

ΤΕΧΝΟΛΟΓΙΑ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

Η Εργασία εξαμήνου:

Πληροφοριακό σύστημα μηχανογράφησης εταιρείας μουσικών παραγωγών

Μέλη ομάδας:

Τσερετέλι Γιώργος 164840 (tsereteligorgos@gmail.com)
Ελένη Μπαρμπαγιάννη 185239 (lena.mparmpa@gmail.com)
Κωνσταντίνος Τροκάνας 185418(trokanas.k@outlook.com)

~~~~~ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ~~~~~

Εισαγωγή	2
Περιγραφή Βάσης	3
Περιγραφή Εφαρμογής	5
Σελίδα Πελάτη(Customer Page)	6
Καλλιτέχνες(Artists)	6
Δισκογραφία(Albums) και Συναυλίες (Concerts)	8
Τραγούδια(Songs)	10
Πλαίσιο Αναζήτησης (Search)	11
Σελίδα Διαχειριστή (Admin Page)	14
Κουμπιά (Buttons) και λειτουργίες τους	17
Προσθήκη Καλλιτέχνη (Add Artist)	17
Διαγραφή Καλλιτέχνη (Delete Artist)	18
Προσθήκη Τραγουδιού (Add Song)	19
Διαγραφή Τραγουδιού (Delete Song)	19
Ενημέρωση Τραγουδιού (Update Song)	20
Ενημέρωση Δίσκου (Update Album)	20
Πίνακας Πωλήσεων (Total Sales)	20
Αρχείο Log File	21
Επίλογος	23

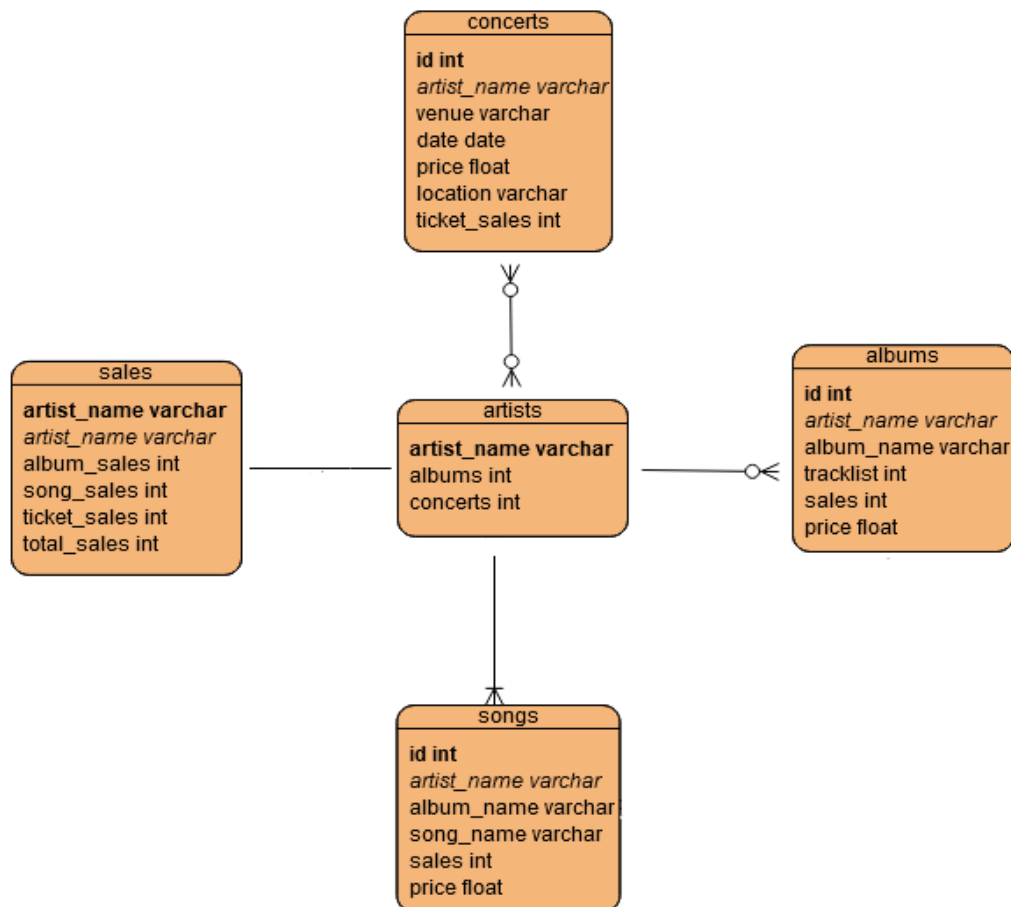
Εισαγωγή

Η εφαρμογή που φτιάξαμε είναι ένα σύστημα μηχανογράφησης εταιρείας μουσικών παραγωγών, το οποίο χωρίζεται σε δύο κατηγορίες: στο ***Admin Page*** και στο ***Customer Page***.

Η βάση λειτουργεί με γνώμονα τις δύο αυτές κατηγορίες, στις οποίες ο διαχειριστής (admin) μπορεί να επεξεργάζεται και να διαγράφει περιεχόμενα από τη βάση και πιο συγκεκριμένα μπορεί να προσθέτει καινούργια τραγούδια, τραγουδιστές, άλμπουμ και συναυλίες. Επιπρόσθετα, ο διαχειριστής όπως και ο πελάτης (customer), έχουν τη δυνατότητα να βλέπουν τους τραγουδιστές, τα άλμπουμ τους και τα τραγούδια που έχει κάθε άλμπουμ, μαζί με άλλα δεδομένα που υπάρχουν στη βάση, όπως οι πωλήσεις και το κόστος των άλμπουμ και των τραγουδιών τους.

Τέλος, στην εφαρμογή μας υπάρχει η δυνατότητα αναζήτησης του τραγουδιστή, των άλμπουμ και τραγουδιών του κάθε καλλιτέχνη. Παρακάτω αναλύεται η λειτουργικότητα και τα στοιχεία της βάσης καθώς και η λειτουργικότητα της εφαρμογής.

Περιγραφή Βάσης



Στην βάση υπάρχουν 5 πίνακες με ονόματα: **artists**, **sales**, **albums**, **songs** και **concerts**.

Αρχικά, στον πίνακα **artists** έχουμε τα πεδία για το όνομα του τραγουδιστή, τον αριθμό των άλμπουμ του, και των συναυλιών. Το πεδίο **artist_name** είναι το **κύριο κλειδί**.

Στον πίνακα **albums** υπάρχουν τα πεδία id, όνομα τραγουδιστή, όνομα του άλμπουμ, τον αριθμό των τραγουδιών στο άλμπουμ, πόσες φορές έχει αγοραστεί το άλμπουμ και η τιμή του. Το **id** είναι πάλι το **κύριο κλειδί**.

Στον πίνακα **songs** υπάρχει το id, το όνομα τραγουδιστή, το όνομα του άλμπουμ, το όνομα τραγουδιού, πόσες φορές έχει πωληθεί το τραγούδι και τέλος η τιμή του. Το **id** είναι ξανά το **κύριο κλειδί**.

Στον πίνακα **concerts** υπάρχει το id, το όνομα τραγουδιστή, το όνομα του σταδίου όπου πραγματοποιείται η συναυλία, η ημερομηνία, η τιμή των εισιτηρίων, η τοποθεσία όπου γίνεται η συναυλία και οι συνολικές πωλήσεις εισιτηρίων για αυτήν την συναυλία. Το **id** είναι το **κύριο κλειδί**.

Στον πίνακα **sales** έχουμε το όνομα του τραγουδιστή, τις συνολικές πωλήσεις των άλμπουμ, τις συνολικές πωλήσεις των τραγουδιών, τις συνολικές πωλήσεις εισιτηρίων, το σύνολο όλων των πωλήσεων μαζί. Όλοι οι πίνακες έχουν το **artist_name** από τον πίνακα artists ως **ξένο κλειδί**.

Οι πίνακες **artists-albums** και **artists-songs** έχουν σχέση ένα προς πολλά. Δηλαδή, ένας artist μπορεί να έχει πολλά άλμπουμ και τραγούδια, αλλά τα τραγούδια και τα άλμπουμ δεν μπορούν να έχουν πολλούς artists.

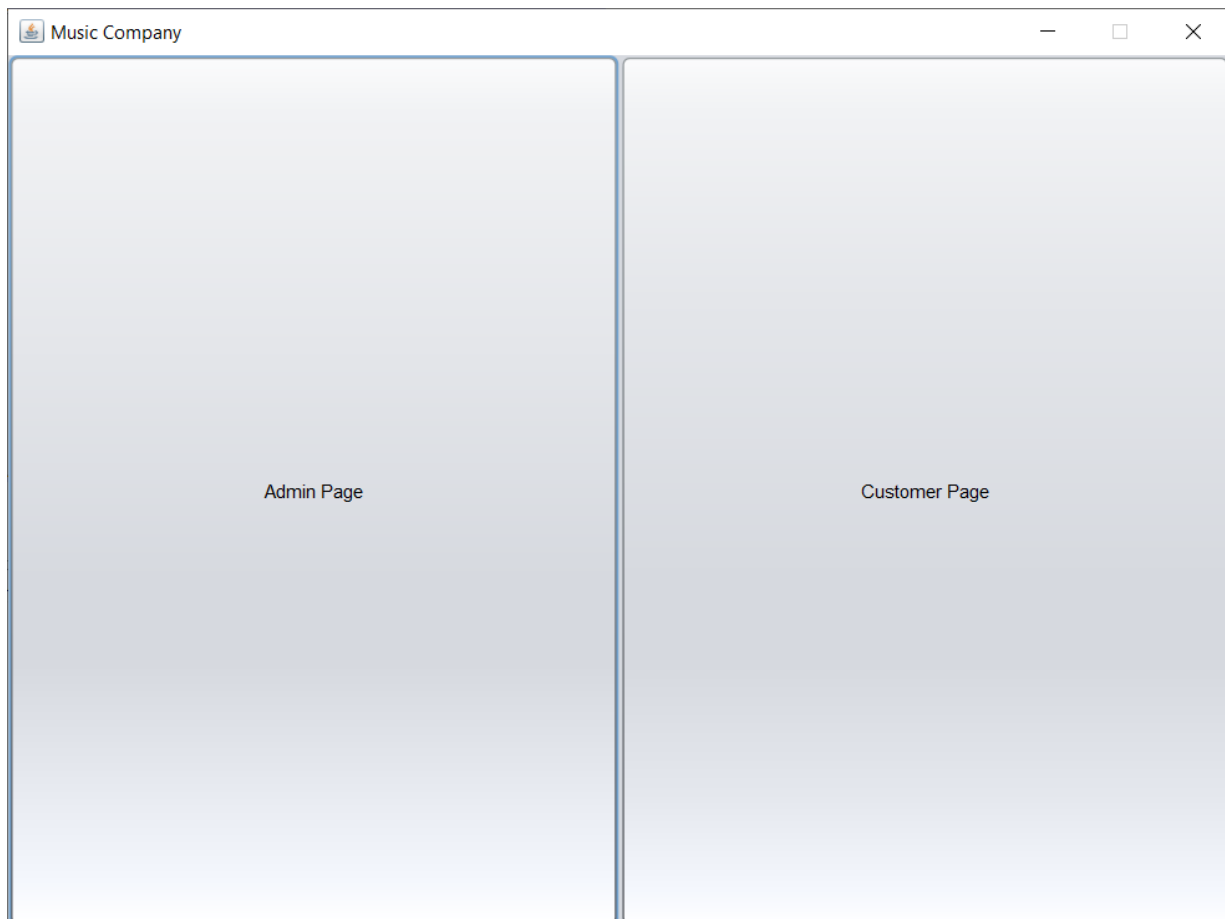
Οι πίνακες **artists-sales** έχουν σχέση ένα προς ένα. Δηλαδή, ένας artist μπορεί να έχει ένα από κάθε πεδίο, και κάθε πεδίο μπορεί να ανήκει σε έναν μόνο artist.

Οι πίνακες **artists-concerts** έχουν σχέση πολλά προς πολλά. Δηλαδή, πολλοί artists μπορούν να έχουν πολλές συναυλίες και πολλές συναυλίες μπορούν να έχουν πολλούς artists.

Περιγραφή Εφαρμογής

Αρχικά, έχουμε ένα πεδίο **jFrame home.java** στο οποίο εμφανίζεται ως τίτλος το όνομα της εταιρείας και από κάτω υπάρχουν δύο κουμπιά (buttons) με **jLabel**: **Admin Page** και **Customer Page**.

Πατώντας το κουμπί **Customer Page**, θα κλείσει το τρέχον **jFrame home.java** και θα ανοίξει το πεδίο **jFrame ArtistsTable.java**. Αντιθέτως, αν πατήσουμε το κουμπί **Admin Page** κλείσει το τρέχον **jFrame home.java** και θα ανοίξει το **Admin.java**.

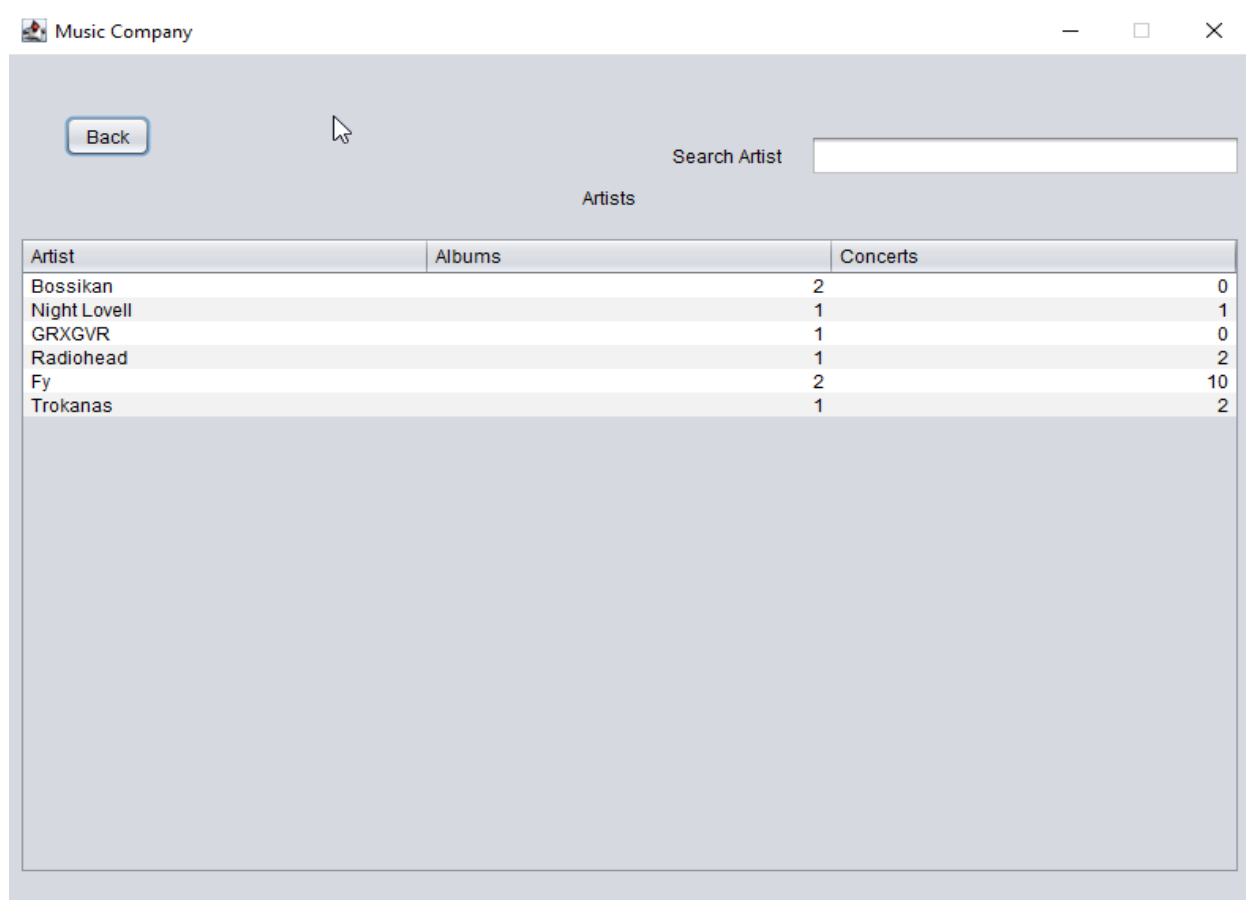


Σελίδα Πελάτη(Customer Page)

Η σελίδα πελάτη, εμπεριέχει τους πίνακες των καλλιτεχνών της εταιρείας, τα άλμπουμ των καλλιτεχνών αυτών καθώς και τις ανερχόμενες συναυλίες τους και τέλος τα τραγούδια που περιέχει κάθε άλμπουμ, όπως φαίνεται και αναλυτικότερα παρακάτω:

Καλλιτέχνες(Artists)

Όπως ήδη προαναφέρθηκε, μόλις πιάσουμε το κουμπί customer page μας μεταφέρει στο πεδίο JFrame ArtistsTable.java, μέσα στο οποίο υπάρχει ο πίνακας **Artists** με τα εξής πεδία: **Artist**, **Albums** και **Concerts**.



The screenshot shows a Java Swing window titled "Music Company". It features a "Back" button, a "Search Artist" text field, and a table labeled "Artists". The table has three columns: "Artist", "Albums", and "Concerts". The data in the table is as follows:

Artist	Albums	Concerts
Bossikan	2	0
Night Lovell	1	1
GRXGVR	1	0
Radiohead	1	2
Fy	2	10
Trokanas	1	2

Στα πεδία αυτά, εμφανίζονται οι τραγουδιστές που υπάρχουν στην βάση και στη συνέχεια τα αντίστοιχα πεδία θα γεμίσουν με τα δεδομένα που υπάρχουν στον **SQL** πίνακα “artists”, το γέμισμα γίνεται καλώντας μια **Stored Procedure**.

```
CREATE OR REPLACE FUNCTION selectartists()  
  RETURNS SETOF artists  
  LANGUAGE sql  
AS $$  
SELECT artist_name, albums, concerts FROM artists  
$$
```

```
statement := dbConnection.createStatement();  
String selectArtists = "SELECT * FROM selectArtists()";  
  
result = statement.executeQuery(selectArtists);
```

Τέλος, να μην παραλειφθεί πως πάνω αριστερά υπάρχει ένα κουμπί **Back**, το οποίο μας μεταφέρει πίσω στο αρχικό πεδίο JFrame home.java και πάνω δεξιά υπάρχει ένα JLabel **Search Artist** και ένα JTextField όπου μπορεί να γίνει η αναζήτηση τραγουδιστή από τον πίνακα που μας δίνεται, το οποίο θα αναλύσουμε στη συνέχεια.

Δισκογραφία(Albums) και Συναυλίες (Concerts)

Ενώ βρισκόμαστε ακόμη στον πίνακα με τα ονόματα των καλλιτεχνών, αν επιλέξουμε κάποιο από τα ονόματα των καλλιτεχνών που εμφανίζονται στον πίνακα, θα κλείσει το τρέχων πεδίο (jFrame) και στη συνέχεια θα εμφανιστεί ένα καινούργιο πεδίο (jFrame). Το JFrame αυτό, θα περιέχει έναν πίνακα με τα πεδία: **Album**, **Tracklist**, **Sales** και **Price** καθώς και έναν πίνακα με τις συναυλίες του καλλιτέχνη με τα πεδία: **Venue**, **Location**, **Date**, **Ticket Sales** και **Price**.

Music Company

Back

Search Album:

Albums

Album	Tracklist	Sales	Price
Concept Vague		15	1500

Concerts

Venue	Location	date	ticket sales	Price
Staples Center	1111 S Figueroa St, Los ...	2022-01-14	17300	30

Στο πεδίο θα εμφανιστούν τα άλμπουμ που έχει ο τραγουδιστής τον οποίο επιλέξαμε και τα πεδία θα γεμίσουν με τα δεδομένα που υπάρχουν στον SQL πίνακα Albums. Το γέμισμα γίνεται και εδώ καλώντας μια **Stored Procedure**, όπως φαίνεται παρακάτω.


```

CREATE OR REPLACE FUNCTION selectAlbums(character varying)
  RETURNS SETOF albums
  LANGUAGE sql
AS $$
SELECT *
FROM albums WHERE artist_name = $1;
$$

statement := dbConnection.createStatement();
String selectAlbums := "SELECT * FROM selectAlbums(?)";
prestatement := dbConnection.prepareStatement(selectAlbums);
prestatement.setString(1, ArtistName);
result := prestatement.executeQuery();

```

Όπως προαναφέρθηκε, κάτω από τον πίνακα Albums υπάρχει ο πίνακας **Concerts** με τα πεδία **Venue**, **Location**, **Date**, **Ticket Sales** και **Price**, όπου εμφανίζονται οι συναυλίες που πρόκειται να πραγματοποιηθούν, για τον καλλιτέχνη που έχουμε επιλέξει. Τα αντίστοιχα πεδία θα γεμίσουν με τα δεδομένα που υπάρχουν στον **SQL** πίνακα **Concerts**. Το γέμισμα για ακόμη μια φορά γίνεται καλώντας μια **Stored Procedure**.

```

CREATE OR REPLACE FUNCTION selectconcerts(character varying)
  RETURNS SETOF concerts
  LANGUAGE sql
AS $$
SELECT * FROM concerts WHERE artist_name = $1;
$$

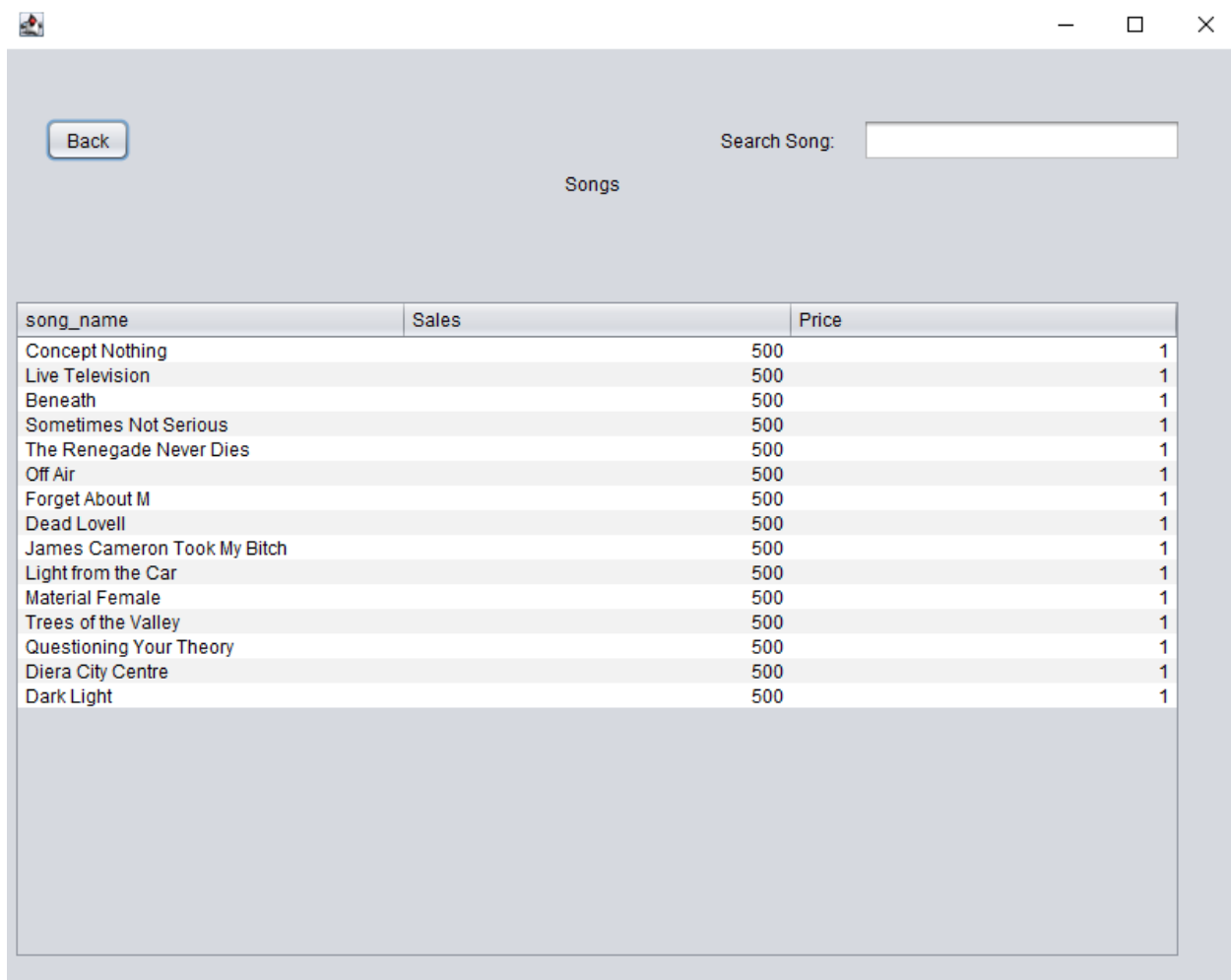
String selectConcerts := "SELECT * FROM selectConcerts(?)";
prestatement1 := dbConnection.prepareStatement(selectConcerts);
prestatement1.setString(1, ArtistName);
result := prestatement1.executeQuery();

```

Όπως και στο προηγούμενο JFrame, πάνω αριστερά υπάρχει το κουμπί **Back** το οποίο μας μεταφέρει στο προηγούμενο JFrame ArtistsTable.java και πάνω δεξιά υπάρχει ένα JLabel **Search Album** και ένα JTextField όπου μπορεί να γίνει η αναζήτηση των άλμπουμ που έχει ο συγκεκριμένος τραγουδιστής.

Τραγούδια(Songs)

Όπως και με όλα τα προηγούμενα jFrames, πατώντας πάνω σε κάποιο από τα πεδία από τον πίνακα **Albums**, θα κλείσει το τρέχων πεδίο (jFrame) και θα εμφανιστεί ένα καινούργιο πεδίο (jFrame) SongsTable.java με ένα πινάκα με τα πεδία: **Song name**, **Sales** και **Price** όπου εμφανίζονται τα τραγούδια που έχει το άλμπουμ του τραγουδιστή που επιλέξαμε, πόσες πωλήσεις των τραγουδιών έχουν γίνει και η τιμή του κάθε κομματιού.



The screenshot shows a Java Swing window titled "SongsTable.java". It contains a "Back" button on the left, a "Search Song:" label with an empty text input field on the right, and a table titled "Songs" in the center. The table has three columns: "song_name", "Sales", and "Price". It lists 15 songs, each with a sales value of 500 and a price of 1.

song_name	Sales	Price
Concept Nothing	500	1
Live Television	500	1
Beneath	500	1
Sometimes Not Serious	500	1
The Renegade Never Dies	500	1
Off Air	500	1
Forget About M	500	1
Dead Lovell	500	1
James Cameron Took My Bitch	500	1
Light from the Car	500	1
Material Female	500	1
Trees of the Valley	500	1
Questioning Your Theory	500	1
Diera City Centre	500	1
Dark Light	500	1

Τα πεδία του πίνακα θα γεμίσουν με τα δεδομένα που υπάρχουν στον **SQL** πίνακα Songs και το γέμισμα γίνεται καλώντας μια **Stored Procedure**.

```
CREATE OR REPLACE FUNCTION selectsongs(character varying)
  RETURNS SETOF songs
  LANGUAGE sql
AS $$
SELECT *
FROM songs WHERE album_name = $1;
$$

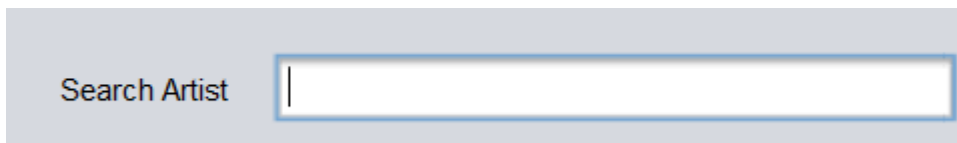
String selectSongs := "SELECT * FROM selectSongs (?) ";
prestatement := dbConnection.prepareStatement(selectSongs);
prestatement.setString(1, albumTest);
```

Όπως και στο προηγούμενο JFrame, πάνω αριστερά υπάρχει ένα κουμπί **Back** το οποίο μας μεταφέρει πίσω στο JFrame AlbumsTable.java και πάνω δεξιά υπάρχει ένα JLabel **Search Song** και ένα JTextField όπου μπορεί να γίνει η αναζήτηση τραγουδιών που ανήκουν στο άλμπουμ που έχουμε επιλέξει.

Πλαίσιο Αναζήτησης (Search)

Βλέπουμε ότι κατά την ανάπτυξη της εφαρμογής μας, σκοπός είναι η εύκολη και γρήγορη αναζήτηση των καλλιτεχνών, των άλμπουμ του καλλιτέχνη και των τραγουδιών του άλμπουμ από τον πελάτη.

Κάτι τέτοιο επιτυγχάνεται με ένα **πλαίσιο αναζήτησης (JTextField)**, το οποίο επιτρέπει στον χρήστη να πληκτρολογήσει το όνομα των πληροφοριών που ψάχνει και θα του επιστρέψει τις πληροφορίες που αναζητά ή το κατάλληλο μήνυμα μη εύρεσης δεδομένων, αν η αναζήτηση του χρήστη δεν υπάρχει.



Back

Search Artist

Artists

Artist	Albums	Concerts
Bossikan	2	0

Το πρόγραμμα δεν βρίσκει μόνο τους καλλιτέχνες όταν το όνομα τους γραφτεί έτσι όπως ακριβώς είναι, αλλά έχει τη δυνατότητα να βρει και τους καλλιτέχνες που η φράση αναζήτησής που πληκτρολογεί ο χρήστης εμπεριέχεται στο όνομά τους.

Back

Search Artist

Artists

Artist	Albums	Concerts
Radiohead	1	2

Το οποίο επιτυγχάνεται με τη χρήση ενός **Stored Procedure** με αυτόν τον τρόπο:

```
String findArtist = "SELECT * FROM findartist(?)";

PreparedStatement prestatement = dbConnection.prepareStatement(findArtist);
String searchQuery = "%" + searchBox.getText() + "%";
prestatement.setString(1, searchQuery);
result = prestatement.executeQuery();
```

```
CREATE OR REPLACE FUNCTION findartists(character varying, character varying)
RETURNS SETOF artists
LANGUAGE SQL AS $$
    SELECT *
    FROM artists WHERE LOWER(artist_name) LIKE LOWER($1);
$$
```

Τέλος, σε περίπτωση άκυρης αναζήτησης από τον χρήστη, το πρόγραμμα θα εμφανίσει **μήνυμα λάθους** σε ένα καινούργιο παράθυρο, ενημερώνοντας τον χρήστη ότι η αναζήτηση που έχει δώσει δεν υπάρχει .



```
if(result.isBeforeFirst() == false) {  
    System.out.println("There is not an artist with this name.");  
    JOptionPane.showMessageDialog(null, "There is not an artist with this name.");  
}
```

Στο **jFrame AlbumsTable.java** το **Stored Procedure** που χρησιμοποιείται είναι :

```
CREATE OR REPLACE FUNCTION findalbum(character varying, character varying)  
RETURNS SETOF albums  
LANGUAGE SQL AS $$  
    SELECT *  
    FROM albums WHERE artist_name = $1  
    AND LOWER(album_name) LIKE LOWER($2);  
$$
```

Στο **jFrame SongsTable.java** το **Stored Procedure** που χρησιμοποιείται είναι :

```

CREATE OR REPLACE FUNCTION findsong(character varying, character varying)
RETURNS SETOF songs
LANGUAGE SQL AS $$
    SELECT *
    FROM songs WHERE artist_name = $1
    AND LOWER(song_name) LIKE LOWER($2);
    $$

```

Με ένα `(jTextField)`, λοιπόν, στους πίνακες **ArtistsTable.java**, **AlbumsTable.java**, **SongsTable.java**, δίνεται η δυνατότητα στον χρήστη να αναζητήσει εύκολα και γρήγορα για τον καλλιτέχνη, το άλμπουμ ή το τραγούδι που ψάχνει.

Σελίδα Διαχειριστή (Admin Page)

Η σελίδα διαχειριστή, εμπεριέχει τα συστήματα διαχείρισης της εταιρείας για την εισαγωγή, επεξεργασία και διαγραφή τραγουδιών, καλλιτεχνών και άλμπουμ όπως φαίνεται και αναλυτικότερα παρακάτω:

The Admin Panel interface is divided into three main sections for managing music data: Song, Artist, and Album. Each section contains input fields for adding or updating records, and buttons for 'Add', 'Update', and 'Delete'.

Song Section: Includes input fields for Song, Artist, Album, Sales, and Price. Buttons: Add, Update, Delete.

Artist Section: Includes input fields for Artist, Albums, and Concerts. Buttons: Add, Delete.

Album Section: Includes input fields for Artist, Album, Tracklist, Sales, and Price. Button: Update.

Summary Tables: Two tables are displayed on the right side of the panel.

id	Song	Artist	Album	Sales	Price
1	4real	Bossikan	Ergo 1	200	1,99
2	Mavri Poli	Bossikan	Ergo 1	200	1,99
3	Bad Kaif	Bossikan	Ergo 1	200	1,99
4	Tussin	Bossikan	Ergo 1	200	1,99
5	Money Da...	Bossikan	Ergo 1	200	1,99
7	XIONANT...	Bossikan	Ergo 2	300	1,99
8	OXI DEN ...	Bossikan	Ergo 2	300	1,99
9	WILD REST	Bossikan	Ergo 2	300	1,99
10	BOUNCE	Bossikan	Ergo 2	300	1,99
11	PROVLIM...	Bossikan	Ergo 2	300	1,99
12	BIG CHE...	Bossikan	Ergo 2	300	1,99
13	MONEYF...	Bossikan	Ergo 2	340	1,99
14	KATARES	Bossikan	Ergo 2	330	1,99

id	Artists	Album	Tracklist	Sales	Price
1	Bossikan	Ergo 1	1	2000	10
2	Bossikan	Ergo 2	1	2500	15
3	Night Lovell	Concept V...	1	1500	10
4	GRXGVR	4Ever You...	1	500	10
5	Radiohead	A Moon Sh...	1	5000	15

At the bottom right, there are buttons: Home, Total Sales, Reset, and Exit.

Με ένα κλικ, το κουμπί admin page (**btnAdmin**) μας μεταφέρει στο πεδίο JFrame Admin.java, μέσα στο οποίο υπάρχει:

- ο πίνακας **Songs** με τα εξής πεδία: **Song**, **Artist**, **Album**, **Sales** και **Price**.
- ο πίνακας **Albums** με τα εξής πεδία: **id**, **Artist**, **Album**, **Tracklist**, **Sales** και **Price**
- Ένα κουμπί **Home** το οποίο μας επιστρέφει στο αρχικό **JFrame Home.java**
- ένα κουμπί **Reset** το οποίο καθαρίζει τα πεδία από τις φόρμες
- ένα κουμπί **Exit** το οποίο κλείνει την εφαρμογή
- ένα κουμπί **Total Sales** που εμφανίζει έναν πίνακα με τις συνολικές πωλήσεις

Πίνακας Τραγουδιών

id	Song	Artist	Album	Sales	Price
1	4real	Bossikan	Ergo 1	200	1.99
2	Mavri Poli	Bossikan	Ergo 1	200	1.99
3	Bad Kaif	Bossikan	Ergo 1	200	1.99
4	Tussin	Bossikan	Ergo 1	200	1.99
5	Money D...	Bossikan	Ergo 1	200	1.99
7	XIONANT...	Bossikan	Ergo 2	300	1.99
8	OXI DEN ...	Bossikan	Ergo 2	300	1.99
9	WILD RE...	Bossikan	Ergo 2	300	1.99
10	BOUNCE	Bossikan	Ergo 2	300	1.99
11	PROVLIM...	Bossikan	Ergo 2	300	1.99
12	BIG CHE...	Bossikan	Ergo 2	300	1.99
13	MONEYF...	Bossikan	Ergo 2	340	1.99
14	KATARES	Bossikan	Ergo 2	330	1.99
15	NAPOLI	Bossikan	Ergo 2	320	1.99
16	Concept ...	Night Lov...	Concept ...	500	1.99
17	Live Tele...	Night Lov...	Concept ...	500	1.99
18	Beneath	Night Lov...	Concept ...	500	1.99
19	Sometim...	Night Lov...	Concept ...	500	1.99
20	The Ren...	Night Lov...	Concept ...	500	1.99
21	Off Air	Night Lov...	Concept ...	500	1.99
22	Forget Ab...	Night Lov...	Concept ...	500	1.99
23	Dead Lov...	Night Lov...	Concept ...	500	1.99
24	James C...	Night Lov...	Concept ...	500	1.99
25	Light fro	Night Lov	Concept	500	1.99

Πίνακας Δίσκων

id	Artists	Album	Tracklist	Sales	Price
1	Bossikan	Ergo 1	1	2000	10
2	Bossikan	Ergo 2	1	2500	15
3	Night Lovell	Concept V...	1	1500	10
4	GRXGVR	4Ever You...	1	500	10
5	Radiohead	A Moon Sh...	1	5000	15

Στο ίδιο **jFrame Admin.java** βρίσκεται και η φόρμα διαχείρισης της βάσης, όπου γίνεται η **προσθήκη**, η **επεξεργασία** και η **διαγραφή** των δεδομένων.

Η φόρμα έχει χωριστεί σε τρεις υποδεέστερες **Song**, **Artist** και **Album** καθώς έτσι επιτυγχάνεται η σωστή λειτουργία της σελίδας διαχείρισης.

Στα πεδία αυτά, υπάρχουν τα **jTextFields** που αντιπροσωπεύουν τα columns των tables Songs για το πεδίο Add Song και το table Artist για το πεδίο Add Artist αντίστοιχα. Ο admin στην συνέχεια θα γεμίσει τα αντίστοιχα πεδία με τα δεδομένα τα οποία θα τροποποιήσουν την βάση ανάλογα.

The screenshot displays a Java Swing window titled 'Admin.java' with three main panels for managing data:

- Song Panel:** Contains input fields for 'Song', 'Artist', 'Album', 'Sales', and 'Price'. At the bottom are buttons for 'Add', 'Update', and 'Delete'.
- Artist Panel:** Contains input fields for 'Artist', 'Albums', and 'Concerts'. At the bottom are buttons for 'Add' and 'Delete'.
- Album Panel:** Contains input fields for 'Artist', 'Album', 'Tracklist', 'Sales', and 'Price'. At the bottom is an 'Update' button.

Στο πεδίο **Song** υπάρχουν τρία **JButton**, Add , Update και Delete με μεταβλητά ονόματα **btnAddSong** , **btnUpdate** και **btnDelete**.

Κουμπιά (Buttons) και λειτουργίες τους

Προσθήκη Καλλιτέχνη (Add Artist)

Για να μπορεί ο διαχειριστής να εισάγει καινούργιο τραγούδι κάποιου Artist, πρέπει πρώτα να εισάγει τον Artist μέσω της φόρμας Add Artist όπου εισάγει το όνομα, τα άλμπουμ που έχει ο Artist και τις συναυλίες. Αφού εισάγει τον Artist, τότε μπορεί να εισάγει και τραγούδια που ανήκουν σε αυτόν τον Artist, μαζί με το όνομα άλμπουμ, της πώλησης που έχει το τραγούδι, και την τιμή του τραγουδιού.

```
String insertArtist = "insert into artists(artist_name,albums,concerts) values(?,?,?)";
prestatement = dbConnection.prepareStatement(insertArtist);
prestatement.setString(1, txtartista.getText());
prestatement.setInt(2, albums);
prestatement.setInt(3, concerts);
prestatement.executeUpdate();
myConnection();

JOptionPane.showMessageDialog(this, "Artist Added");
```

Διαγραφή Καλλιτέχνη (Delete Artist)

Πατώντας πάνω σε κάποιο τραγούδι από τον πίνακα στο Admin. java, το όνομα του Artist εμφανίζεται στις φόρμες εισόδου, ενώ πατώντας “Delete” στην φόρμα Artist θα εμφανιστεί ένα Alert Box που ρωτάει αν είστε σίγουρος για την διαγραφή, αν ναι τότε ο Artist, μαζί με όλα τα τραγούδια και τα άλμπουμ του, θα διαγραφεί από την βάση.

```
.Class.forName(driverClassName);  
.dbConnection = DriverManager.getConnection(url, .username, .passwd);  
.String deleteartistfromsongs = "delete from songs where artist_name = .?";  
.String deleteartistfromalbums = "delete from albums where artist_name = .?";  
.String deleteartistfromartists = "delete from artists where artist_name = .?";  
.prestatement = .dbConnection.prepareStatement(deleteartistfromsongs);  
.  
.prestatement.setString(1,txtartist.getText());  
.prestatement.executeUpdate();  
.prestatement = .dbConnection.prepareStatement(deleteartistfromalbums);  
.  
.prestatement.setString(1,txtartist.getText());  
.prestatement.executeUpdate();  
.prestatement = .dbConnection.prepareStatement(deleteartistfromartists);  
.  
.prestatement.setString(1,txtartist.getText());  
.prestatement.executeUpdate();  
  
.JOptionPane.showMessageDialog(this, "Artist Deleted");  
.myConnection();
```

Προσθήκη Τραγουδιού (Add Song)

Με κλικ στο Add, ο κώδικας θα ελέγξει αν το τραγούδι που πρόκειται να εισαχθεί στην βάση υπάρχει ήδη.

```
//CHECK IF SONG EXISTS
String songname = txtsong.getText();
String artistname = txtartist.getText();

statement = dbConnection.createStatement();
result = statement.executeQuery("select * from selectallsongs()");
while(result.next()){
    String songfromlist = result.getString("song_name");
    String artistn = result.getString("artist_name");
    if(songfromlist.compareTo(songname) == 0 && artistn.compareTo(artistname) == 0){
        System.out.println("!!!");
        songExists = true;
    }
}
```

Σε περίπτωση που υπάρχει η εισαγωγή δεν θα πραγματοποιηθεί. Αν δεν υπάρχει το τραγούδι, τότε θα πραγματοποιηθεί έλεγχος για το αν υπάρχει το άλμπουμ μέσα στην βάση και αν δεν υπάρχει, τότε θα δημιουργηθεί και θα προστεθεί στον πίνακα Albums. Η στήλη tracklist θα αρχικοποιηθεί ως "1" και μετά το τραγούδι θα εισαχθεί στην βάση. Αν το άλμπουμ υπάρχει ήδη μέσα στην βάση, τότε το "tracklist" του άλμπουμ θα μεγαλώσει κατά 1 και μετά θα προστεθεί το τραγούδι.

Διαγραφή Τραγουδιού (Delete Song)

Για να γίνει διαγραφή κάποιου τραγουδιού, πρέπει να γίνει επιλογή του τραγουδιού που θέλουμε να διαγράψουμε από τον πίνακα με τα τραγούδια. Πατώντας το τραγούδι, τα στοιχεία του θα εμφανιστούν στην φόρμα Song, πατώντας το κουμπί Delete το τραγούδι θα διαγραφεί και το tracklist από το άλμπουμ όπου υπήρχε το τραγούδι θα μειωθεί κατά 1.

Ενημέρωση Τραγουδιού (Update Song)

Αν κατά την εισαγωγή του τραγουδιού έγινε κάποιο λάθος, υπάρχει η δυνατότητα ενημέρωσης των στοιχείων του. Πατώντας πάνω στο τραγούδι, τα στοιχεία του θα εμφανιστούν στην φόρμα Song, ενώ αλλάζοντας τα στοιχεία και πατώντας το κουμπί Update, τα στοιχεία του τραγουδιού θα αντικατασταθούν με τα καινούργια.

Ενημέρωση Δίσκου (Update Album)

Μετά την εισαγωγή του τραγουδιού, εισάγεται και ένα άλμπουμ στον πίνακα άλμπουμ με τα πεδία id, artist_name, album_name, tracklist, sales και price. Όλα τα πεδία εισάγονται με τα σωστά δεδομένα, εκτός από τα πεδία sales και price, όπου τα πεδία αυτά έχουν τιμές 0. Για αυτόν τον λόγο υπάρχει η φόρμα Album στην οποία γίνεται η επεξεργασία του άλμπουμ και η εισαγωγή των σωστών δεδομένων. Πατώντας ένα άλμπουμ από τον πίνακα, τα στοιχεία του θα εμφανιστούν στην φόρμα Album. Ο διαχειριστής μπορεί να αλλάξει τα στοιχεία του κάθε πεδίου, πατώντας το κουμπί Update τα καινούργια στοιχεία θα μπουν στην βάση.

Πίνακας Πωλήσεων (Total Sales)

Πατώντας στο κουμπί “**Total sales**” κλείνει το τρέχον JFrame και ανοίγει ένα άλλο όπου εμφανίζεται ένας πίνακας με τα πεδία **Artist, Album Sales, Song Sales, Ticket Sales**, όπου εμφανίζονται οι συνολικές πωλήσεις Άλμπουμ, Τραγουδιών και εισιτηρίων όπου έχει ο κάθε καλλιτέχνης. Πατώντας το κουμπί **Back** γίνεται η επιστροφή στον JFrame Admin. java.

Αρχείο Log File

Χρησιμοποιώντας το αρχείο Log File αποθηκεύουμε τις αλλαγές που έχουμε κάνει στους διάφορους πίνακες , μέσα σε διαφορετικούς πίνακες, που ονομάζονται **artists_log_file**, **albums_log_file**, **songs_log_file**, **concerts_log_file** και **sales_log_file**, αντίστοιχα.

Παράδειγμα του αρχείου: Artists_log_file.

Αρχικά, έχουμε ένα πίνακα **artists_log_file**, που περιέχει τις στήλες **operation** και **oper_time** οι οποίες αποθηκεύουν την πράξη που έχει γίνει στους πίνακες Artist, insert, update και delete, και ποτέ έχει πραγματοποιηθεί η πράξη. Επιπρόσθετα, περιέχει τις στήλες **artist_name**, **albums** και **concerts**, οι οποίες είναι οι στήλες που υπάρχουν και στον πίνακα Artist.

Ακόμη, δημιουργούμε ένα Function/Procedure **createlogartists()**, όπου:

- Αν κάνουμε delete έναν καλλιτέχνη από τον πίνακα Artists, γίνεται insert στον πίνακα **artists_log_file** ο καλλιτέχνης που κάναμε delete ,τα στοιχεία του και η ώρα που έγινε η διαγραφή
- Αν κάνουμε insert έναν καινούργιο καλλιτέχνη στον πίνακα Artists τότε ο καλλιτέχνης προστίθεται και στον πίνακα **artists_log_file**, όπως και η ώρα που έγινε το insert
- Αν κάνουμε update ένα καλλιτέχνη στον πίνακα Artists τότε κάνουμε insert στον πίνακα **artists_log_file** τα καινούργια στοιχεία του Artist και την ώρα που έγινε το update

Τέλος, υπάρχει ένα Trigger **keeplogartists** όπου μετά από ένα insert, update ή delete στον πίνακα artists, εκτελείται η διαδικασία **createlogartists()**.

Όλα τα υπόλοιπα **log_file** λειτουργούν με παρόμοιο τρόπο.

Αντίστοιχη φωτογραφία των **Artists_log_file**, **createlogartists()**, **keeplogartists**:

```
CREATE TABLE artists_log_file (  
operation char(1) NOT NULL,  
oper_time timestamp NOT NULL,  
artist_name varchar(255) NOT NULL,  
albums int NOT NULL,  
concerts int NOT NULL);  
  
CREATE OR REPLACE FUNCTION createlogartists()  
RETURNS TRIGGER AS $$  
BEGIN  
IF (TG_OP = 'DELETE') THEN  
INSERT INTO artists_log_file SELECT 'D', now(), OLD.artist_name, OLD.albums, OLD.concerts;  
RETURN OLD;  
ELSIF (TG_OP = 'UPDATE') THEN  
INSERT INTO artists_log_file SELECT 'U', now(), NEW.artist_name, NEW.albums, NEW.concerts;  
RETURN NEW;  
ELSIF (TG_OP = 'INSERT') THEN  
INSERT INTO artists_log_file SELECT 'I', now(), NEW.artist_name, NEW.albums, NEW.concerts;  
RETURN NEW;  
END IF;  
RETURN NULL;  
END;  
$$  
LANGUAGE plpgsql;  
  
CREATE TRIGGER keeplogartists  
AFTER INSERT OR UPDATE OR DELETE ON artists  
FOR EACH ROW EXECUTE PROCEDURE createlogartists();  
|
```

Επίλογος

Με το πέρας της εργασίας μας και έπειτα από πολύωρη ενασχόληση με αυτή ,αποκομίσαμε αρκετά και σημαντικά πράγματα. Αρχικά, κατανοήσαμε πως λειτουργεί ένα σύστημα βάσεων δεδομένων και πως να δημιουργούμε βάσεις χρησιμοποιώντας το εργαλείο `pgAdmin` .Επιπρόσθετα, μάθαμε πως να δημιουργούμε μία εφαρμογή μέσω της γλώσσας προγραμματισμού `java`, η οποία μας επιτρέπει να προσθέτουμε και να επεξεργαζόμαστε δεδομένα σε μία βάση. Τέλος, να μην παραλειφθεί ότι κατα τη διάρκεια της ενασχόλησης μας με την εργασία ,καλλιεργήθηκε το ομαδικό πνεύμα και η συνεργασία των μελών της ομάδας.