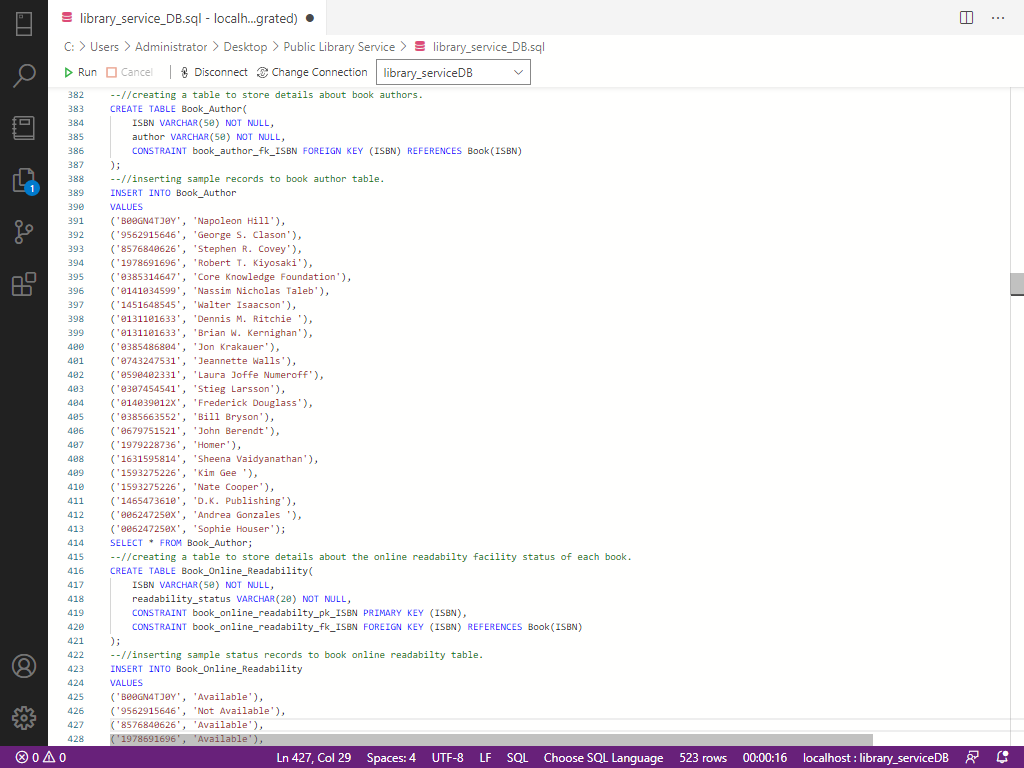
# DATABASE CREATION

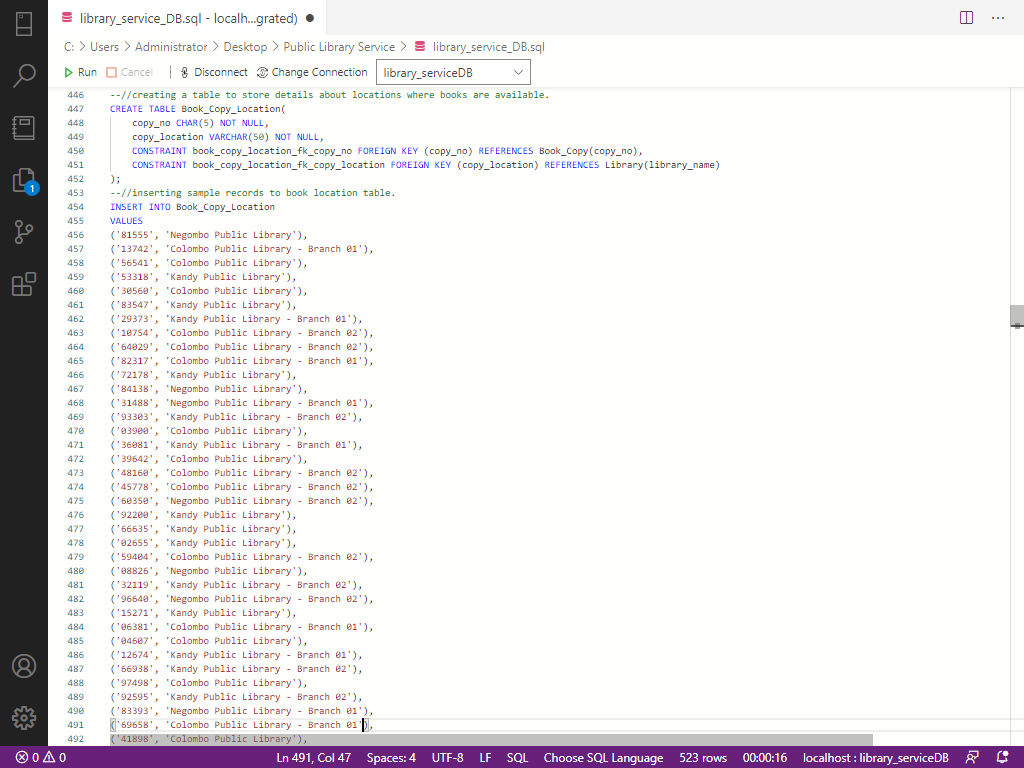
# CREATE TABLE STATEMENTS

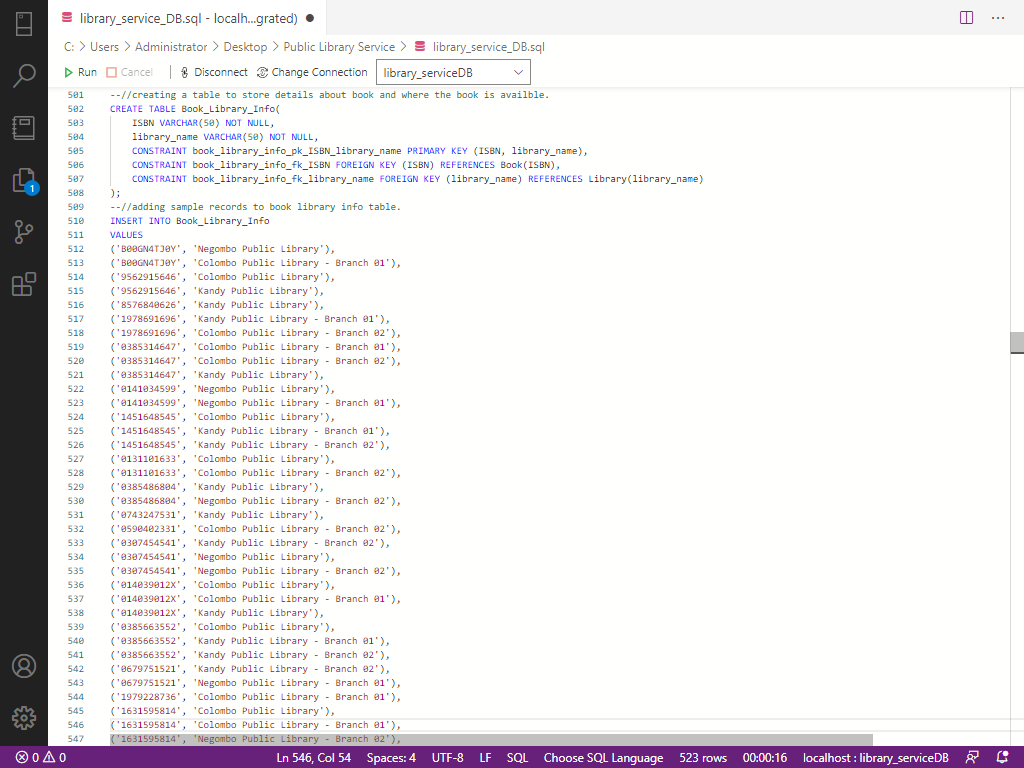


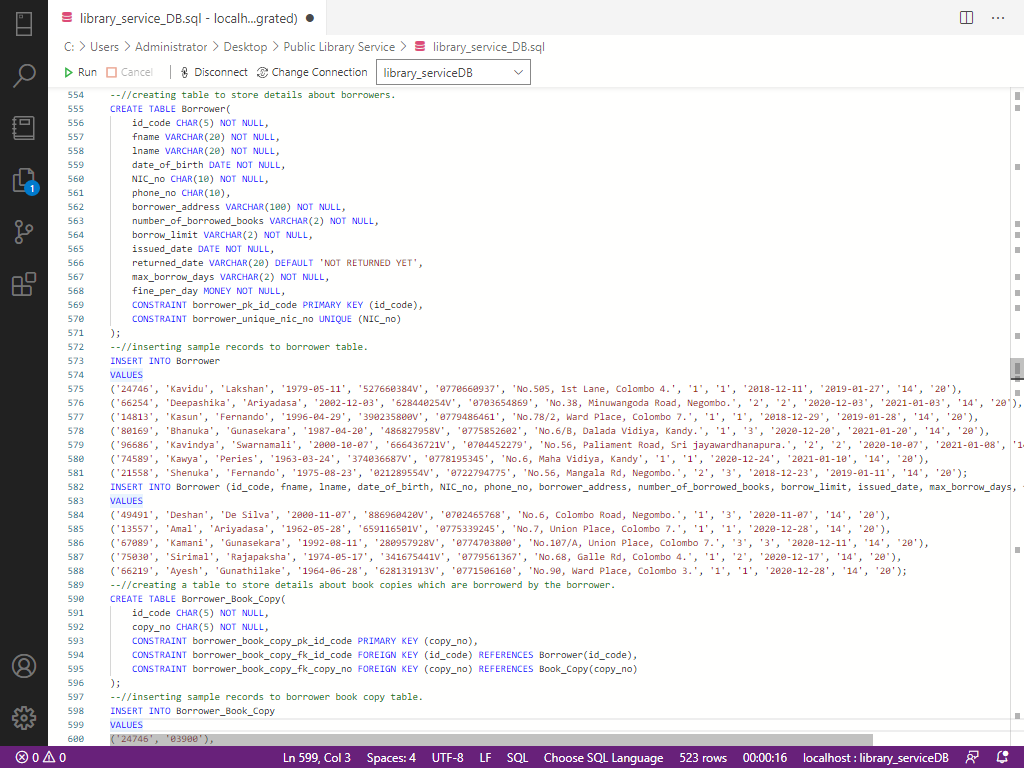


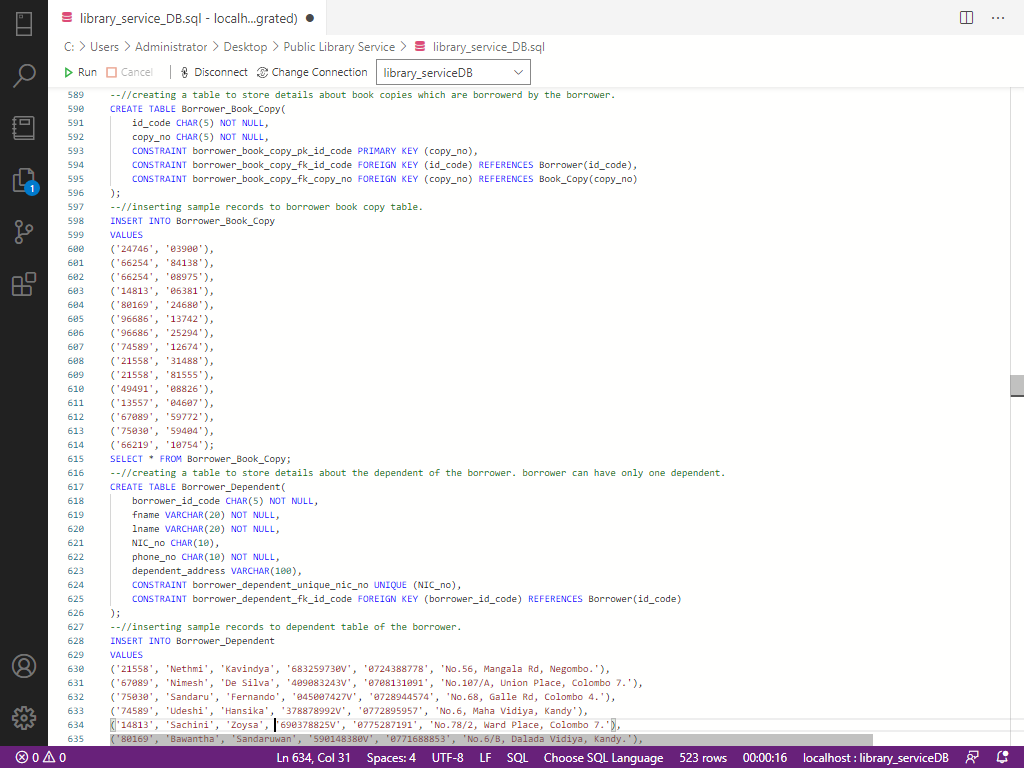




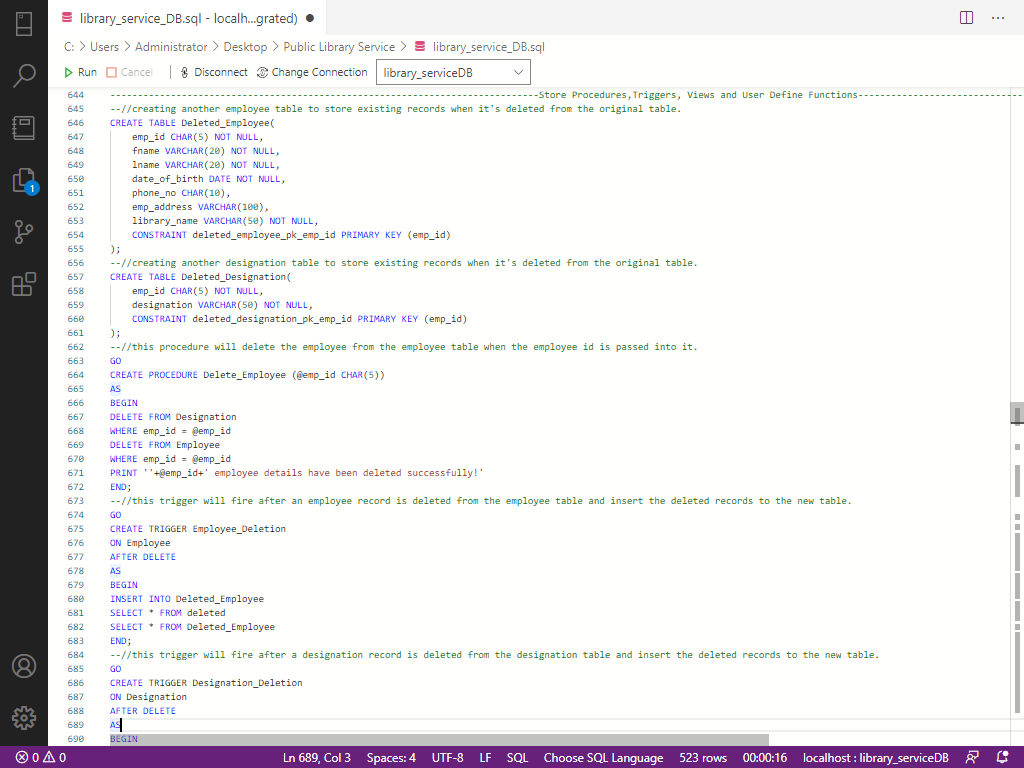


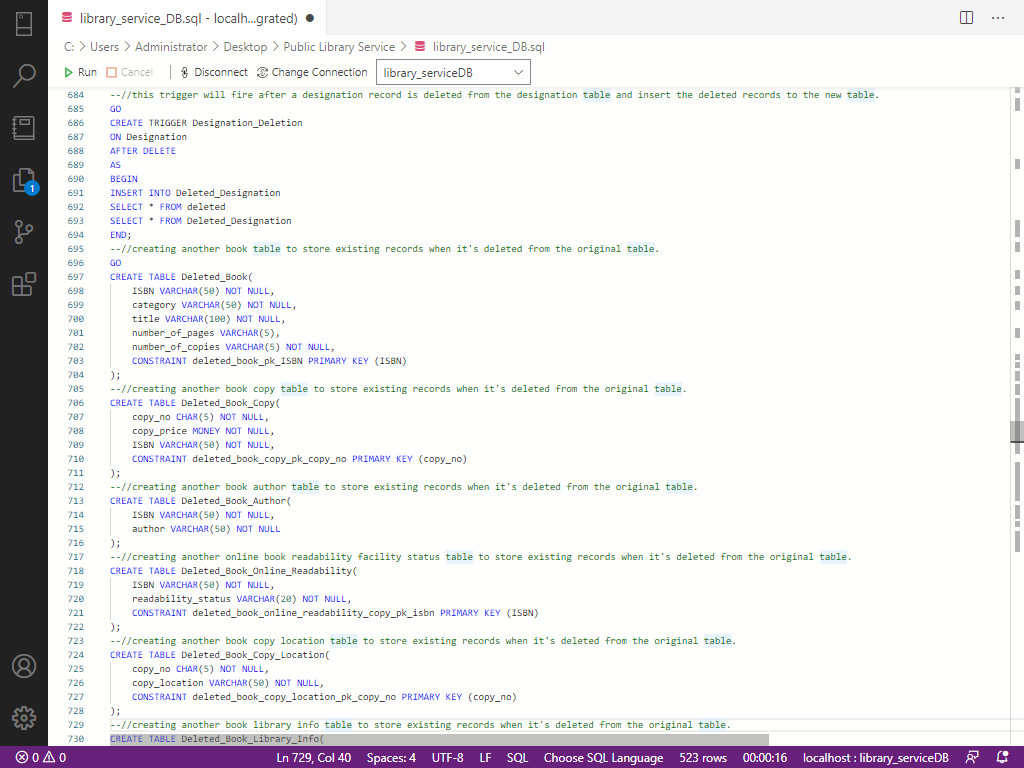


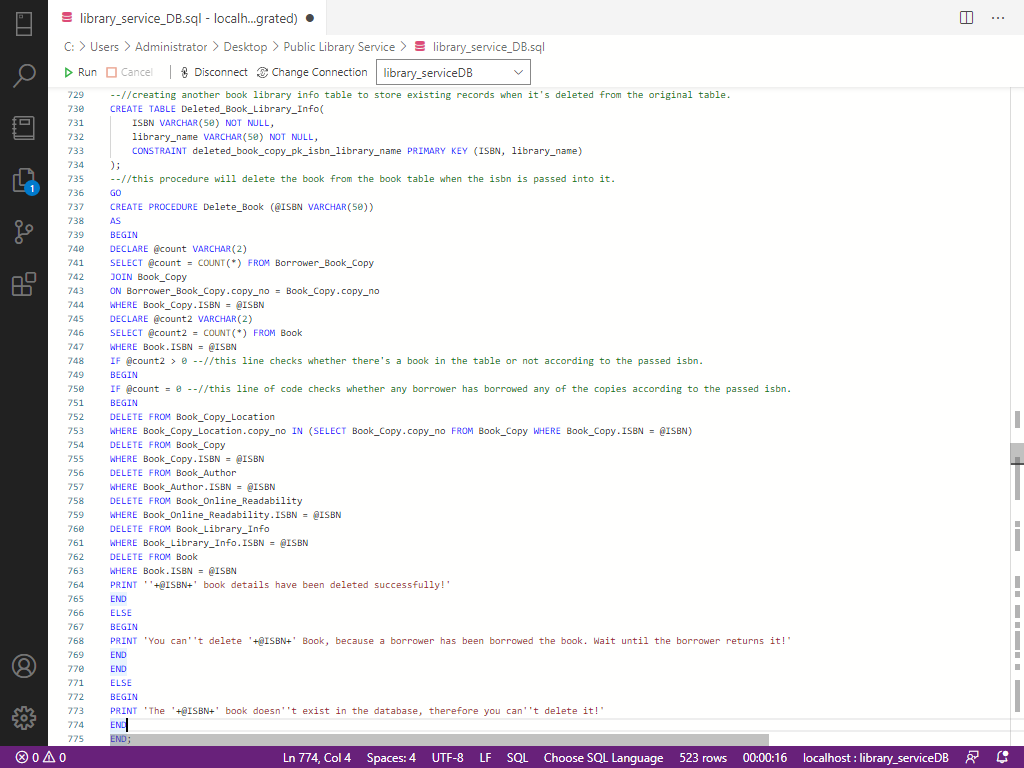


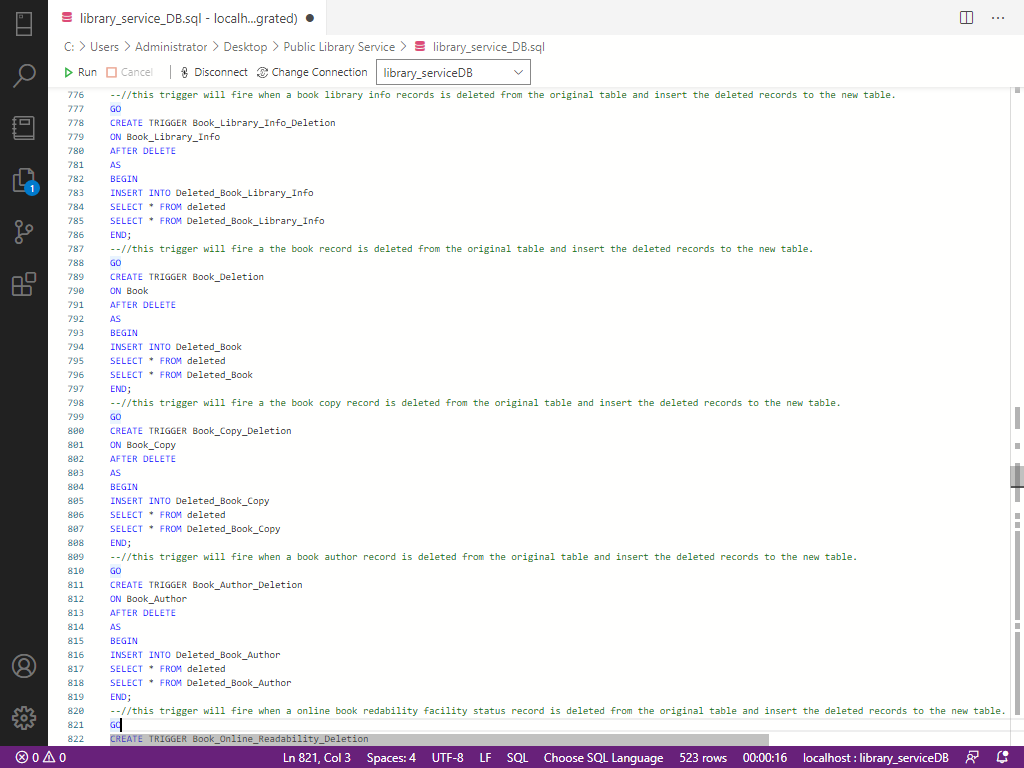


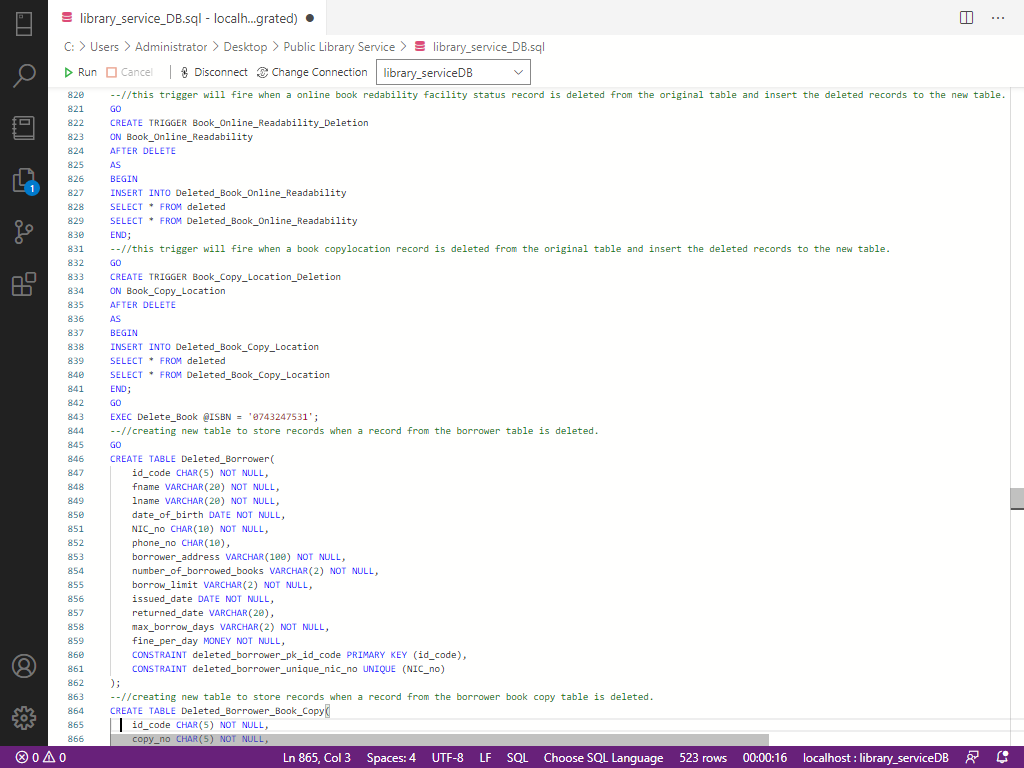
**STORE PROCEDURES AND TRIGGERS**

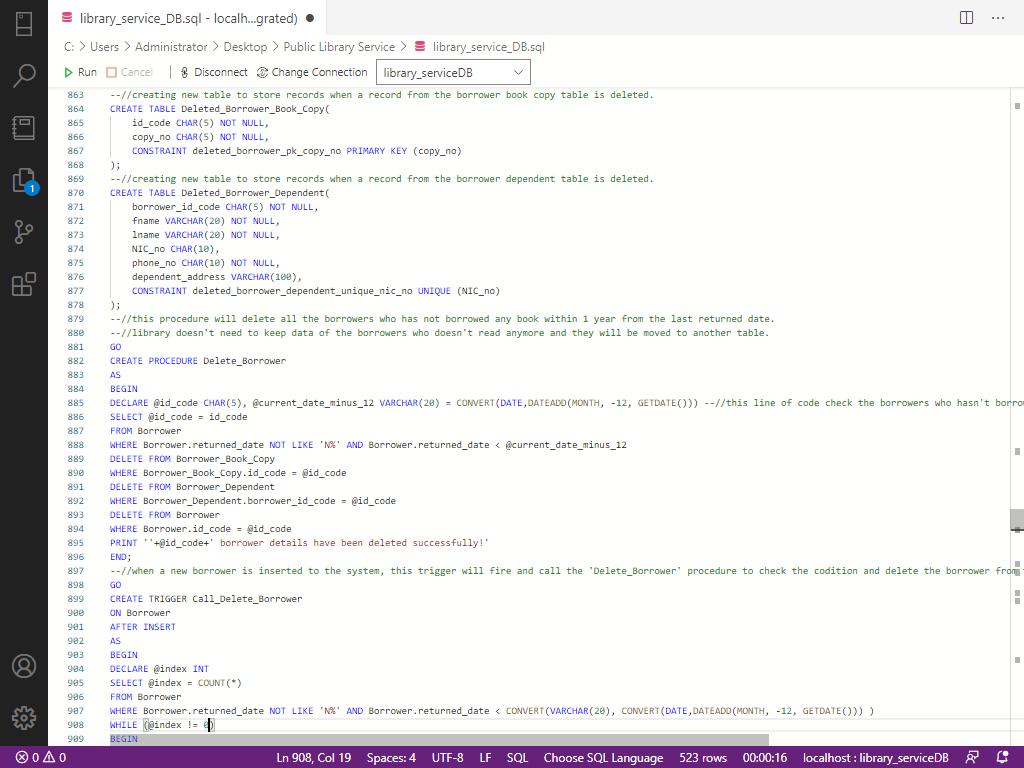


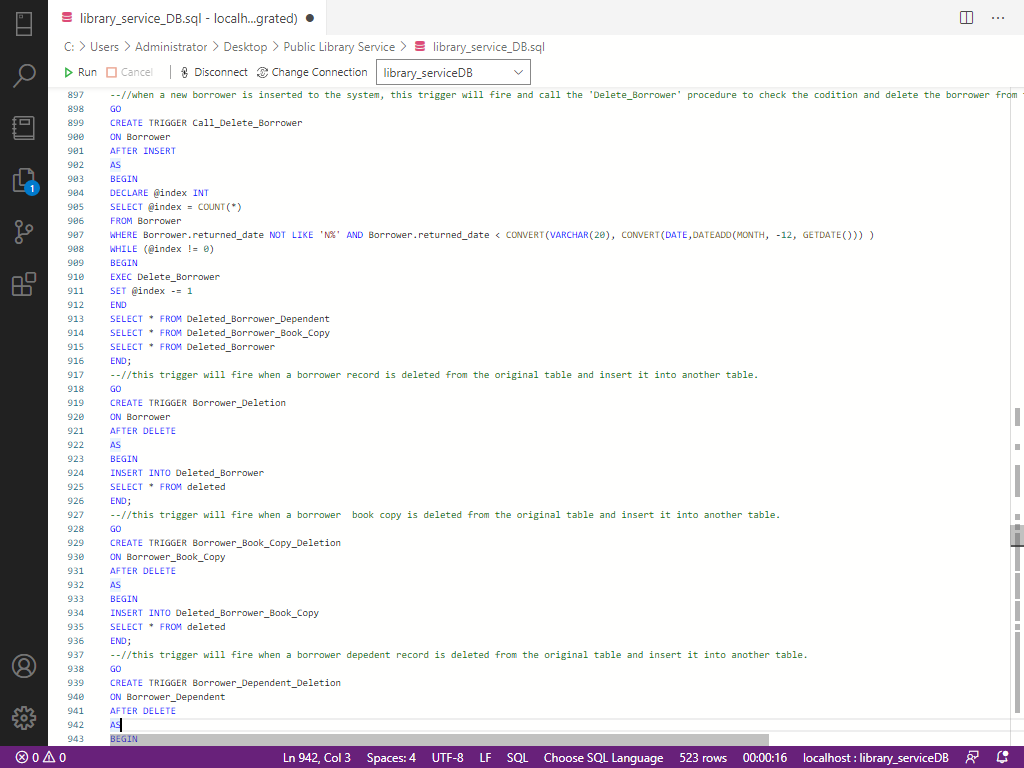




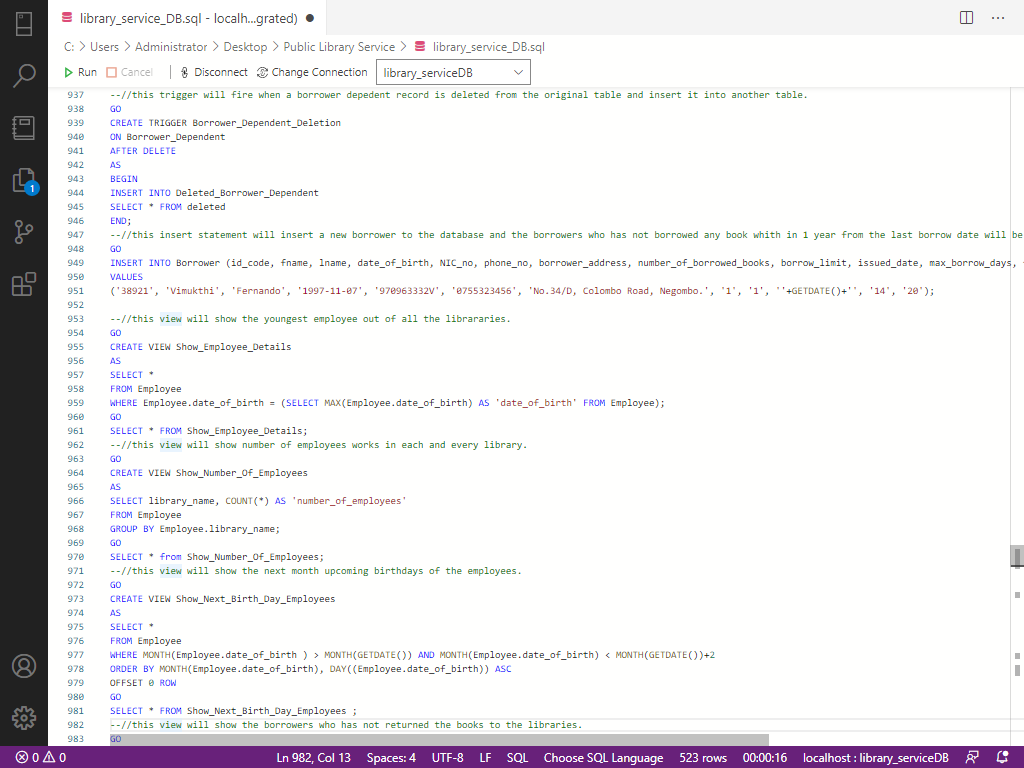


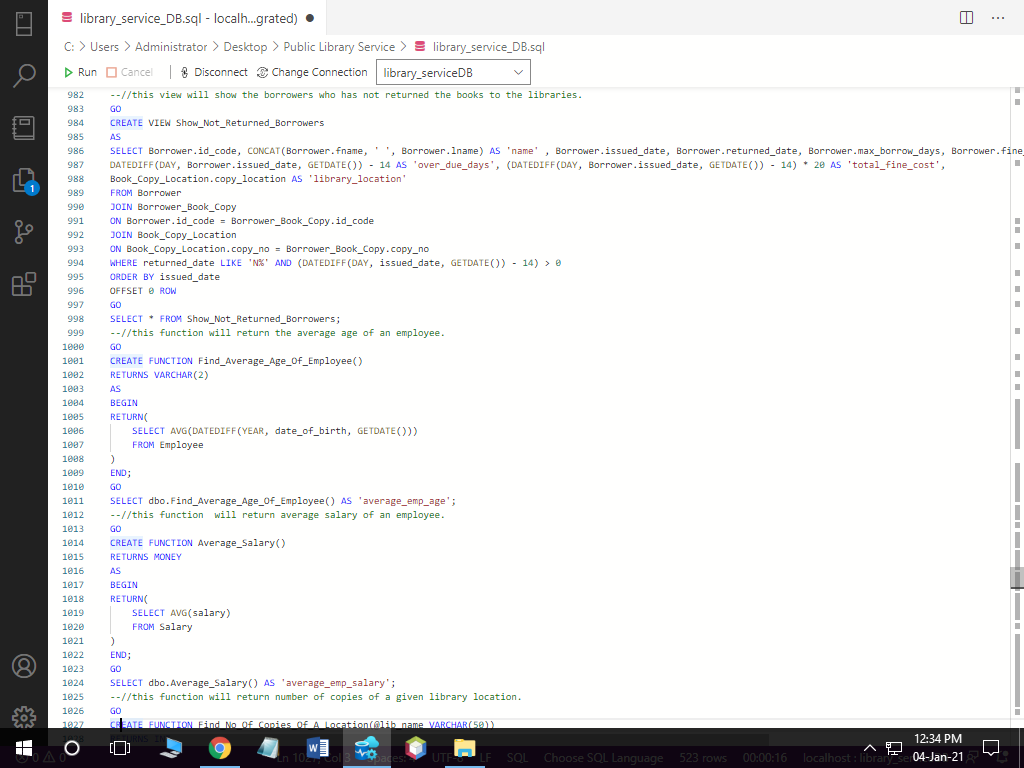


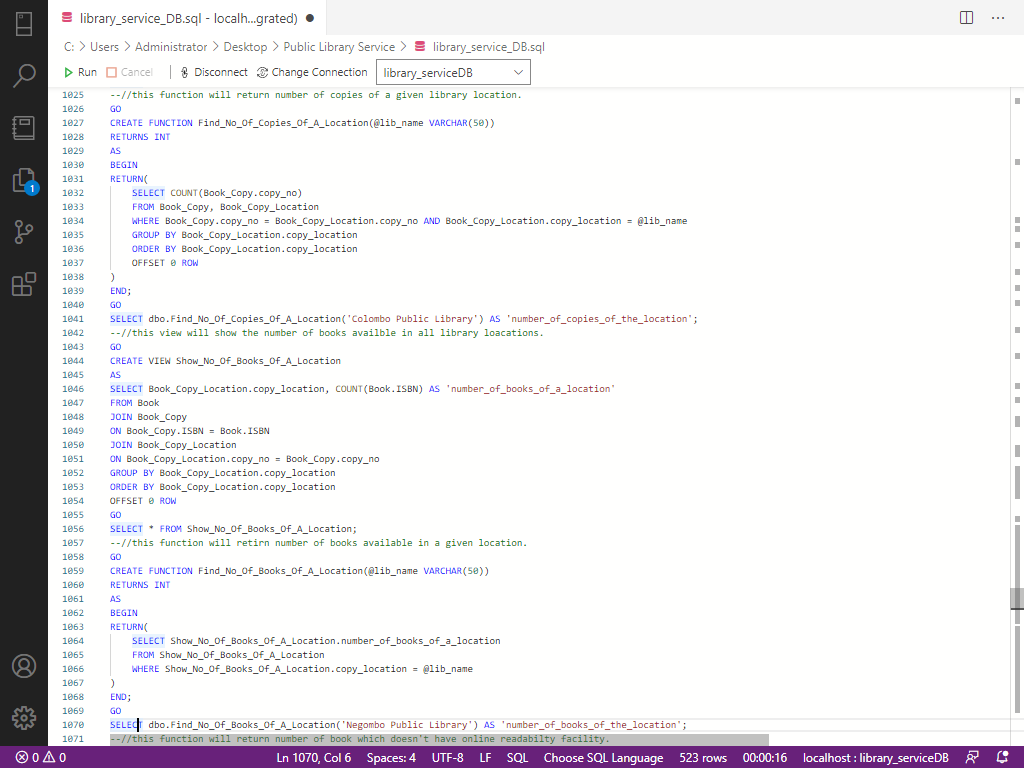


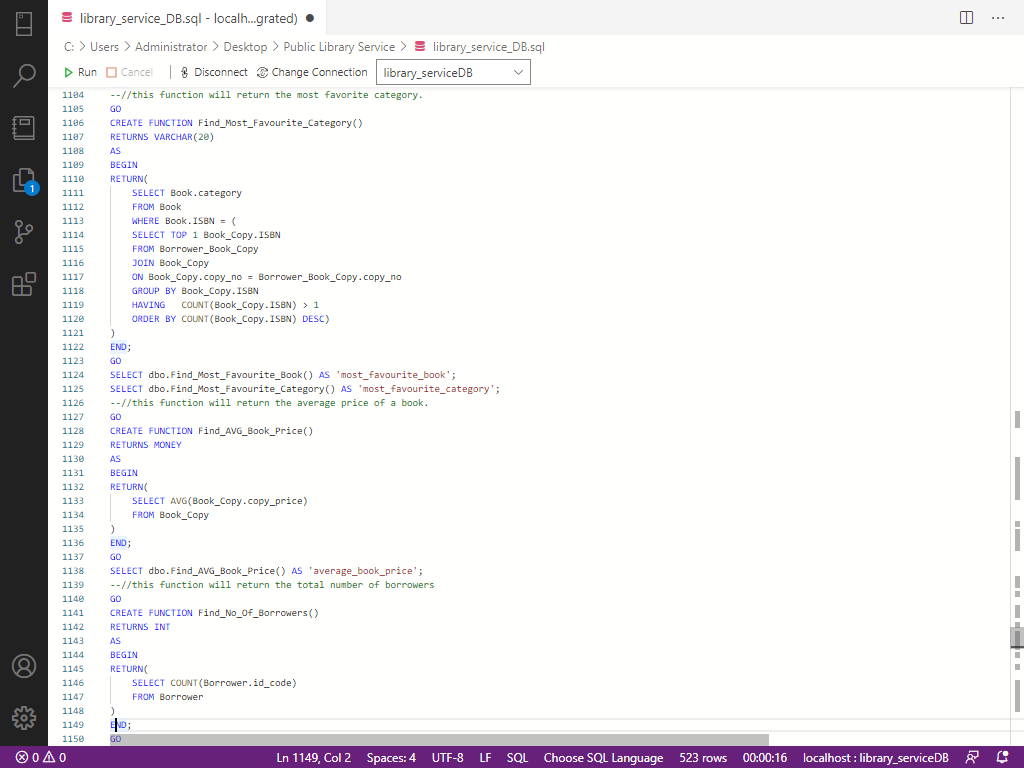


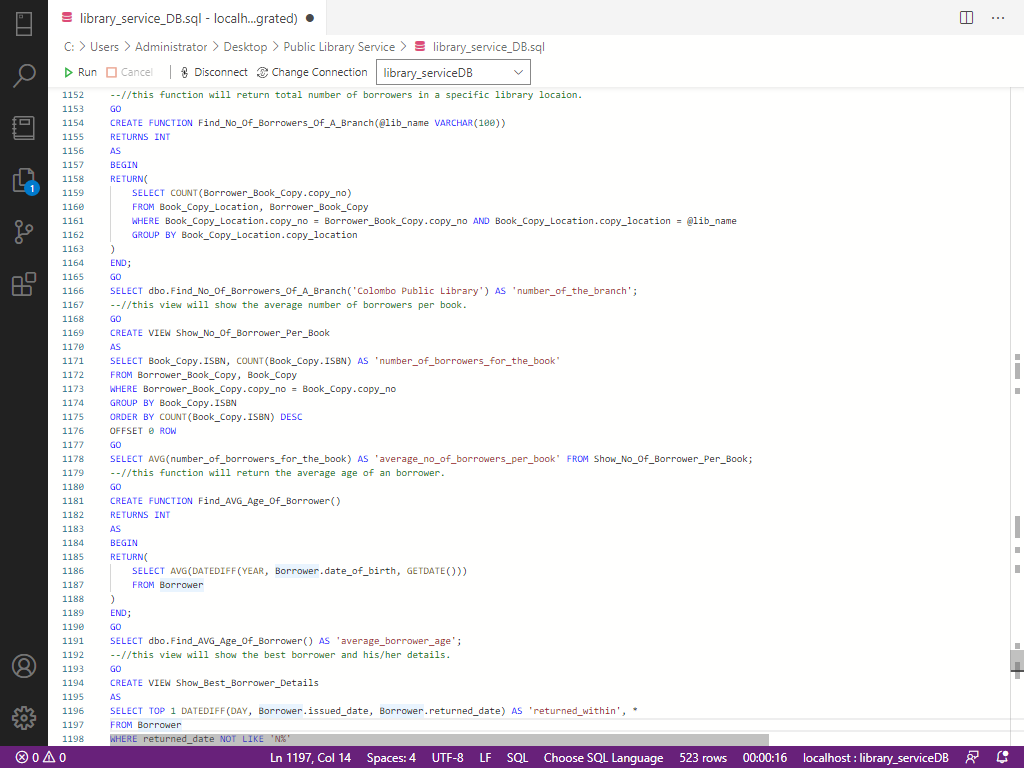
# USER DEFINED FUNCTIONS (UDF)

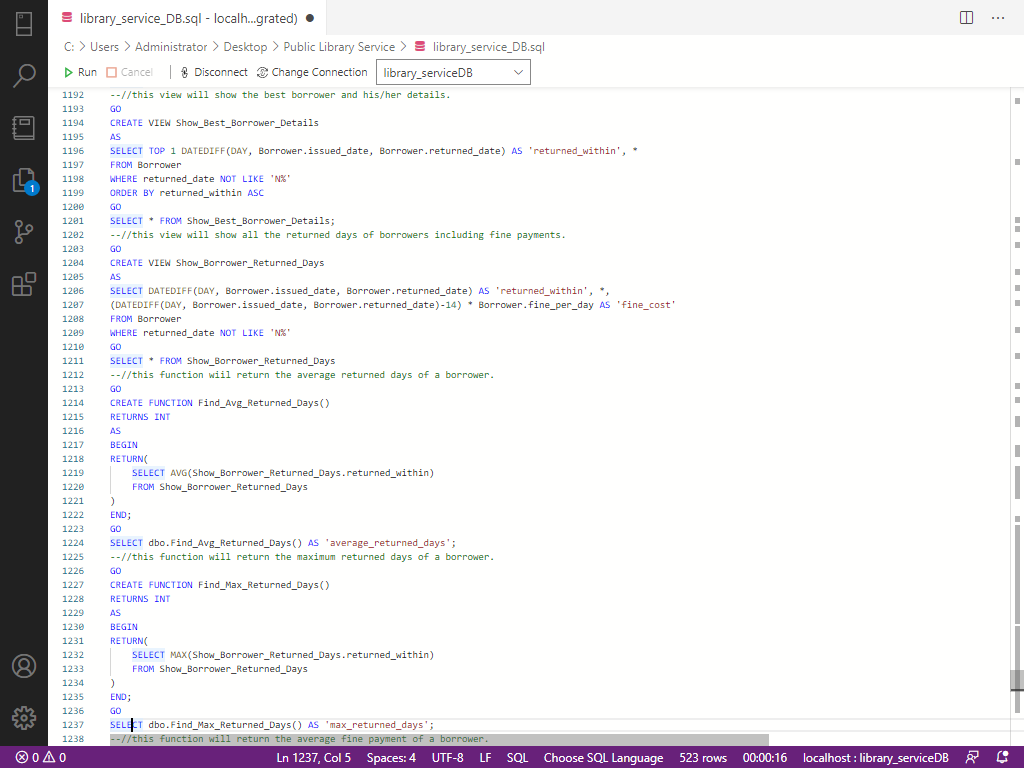


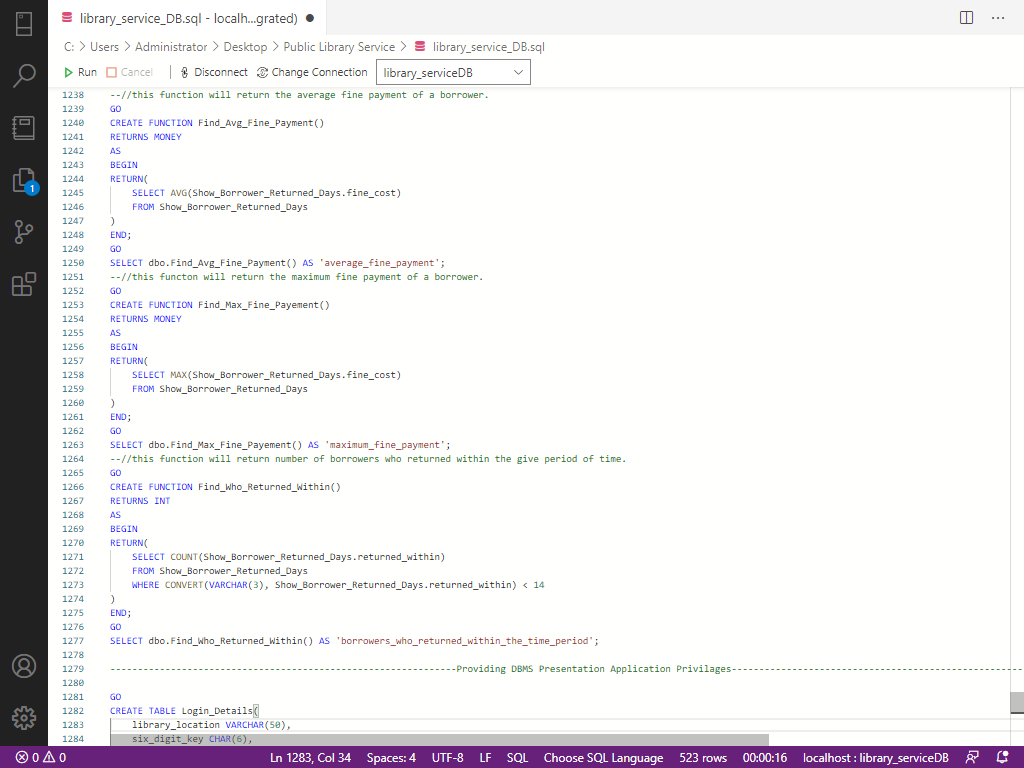








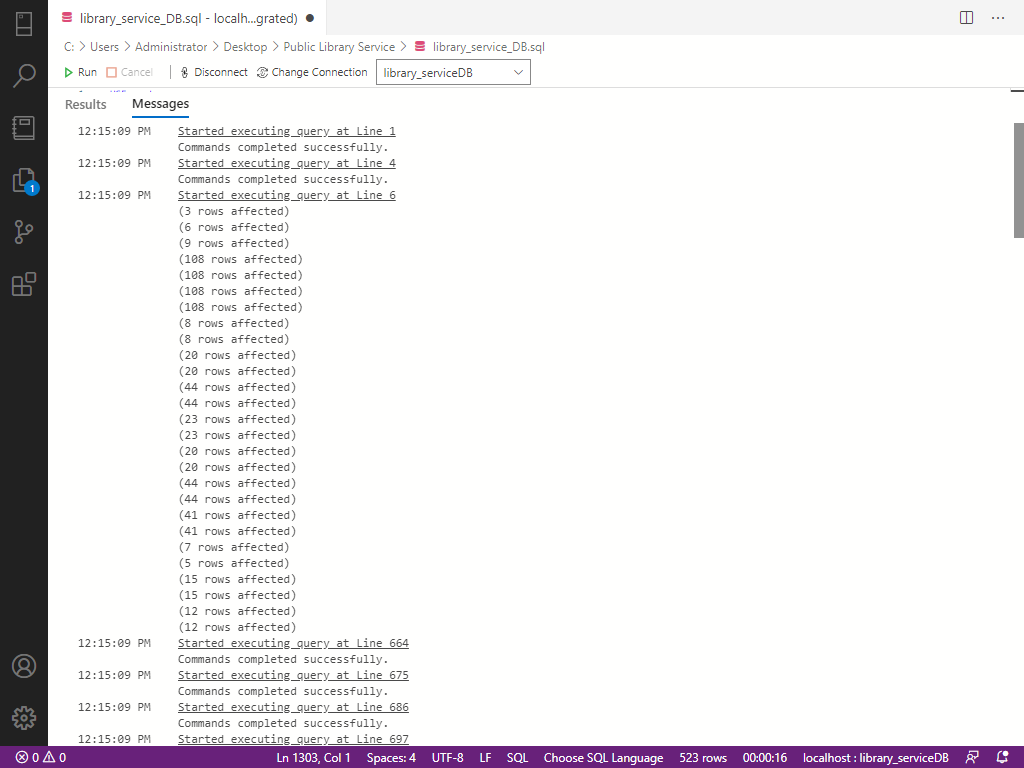


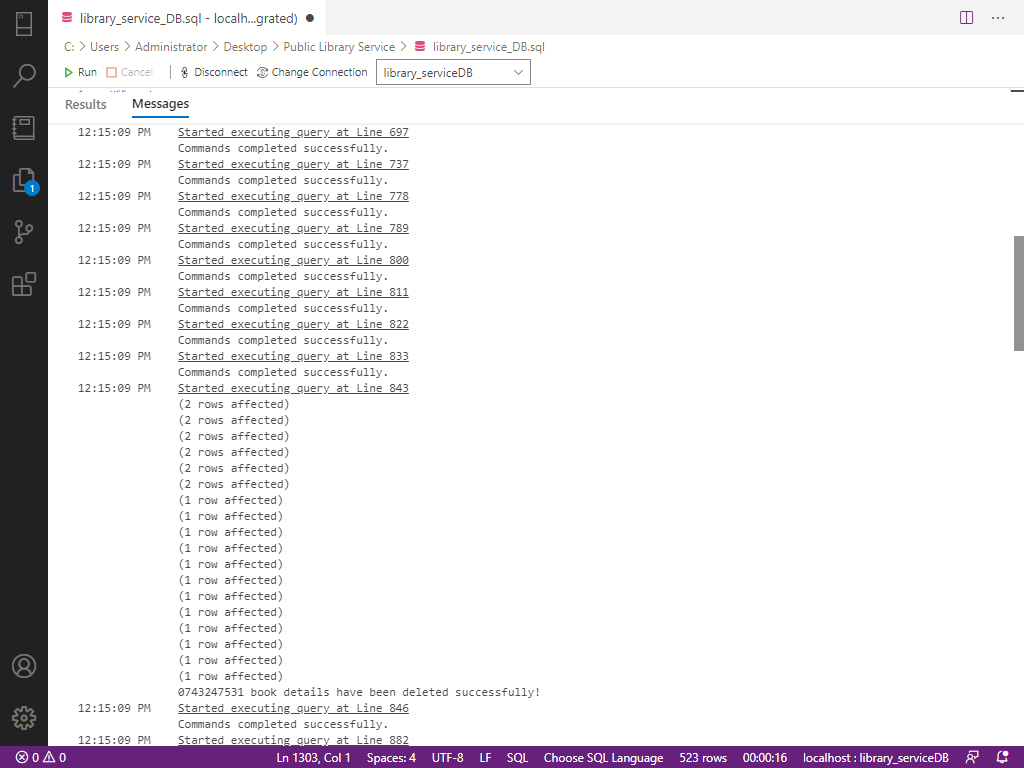


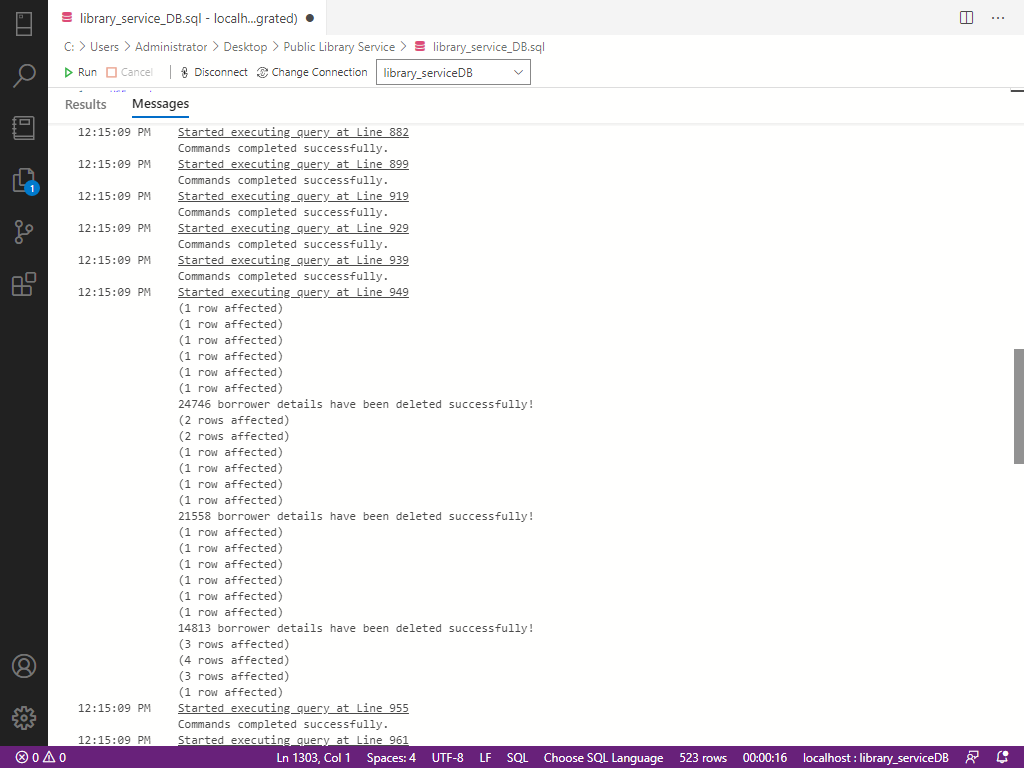
**TABLE CREATION FOR APPLIATION LOGINS**

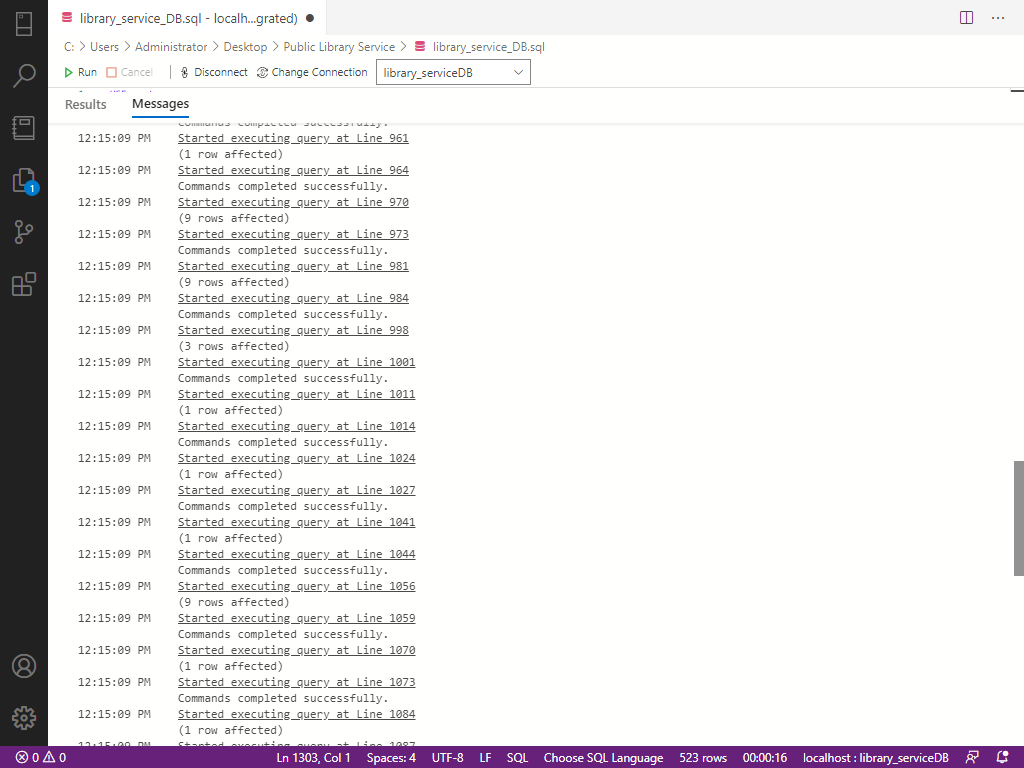


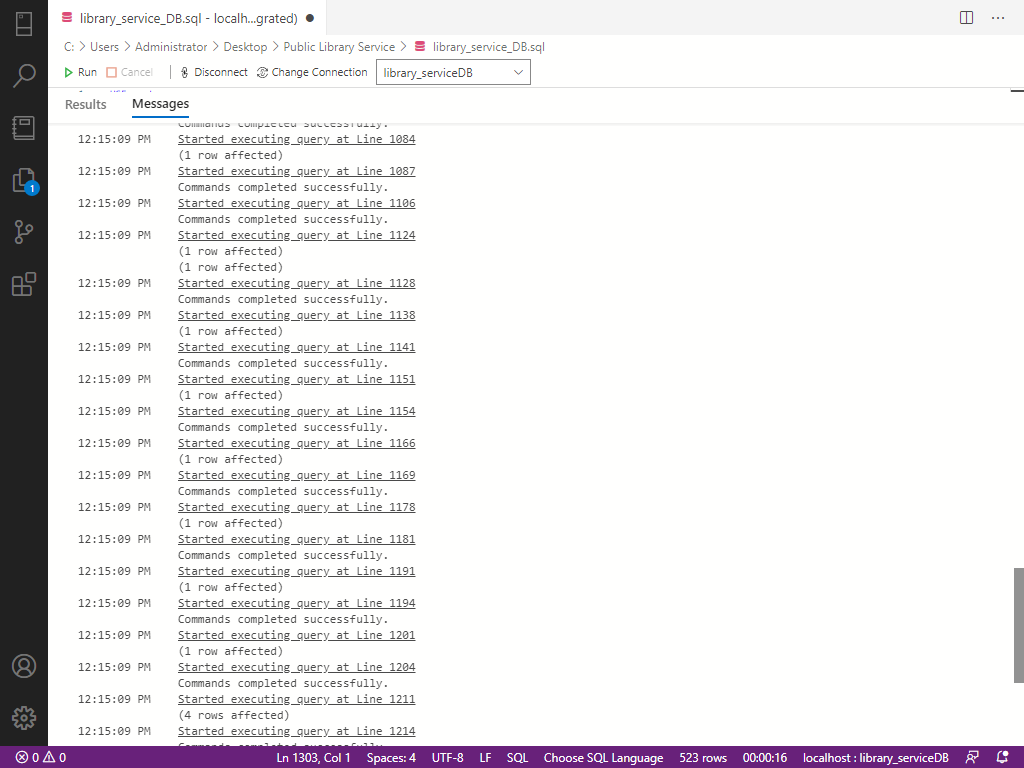
**SUCCESSFULLY EXCECUTED RESULTS**

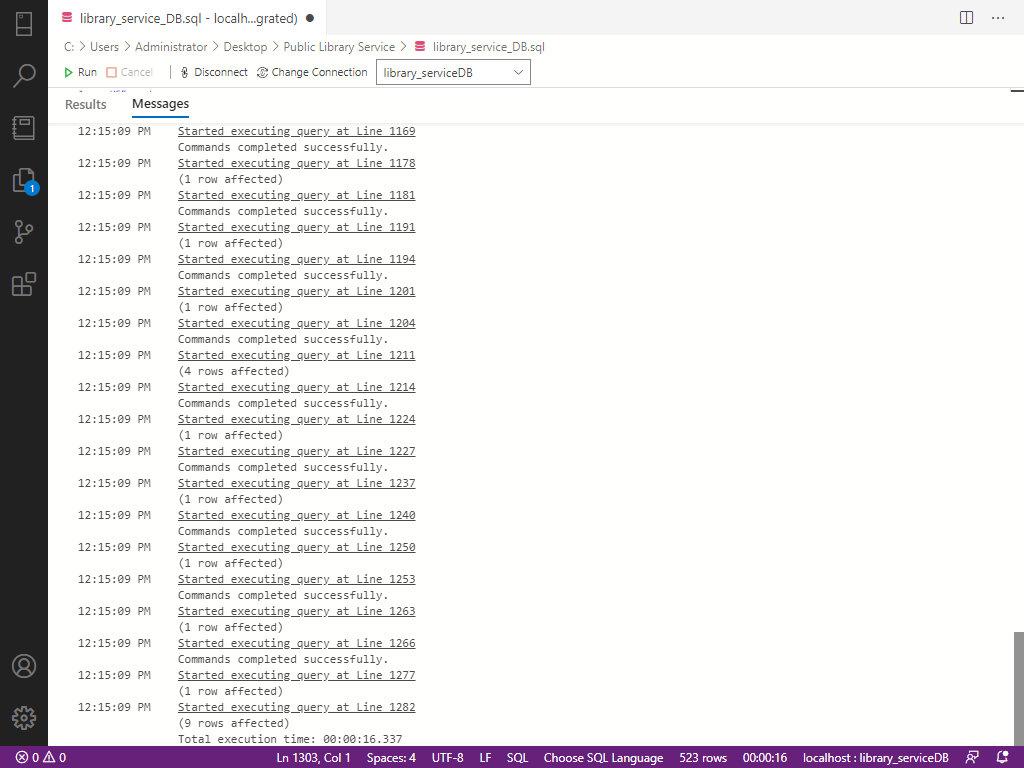
****

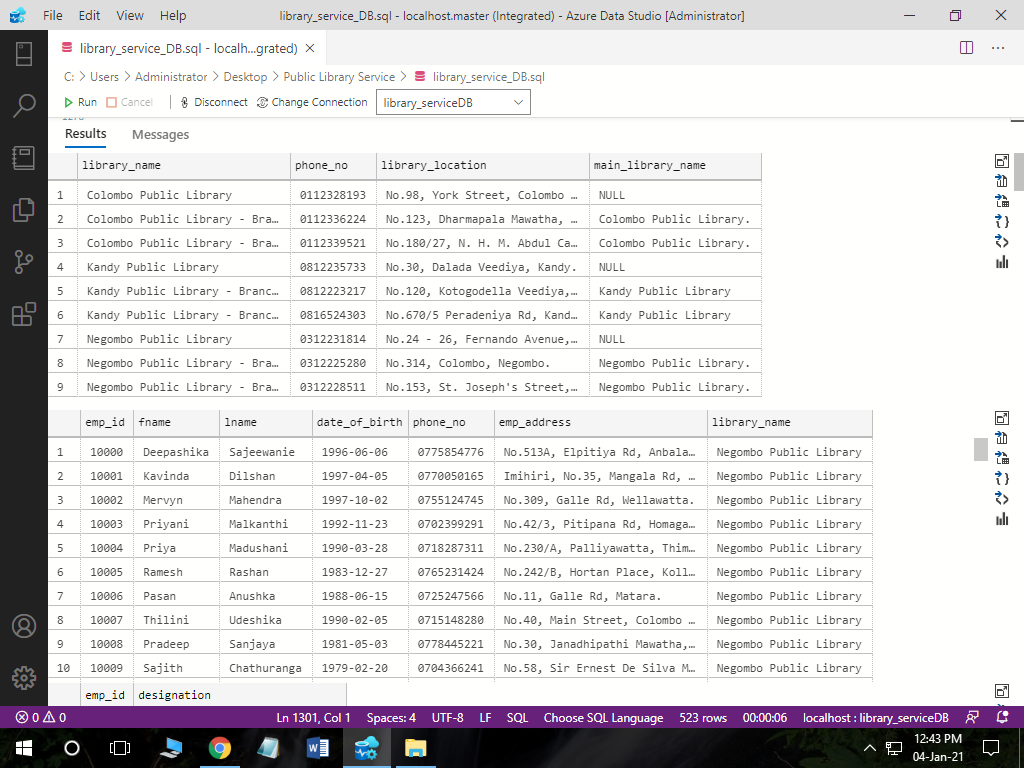
****

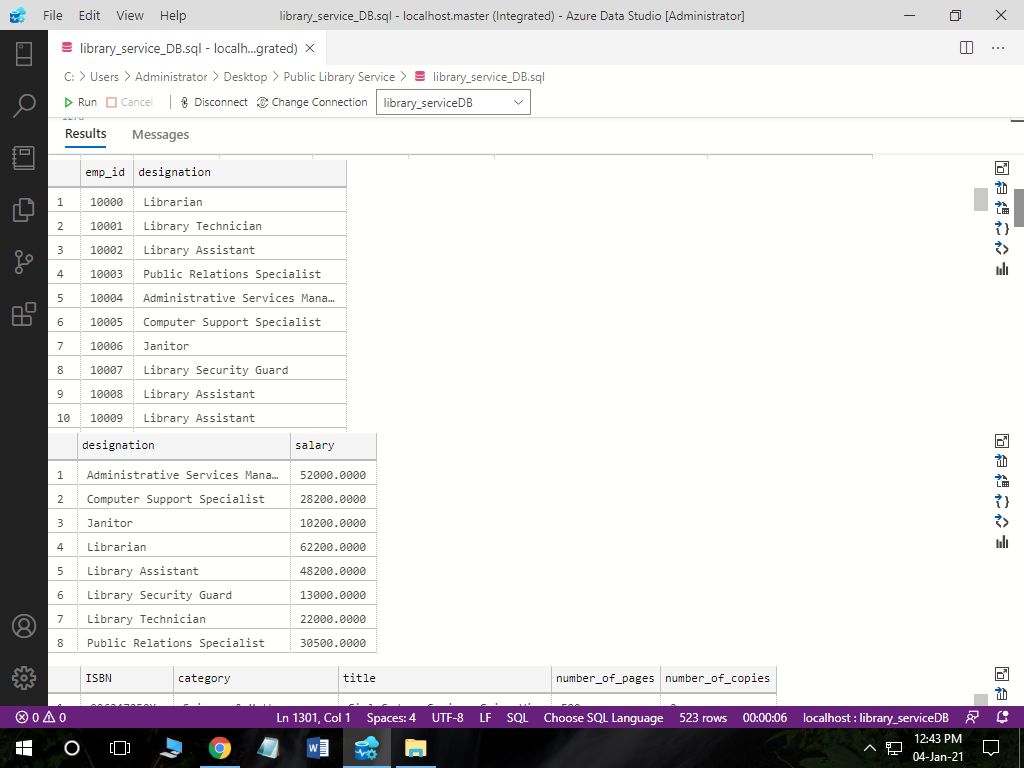
****

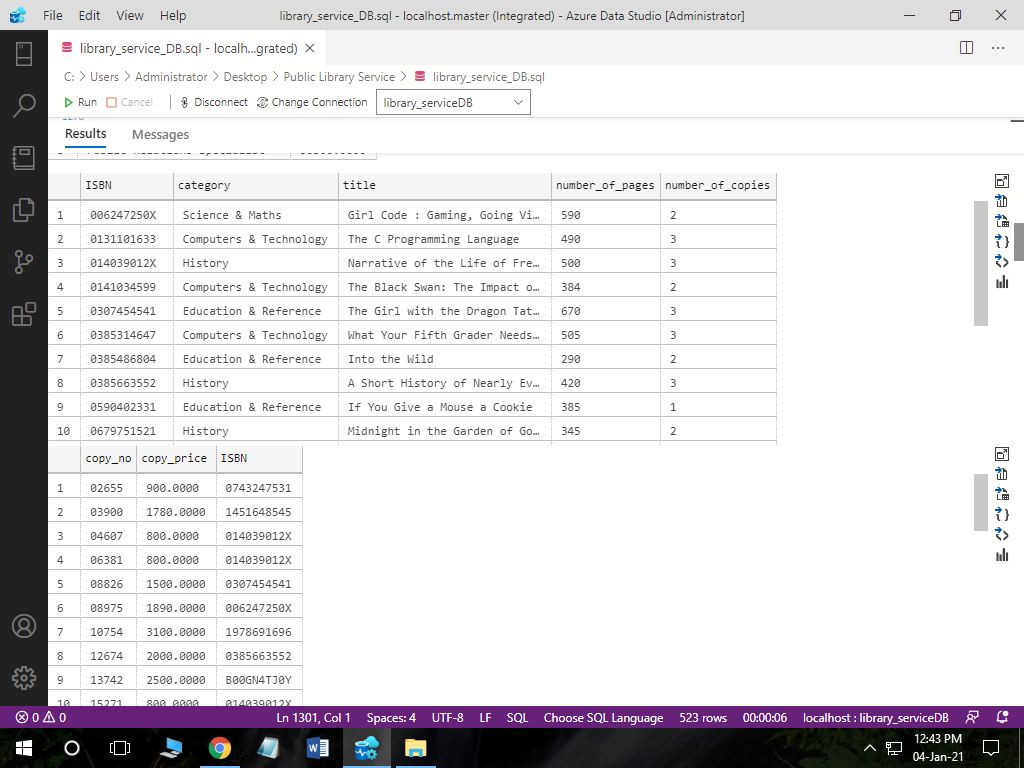
****

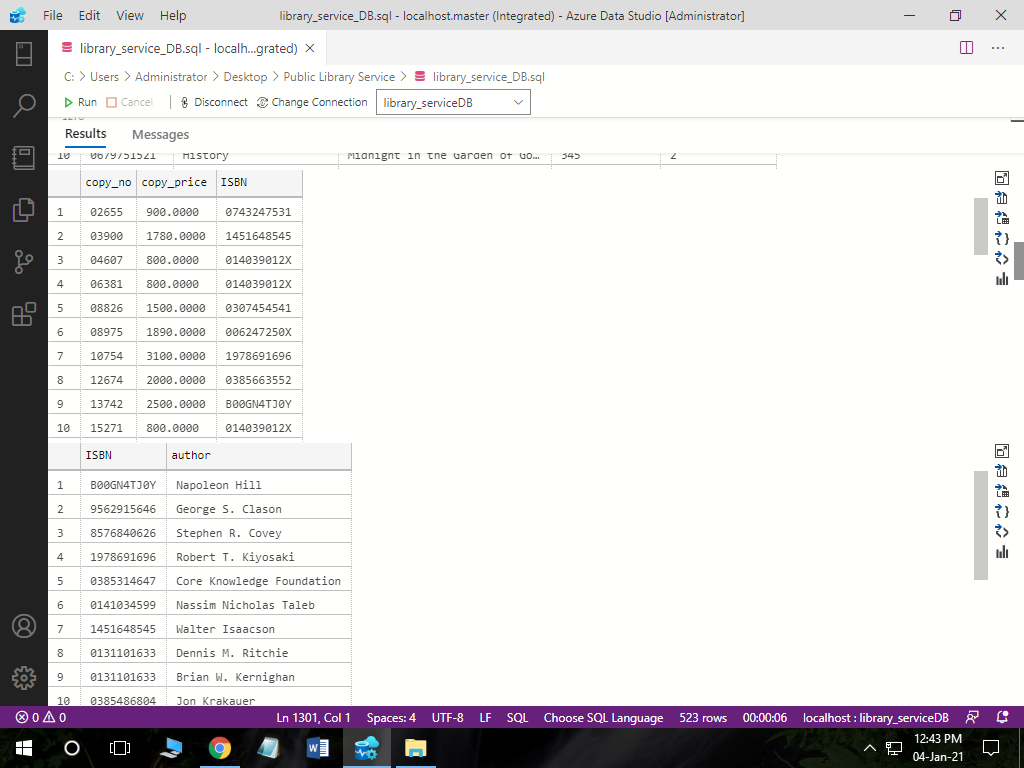
****

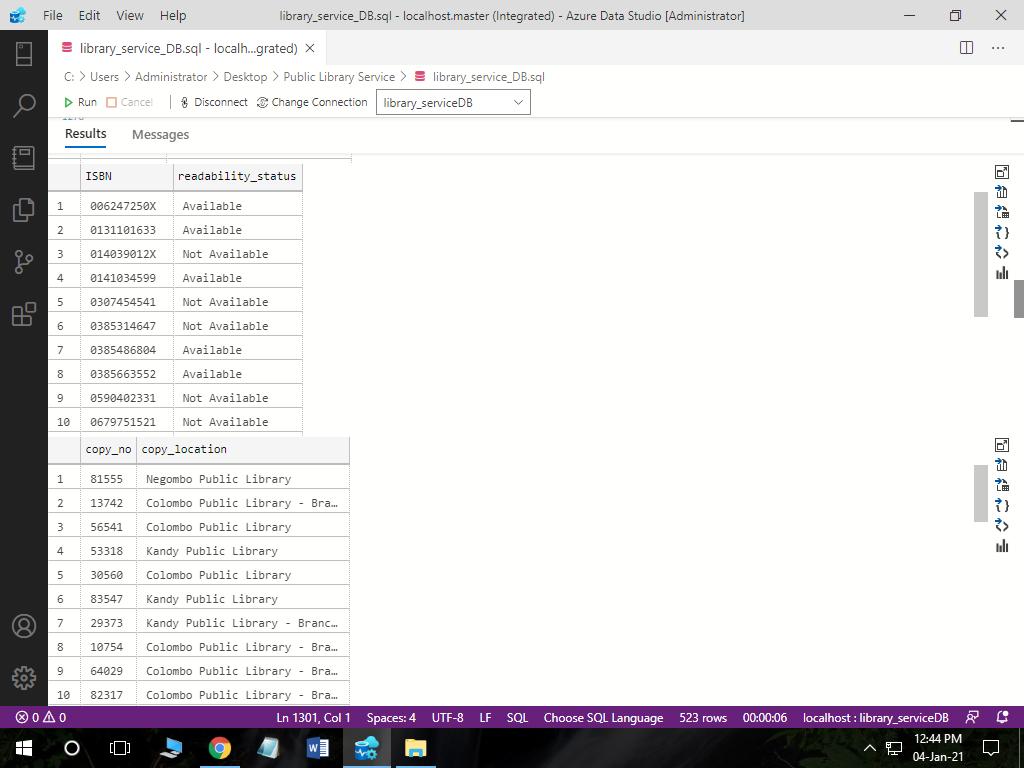
****

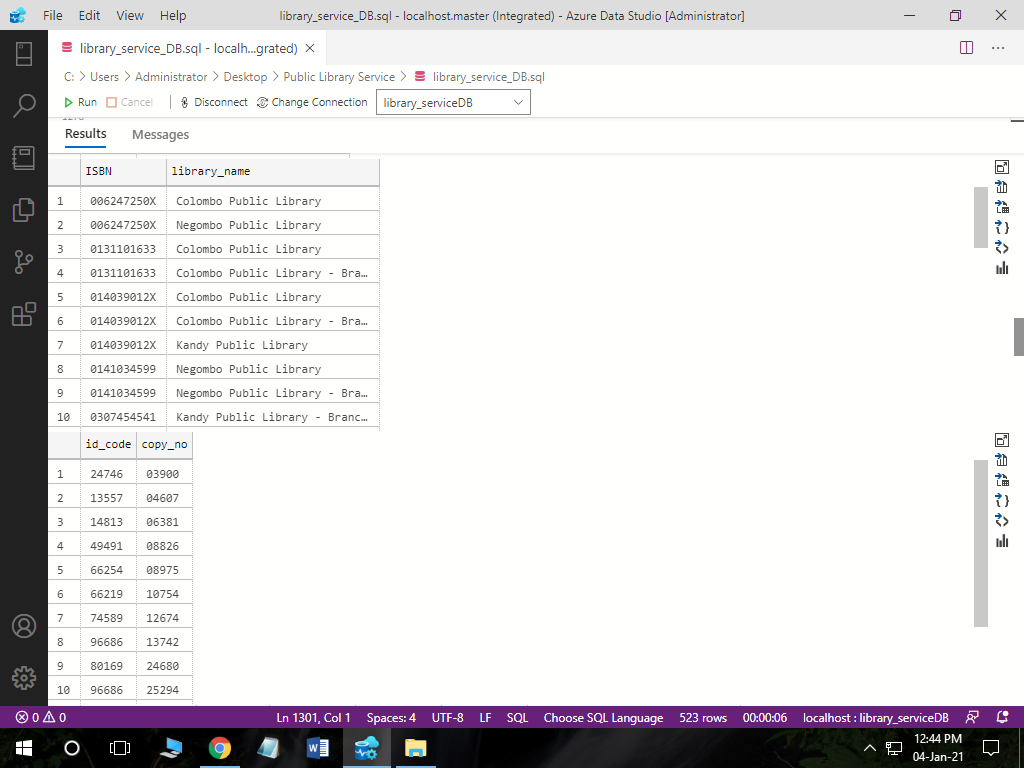


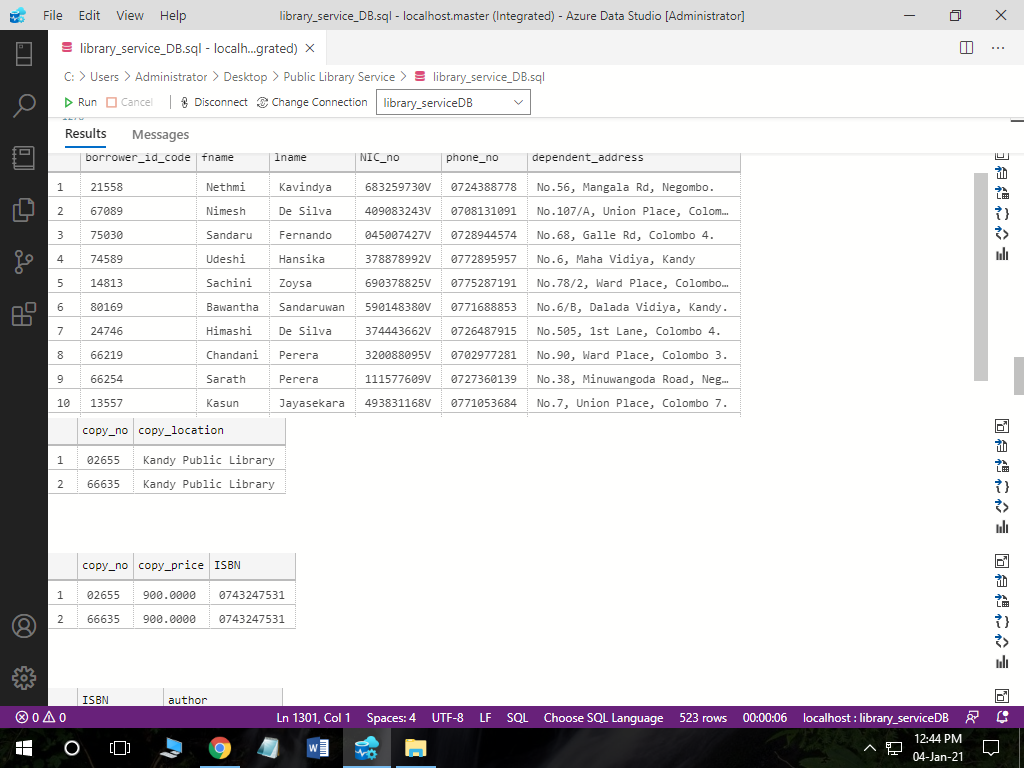


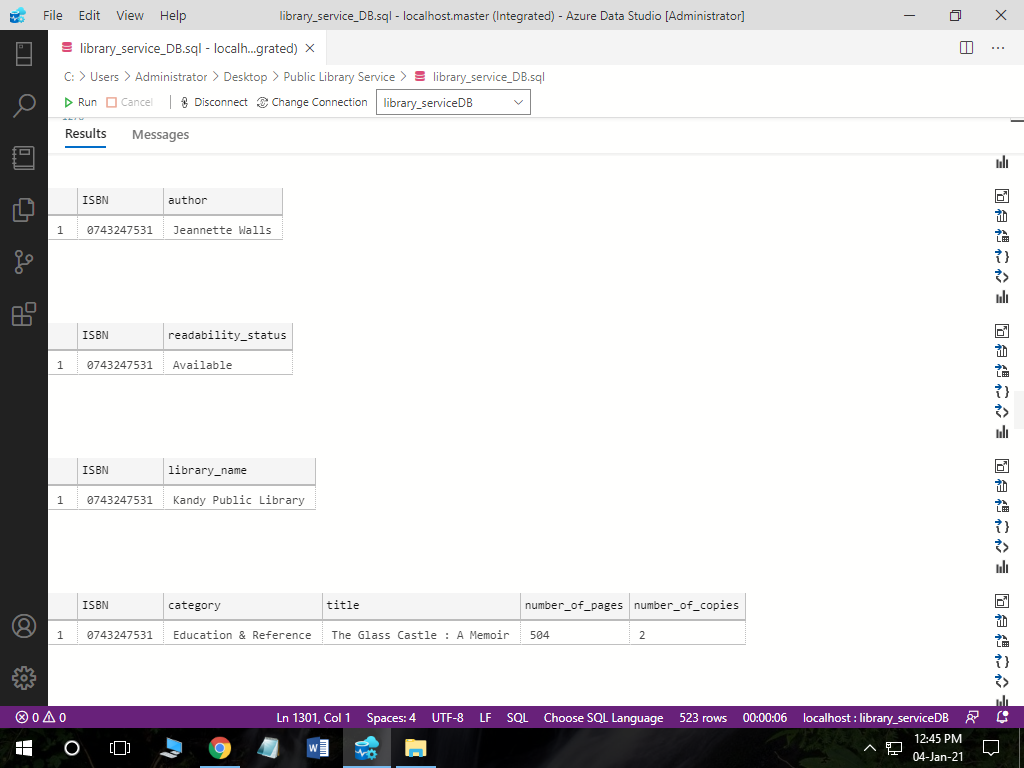


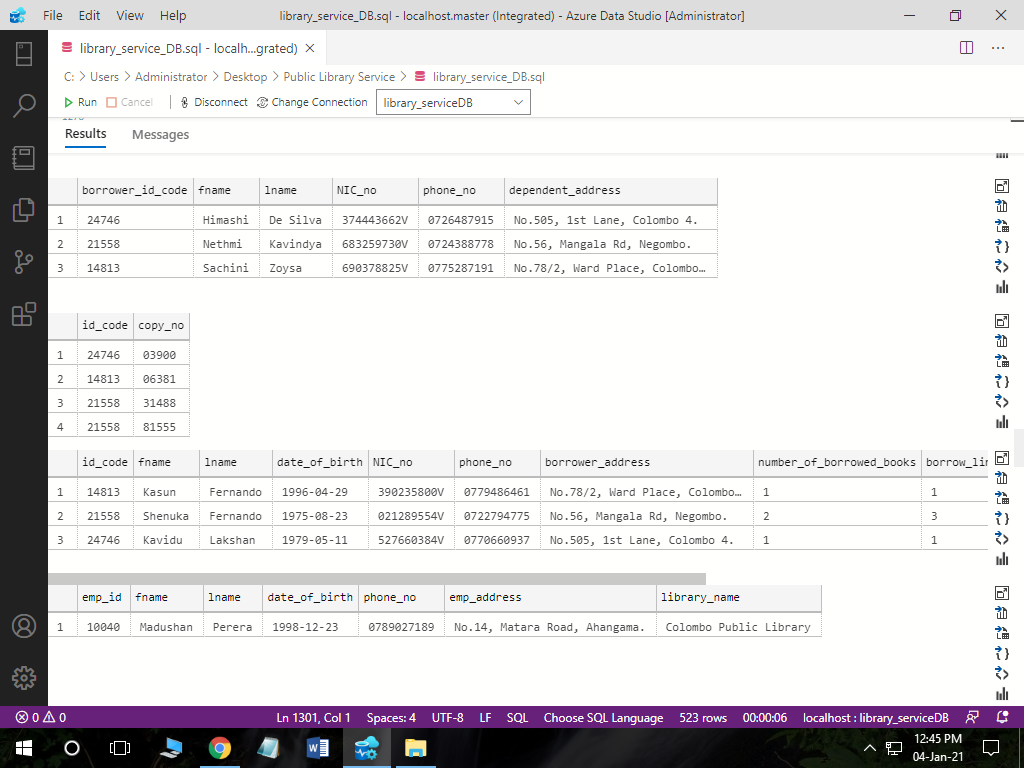


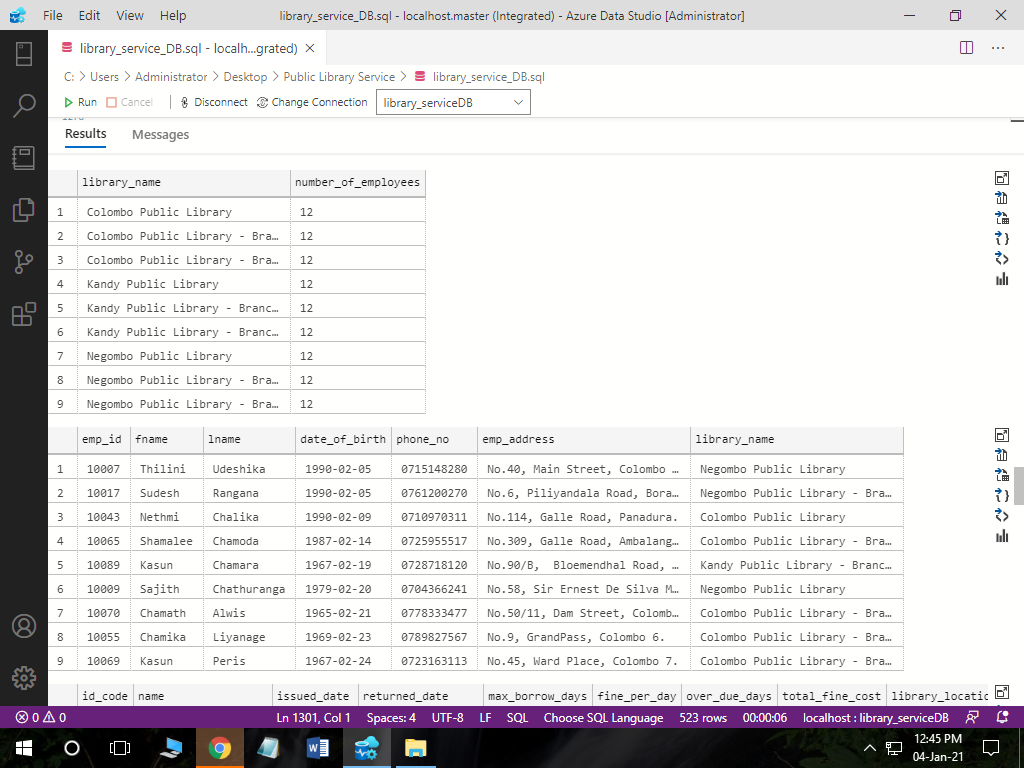


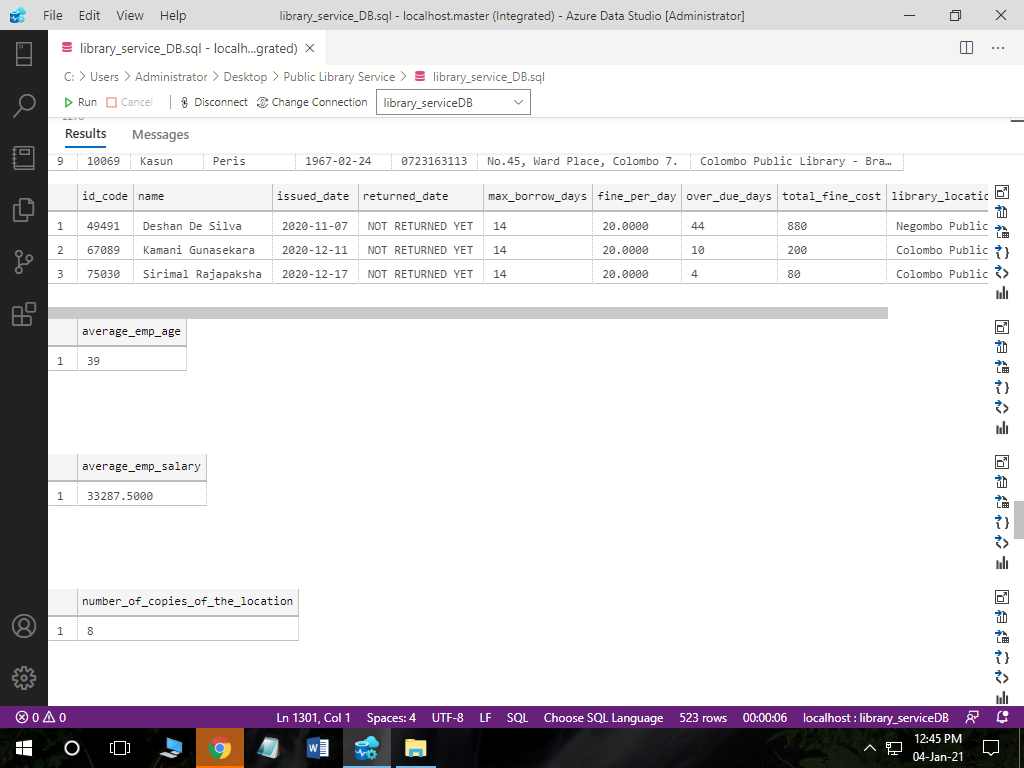


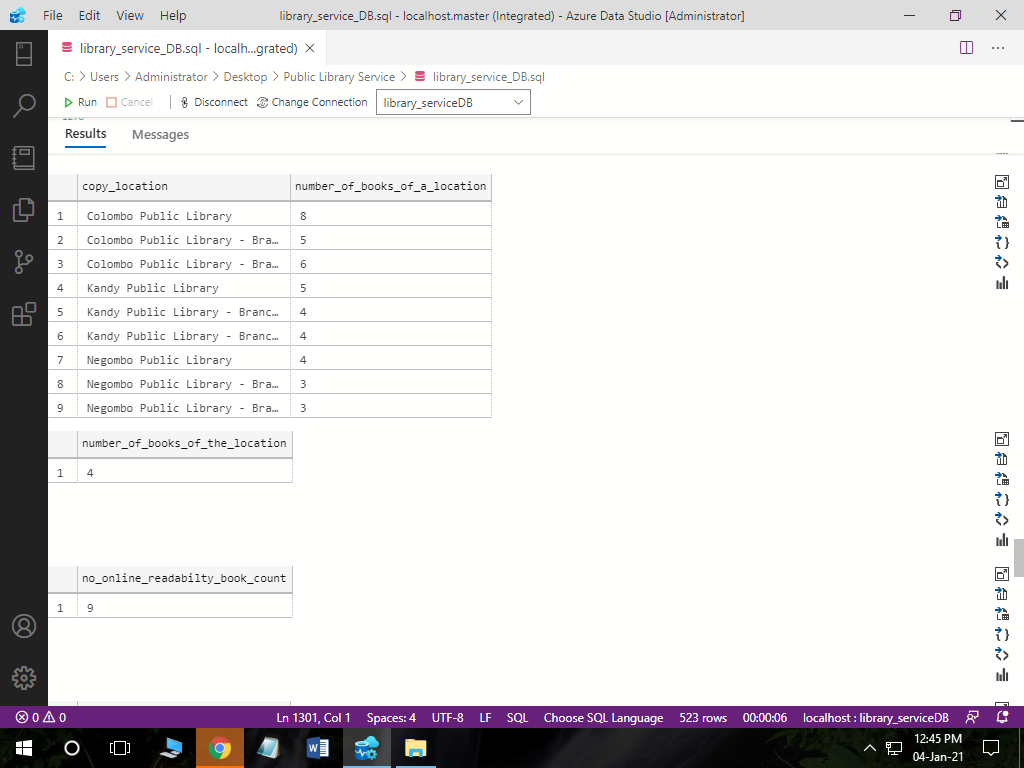


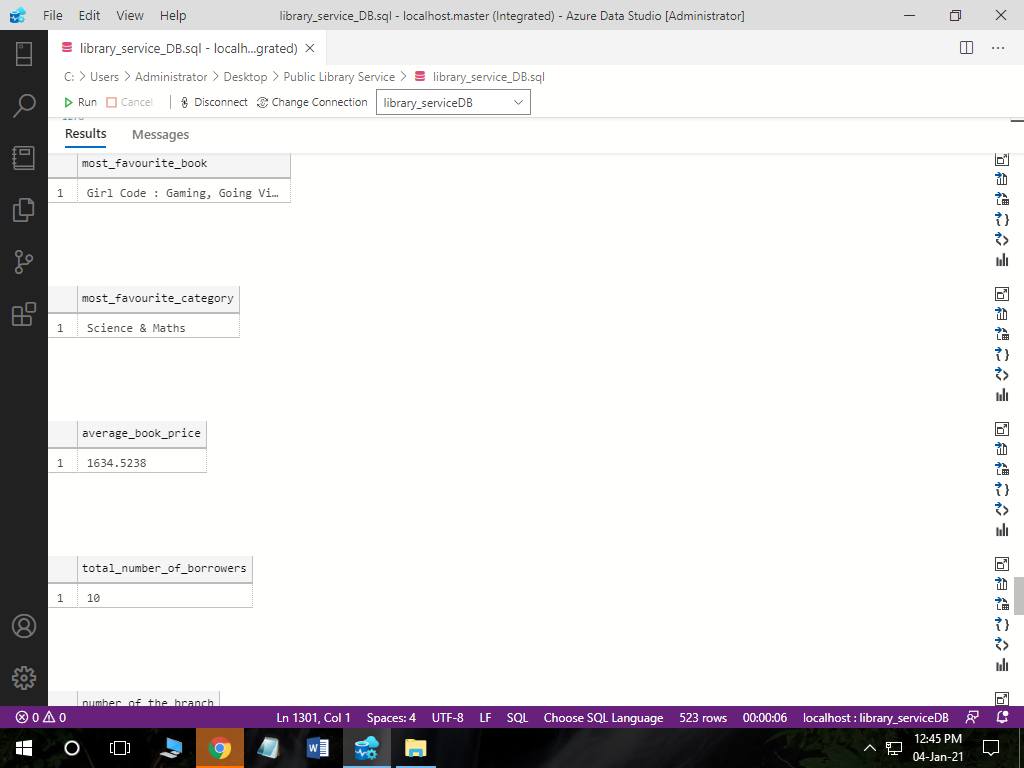


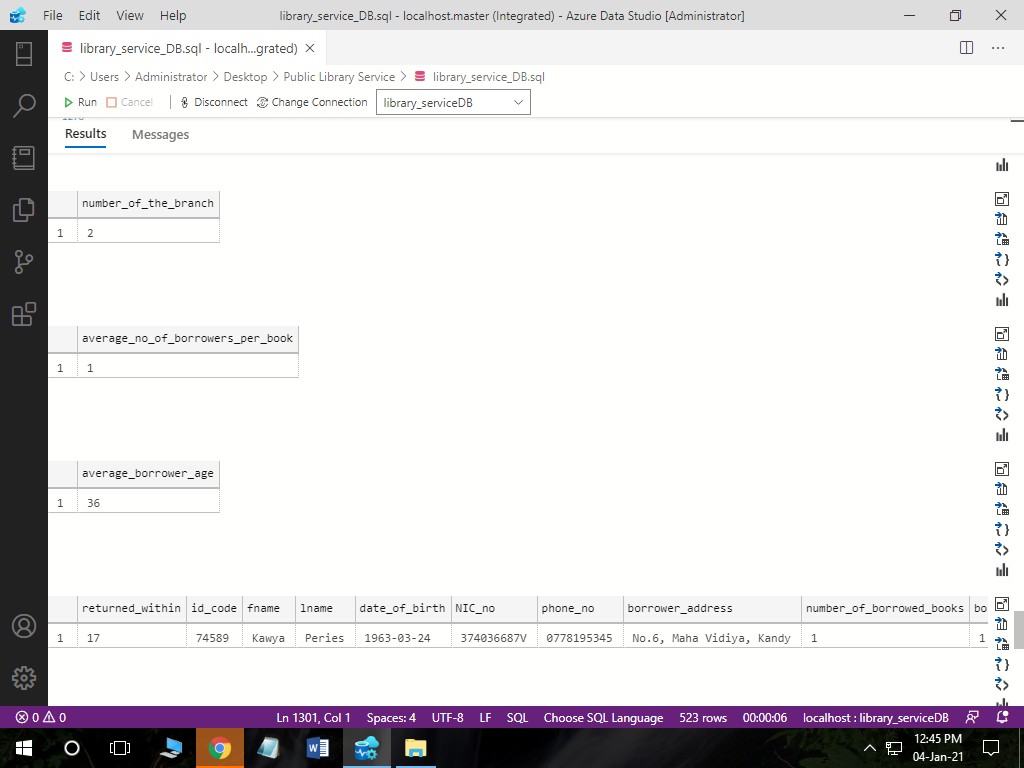


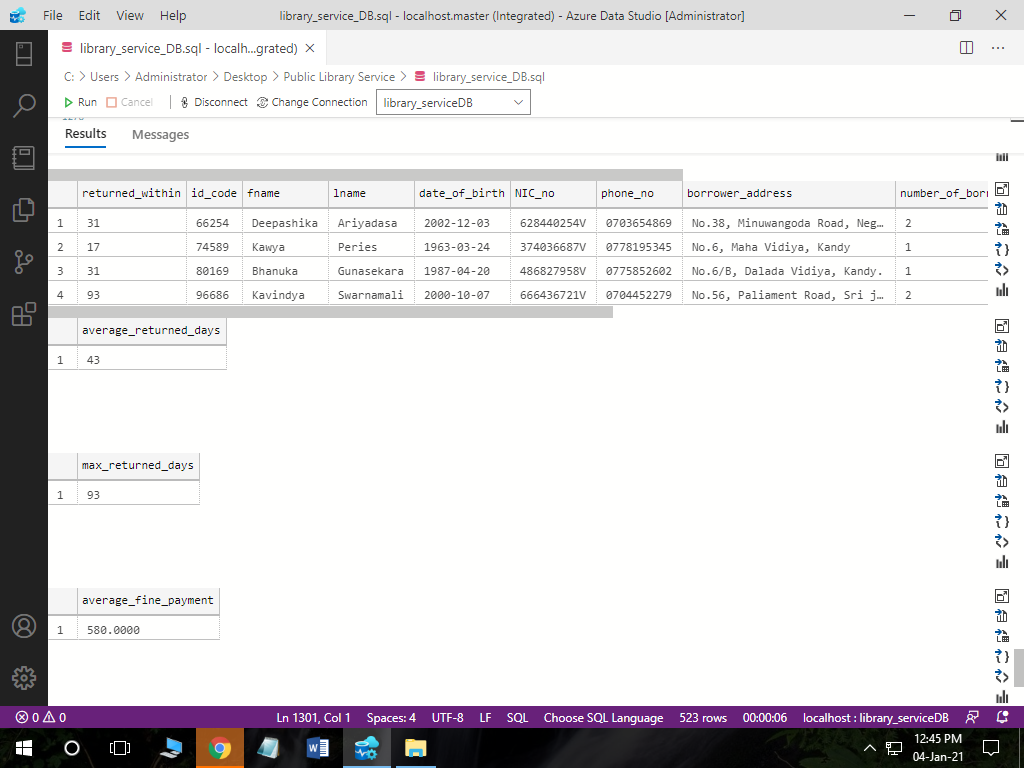


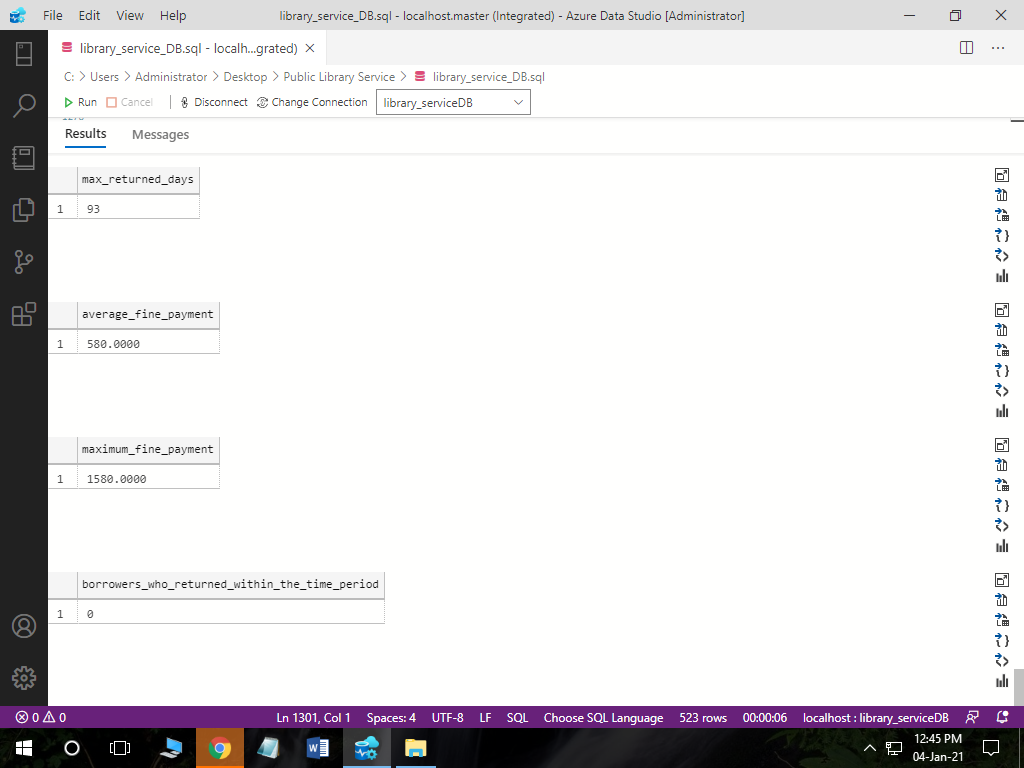












**CRITICAL EVALUATIONS**

* When creating Database I had to use Master keyword to drop the database when needed because I can’t drop it when already using the library\_serviceDB database. But when creating for the first time that is okay not to use master.
* Once I forget to mention the primary key to the table, but nothing was wrong with it until I use that that table’s attribute as a foreign key to another table because an error occurred.
* When creating user defined function, I wanted to create one with no parameters so I didn’t mention brackets after the function name so I realized there was an error. So then I found out that I have to put null brackets even if there’s no parameters to pass.
* Sometimes when you don’t know whether the store procedure, view, trigger or user-defined functions may work when you execute the whole function or store procedure at once, sometimes I didn’t get any results when executing them, so I had to use a trick when creating all off these. That was Query statements must be executed first before adding them inside the store procedure, function, trigger or whatever it is and that can leads to identify problem that may occur during execution properly and easily.
* That’s how I wrote more than 1000 lines with no errors.

**FUTURE IMPLEMENTATIONS**

* To send an email to borrower when the maximum returned date is closer.
* To send birth day wishes to all the borrowers who has upcoming birth days.

**THE END | THANK YOU**