

Aaron Vance & Kaya Pina

CS365

Prof. Islam

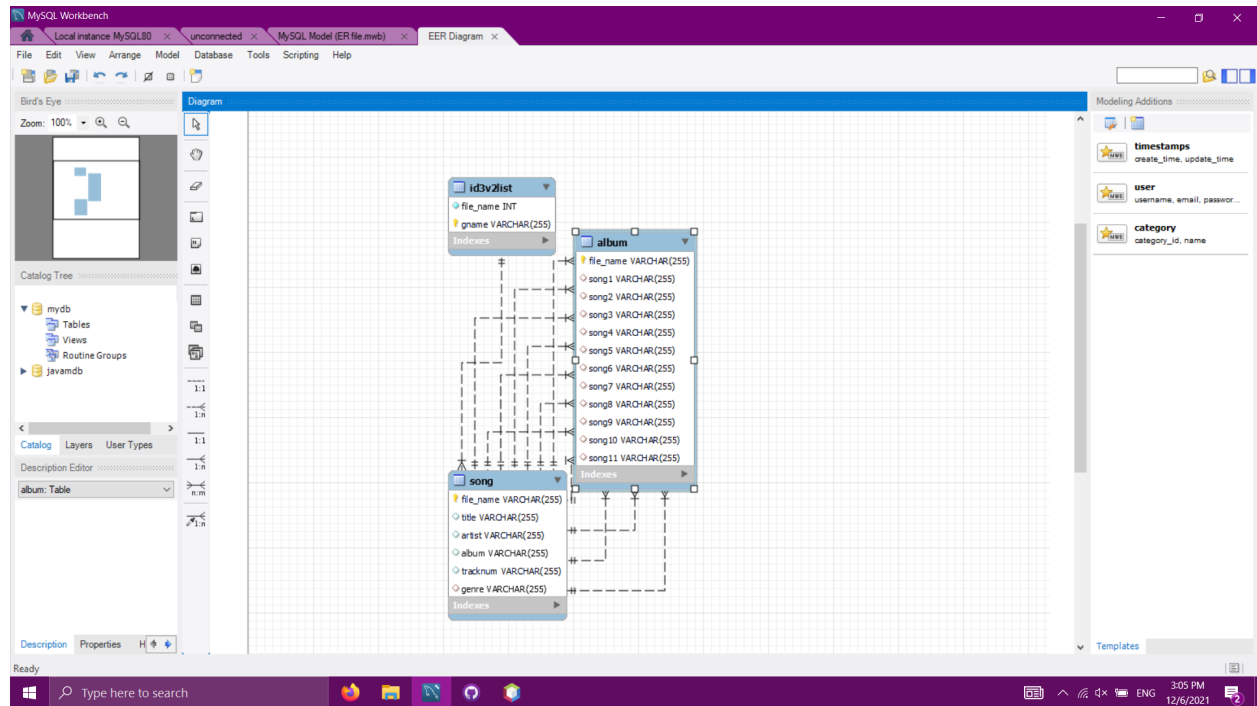
6 December 2021

### Term Project

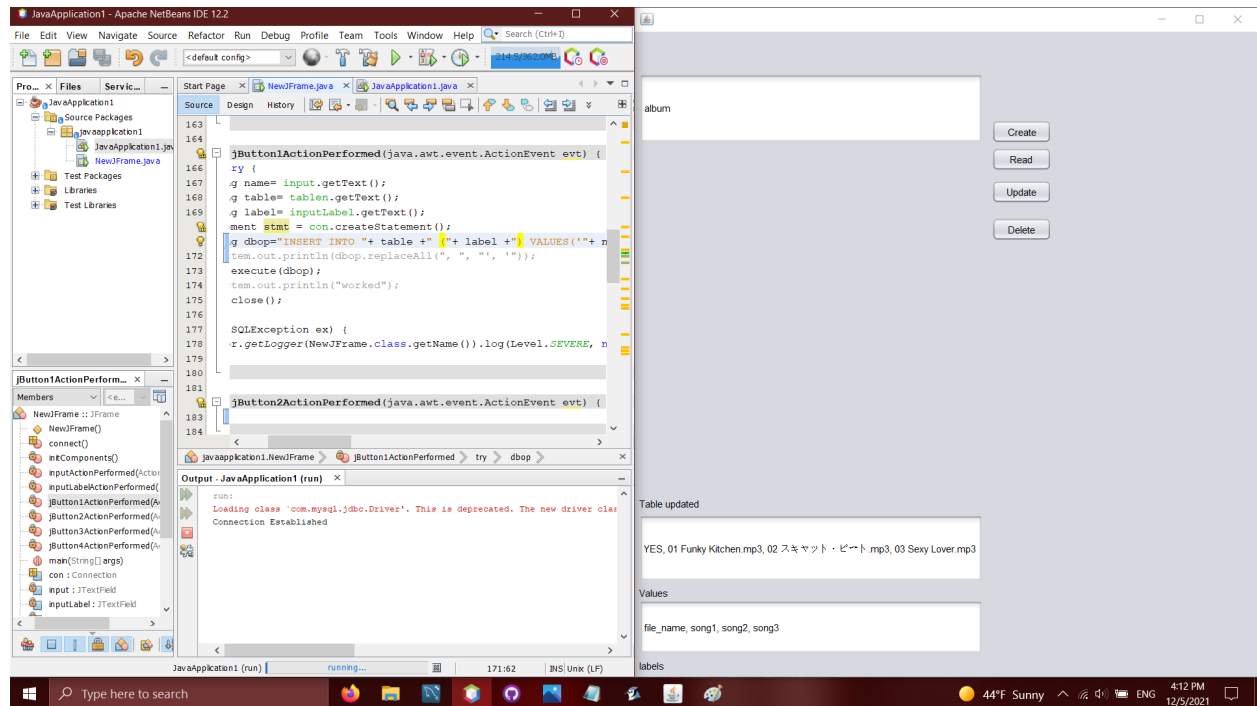
The basis of this project was to make an interactive database that is used to organize music files such as: artist, album, genre, track numbers, etc. This was based on observations about the iD3v2.4 metadata header and its shortcomings. It is the standardized format for audio files. For this project specifically, we used Java and SQL in which a connector was downloaded so Java and SQL were able to communicate with one another. In the Java file, we coded it with create, read, update, and delete statements that the user could operate from a GUI made using JFrame. Java would then successfully connect to an existing database in SQL and relay the CRUD operations. Within each of these statements in Java, multiple queries were executed as well. This is limited by the GUI, we did not want to copy and paste SQL syntax directly, that would render the GUI redundant. The GUI instead has the user input formatted plain text, which the Java application formats to make queries. Because of this, the GUI favors simple queries.

Within the SQL files is where the tables are made with column names and then further populated with the values for those columns. In this specific case, a table “song” was created with the column names: file\_name, title, artist, album, track number, and genre. Another table “album” was created with the column names: file\_name, which is used to store the album name, and song1-song11. Song1-song11 are foreign keys and need to be in the song table under file\_name. A genre table was also made, it has two columns for an identifying number and a name of the genre. This was made using the iD3v2.4 specifications from the winamp 5.6 update. A script was then created where each of these tables and they were populated with their respective values.

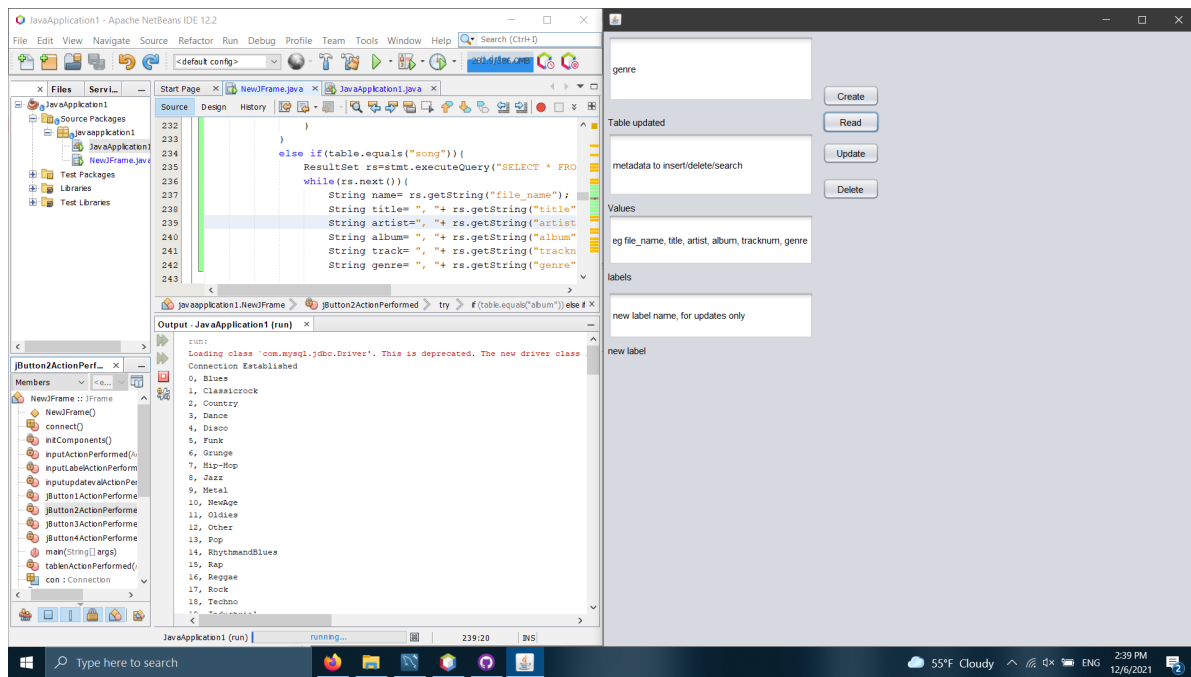
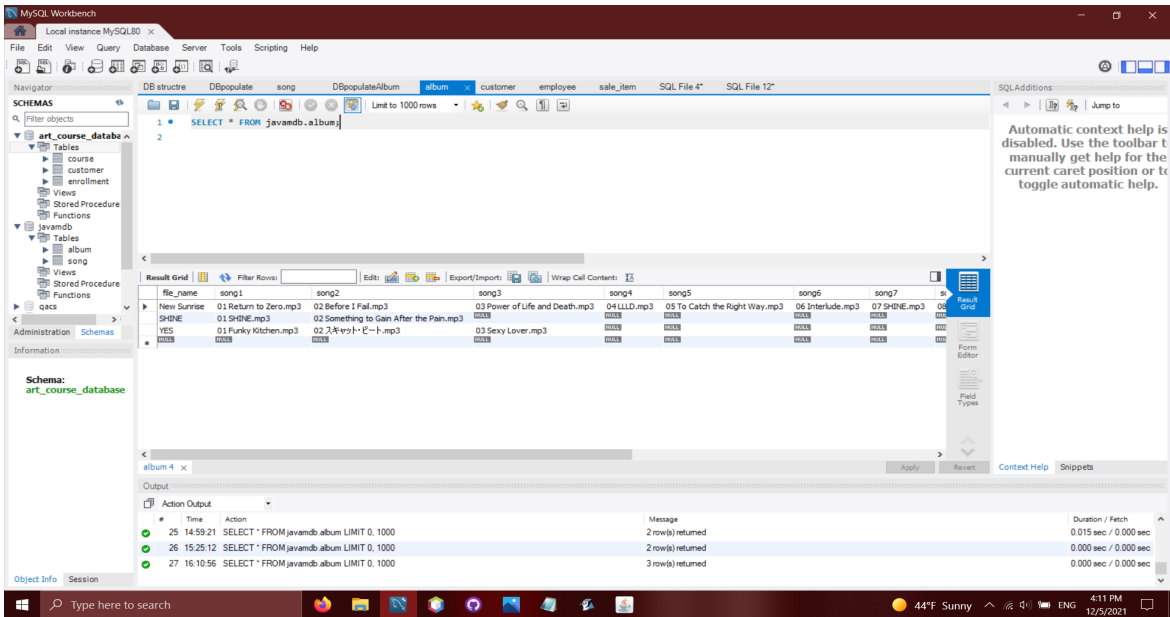
## Screenshot 1: ER Diagram



## Screenshot 2: Create



## Screenshot 3: Read



Screenshot 4: Update

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

DB structure\* DBpopulate song DBpopulateAlbum album DBid3 genre customer employee sale\_item SQL File 4\* SQL File 12\*

1 • SELECT \* FROM javamdb.album;  
2 • update album set song4=null where file\_name='YES';

Limit to 1000 rows

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid

file_name	song1	song2	song3	song4	song5	song6	song7	song8
New Sunrise	01 Return to Zero.mp3	02 Before I Fall.mp3	03 Power of Life and Death.mp3	04 LLLD.mp3	05 To Catch the Right Way.mp3	06 Interlude.mp3	07 SHINE.mp3	08
SHINE	01 SHINE.mp3	02 Something to Gain After the Pain.mp3						
YES	01 Funky Kitchen.mp3	02 2.4444444444444444.mp3	03 Sexy Lover.mp3					

Table: genre

Columns:

- file\_name int PK
- gname varchar(255)

Output

#	Time	Action	Message	Duration / Fetch
233	18:50:31	update album set song4=null where file_name='YES'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.016 sec
234	18:50:35	SELECT * FROM javamdb.album LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
235	19:06:26	SELECT * FROM javamdb.album LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

JavaApplication1 - Apache NetBeans IDE 12.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl-F)

Source

```

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        String name= input.getText();
        String table= tablen.getText();
        String label= inputLabel.getText();
        String updateval= inputupdateval.getText();
        Statement stat = con.createStatement();
        String dbop="update "+ table +" set "+label+"="+updateval;
        System.out.println(dbop);
        stat.executeUpdate(dbop);
        //System.out.println("worked");
        stat.close();
    } catch (SQLException ex) {
        Logger.getLogger(NewJFrame.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

Output - JavaApplication1 (run)

```

run:
Loading class 'com.mysql.jdbc.Driver'. This is deprecated. The new driver class
Connection Established
update album set song4='Back in Black.mp3' where file_name='YES';

```

album

Table updated

YES

Values

song4

labels

Back in Black.mp3

new label

Create

Read

Update

Delete

JavaApplication1 (run)

MySQL Workbench - Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas, art\_course\_database, javamdb, album, genre, song, Views, Stored Procedures, Functions, qacs, queen\_anne\_shop\_data, sakila, sys, world

Table: genre  
Columns: file\_name (int PK), gname (varchar(255))

Query: 1. SELECT \* FROM javamdb.album; 2. update album set song4=null where file\_name='YES';

Result Grid: 7 rows, 8 columns (file\_name, song1, song2, song3, song4, song5, song6, song7)

Output: 234 18:50:35 SELECT \* FROM javamdb.album LIMIT 0, 1000 3 row(s) returned 0.000 sec / 0.000 sec

Screenshot 4: Delete

MySQL Workbench - Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

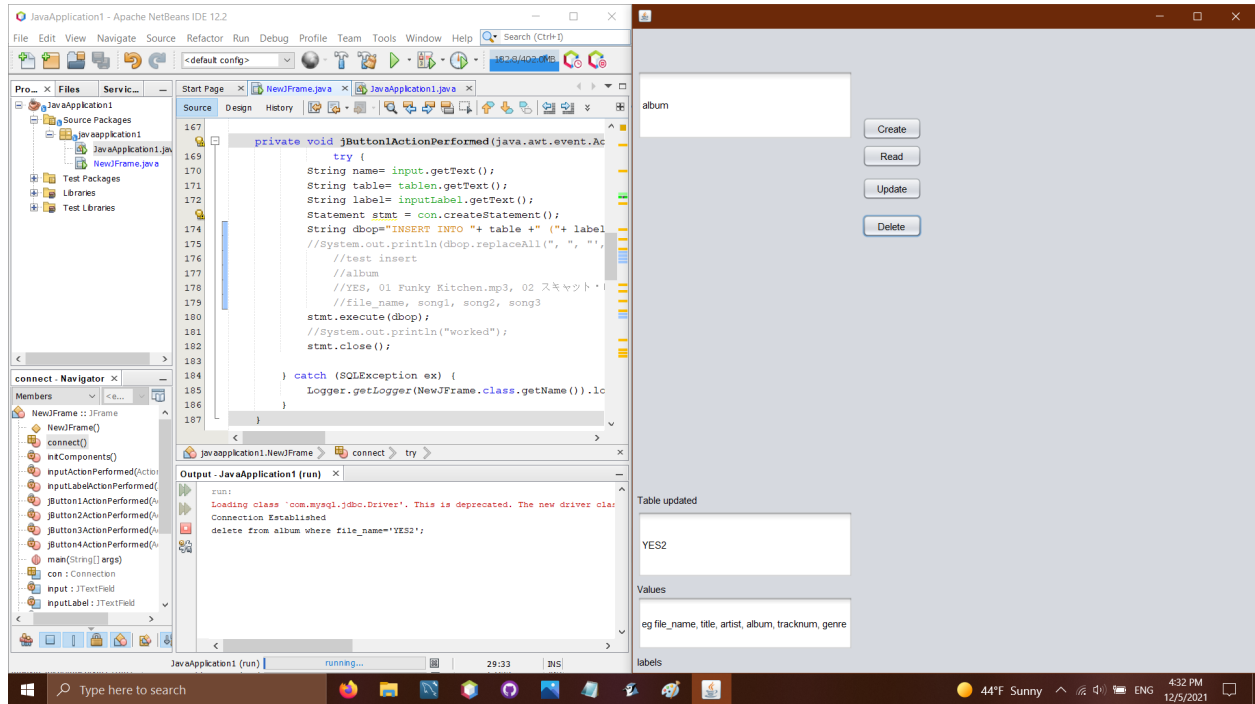
Navigator: Schemas, art\_course\_database, javamdb, album, genre, song, Views, Stored Procedures, Functions, qacs, queen\_anne\_shop\_data, sakila, sys, world

Table: genre  
Columns: file\_name (int PK), gname (varchar(255))

Query: 1. SELECT \* FROM javamdb.album; 2. update album set song4=null where file\_name='YES';

Result Grid: 7 rows, 8 columns (file\_name, song1, song2, song3, song4, song5, song6, song7)

Output: 27 16:10:56 SELECT \* FROM javamdb.album LIMIT 0, 1000 3 row(s) returned 0.000 sec / 0.000 sec



**Screenshot 6: Schema Diagram**

## Schema Diagram

---

Kaya Pina | December 6, 2021

