

CPSC 304 Project Cover Page

Milestone #: 4

Date: April 5, 2023

Group Number: 25

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Bryan Hui	53984746	x8f3l	bryanhui77@hotmail.com
Dhrubo Karmaker	13198379	w5u9r	dhruv.karmaker@gmail.com
Harper Kim	45579521	p0f7q	wowhkk@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 1, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia.

Description

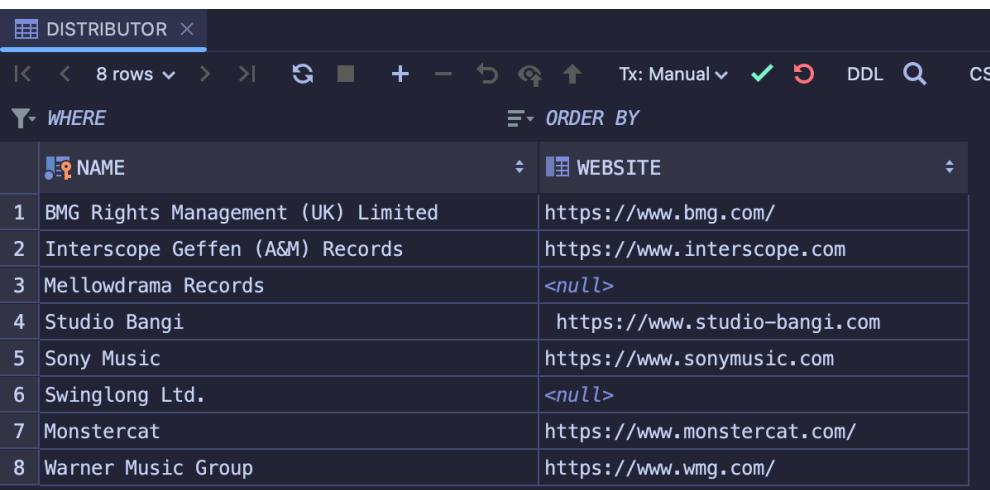
We have created a database that reflects models music streaming platforms. Our front-end application for the database allows users to search for and interact with music releases, such as adding songs to their library or playlists. Streaming service admins are able to view and manage content interactions, payments, and select tables from the database through the application as well.

Schema Differences

We have removed the ‘Following’ table and ‘MonthlyListeners’ attribute from the ‘Artist’ table as we believe that they are not necessary for our project to deliver a solid feature set. These removals do not affect the use-cases of our application in any way.

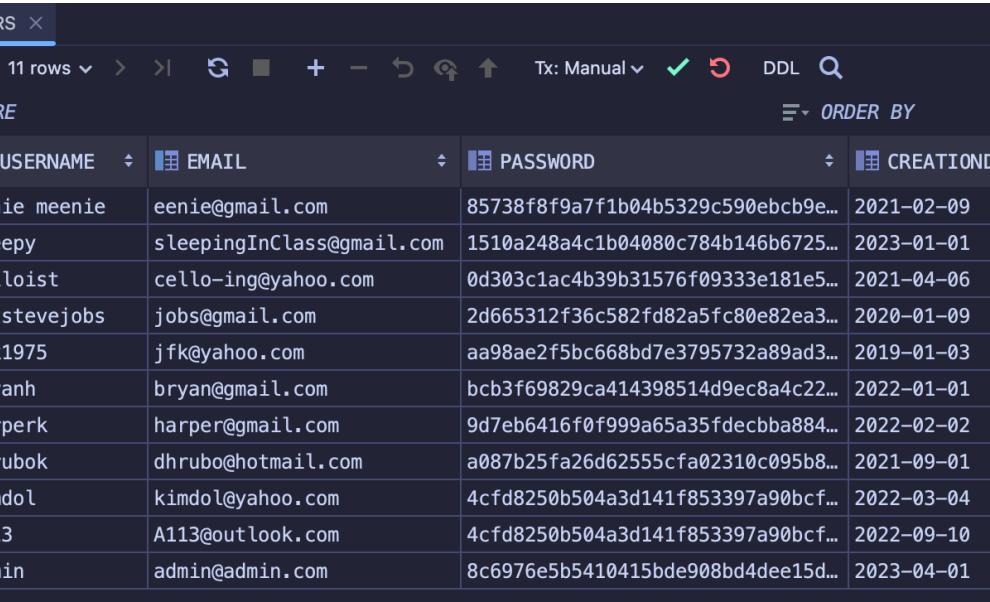
Schema Screenshots

Here are the screenshots of schemas after the SQL script is run.



The screenshot shows the 'DISTRIBUTOR' table with 8 rows of data. The columns are 'NAME' and 'WEBSITE'. The data is as follows:

	NAME	WEBSITE
1	BMG Rights Management (UK) Limited	https://www.bmg.com/
2	Interscope Geffen (A&M) Records	https://www.interscope.com
3	Mellowdrama Records	<null>
4	Studio Bangi	https://www.studio-bangi.com
5	Sony Music	https://www.sonymusic.com
6	Swinglong Ltd.	<null>
7	Monstercat	https://www.monstercat.com/
8	Warner Music Group	https://www.wmg.com/



The screenshot shows the 'USERS' table with 11 rows of data. The columns are 'USERNAME', 'EMAIL', 'PASSWORD', and 'CREATIONDATE'. The data is as follows:

	USERNAME	EMAIL	PASSWORD	CREATIONDATE
1	eenie meenie	eenie@gmail.com	85738f8f9a7f1b04b5329c590ebcb9e...	2021-02-09
2	sleepy	sleepingInClass@gmail.com	1510a248a4c1b04080c784b146b6725...	2023-01-01
3	celloist	cello-ing@yahoo.com	0d303c1ac4b39b31576f09333e181e5...	2021-04-06
4	notstevejobs	jobs@gmail.com	2d665312f36c582fd82a5fc80e82ea3...	2020-01-09
5	jfk1975	jfk@yahoo.com	aa98ae2f5bc668bd7e3795732a89ad3...	2019-01-03
6	bryanh	bryan@gmail.com	bcb3f69829ca414398514d9ec8a4c22...	2022-01-01
7	harperk	harper@gmail.com	9d7eb6416f0f999a65a35fdecbb884...	2022-02-02
8	dhrubok	dhrubo@hotmail.com	a087b25fa26d62555cf02310c095b8...	2021-09-01
9	kimdol	kimdol@yahoo.com	4cf8250b504a3d141f853397a90bc...	2022-03-04
10	A113	A113@outlook.com	4cf8250b504a3d141f853397a90bc...	2022-09-10
11	admin	admin@admin.com	8c6976e5b5410415bde908bd4dee15d...	2023-04-01

FREEUSER	
WHERE	
1 harperk	13
2 dhrubok	16
3 kimadol	0
4 jfk1975	13
5 eenie meenie	6

PREMIUMUSER			
WHERE			
ORDER BY			
1 bryanh	2022-03-01	2023-03-01	
2 A113	2022-02-01	2023-02-01	
3 sleepy	2023-03-01	2024-02-01	
4 celloist	2023-03-01	2024-02-01	
5 notstevejobs	2021-03-01	2022-02-01	
6 admin	2023-04-01	2024-04-01	

RELEASES					
WHERE ORDER BY					
1	1 Whenever You Need Somebody	Album	1987-11-12	https://cdn.spotube.com/img/rickroll.png	BMG Rights Management (UK) Limi...
2	2 Astro Lounge	Album	1999-06-08	https://cdn.spotube.com/img/shrek.png	Interscope Geffen (A&M) Records
3	3 Curb Your Enthusiasm (Music Fro...	Album	2006-04-17	https://cdn.spotube.com/img/frolic.png	Mellowdrama Records
4	4 Tritio Jatra	Album	2006-04-17	https://cdn.spotube.com/img/nemesis.png	Studio Bangi
5	5 Superunknown	EP	2006-04-17	https://cdn.spotube.com/img/soundgarden.png	Sony Music
6	6 Take On Me (2017 Acoustic)	Single	1984-09-01	https://cdn.spotube.com/img/aha.png	Swinglong Ltd.
7	7 Meowfest	Album	2003-10-10	https://cdn.spotube.com/img/kitty.png	Monstercat
8	8 The Barker is in Town	EP	2003-01-01	https://cdn.spotube.com/img/doggo.png	Monstercat
9	9 Dogs Don't Suck	Single	2003-10-10	https://cdn.spotube.com/img/friends.png	Monstercat
10	10 Only Me	EP	2015-02-27	https://cdn.spotube.com/img/notuu.png	Warner Music Group
11	11 Sonder	EP	2015-08-12	https://cdn.spotube.com/img/sonder.png	Warner Music Group
12	12 Not A Chance	Single	2012-05-29	https://cdn.spotube.com/img/nac.png	Monstercat
13	13 Zeka	Single	2018-12-31	https://cdn.spotube.com/img/zeka.png	Sony Music
14	14 Basic	Single	2018-12-31	https://cdn.spotube.com/img/basic.png	Sony Music

SONG						
WHERE ORDER BY						
1	1	1	Never Gonna Give You Up	214	Pop	213422997
2	1	2	Whenever You Need Somebody	244	Pop	1842825
3	1	3	Together Forever	206	Pop	1504161
4	1	4	It Would Take a Strong Strong ...	221	Pop	805086
5	1	5	The Love Has Gone	260	Pop	316040
6	1	6	Don't Say Goodbye	249	Pop	84133
7	1	7	Slipping Away	242	Pop	544835
8	1	8	No More Looking for Love	243	Pop	415202
9	1	9	You Move Me	243	Pop	949624
10	1	10	When I Fall in Love	183	Pop	595731
11	2	1	Who's There	214	Alternative	150983
12	2	2	Diggin' Your Scene	189	Alternative	3448250
13	2	3	I Just Wanna See	183	Alternative	5094786
14	2	4	Waste	205	Alternative	8007989
15	2	5	All Star	190	Alternative	692894289
16	2	6	Satellite	218	Alternative	5261034
17	3	1	Frolic	210	Soundtrack	5359039
18	3	2	Bubba Dub Bossa	108	Soundtrack	92825
19	3	3	Beach Parade	100	Soundtrack	91884
20	3	4	For Whom the Bell Tolls	64	Soundtrack	93682
21	3	5	The Stranger	142	Soundtrack	77515
22	4	1	Kobe	254	Rock	249567
23	4	2	Bir	401	Rock	506029
24	4	3	Tritio Jatra	254	Rock	645791
25	4	4	Nirbashon	234	Rock	5359039
26	5	1	Let Me Down	313	Hard Rock	508989

25		4	4	Nirbashon		234	Rock	5359039
26		5	1	Let Me Down		313	Hard Rock	508989
27		5	2	My Wave		304	Hard Rock	623314
28		5	3	Fell On Black Days		396	Hard Rock	726962
29		6	1	Take On Me (2017 Acoustic)		184	Pop	6666852
30		7	1	Start Hop, Jump		294	Dance	1412892
31		7	2	Pounce Bounce House		287	Dance	3295024
32		7	3	Pawitivity		124	Pop	12895982
33		7	4	Tails Up, Tails Down		252	Dance	2932104
34		7	5	Meowrr		244	Alternative	292501
35		7	6	Just a Mere Kitten		355	Ballad	214214901
36		7	7	Show Your Stripes		294	Dance	125292
37		7	8	Not 4 You		204	Dance	25920
38		7	9	Kickback, Relax		292	Dance	242425
39		7	10	Meowrr Part 2		189	Dance	1412892
40		7	11	High on Catnip St.		193	Hip-hop	12591250
41		7	12	KitKat		290	Dance	215912590
42		7	13	Unlucky		182	Dance	12590125
43		7	14	Dogs Suck		450	Pop	12590125
44		7	15	Paws Off My Ending		900	Trance	2592050
45		8	1	Woof, Poof Roof		241	Electronic	1412892
46		8	2	The Barker is in Town		192	Hip-hop	3295024
47		8	3	Barked Up the Wrong Tree		285	Electronic	12895982
48		8	4	Cone of Shame		301	Classical	2932104
49		8	5	Wag, Wag		127	Pop	292501
50		9	1	Dogs Don't Suck		193	Dance	12521521
51		10	1	Would You		209	Pop	24121453
52		10	2	Only Me		124	Pop	9358249
53		10	3	Be Fine		158	Pop	2920959025
54		10	4	Blue UU		148	Pop	428948429
55		11	1	Sonder		294	Alternative	2392498
56		12	1	Not A Chance		359	Trance	29485299
57		13	1	Zeka		180	Bluegrass	2727920
58		14	1	Basic		153	Pop	2727920

ARTIST X

CREATES X

WHERE

	ID	NAME
1	1	Rick Astley
2	2	Smash Mouth
3	3	Curb Your Enthusiasm
4	4	Luciano Michelini
5	5	Nemesis
6	6	Soundgarden
7	7	a-ha
8	8	DJ KitKat
9	9	Mr. Barker
10	10	Not UU
11	11	Zeka
12	12	John Doe

RELEASEID ARTISTID

RELEASEID	ARTISTID
1	1
2	2
3	3
4	3
5	4
6	5
7	6
8	7
9	8
10	9
11	9
12	10
13	11
14	12
15	13
16	14

FEATUREDIN X

FEATUREDIN

WHERE ORDER BY

ARTISTID	RELEASEID	TRACKNUM
1	1	1
2	2	1
3	3	2
4	4	4
5	5	3

POSTALCOD...TYPROVINCE

WHERE ORDER BY

	CITY	PROVINCE	POSTALCODE
1	Vancouver	BC	V6E7F3
2	Vancouver	BC	V4E7F3
3	Vancouver	BC	V6E5E3
4	Vancouver	BC	V6E789
5	Vancouver	BC	V8E7K3
6	Vancouver	BC	V6E7K3
7	Vancouver	BC	V6E1K2
8	Calgary	AB	G5E5C6
9	Winnipeg	MB	H4R7J8
10	Toronto	ON	K9L0H1

CARDTABLE

WHERE ORDER BY

	CARDCOMPANY	CARDNUM	CARDEXPIRY
1	Visa	2394235323332438	2025-01-01
2	Amex	5434235323332438	2026-01-01
3	Mastercard	1234353233324332	2027-01-01
4	Visa	7893353233324342	2025-02-01
5	Amex	8989353233324023	2023-07-01

BILLINGADDRESS

WHERE ORDER BY

	STREETNUM	STREET	POSTALCODE
1	1234	Cambie st.	V4E7F3
2	1234	Camosun St.	V6E789
3	1234	Hastings st.	V8E7K3
4	1234	Joyce st.	V6E7F3
5	2345	Kenneth st.	G5E5C6
6	2829	Cambie st.	V6E7F3
7	2948	Cambie st.	V4E7F3
8	3235	Courtney St.	V6E5E3
9	3235	Joyce st.	V6E7F3
10	7843	Cambie st.	V6E7F3
11	7843	Manitoba st.	V6E1K2
12	8343	Montgomery st.	H4R7J8
13	8583	Elliot st.	K9L0H1

USERPAYMENT
X

<
< 5 rows
>
>|
↶
↷
↶ ↷ ↸ ↺ ↻
Tx: Manual
✓
↻
DDL
🔍

WHERE
ORDER BY

	ID	AMOUNT	Date	CARDNUM	USERNAME	STREETNUM	STREET	POSTALCODE
1	1	7.00	2022-03-01	2394235323332438	bryanh	3235	Joyce st.	V6E7F3
2	2	11.00	2022-03-01	5434235323332438	sleepy	1234	Joyce st.	V6E7F3
3	3	12.00	2022-02-01	1234353233324332	A113	1234	Cambie st.	V4E7F3
4	4	15.00	2022-02-01	7893353233324342	notstevejobs	1234	Cambie st.	V4E7F3
5	5	15.00	2022-02-01	8989353233324023	celloist	7843	Cambie st.	V6E7F3

COMPANYPAYMENT
X

<
< 5 rows
>
>|
↶
↷
↶ ↷ ↸ ↺ ↻
Tx: Manual
✓
↻
DDL
🔍

WHERE
ORDER BY

	ID	AMOUNT	Date	STREETNUM	STREET	POSTALCODE	DISTRIBUTORNAME
1	1	7000.00	2022-03-01	3235	Courtney St.	V6E5E3	Sony Music
2	2	1100.00	2022-03-01	1234	Camosun St.	V6E789	Studio Bangi
3	3	1200.00	2022-02-01	1234	Cambie st.	V4E7F3	BMG Rights Management (UK) Limi...
4	4	1500.00	2022-02-01	1234	Hastings st.	V8E7K3	Interscope Geffen (A&M) Records
5	5	1500.00	2022-02-01	7843	Manitoba st.	V6E1K2	Mellowdrama Records

ADDSLTOlibrary
X

<
< 16 rows
>
>|
↶
↷
↶ ↷ ↸ ↺ ↻
Tx: Manual
✓
↻
DDL
🔍

WHERE
ORDER BY

	RELEASEID	TRACKNUM	USERNAME	DOWNLOADED	LIKED
1		1	bryanh	0	1
2		1	bryanh	0	0
3		2	kimdol	0	1
4		4	sleepy	1	0
5		3	A113	1	1
6		4	A113	1	1
7		4	admin	1	1
8		4	eenie meenie	1	1
9		4	bryanh	1	1
10		4	harperk	1	1
11		4	dhrubok	1	1
12		4	kimdol	1	1
13		4	celloist	1	1
14		4	notstevejobs	1	1
15		4	jfk1975	1	1
16		4	sleepy	1	1

PLAYLIST

5 rows

	USERNAME	NAME
1	A113	alt jam
2	bryanh	lofi
3	eenie meenie	90s jam
4	jfk1975	80s glam metal
5	kimdol	study-session

PLAYLISTISIN

5 row: PLAYLISTISIN

WHERE ORDER BY

	USERNAME	NAME	RELEASEID	TRACKNUM
1	bryanh	lofi	1	1
2	bryanh	lofi	1	2
3	eenie meenie	90s jam	4	4
4	jfk1975	80s glam metal	3	1
5	kimdol	study-session	2	4

SQL Queries

INSERT

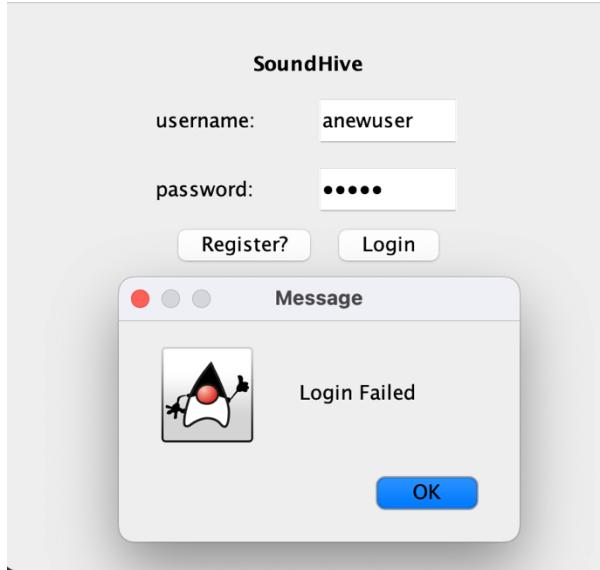
Query:

```
INSERT INTO USERS VALUES (<username>, <email>, <hashed-password>, <registration-date>);
```

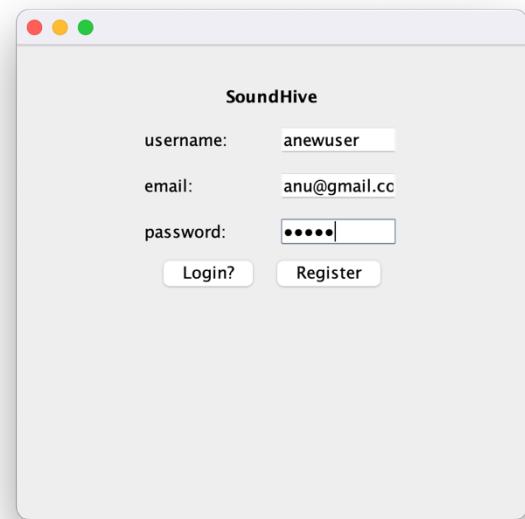
Location: src/backend/LoginRegistrationEndpoints.java/register(), Line 30

Description: Allows users to register an account with their username, email address, and password. This query uses prepared statements and hashed passwords to conceal sensitive information.

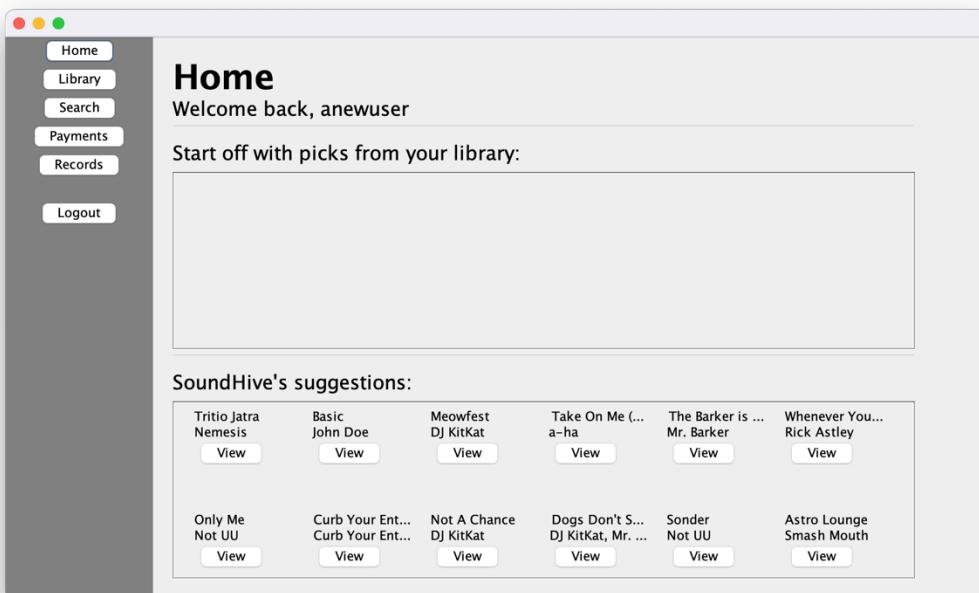
Before:



During:



After



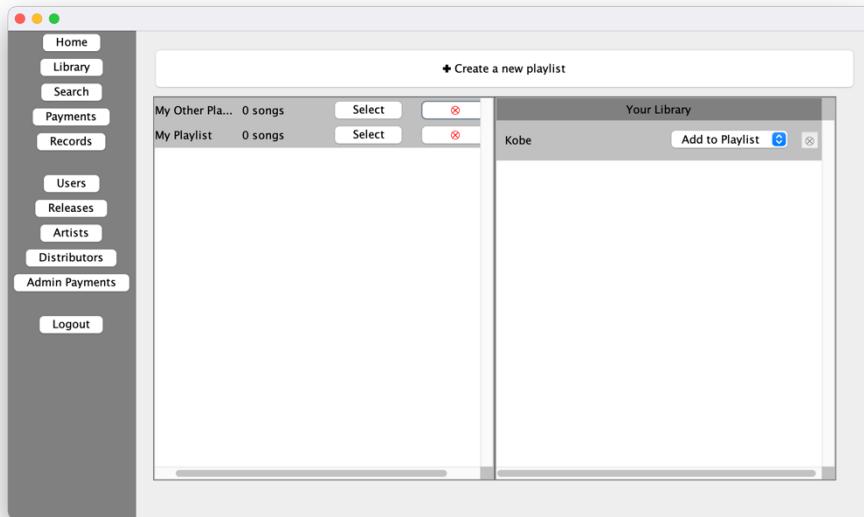
DELETE

```
"DELETE FROM Playlist p WHERE p.Username = ? AND p.Name = ?"
```

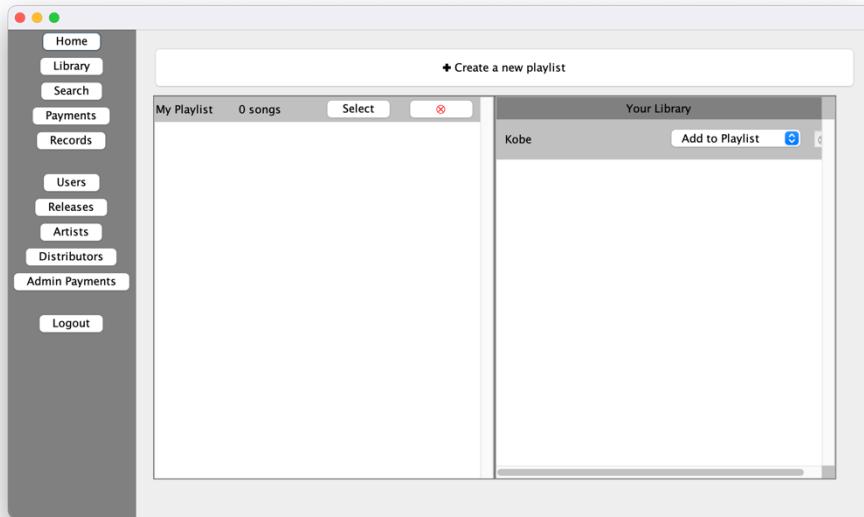
Location: src/backend/LibraryEndpoints.java/deletePI(), Line 150

Description: Allows a user to delete a playlist of their choice. Each playlist is represented as a panel in the library page on which there is an x button to delete the playlist of their choice.

Before:



After:



UPDATE

Query:

```
UPDATE Song SET
```

```
Name = '<name>', Duration = <duration>, Genre = '<genre>', Plays = <plays>  
WHERE ReleaseID = <release-id> AND TrackNum = <track-number>;
```

Location: src/backend/SongEndpoints.java/editSong(), Line 53

Description: Allows admins to update song information.

Before:

The screenshot shows a web application interface titled "Manage: The Barker is in Town". On the left is a sidebar with navigation links: Home, Library, Search, Payments, Records, Users, Releases, Artists, Distributors, Admin Payments, and Logout. The main content area has tabs for "Add Songs" and "Manage Artists". Below the tabs is a table listing five songs:

Track #	Name	Artist	Genre	Duration (s)	Actions
1	Woof, Poof Roof	Mr. Barker	Electronic	241	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
2	The Barker is in Town	Mr. Barker	Hip-hop	180	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
3	Barked Up the Wrong Tree	Mr. Barker	Electronic	216	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
4	Cone of Shame	Mr. Barker	Classical	216	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
5	Wag, Wag	Mr. Barker	Pop	120	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

To the right of the table is a modal window titled "Song Info" with the following fields:

Field	Value
Release ID:	8
Track #:	1
Name:	Woof, Poof Roof
Duration (s):	241
Genre:	Electronic
# of Plays:	1412892

After:

The screenshot shows the same web application interface after updating a song. The list of songs now shows the changes:

Track #	Name	Artist	Genre	Duration (s)	Actions
1	Zoof, Woof Hoof	Mr. Barker	Pop	180	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
2	The Barker is in Town	Mr. Barker	Hip-hop	180	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
3	Barked Up the Wrong Tree	Mr. Barker	Electronic	216	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
4	Cone of Shame	Mr. Barker	Classical	216	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
5	Wag, Wag	Mr. Barker	Pop	120	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

The "Song Info" modal still displays the previous values (Release ID: 8, Track #: 1, Name: Woof, Poof Roof, Duration (s): 241, Genre: Electronic, # of Plays: 1412892), indicating that the update was not yet saved.

Selection

Query:

```
SELECT DISTINCT Name, Website FROM Distributor WHERE  
LOWER(Name) LIKE LOWER('%<keywords>%') ORDER BY Name ASC;
```

Location: src/backend/DistributorEndpoints.java/SearchDistributors(), Line 43

Description: Allows admins search for distributors with names that contain specified keywords.

Before:

Name	Website	Actions
BMG Rights Management (UK) Limited	https://www.bmg.com/	[Edit] [Delete]
Interscope Geffen (A&M) Records	https://www.interscope.com	[Edit] [Delete]
Mellowdrama Records		[Edit] [Delete]
Monstercat	https://www.monstercat.com/	[Edit] [Delete]
Sony Music	https://www.sonymusic.com	[Edit] [Delete]
Studio Bangi	https://www.studio-bangi.com	[Edit] [Delete]
Swinglong Ltd.		[Edit] [Delete]
Warner Music Group	https://www.wmg.com/	[Edit] [Delete]

After:

Name	Website	Actions
Sony Music	https://www.sonymusic.com	[Edit] [Delete]
Warner Music Group	https://www.wmg.com/	[Edit] [Delete]
Interscope Geffen (A&M) Records	https://www.interscope.com	[Edit] [Delete]
Mellowdrama Records		[Edit] [Delete]

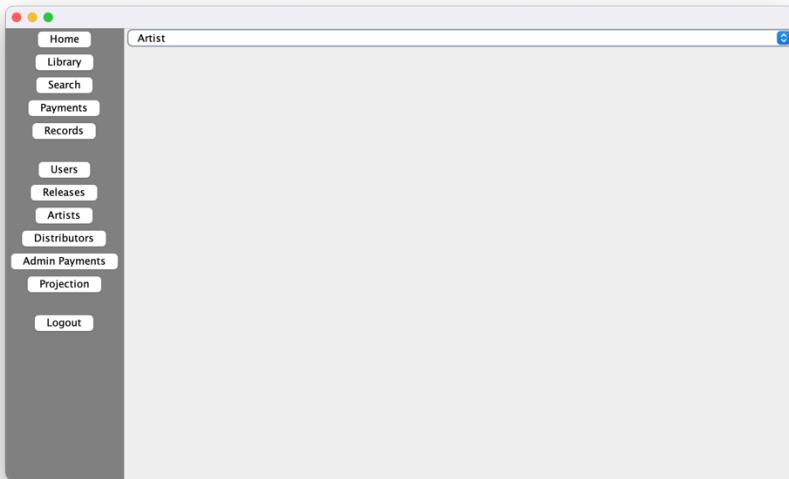
Projection

```
SELECT <selected-columns> FROM <selected-table>
```

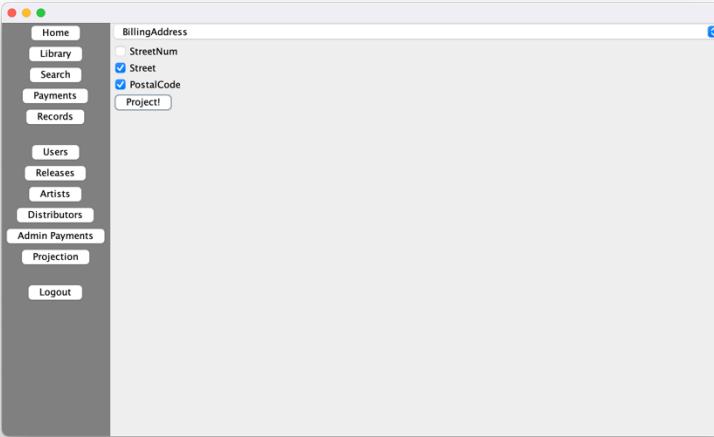
Location: src/backend/ProjectionEndpoint/getProjection(), Line 13

Description: Allows the admin to choose what they want to know about any table in the database. The admin can select which columns they would like to view.

Before:



During:



After:

Street	BillingAddress
Whatever	V8I0P
Cambie st.	V4E7F3
Camosun St.	V6E789
Hastings st.	V8E7K3
Joyce st.	V6E7F3
Kenneth st.	G5E5C6
Cambie st.	V6E7F3
Cambie st.	V4E7F3
Courtney St.	V6E5E3
Joyce st.	V6E7F3
Cambie st.	V6E7F3
Manitoba st.	V6E1K2
Montgomery st.	H4R7J8
Elliot st.	K9L0H1

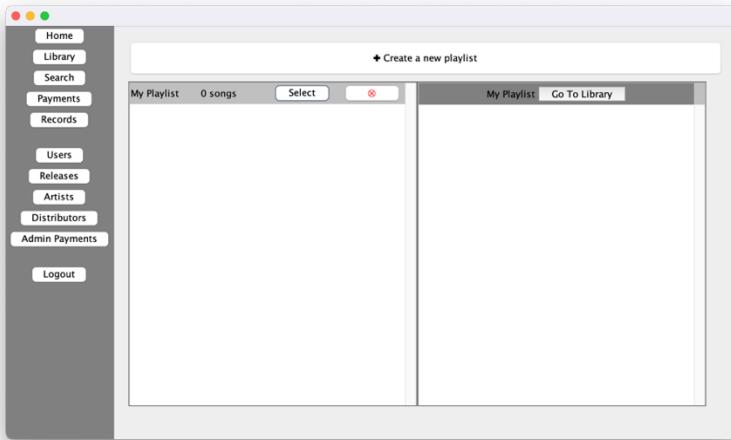
Join

```
SELECT s.ReleaseID, s.TrackNum, s.Name, s.Duration, s.Genre  
FROM Song s, PlaylistIsIn pi  
WHERE s.ReleaseID = pi.ReleaseID AND pi.TrackNum = s.TrackNum AND pi.Username =  
<username> AND pi.Name = <playlist-name>
```

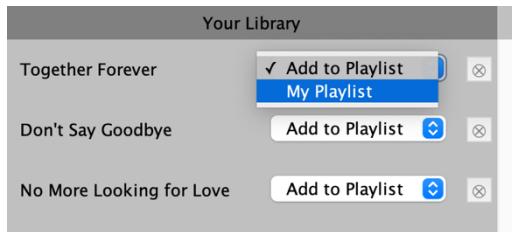
Location: src/backend/LibraryEndpoint.java/getPlaylistSongs(), Line 59

Description: allows user to get information about all songs belonging to a playlist, requiring a join between the song entity and the playlistisIn entity. This can be done in the library page.

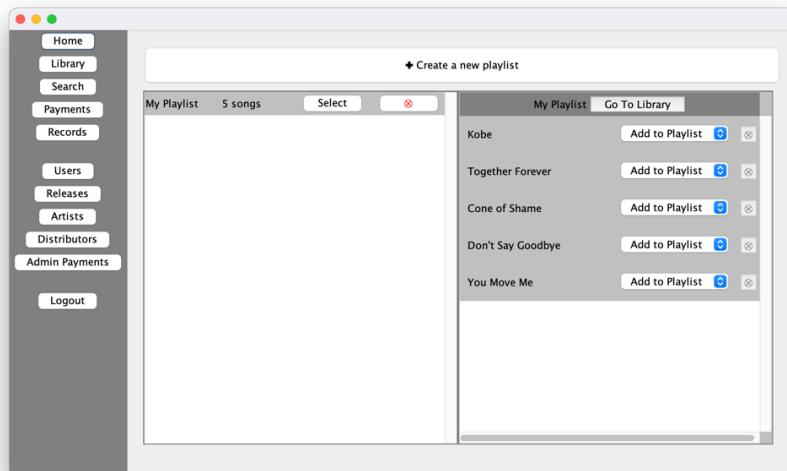
Before:



During:



After:



Aggregation with GROUP BY

```
"SELECT <aggregate identifier> as tabOne, COUNT(*) as count  
FROM <qualification> <WHERE clause, if needed>  
GROUP BY <grouping-list>"
```

Location: src/backend/QueryEndpoints/countConditionsGroupBy(), Line 174

Description: Allows user to choose what aggregate information they want to know about users, such as the number of playlists or songs in library each user has or how many users were created in a given year. Options provided by a dropdown in users page.

Before:

Generic User lookup:		Free Users	Premium Users
eenie meenie	eenie@gmail.com	Select	
sleepy	sleepingInClass@gmail.com	Select	
celloist	cello-ing@yahoo.com	Select	
notstevejobs	jobs@gmail.com	Select	
jfk1975	jfk@yahoo.com	Select	

Information about Users by Condition: Select Condition Submit

A113	1
admin	1
bryanh	1
eenie meenie	1
jfk1975	1

During:

Information about Users by Condition: ✓ Select Condition 1. Creation Date 2. # of Playlists 3. # of Songs in Library Submit

sleepy	
kimdol	

After:

Generic User lookup:		Free Users	Premium Users
eenie meenie	eenie@gmail.com	Select	
sleepy	sleepingInClass@gmail.com	Select	
celloist	cello-ing@yahoo.com	Select	
notstevejobs	jobs@gmail.com	Select	
jfk1975	jfk@yahoo.com	Select	

Information about Users by Condition: 3. # of Songs in Library Submit

sleepy	2
kimdol	2
celloist	1
notstevejobs	1
A113	2

The following three queries share the same image. If more entries qualify, they will be shown in this tab as well.

Before:



After:



Aggregation with HAVING

Query:

```
SELECT r.Name as Name
FROM Releases r, Song s
WHERE r.ID = s.ReleaseID
GROUP BY r.Name
HAVING SUM(s.Duration) > 3600;
```

Location: src/backend/ReleaseEndpoints.java/getReleasesOverHour(), Line 126

Description: Allows users to see which releases have runtimes of over 60 minutes.

Nested Aggregation with GROUP BY

Query:

```
CREATE VIEW CountSongsPerArtist
AS SELECT ID, Name, COUNT(*) as Count
FROM ((SELECT a1.ID as ID, a1.Name, c1.ReleaseID as RID, s1.TrackNum FROM Artist a1,
Song s1, Creates c1
WHERE a1.ID = c1.ArtistID AND c1.ReleaseID = s1.ReleaseID)
UNION (SELECT a2.ID as ID, a2.Name, s2.ReleaseID as RID, s2.TrackNum FROM Artist a2,
Song s2, FeaturedIn f2
WHERE a2.ID = f2.ArtistID AND s2.ReleaseID = f2.ReleaseID AND s2.TrackNum =
f2.TrackNum))
GROUP BY ID, Name;
```

```
SELECT c1.Name
FROM CountSongsPerArtist c1
WHERE c1.Count >= ALL (SELECT c2.Count FROM CountSongsPerArtist c2);
```

Location: src/backend/ArtistEndpoints.java/getArtistsMostSongs(), Line 91

Description: Allows users to see which artists have the highest number of songs contributed, created, and featured in.

Division

Query:

```
SELECT Name
FROM Song S
WHERE NOT EXISTS ((  
    SELECT Username FROM Users U) MINUS (
        SELECT A.Username
        FROM AddsToLibrary A
        WHERE A.Liked=1 AND A.ReleaseID = S.ReleaseID AND A.TrackNum = S.TrackNum
    ))
```

Location: src/backend/ReleaseEndpoints.java/getGoatedSongs(), Line 34

Description: Allows users to see which songs have been liked by every single user.