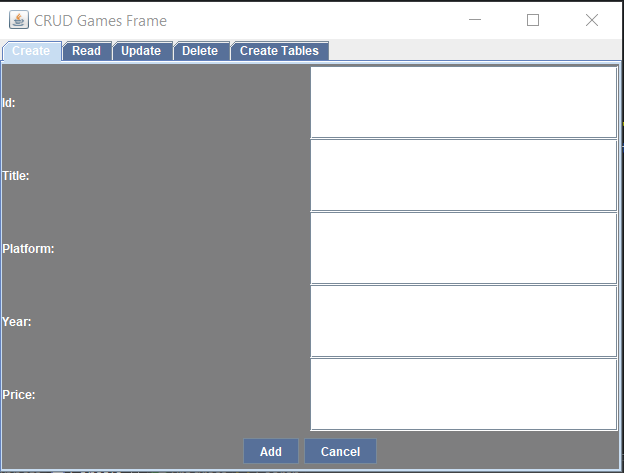
Donal Crotty

A00216737  Software Design (Cloud Computing)

Distributed systems assignment report

# GUI

The GUI I designed for this assignment to act as the client consists of a tabbed pane layout with 5 panes; Create, Read, Update, Delete and Create Tables.



# Create

The user can add a new game by filling out the relevant fields. They must give the game an ID, Title, Platform, Year and Price. Once the add button is clicked, a HTTP POST request is sent, and the game is added to the database as can be seen in the example documented in the below screenshots. Once the user submits the form, the fields will be cleared to give the user an indication that the PST request has been sent.

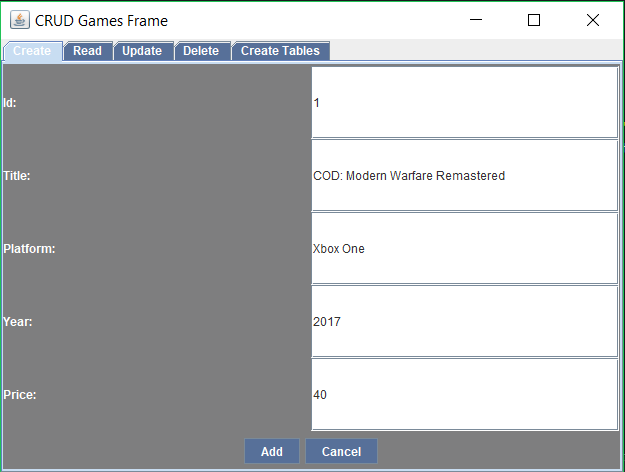


Figure - Create Tab- Add new game



Figure - 204 Status Code has been returned- POST success

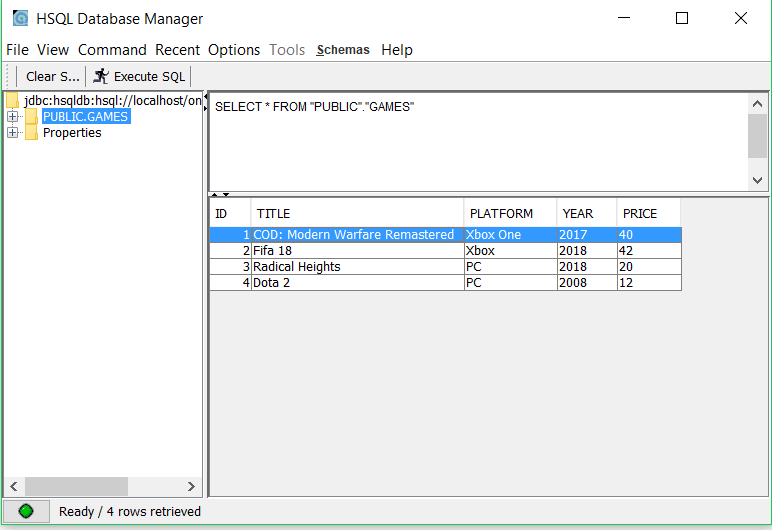


Figure - Game has been added to the DB

# Read

The user can retrieve all records using a GET request and also retrieve a single record using a GET by ID request. To return all records in the DB, the user just has to leave the ID field blank in the Read tabbed pane. For the user to retrieve a single request the user must specify an ID.

In the below example, all records in the database are shown and an example of the GET all and GET by id requests also.

For the read, I was unable to get the results to display in the correct format within the GUI for some unknown reason, however in the console they displayed. I feel as though the way I have implemented it should work, however I was unable to find any issue with it. Within the GUI, I could get the results to populate a textbox in XML format, but not after parsing it. I believe my issue may be within the ParseGame.java file.

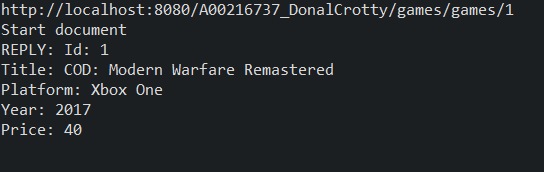


Figure - Get by ID displaying in console

## Update

The user can update their record by specifying the ID of the record they wish to update along with what they would like to update the record to in the Update tabbed pane. For example, I wish to update “Fifa 18” with an ID of 2 to “Fifa 18 UPDATED”. This is achieved by submitting a HTTP Put request to the url: <http://localhost:8080/A00216737_DonalCrotty/games/games/2> via the GUI. If a record with the specified ID does not exist, a POST request instead will be sent creating a new record with the input.

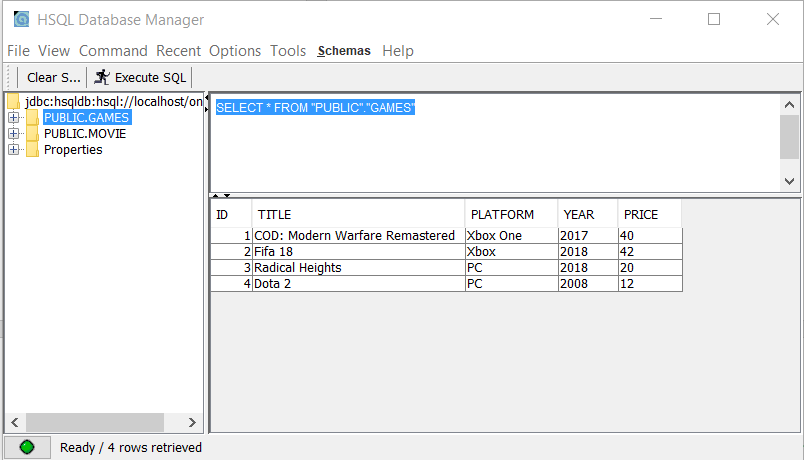


Figure - Table prior to update being performed

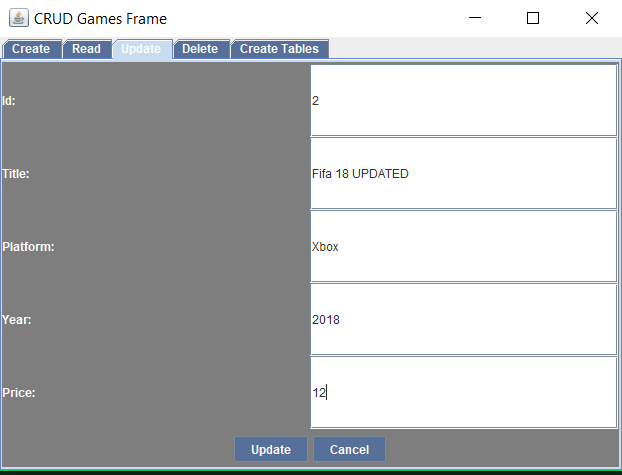


Figure - Update tabbed pane

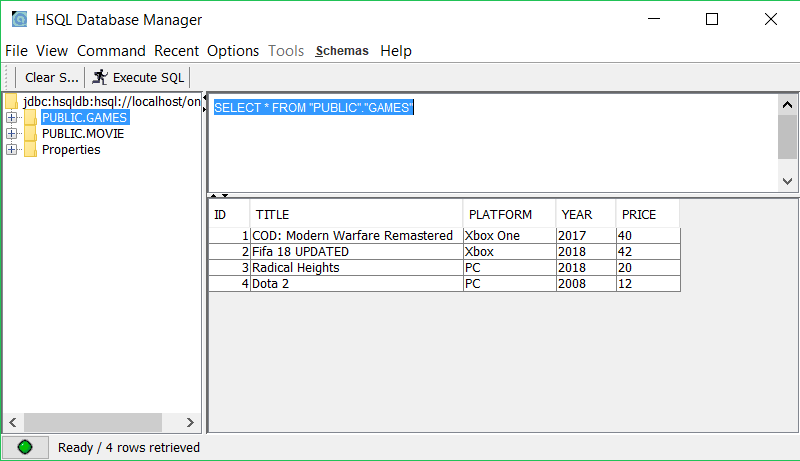


Figure - Record 2's Title has been updated to "Fifa 18 UPDATED"

# Delete

If the user wishes to delete a record they have two options; delete all records or delete one record. To delete all records from the table, they simply click the delete button without specifying an ID. If they wish to only delete one record, they must specify the ID of the record they wish to delete.

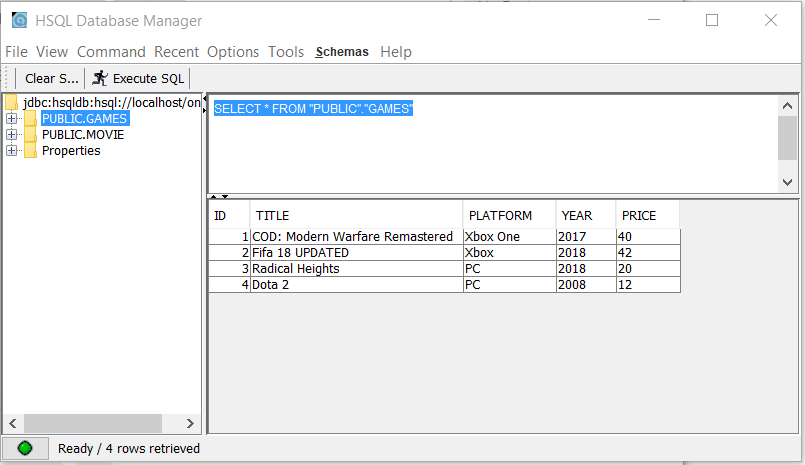


Figure - Table before delete by ID

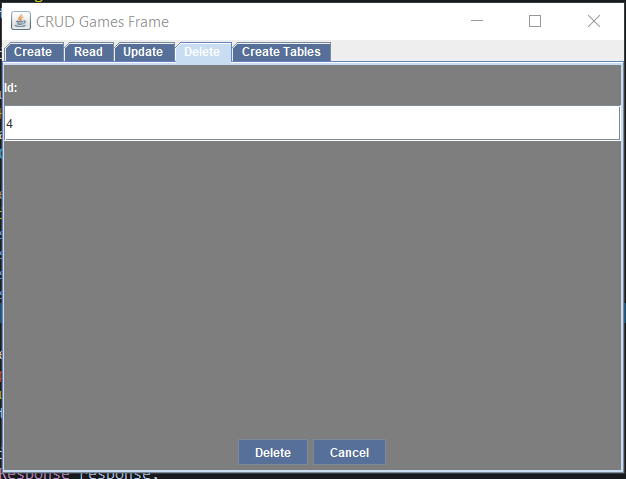


Figure - Delete Record with ID of 4 from DB

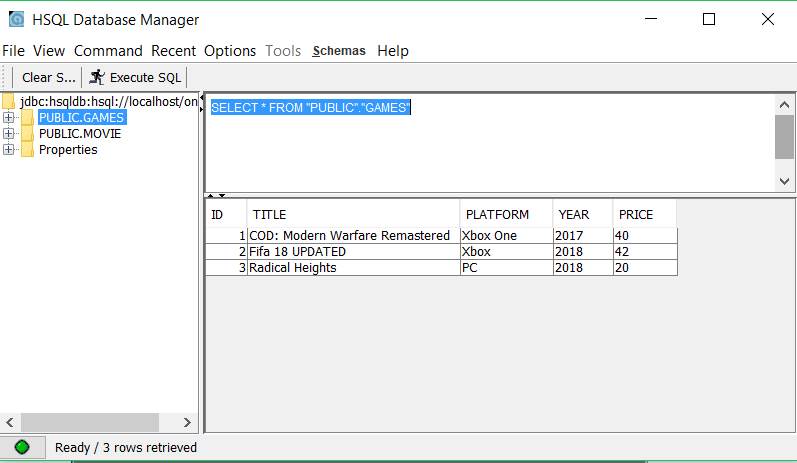


Figure - Record with an ID of 4 has been deleted from the HSQL DB

# Create Tables

The user can create a table during runtime by accessing the Create Tables tabbed pane. They simply click the button and a Movie table will be created and populated with multiple records. If the table already exists, it will be dropped, and a new table will be created.

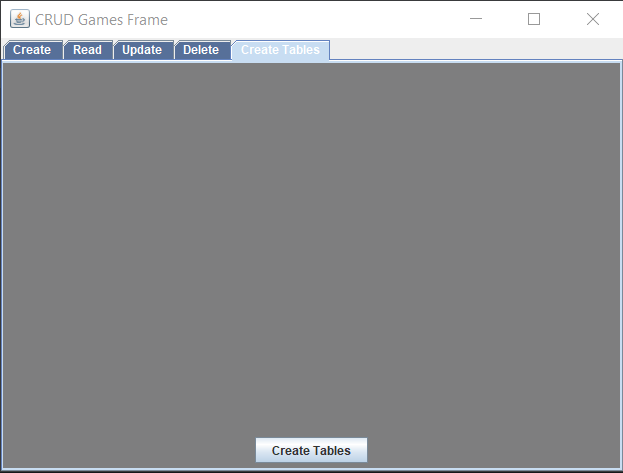


Figure - Create Tables tabbed pane

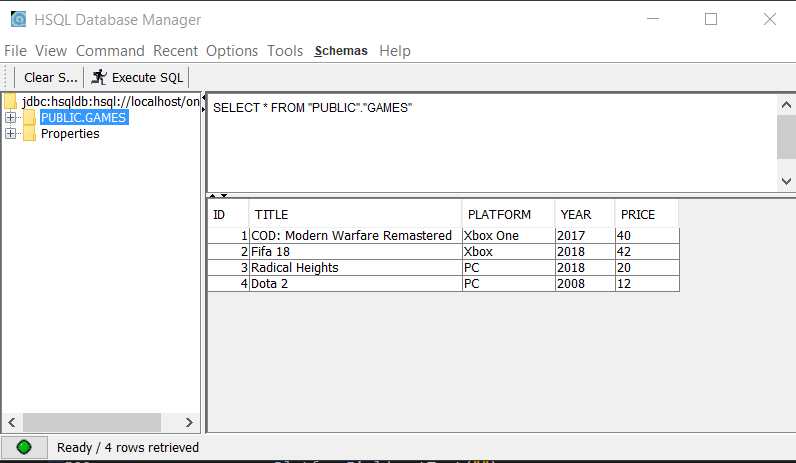


Figure - HSQL DB prior to tables being created during runtime

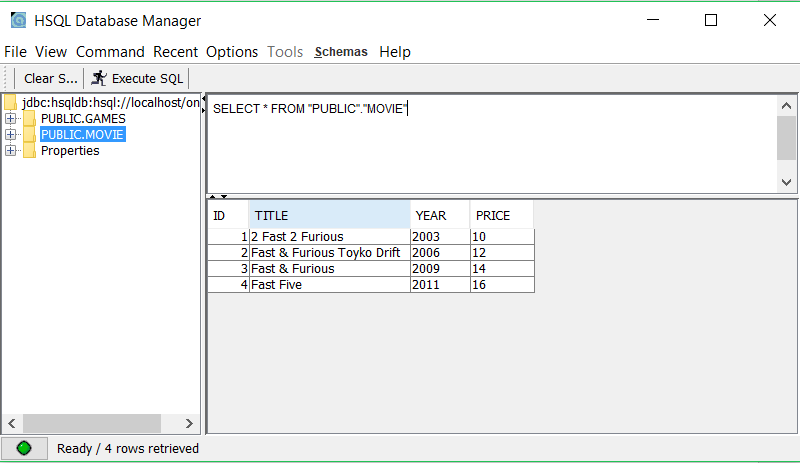


Figure - New Movie Table has been created and populated. Games table still exists also.