Assignment 1   
CPS610 – Database 2  
Wednesday Section – Group 13  
Member #1: Andy Lee (500163559)  
Member #2: Sohrab Soltani (500801172)

# Introduction

In our first assignment, we deploy two local Oracle databases, then utilizing Java via OJDBC to connect and perform add, update, and delete on the databases while following two-phase commit protocol.

The submission consists a zip. First, extract the zip file and open the project using Eclipse. Next, open and run MainProgram.java to launch the program, then follow the instructions displayed on the console menu (see Figure 1). To perform a function, key in any digit of 1 to 5 and press Enter. To exit the program, key in 0 and press Enter. Figure 2 and Figure 3 show two databases running inside a virtual machine, with a table TESTJ respectively, and “name” field with no data.

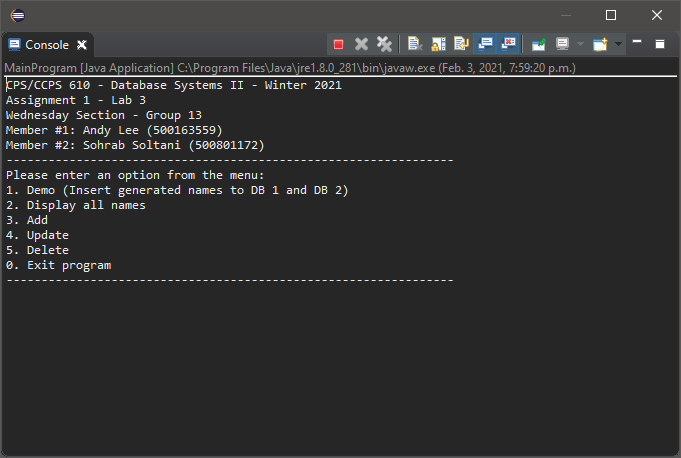


Figure 1 Main program console menu

|  |  |
| --- | --- |
| Figure 2 Table Test J in Virtual Machine vbox-1 | Figure 3 Table Test J in Virtual Machine vbox-2 |

Contents

[Introduction 1](#_Toc63281439)

[Option 1 – Demo inserting names in multithread 4](#_Toc63281440)

[Option 2 – Display all names 9](#_Toc63281441)

[Option 3 – Adding name to databases 10](#_Toc63281442)

[Option 4 – Updating a name in the database 13](#_Toc63281443)

[Option 5 – Deleting name from database 16](#_Toc63281444)

[Option 0 – Exiting program 19](#_Toc63281445)

Table of Figures

[Figure 1 Main program console menu 1](#_Toc63281343)

[Figure 2 Table Test J in Virtual Machine vbox-1 2](#_Toc63281344)

[Figure 3 Table Test J in Virtual Machine vbox-2 2](#_Toc63281345)

[Figure 4 Option 1, first screen 5](#_Toc63281346)

[Figure 5 Option 1, second screen 6](#_Toc63281347)

[Figure 6 Option 1, after demo insert showing database 1 7](#_Toc63281348)

[Figure 7 option 1, after demo insert showing database 2 8](#_Toc63281349)

[Figure 8 option 2, displaying all names in both databases 9](#_Toc63281350)

[Figure 9 option 3, adding a new name 10](#_Toc63281351)

[Figure 10 option 3, added name shown on database 1 11](#_Toc63281352)

[Figure 11 option 3, added name shown on database 2 12](#_Toc63281353)

[Figure 12 option 4, updating name 13](#_Toc63281354)

[Figure 13 option 4, updated name shown in database 1 14](#_Toc63281355)

[Figure 14 option 4, updated name shown in database 2 15](#_Toc63281356)

[Figure 15 option 5, deleting a name 16](#_Toc63281357)

[Figure 16 option 5, deleted name absent in database 1 17](#_Toc63281358)

[Figure 17 option 5, deleted name absent in database 2 18](#_Toc63281359)

[Figure 18 option 0, exiting program 19](#_Toc63281360)

# Option 1 – Demo inserting names in multithread

Option 1 runs a simulation of concurrent users by spawning multiple threads with randomly generate names to be filled into the databases. The algorithm will first verify that the name generated is unique to both databases, and if the name does not exist in either database, the algorithm will perform a two-phase commit to insert the names into the two databases. See Figure 4 and Figure 5 for verbose console output; Figure 6 and Figure 7 for SQL developer output.

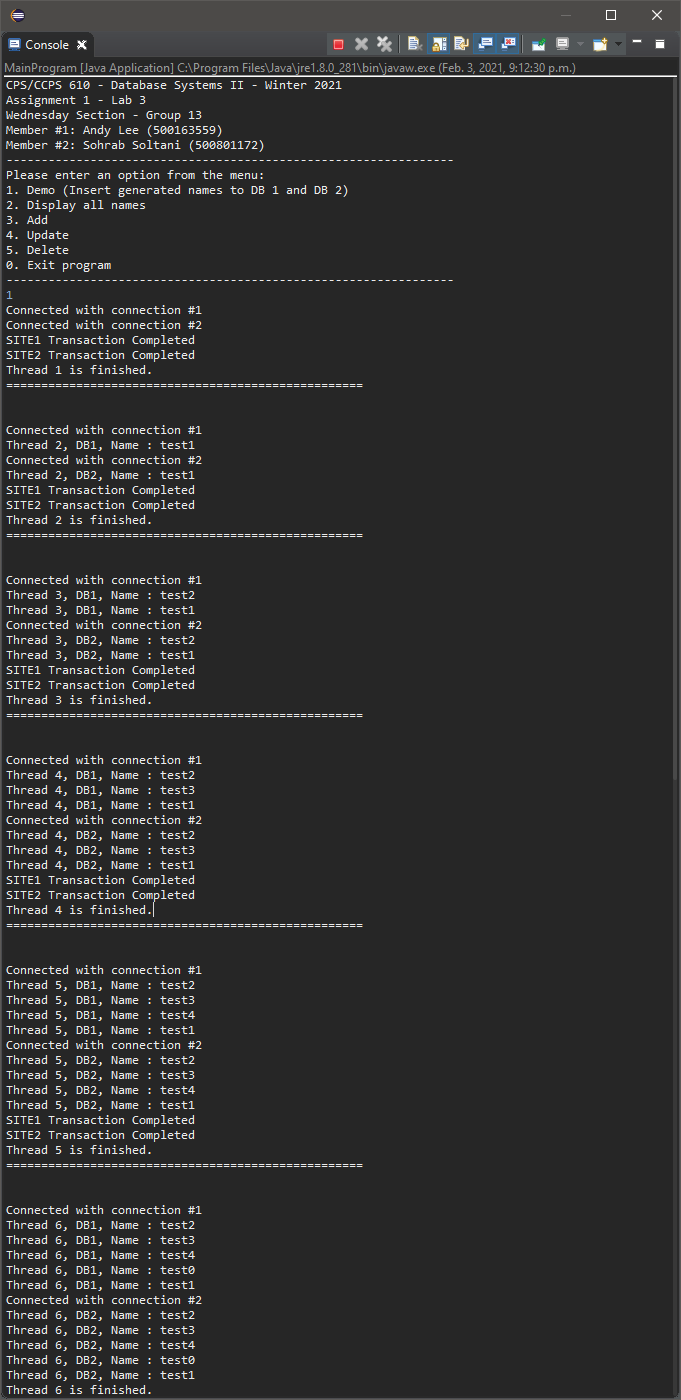


Figure 4 Option 1, first screen

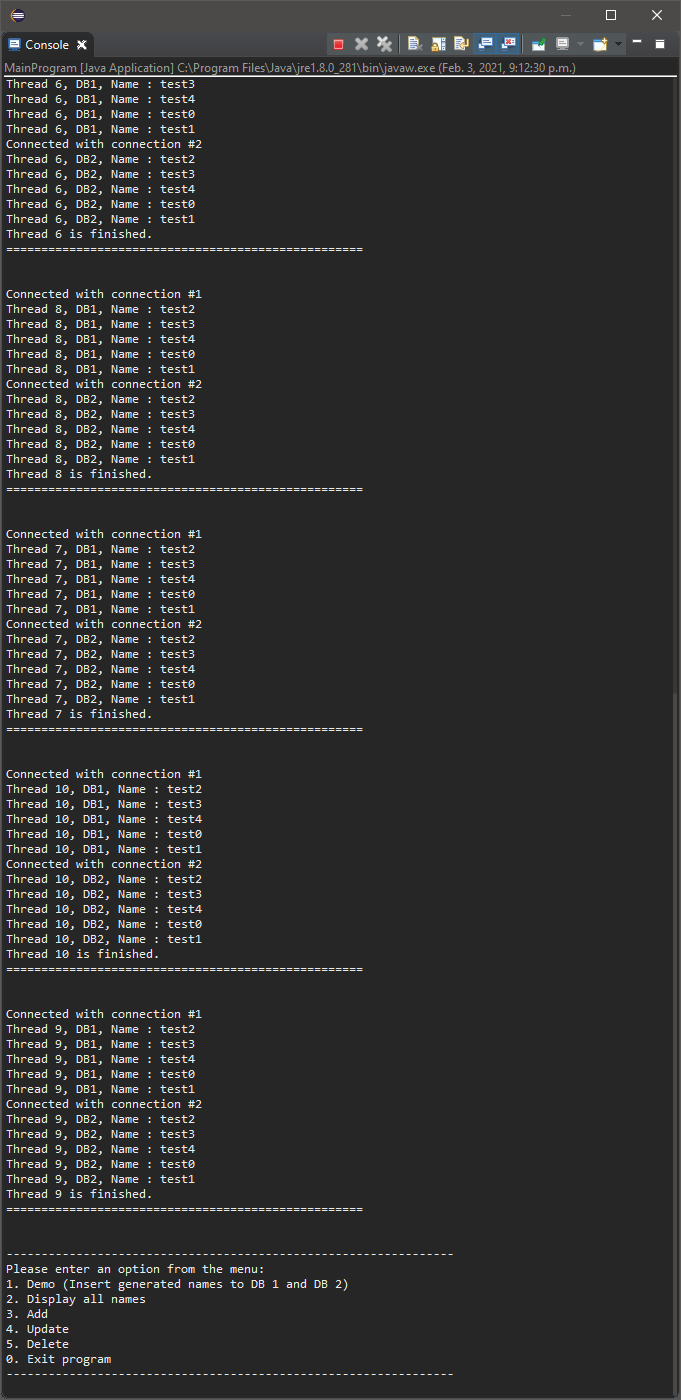


Figure 5 Option 1, second screen

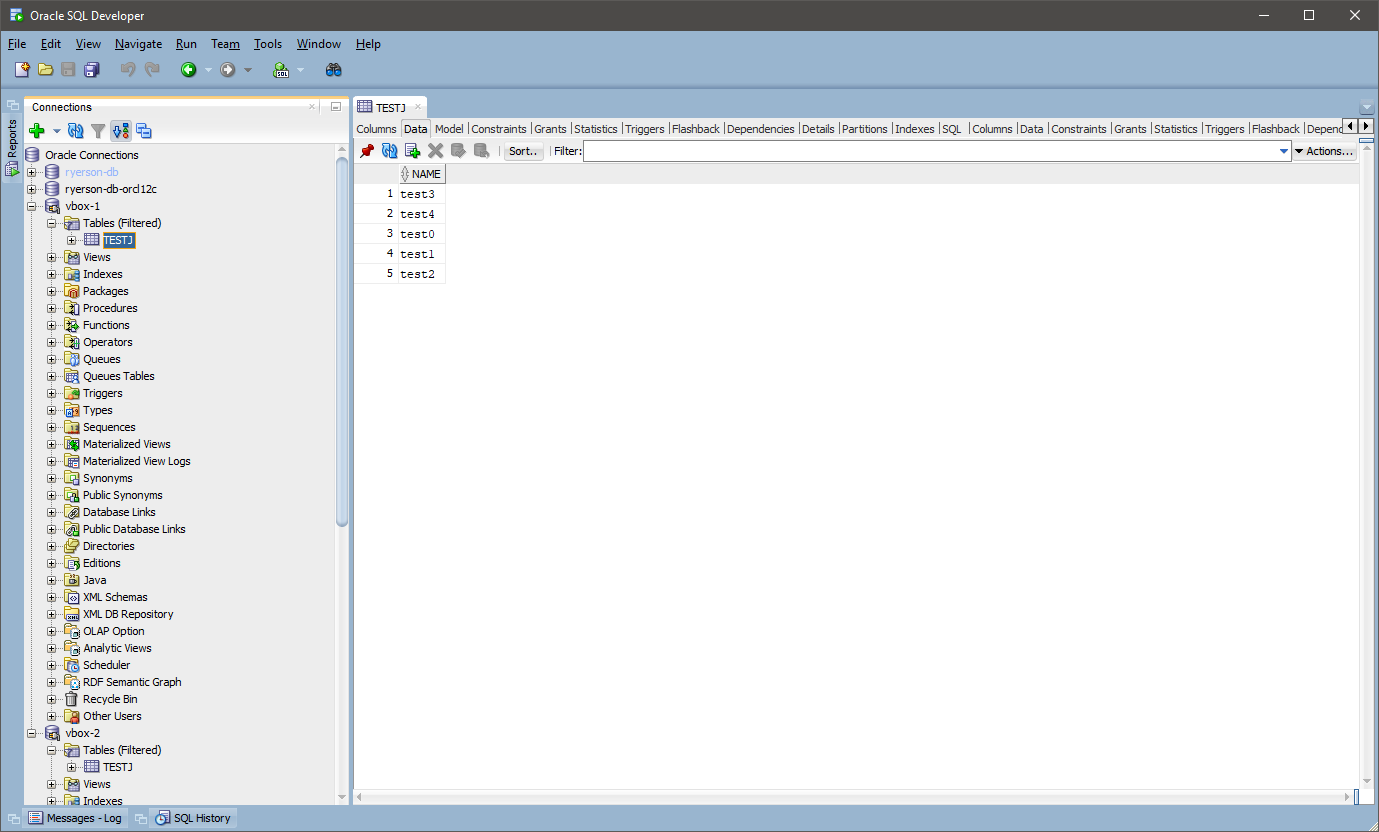


Figure 6 Option 1, after demo insert showing database 1

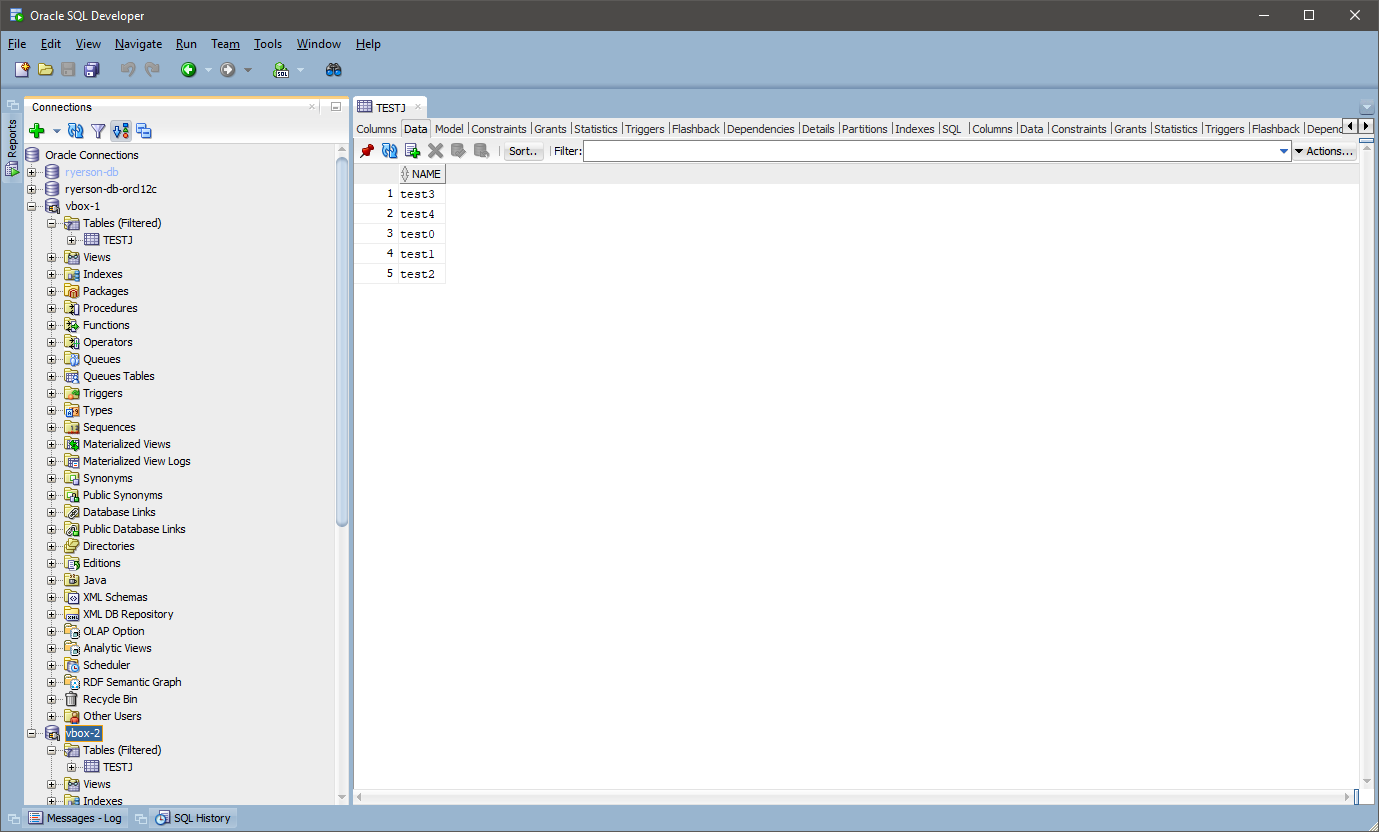


Figure 7 option 1, after demo insert showing database 2

# Option 2 – Display all names

Option 2 displays all names currently stored in both databases. See Figure 8 for console output.

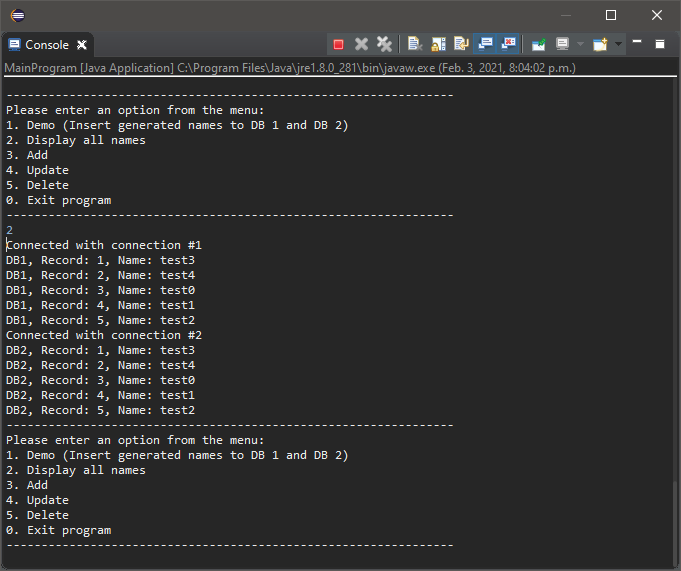


Figure 8 option 2, displaying all names in both databases

# Option 3 – Adding name to databases

Option 3 prompts the user for a name to be added to the database. The algorithm will first verify that the name given is unique in both database before adding. Figure 9 shows the console output for a name that has been successfully added to the database; Figure 10 and Figure 11 shows the databases from SQL developer.

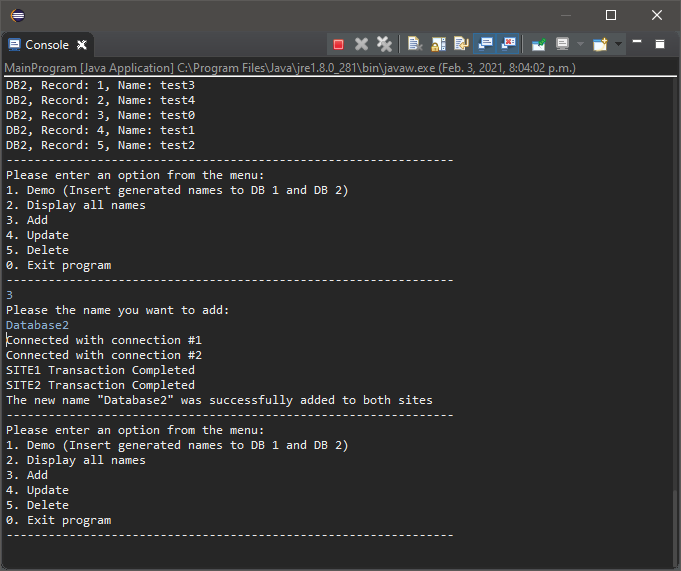


Figure 9 option 3, adding a new name

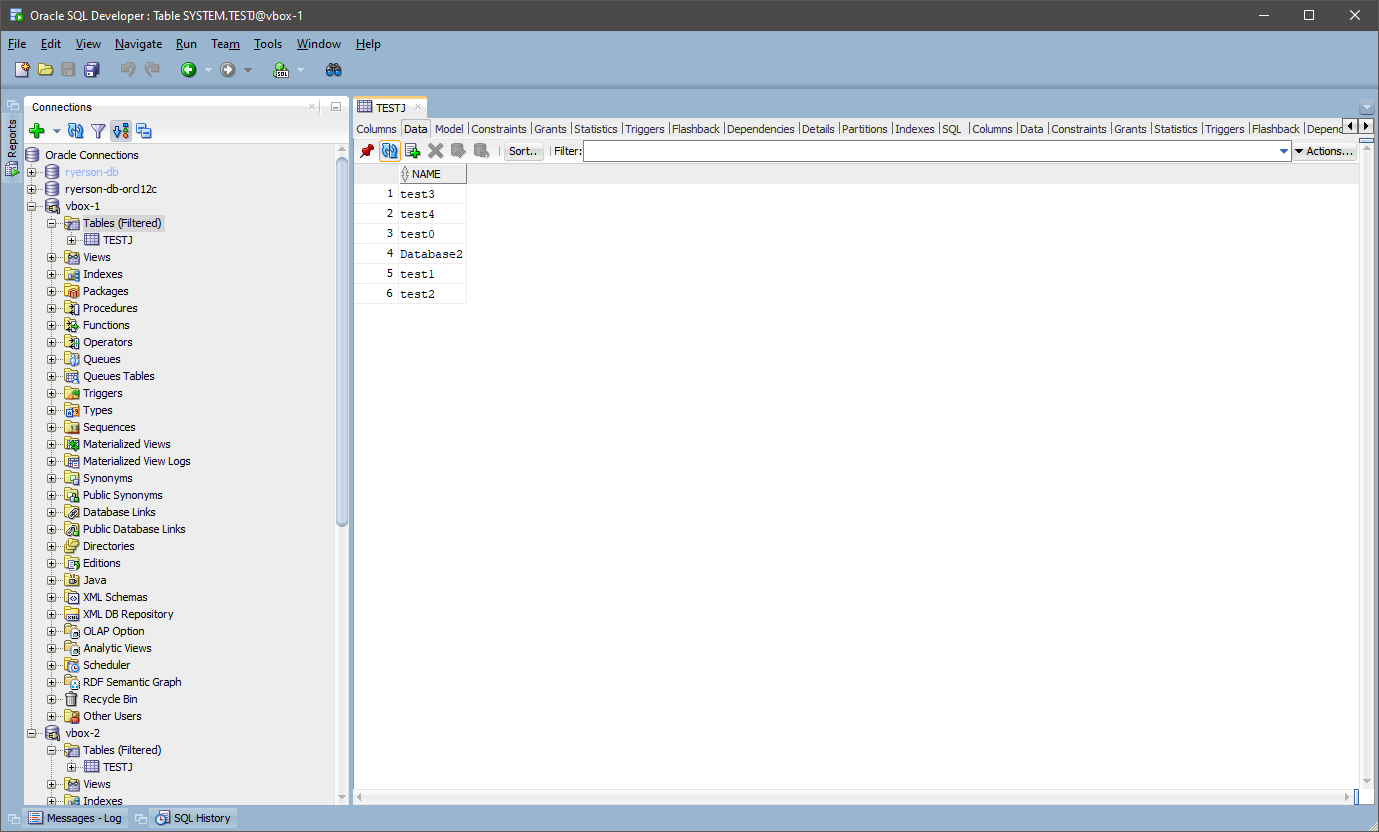


Figure 10 option 3, added name shown on database 1

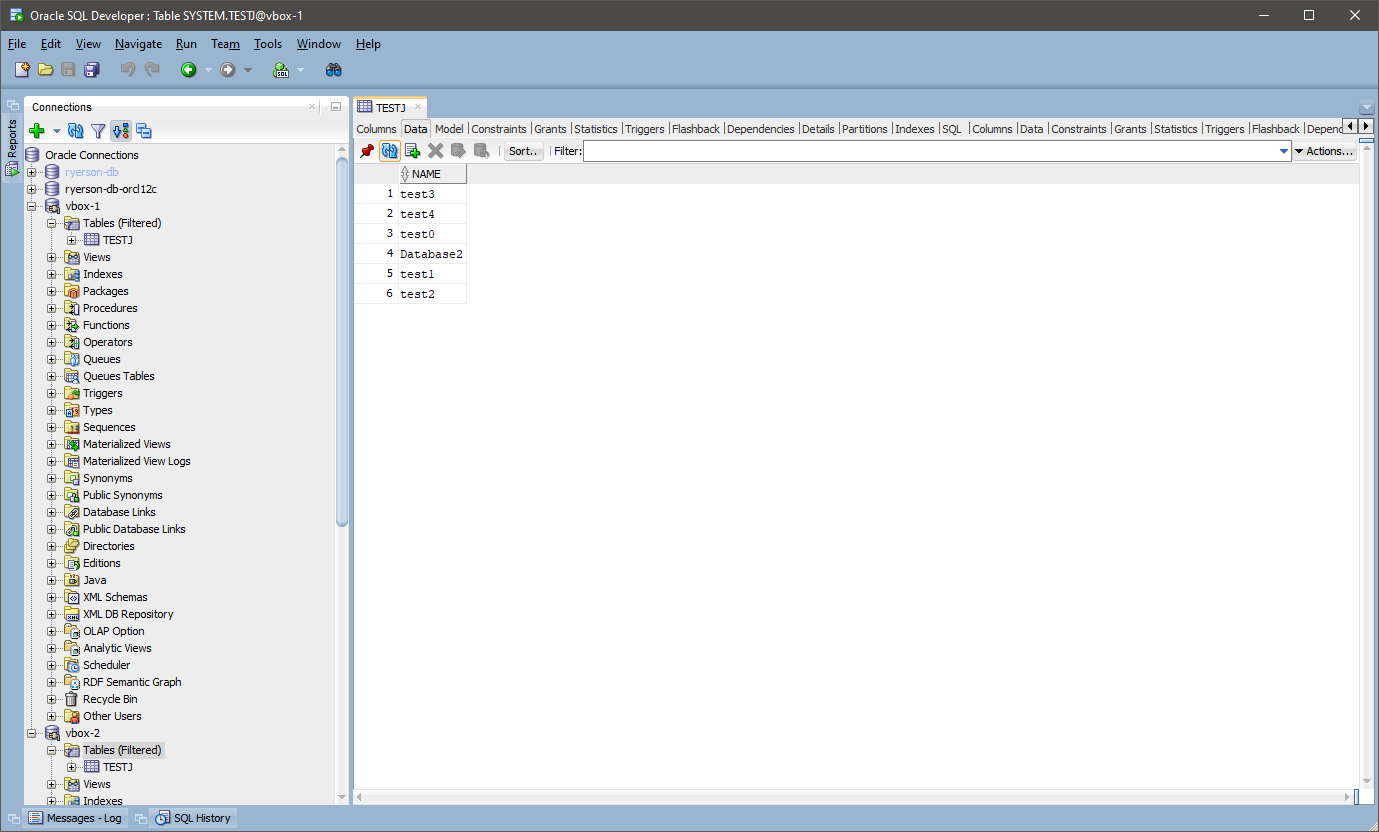


Figure 11 option 3, added name shown on database 2

# Option 4 – Updating a name in the database

Option 4 will display all the names currently stored in the database. From the list, the user chooses which name to update by entering the corresponding digit, then the program will prompt the user to enter the name to be changed, and updates the name on both databases. Figure 12 shows the user successfully changing Database 2 to Database2021; Figure 13 and Figure 14 shows the updated name on SQL developer.

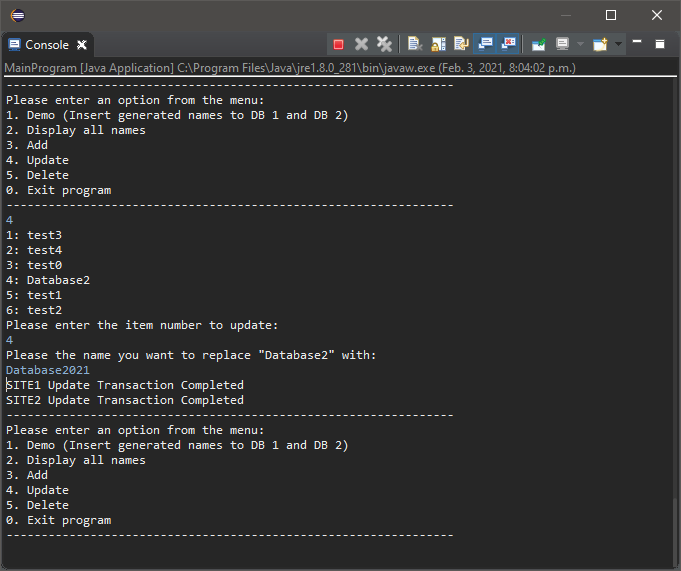


Figure 12 option 4, updating name

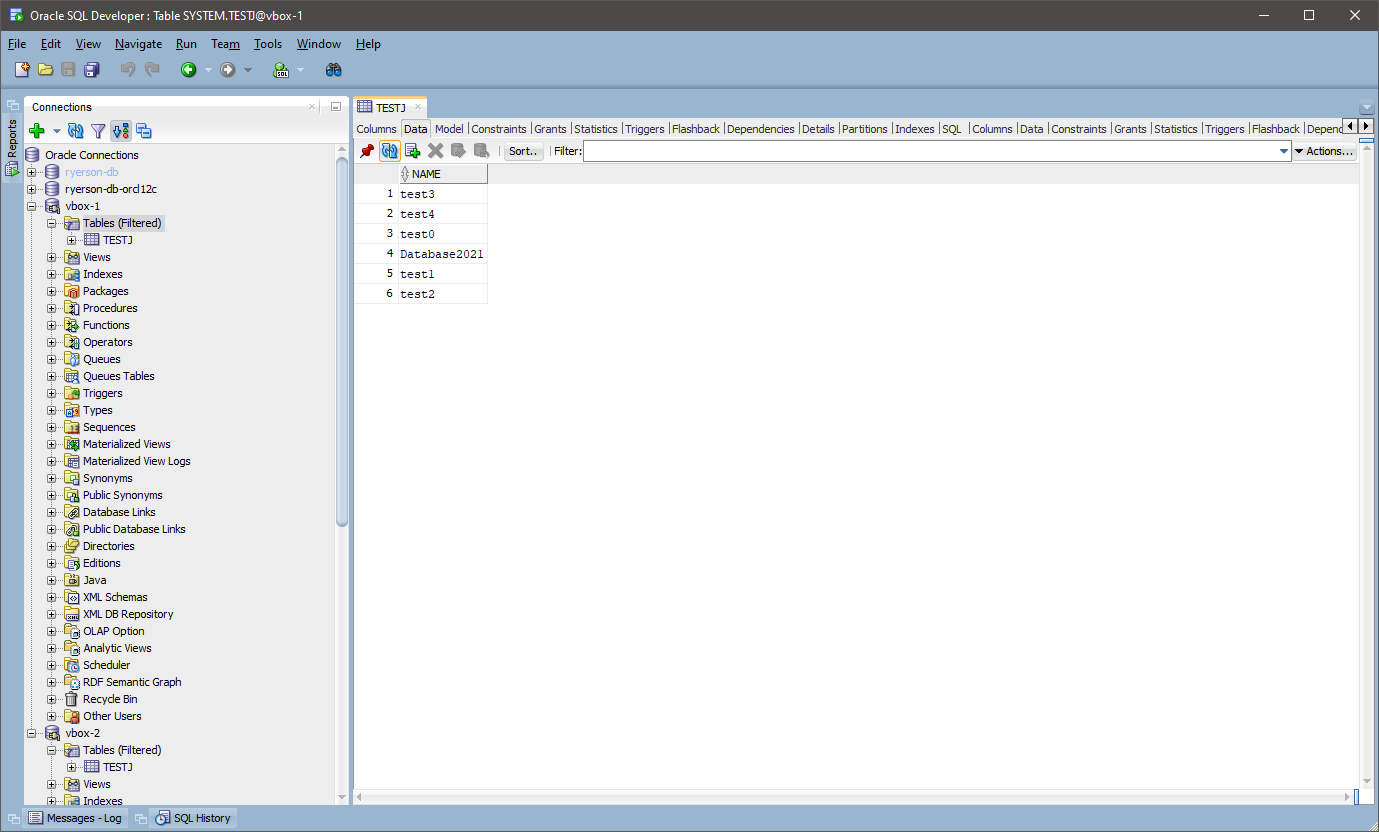


Figure 13 option 4, updated name shown in database 1

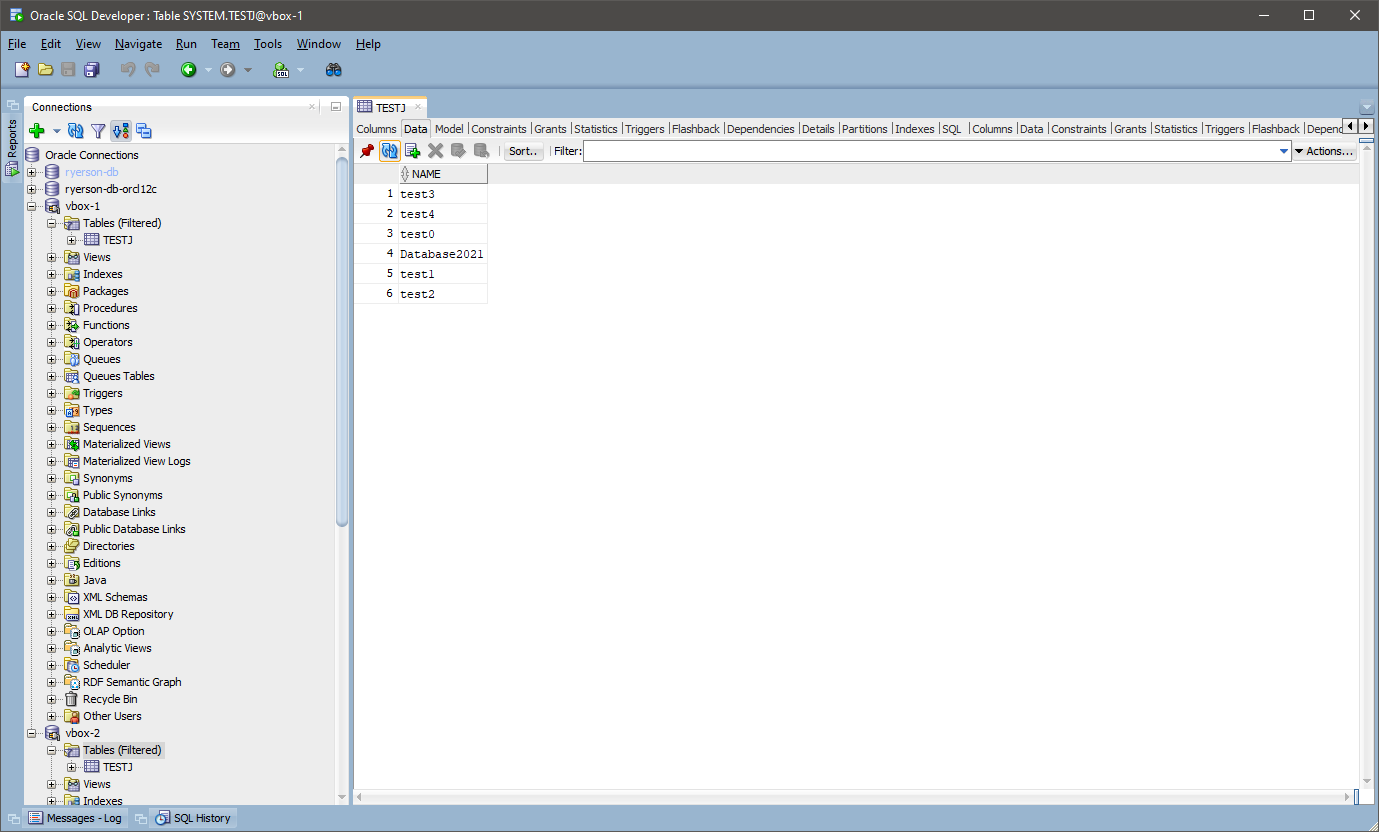


Figure 14 option 4, updated name shown in database 2

# Option 5 – Deleting name from database

Option 5 first displays all the names currently stored in the databases, then the user chooses which name to delete by entering the corresponding digit, and the program deletes the name from both databases. Figure 15 shows the user successfully removing Database2021 from the database; Figure 16 and Figure 17 shows the name removed in SQL developer.

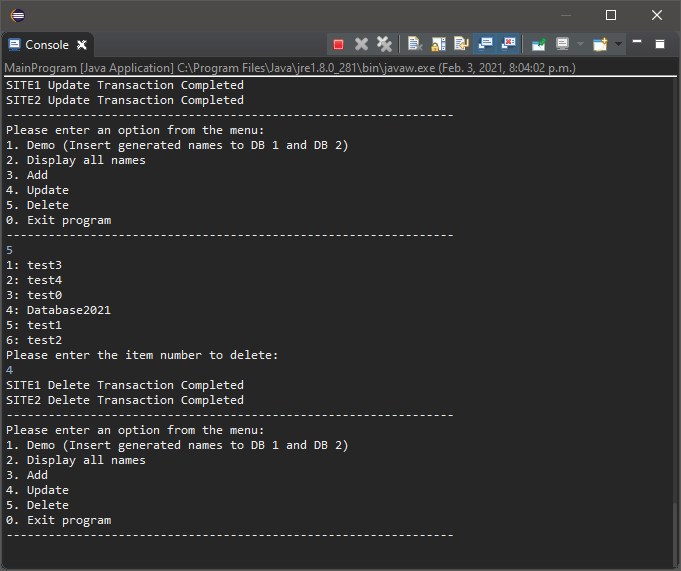


Figure 15 option 5, deleting a name

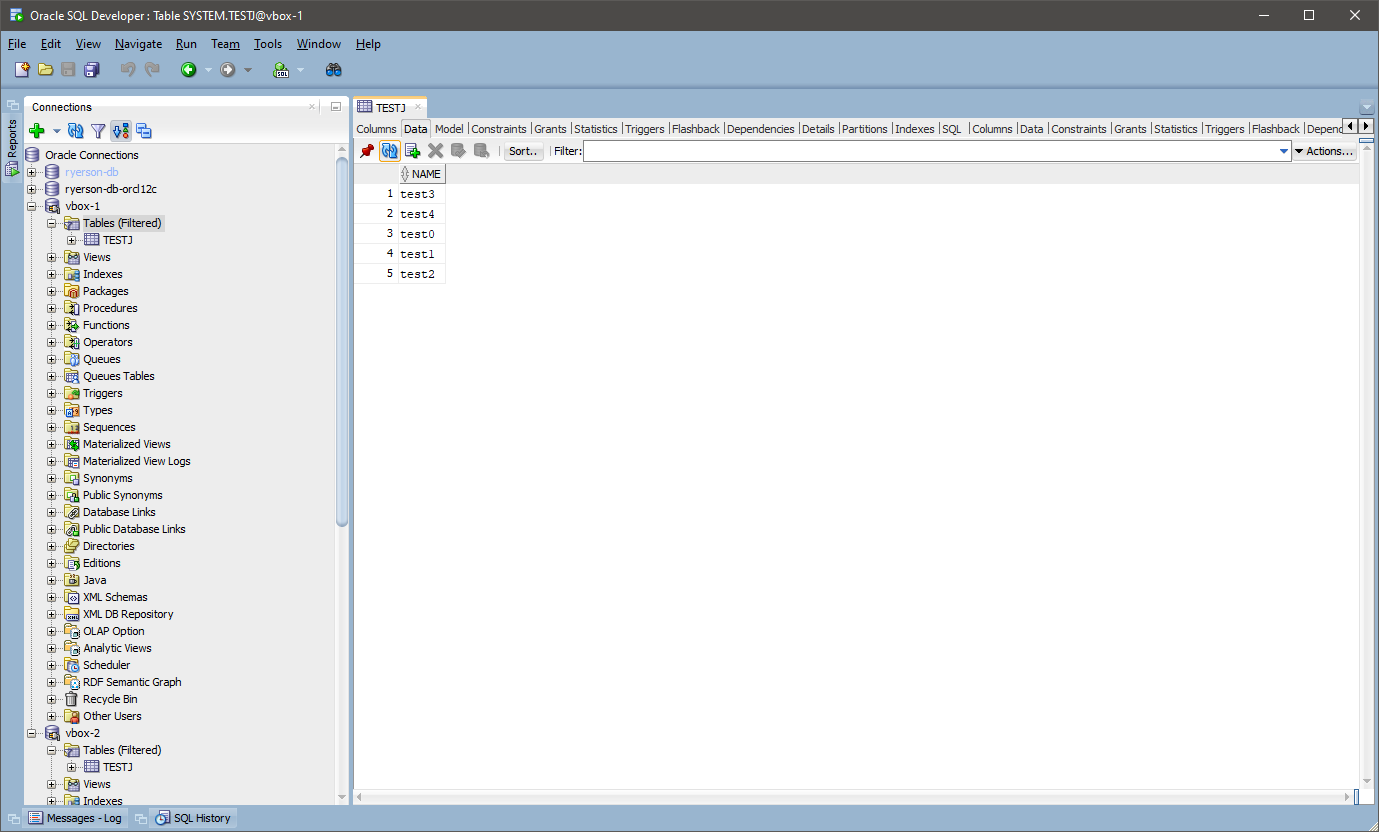


Figure 16 option 5, deleted name absent in database 1

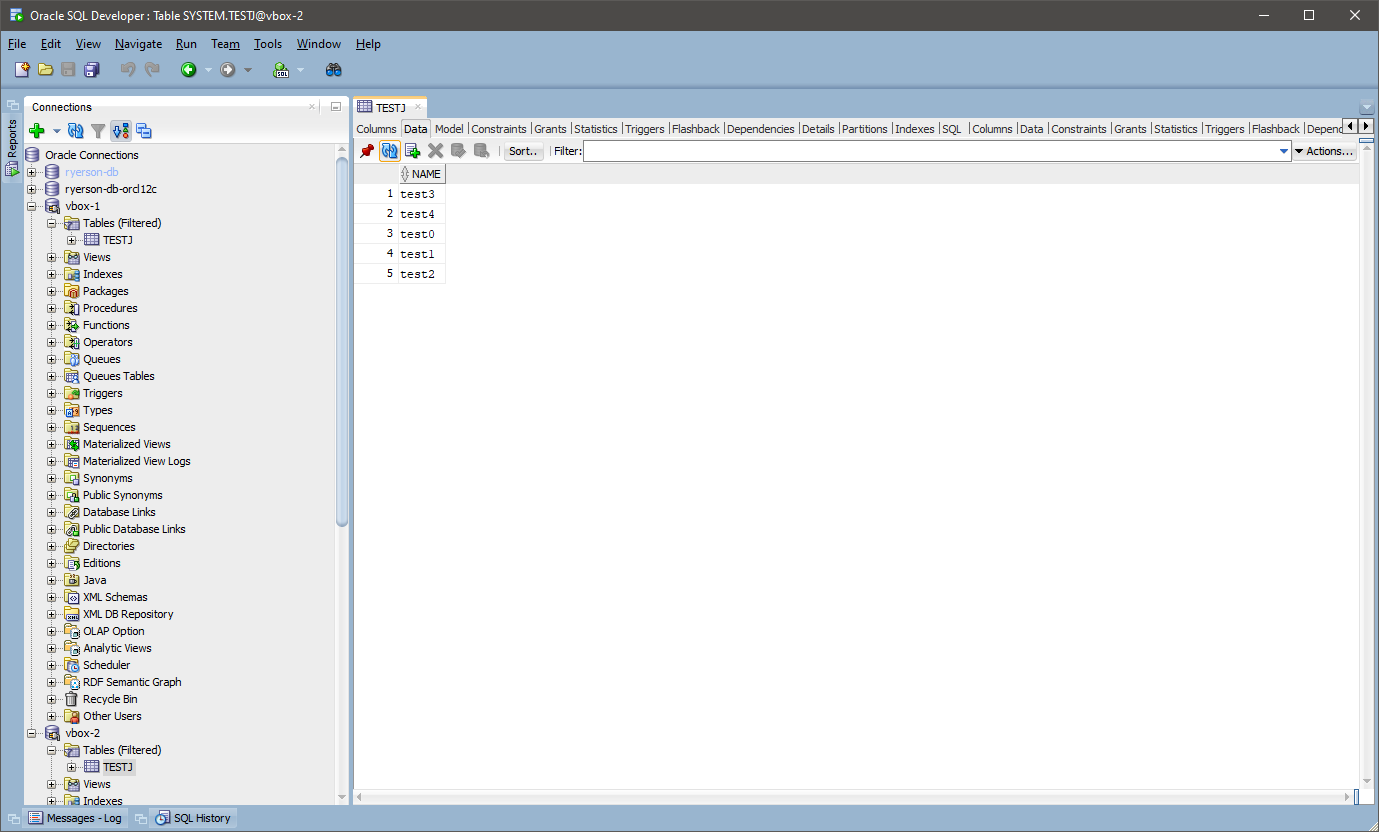


Figure 17 option 5, deleted name absent in database 2

# Option 0 – Exiting program

Option 0 exits and terminates the program.

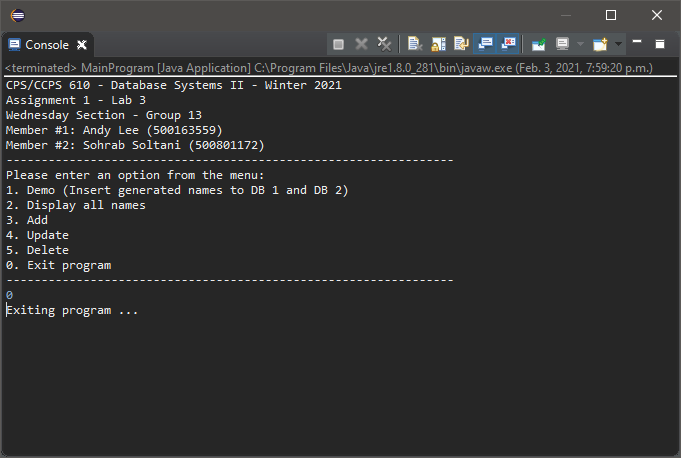


Figure 18 option 0, exiting program