

ΕΦΑΡΜΟΓΗ ΔΙΣΚΟΓΡΑΦΙΚΗΣ ΕΤΑΙΡΕΙΑΣ

Σχεδιασμός και Υλοποίηση της Βάσης Δεδομένων

Ευαγγελία Γκαγκά, 1066528

up1066528@upnet.gr

Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών, Πανεπιστήμιο Πατρών, 26504, Ρίο-Πάτρα

Δημήτριος Μακρής, 1066496

up1066496@upnet.gr

Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών, Πανεπιστήμιο Πατρών, 26504, Ρίο-Πάτρα

Το θέμα της παρούσας εργασίας είναι η δημιουργία βάσης δεδομένων μιας δισκογραφικής εταιρείας. Η μελέτη χωρίζεται σε τρία στάδια, τον σχεδιασμό της βάσης δεδομένων μέσα από την κατανόηση του μικρόκοσμου, την υλοποίηση της βάσης δεδομένων και τέλος την ανάπτυξη λογισμικού γραφικής διεπαφής που επιτρέπει στον χρήστη να αλληλεπιδρά με τη βάση.

Λέξεις Κλειδιά: μικρόκοσμος, βάση δεδομένων, εφαρμογή, γραφική διεπαφή, MySQL, Python, ERD, Relational Schema

Σημείωση: Για οδηγίες εγκατάστασης και εκτέλεσης της εφαρμογής δείτε το αρχείο MANUAL.txt.

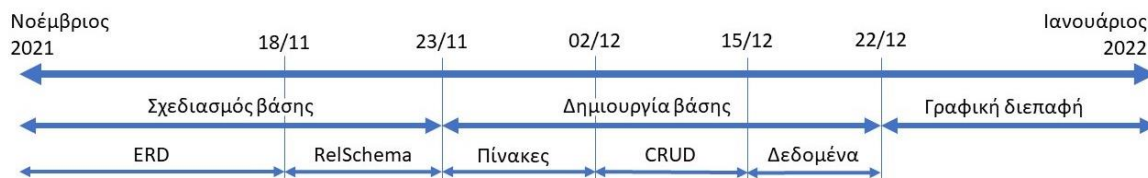
1 ΠΕΡΙΛΗΨΗ

Πρώτο και βασικότερο βήμα για την εκπόνηση της εργασίας είναι η κατανόηση του μικρόκοσμου μιας δισκογραφικής εταιρείας. Αφού αυτό επιτευχθεί, επόμενο βήμα είναι η κατασκευή του Διαγράμματος Οντοτήτων – Συσχετίσεων (Entity Relationship Diagram ή ERD) το οποίο μετασχηματίζεται σε Σχεσιακό Μοντέλο (Relational Schema), ολοκληρώνοντας τον σχεδιασμό της βάσης δεδομένων. Στη συνέχεια, η υλοποίηση βασίστηκε στην σύνταξη των κατάλληλων εντολών MySQL που δημιουργούν τη βάση και το περιεχόμενό της. Κατόπιν, για την εύκολη αλληλεπίδραση οποιουδήποτε χρήστη με τη βάση (εισαγωγή, διαγραφή, ανανέωση δεδομένων κτλ.) δημιουργήθηκε μία εφαρμογή σε γλώσσα προγραμματισμού Python με φιλική γραφική διεπαφή. Το σημαντικότερο βήμα είναι αδιαμφισβήτητα η κατανόηση κάθε πλευράς του μικρόκοσμου και η σωστή δημιουργία του διαγράμματος που τον αναπαριστά. Χωρίς αυτό το στάδιο, είναι αδύνατη η δημιουργία μιας βάσης που να ανταποκρίνεται στο ζητούμενο πρόβλημα.

2 ΟΡΓΑΝΩΣΗ

2.1 Χρονοδιάγραμμα

Προκειμένου να υλοποιηθούν όλες οι λειτουργίες στις ορισμένες προθεσμίες, κρίθηκε χρήσιμη η δημιουργία ενός χρονοδιαγράμματος που θα ανταποκρίνεται στις ανάγκες του πρότζεκτ, το οποίο απεικονίζεται στην Εικ. 1.



Εικόνα 1: Χρονοδιάγραμμα Εργασίας

Σε πρώτη φάση σχεδιάστηκαν τα διαγράμματα που ορίζουν τη βάση (ERD και Relational Schema), τα οποία ενημερώθηκαν και ολοκληρώθηκαν μετά την πρώτη παρουσίαση του πρότζεκτ (23/11). Στη συνέχεια, με τη χρήση MySQL δημιουργήθηκαν η βάση και οι πίνακες που την αποτελούν και ελέγχθηκε η λειτουργία τους μέσω εντολών CRUD (Create, Read, Update & Delete). Τα δεδομένα που χρησιμοποιήθηκαν κατά τον έλεγχο ήταν λίγα. Για να γίνει όμως προσομοίωση πραγματικών συνθηκών, η βάση γέμισε με τυχαία παραγμένα δεδομένα από ιστοσελίδα αντίστοιχης παρεχόμενης υπηρεσίας (βλέπε 6-Εργαλεία). Η εύκολη χρήση της βάσης από οποιονδήποτε χρήστη οδήγησε στη δημιουργία απλής γραφικής διεπαφής που υλοποιεί τις βασικές λειτουργίες της.

2.2 Αρμοδιότητες

Για την ταχύτερη διεκπεραίωση της κάθε ενότητας και παράλληλα την τήρηση του χρονοδιαγράμματος, σε ορισμένες περιπτώσεις κρίθηκε χρήσιμος ο χωρισμός αρμοδιοτήτων. Προφανώς, αυτό δεν συνεπάγεται τη μονομερή ενασχόληση κάποιου μέλους με κάποια ενότητα. Κάθε ενότητα αναπτύχθηκε και εκλέχθηκε και από τα δύο μέλη της ομάδας. Το μέρος του σχεδιασμού του ERD απαιτούσε την ταυτόχρονη συνεργασία και των δύο μελών, γιατί αποτελεί τη βάση για όλες τις υπόλοιπες ενότητες. Στη συνέχεια, ακολουθεί ενδεικτική λίστα με τις αρμοδιότητες του κάθε μέλους:

Μετάβαση από ERD σε Relational Schema – Δημήτριος Μακρής

Δημιουργία βάσης και πινάκων (MySQL) – Ευαγγελία Γκαγκά

Έλεγχος βάσης με δοκιμαστικά δεδομένα – Δημήτριος Μακρής

Σύνταξη εντολών CRUD (queries) – Ευαγγελία Γκαγκά

Παραγωγή τυχαίων δεδομένων – Ευαγγελία Γκαγκά

Σύνδεση MySQL & Python – Δημήτριος Μακρής

Γραφική διεπαφή Α' Μέρος – Δημήτριος Μακρής

Γραφική διεπαφή Β' Μέρος – Ευαγγελία Γκαγκά & Δημήτριος Μακρής

Σύνταξη έκθεσης και λοιπών συνοδευτικών εγγράφων – Ευαγγελία Γκαγκά & Δημήτριος Μακρής

3 ΣΧΕΔΙΑΣΜΟΣ

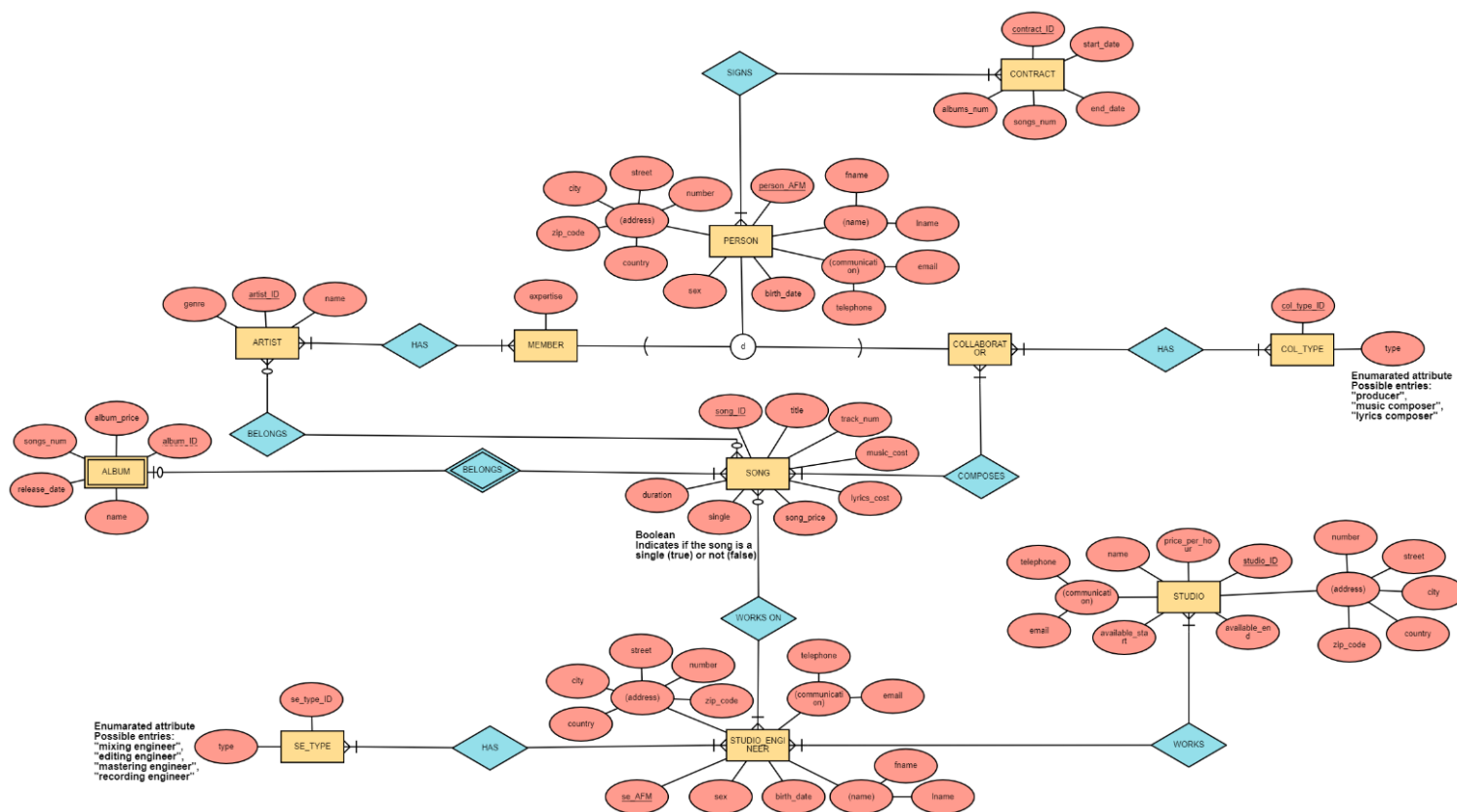
3.1 Μικρόκοσμος

Κυρίαρχος στόχος για τον σχεδιασμό είναι η κατανόηση του μικρόκοσμου. Κύριος λόγος ύπαρξης μιας δισκογραφικής εταιρείας είναι η υπογραφή συμβολαίων με πελάτες-μέλη με σκοπό την παραγωγή τραγουδιών που μπορεί να απαρτίζουν κάποιο άλμπουμ ή να κυκλοφορούν ανεξάρτητα. Κάθε πελάτης της δισκογραφικής εταιρείας μπορεί να είναι μέλος ενός ή περισσότερων συγκροτημάτων ή σόλο καλλιτέχνης ή εξωτερικός συνεργάτης που συμβάλλει στην

παραγωγή κάποιου τραγουδιού. Επιπλέον, διάφοροι φορείς συμμετέχουν στην ηχογράφηση και την επεξεργασία των τραγουδιών. Η δισκογραφική εταιρεία συνεργάζεται με studios σε διάφορες τοποθεσίες, τα οποία έχουν τους δικούς τους μηχανικούς που συμμετέχουν στη διαδικασία.

3.2 Διάγραμμα Οντοτήτων – Συσχετίσεων (ERD)

Το διάγραμμα οντοτήτων συσχετίσεων (Εικ. 2) παρουσιάζει τις σχέσεις μεταξύ των οντοτήτων που ανήκουν στον μικρόκοσμο.



Εικόνα 2: Διάγραμμα Οντοτήτων – Συσχετίσεων (ERD)

Ανάλυση Οντοτήτων:

PERSON: Πελάτης-συνεργάτης δισκογραφικής εταιρείας. Γνωρίσματά του είναι βασικά στοιχεία όσα περιγράφουν ένα άτομο (προσωπικά στοιχεία και στοιχεία επικοινωνίας). Το πρωτεύον κλειδί είναι το ΑΦΜ, διότι είναι μοναδικό για κάθε άτομο. Διακρίνεται σε *member* και *collaborator*.

CONTRACT: Συμβόλαιο που υπογράφουν όλοι οι πελάτες (*person*) της δισκογραφικής εταιρείας. Γνωρίσματά του είναι βασικά στοιχεία που χαρακτηρίζουν τη συνεργασία του με την εταιρεία (ημερομηνίες έναρξης – λήξης, αριθμός τραγουδιών και άλμπουμ).

MEMBER: Μέλος μίας μπάντας ή σόλο καλλιτέχνης. Κληρονομεί τα γνωρίσματα του πελάτη (*person*) και έχει επιπλέον την μουσική του εξειδίκευση (π.χ. *vocals*, *guitar*, *drums* κτλ.).

ARTIST: Καλλιτέχνης με γνωρίσματα το όνομα και το είδος μουσικής του. Αναπαριστά το προφίλ κάθε μέλους (member) στον μουσικό κόσμο. Δηλαδή, κάθε μέλος μιας μπάντας είναι ένα member και όλοι μαζί αποτελούν έναν artist. Κατ' αντιστοιχία, ένας σόλο καλλιτέχνης είναι ένα member και ένας artist (που μπορεί να έχει καλλιτεχνικό όνομα – ψευδώνυμο ή και όχι).

COLLABORATOR: Εξωτερικός συνεργάτης που έχει όλα τα γνωρίσματα του πελάτη (person) και μια ή περισσότερες επιπλέον ιδιότητες που σχετίζονται με την δημιουργία ενός τραγουδιού. Επειδή ένας collaborator μπορεί να έχει από μια έως πολλές ιδιότητες έχει δημιουργηθεί η οντότητα col_type που τις αποδίδει.

COL_TYPE: Τύπος συνεργάτη (collaborator). Οι τιμές που μπορεί να πάρει ο τύπος είναι lyrics composer, music composer και producer.

SONG: Τραγούδι με γνωρίσματα που σχετίζονται με την ταυτότητά του και το κόστος παραγωγής του.

ALBUM: Άλμπουμ που περιέχει τραγούδια (songs). Επιλέχθηκε να είναι ασθενής οντότητα γιατί δεν είναι απαραίτητο ένα τραγούδι να ανήκει σε κάποιο άλμπουμ, γεγονός που εκφράζει μια από τις τάσεις στη σύγχρονη μουσική βιομηχανία.

STUDIO ENGINEER: Μηχανικός που εργάζεται σε studio για την παραγωγή και την επεξεργασία ενός τραγουδιού. Γνωρίσματά του είναι βασικά στοιχεία όσα περιγράφουν ένα άτομο (προσωπικά στοιχεία και στοιχεία επικοινωνίας). Το πρωτεύον κλειδί είναι το ΑΦΜ, διότι είναι μοναδικό για κάθε άτομο.

SE_TYPE: Τύπος μηχανικού (studio engineer). Οι τιμές που λαμβάνει είναι mixing engineer, editing engineer, mastering engineer και recording engineer.

STUDIO: Στούντιο ηχογράφησης και επεξεργασίας τραγουδιών με γνωρίσματα στοιχεία σχετικά με την τοποθεσία, την επικοινωνία και το κόστος ενοικίασης.

Ανάλυση Συσχετίσεων:

PERSON – SINGS – CONTRACT: Οι πελάτες και οι συνεργάτες (person) υπογράφουν συμβόλαιο (contract) με τη δισκογραφική εταιρεία.

ARTIST – HAS – MEMBER: Κάθε μέλος συγκροτήματος (member) αναπαρίσταται με ένα καλλιτεχνικό ψευδώνυμο (artist), το οποίο είναι το όνομα του συγκροτήματος. Στην περίπτωση του σόλο καλλιτέχνη, αυτός (member) αναπαρίσταται με το καλλιτεχνικό όνομα που επιθυμεί (artist).

COLLABORATOR – HAS – COL_TYPE: Κάθε εξωτερικός συνεργάτης (collaborator) έχει μία ή περισσότερες ιδιότητες (col_type) που αφορούν τις αρμοδιότητές του.

SONG – BELONGS TO – ARTIST: Κάθε τραγούδι (song) συνδέεται με κάποιον καλλιτέχνη (artist).

COLLABORATOR – COMPOSES – SONG: Οι εξωτερικοί συνεργάτες (collaborator) συμβάλλουν, ανάλογα με την ιδιότητά τους, στην δημιουργία ενός τραγουδιού (song).

SONG – BELONGS TO – ALBUM: Το άλμπουμ (album) περιέχει τραγούδια (song).

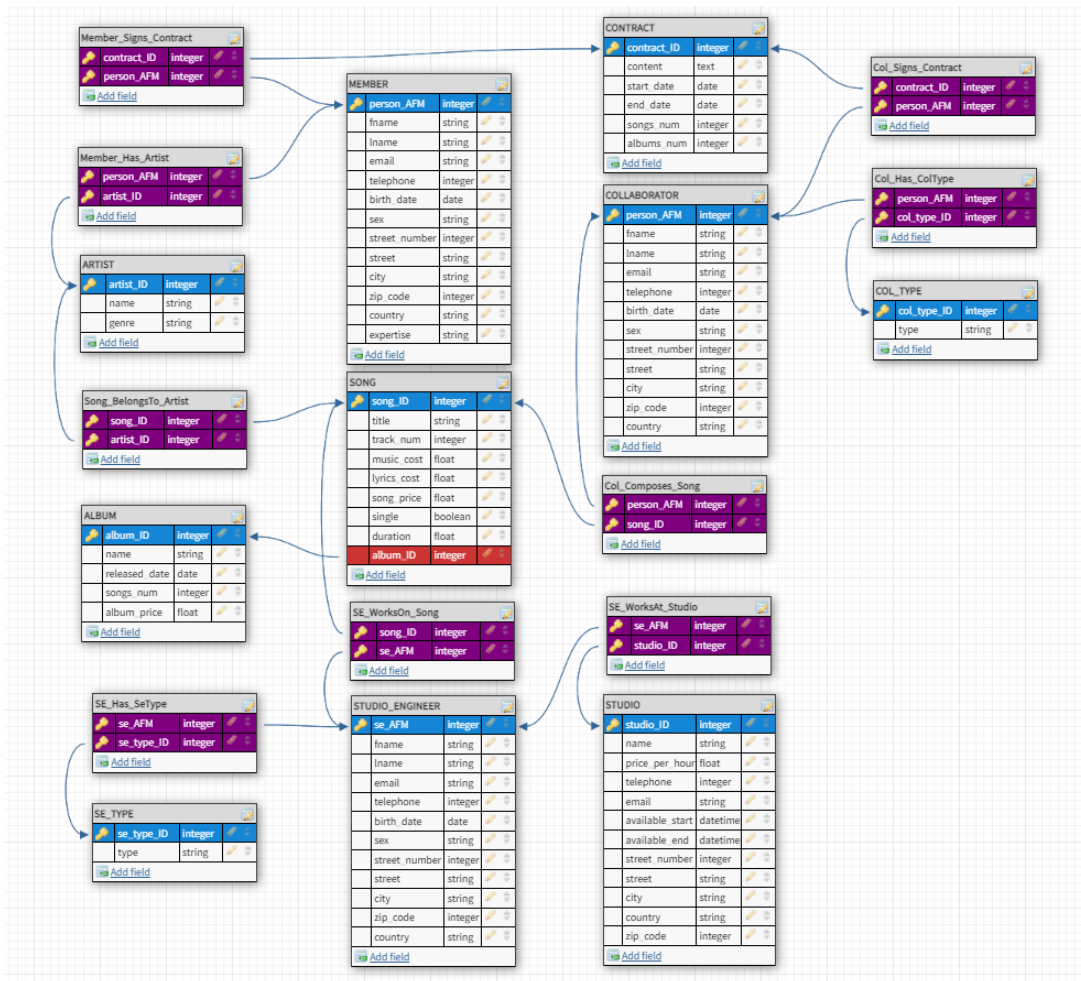
STUDIO_ENGINEER – WORKS ON – SONG: Οι μηχανικοί (studio engineer) εργάζονται στην παραγωγή και την επεξεργασία τραγουδιών (song).

STUDIO_ENGINEER – HAS – SE_TYPE: Κάθε μηχανικός (studio engineer) έχει μία ή περισσότερες ιδιότητες (se_type) που αφορούν τις αρμοδιότητές του.

STUDIO_ENGINEER – WORKS AT – STUDIO: Κάθε μηχανικός (studio engineer) εργάζεται σε κάποιο στούντιο (studio).

3.3 Σχεσιακό Μοντέλο (Relational Schema)

Το σχεσιακό μοντέλο (Εικ. 3) προκύπτει από το διάγραμμα οντοτήτων συσχετίσεων με συνέπεια στη χρήση των κανόνων μετάβασης.



Εικόνα 3: Σχεσιακό Μοντέλο (Relational Schema)

Τα πρωτεύοντα κλειδιά έχουν χρώμα μπλε, τα ξένα κλειδιά κόκκινο και με μωβ αυτά που είναι συνδυασμός πρωτεύοντων και ξένων κλειδιών (που χρησιμοποιούνται για την αναπαράσταση των συσχετίσεων N:N).

4 ΥΛΟΠΟΙΗΣΗ ΒΑΣΗΣ

4.1 Πίνακες

Αρχικά, χρησιμοποιούνται οι κατάλληλες εντολές MySQL για τη δημιουργία της βάσης δεδομένων (Εικ. 4)

```
1 DROP DATABASE discography_company;
2 CREATE DATABASE discography_company;
3 SHOW DATABASES;
4 USE discography_company;
5
```

Εικόνα 4: Εντολές για δημιουργία και χρήση βάσης (αρχείο *create_database.sql*)

Στη συνέχεια, δημιουργήθηκαν όλοι οι πίνακες της βάσης (Εικ. 5). Ενδεικτικά, φαίνονται παρακάτω δύο πίνακες (member, contract) και ο πίνακας που τους συσχετίζει. Το μοτίβο είναι κοινό και για τους υπόλοιπους πίνακες της βάσης.

```
1 DROP TABLE IF EXISTS contract;
2 CREATE TABLE contract (
3     contract_ID INT NOT NULL AUTO_INCREMENT,
4     start_date DATE NOT NULL,
5     end_date DATE NOT NULL,
6     songs_num INT NOT NULL,
7     albums_num INT NOT NULL,
8     PRIMARY KEY (contract_ID)
9 );
10
11 DROP TABLE IF EXISTS member;
12 CREATE TABLE member (
13     person_AFM INT NOT NULL,
14     fname VARCHAR(255) NOT NULL,
15     lname VARCHAR(255) NOT NULL,
16     email VARCHAR(255) NOT NULL,
17     telephone VARCHAR(10) NOT NULL,
18     birth_date DATE NOT NULL,
19     sex ENUM('Male', 'Female') NOT NULL,
20     street_number INT NOT NULL,
21     street VARCHAR(255) NOT NULL,
22     city VARCHAR(255) NOT NULL,
23     zip_code INT NOT NULL,
24     country VARCHAR(255) NOT NULL,
25     expertise VARCHAR(255) NOT NULL,
26     PRIMARY KEY (person_AFM)
27 );
28
29 DROP TABLE IF EXISTS Member_Signs_Contract;
30 CREATE TABLE Member_Signs_Contract(
31     person_AFM INT NOT NULL,
32     contract_ID INT NOT NULL,
33     PRIMARY KEY (person_AFM, contract_ID),
34     FOREIGN KEY (person_AFM) REFERENCES member(person_AFM) ON DELETE CASCADE ON UPDATE CASCADE,
35     FOREIGN KEY (contract_ID) REFERENCES contract(contract_ID) ON DELETE CASCADE ON UPDATE CASCADE
36 );|
```

Εικόνα 5: Δημιουργία πινάκων (αρχείο *create_tables.sql*)

4.2 Δεδομένα

Επόμενο βήμα ήταν ο έλεγχος ορθής λειτουργίας της βάσης. Για να γίνει αυτό, εισήχθησαν δοκιμαστικά δεδομένα όπως αυτά που φαίνονται παρακάτω (Εικ. 6).

```

1 INSERT INTO contract(start_date, end_date, songs_num, albums_num)
2 VALUES ("2010-09-11", "2010-12-01", 12, 1),
3 ("2013-01-01", "2014-01-01", 15, 2),
4 ("2018-07-18", "2019-01-06", 18, 2);
5
6 INSERT INTO member(person_AFM, fname, lname, email, telephone, birth_date, sex, street_number, street, city, zip_code, country, expertise)
7 VALUES (123456, "Lars", "Ulrich", "lars.ulrich@gmail.com", "6912345678", "1963-12-26", "Male", 983, "Kingsway", "Liverpool", 98661, "United Kindom", "drums"),
8 (214856, "James", "Hetfield", "james.hetfield@gmail.com", "6978452103", "1963-08-03", "Male", 2, "Mill Road", "Gloucester", 85496, "United Kindom", "vocals");
9 (456932, "Kirk", "Hammet", "kirk.hammet@gmail.com", "6997841220", "1963-12-26", "Male", 78, "Windsor Road", "Perth", 99451, "United Kindom", "guitar"),
10 (569412, "Robert", "Trujillo", "robert.trujillo@gmail.com", "6960851429", "1964-10-23", "Male", 892, "The Avenue", "Stoke On Trent", 78452, "United Kindom", "base"),
11 (964781, "Syd", "Barret", "syd.barret@gmail.com", "6987452250", "1946-01-06", "Male", 29, "Richmond Road", "Galashiels", 98547, "United Kindom", "vocals"),
12 (748125, "David", "Gilmour", "david.gilmour@gmail.com", "6921457823", "1946-03-06", "Male", 71, "Hermannstrasse", "Kleinniedesheim", 67259, "Germany", "guitar"),
13 (256417, "Nick", "Mason", "nick.mason@gmail.com", "6923147851", "1944-01-27", "Male", 11, "Maßmannsdorfer Chaussee", "Hamburg", 21035, "Germany", "drums"),
14 (321945, "Roger", "Waters", "roger.waters@gmail.com", "6941238574", "1943-11-06", "Male", 39, "Hedemannstasse", "Inzigkofen", 72514, "Germany", "bass"),
15 (164785, "Richard", "Wright", "richard.wright@gmail.com", "6874120359", "1943-07-28", "Male", 19, "Pohlstrasse", "Windhausen", 37539, "Germany", "keyboards"),
16 (789652, "Adele", "Laurie", "adele.laurie@gmail.com", "7458963210", "1988-05-05", "Female", 7, "The Green", "Plymouth", 968541, "United Kindom", "vocals");
17
18 INSERT INTO artist(name, genre)
19 VALUES ("Metallica", "metal"),
20 ("Pink Floyd", "progressive rock"),
21 ("Adele", "pop");
22
23 INSERT INTO collaborator(person_AFM, fname, lname, email, telephone, birth_date, sex, street_number, street, city, zip_code, country)
24 VALUES (546489, "James", "Brown", "james.brown@gmail.com", "6986897541", "1981-03-04", "Male", 385, "Roslyn Ave", "Akron", 44320, "Canada"),
25 (786675, "Alice", "Mayers", "alice.mayers@gmail.com", "6955674532", "1998-08-10", "Female", 32, "Springfield Road", "Glasgow", 45667, "United Kindom"),
26 (897612, "Kira", "Young", "kira.young19@gmail.com", "6956473423", "1974-09-04", "Female", 61, "Green Lane", "Stockport", 67754, "Spain");
27
28 INSERT INTO col_type(type)
29 VALUES ("Producer"),
30 ("Lyrics Composer"),
31 ("Producer");
32
33 INSERT INTO album(name, released_date, songs_num, album_price)
34 VALUES ("21", "2011-01-24", 12, 15.99),
35 ("The Black Album", "1991-09-07", 13, 19.99);
36

```

Εικόνα 6: Εισαγωγή δοκιμαστικών δεδομένων

Παρ' όλα αυτά, μια πραγματική βάση δεδομένων αποτελείται από χιλιάδες, ακόμα και εκατομμύρια δεδομένα σε κάθε πίνακα. Για αυτό το λόγο δημιουργήθηκαν εκατοντάδες τυχαία δεδομένα με τον τρόπο που φαίνεται στην παρακάτω εικόνα (Εικ. 7). Προφανώς, ήταν απαραίτητος ο σωστός περιορισμός των παραμέτρων ώστε να αποφευχθούν λάθη. Τελικά, τα δεδομένα εισάγονται στη βάση μέσω του αρχείου *insert_data.sql* (Εικ. 8).

Fill Database
Generate Data
My Projects
DB Generator
New
Donate
Contact
202 sec

Step 2: Fill Tables with dummy data

1. Database structure
2. Generate data
3. Export database

Step 2 - Fill Tables
List Grid

Current Table: member 0 rows

Field / Type [Key]	Generate	Parameters	Unique	Optional
Allow Null / Default Value	Select type of data to be generated for every column	Comma separated		
person_AFM int(11) [PRI] NO	randomNumber(\$from, \$to) // 39049	100000, 999999	<input checked="" type="checkbox"/>	<input type="checkbox"/>
fname varchar(255) NO	firstName(\$gender = null 'male' 'female') // 'Maynard'		<input type="checkbox"/>	<input type="checkbox"/>
lname varchar(255) NO	lastName // 'Zulauf'		<input type="checkbox"/>	<input type="checkbox"/>
email varchar(255) NO	email // 'kshlerin@collins.com'		<input type="checkbox"/>	<input type="checkbox"/>
telephone varchar(10) NO	phoneNumber // '132-149-0269x3767'		<input type="checkbox"/>	<input type="checkbox"/>
birth_date date NO	date(\$format = 'Y-m-d', \$max = 'now') // '1979-06-09'	Y-m-d,	<input type="checkbox"/>	<input type="checkbox"/>
sex enum('Male', 'Female') NO	randomElement(\$array = array("a", "b", "c")) // .randomElement	Male, Female	<input type="checkbox"/>	<input type="checkbox"/>
street_number int(11) NO	randomNumber(\$from, \$to) // 39049		<input type="checkbox"/>	<input type="checkbox"/>
street varchar(255) NO	-		<input type="checkbox"/>	<input type="checkbox"/>
city varchar(255) NO	city // 'West Judge'		<input type="checkbox"/>	<input type="checkbox"/>
zip_code int(11) NO	lexify(\$string = "????") // wgts		<input type="checkbox"/>	<input type="checkbox"/>
country varchar(255) NO	randomNumber(\$from, \$to) // 39049		<input type="checkbox"/>	<input type="checkbox"/>

Εικόνα 7: Δημιουργία τυχαίων δεδομένων

```

1 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (1, 'enim', '2022-04-30', 10, '8.74');
2 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (2, 'eos', '2020-11-28', 12, '30.71');
3 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (3, 'minus', '2026-05-14', 7, '36.92');
4 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (4, 'dolorum', '2030-10-17', 10, '22.51');
5 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (5, 'eveniet', '2023-12-26', 9, '22.72');
6 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (6, 'esse', '2021-08-26', 10, '13.57');
7 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (7, 'illo', '2019-06-26', 11, '6.65');
8 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (8, 'fugit', '2015-02-01', 9, '39.3');
9 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (9, 'molestiae', '2015-05-25', 11, '33.62');
10 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (10, 'officia', '2018-07-14', 11, '35.34');
11 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (11, 'nihil', '2029-03-21', 9, '26.23');
12 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (12, 'veritatis', '2015-11-16', 11, '30.17');
13 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (13, 'sit', '2017-11-28', 12, '7.81');
14 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (14, 'et', '2031-11-03', 8, '6.37');
15 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (15, 'nemo', '2016-07-30', 10, '5.95');
16 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (16, 'labore', '2025-02-09', 10, '37.08');
17 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (17, 'corporis', '2025-01-30', 12, '9.91');
18 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (18, 'ipsam', '2022-04-19', 5, '16.56');
19 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (19, 'voluptatem', '2016-10-29', 7, '16.47');
20 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (20, 'cupiditate', '2012-11-10', 5, '34.03');
21 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (21, 'veniam', '2029-04-04', 5, '6.05');
22 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (22, 'ut', '2029-09-07', 6, '20.27');
23 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (23, 'et', '2013-05-12', 11, '25.4');
24 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (24, 'impedit', '2015-08-03', 10, '18.65');
25 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (25, 'magni', '2026-06-13', 9, '38.3');
26 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (26, 'ut', '2014-07-12', 12, '29.47');
27 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (27, 'eius', '2031-12-20', 10, '35.57');
28 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (28, 'totam', '2013-12-23', 11, '11.45');
29 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (29, 'qui', '2015-07-19', 8, '24.19');
30 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (30, 'quo', '2013-09-19', 6, '15.54');
31 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (31, 'asperiores', '2024-03-23', 5, '9.23');
32 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (32, 'et', '2019-09-20', 11, '29.48');
33 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (33, 'omnis', '2023-03-10', 12, '8.71');
34 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (34, 'voluptatibus', '2018-10-25', 7, '28.67');
35 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (35, 'sunt', '2021-11-25', 12, '29.5');
36 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (36, 'doloremque', '2020-12-23', 9, '6.55');
37 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (37, 'at', '2026-04-03', 7, '32.04');
38 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (38, 'et', '2012-05-01', 5, '13.74');
39 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (39, 'quos', '2010-05-26', 10, '17.15');
40 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (40, 'nisi', '2024-10-20', 7, '22.91');
41 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (41, 'qui', '2022-08-10', 11, '33.36');
42 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (42, 'perspiciatis', '2025-06-07', 6, '37.24');
43 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (43, 'quo', '2025-07-19', 8, '25.27');
44 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (44, 'laborum', '2022-06-04', 10, '6.81');
45 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (45, 'sint', '2031-01-20', 6, '37.08');
46 INSERT INTO 'album' ('album_ID', 'name', 'released_date', 'songs_num', 'album_price') VALUES (46, 'et', '2010-10-29', 10, '37.7');

```

Εικόνα 8: Εισαγωγή τυχαίων δεδομένων (αρχείο *insert_data.sql*)

5 ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ

5.1 Σύνδεση MySQL - Python

Για τη σύνδεση της MySQL με την Python χρησιμοποιήθηκε η βιβλιοθήκη `mysql.connector`. Η σύνδεση υλοποιείται από την κλάση `DbConnection`, η οποία περιέχει τις κατάλληλες συναρτήσεις για σύνδεση και αποσύνδεση με τη βάση, εκτέλεση queries και έλεγχο σφαλμάτων. Ενδεικτικά φαίνεται ο κώδικας που υλοποιεί τη σύνδεση στην Εικ. 9.


```

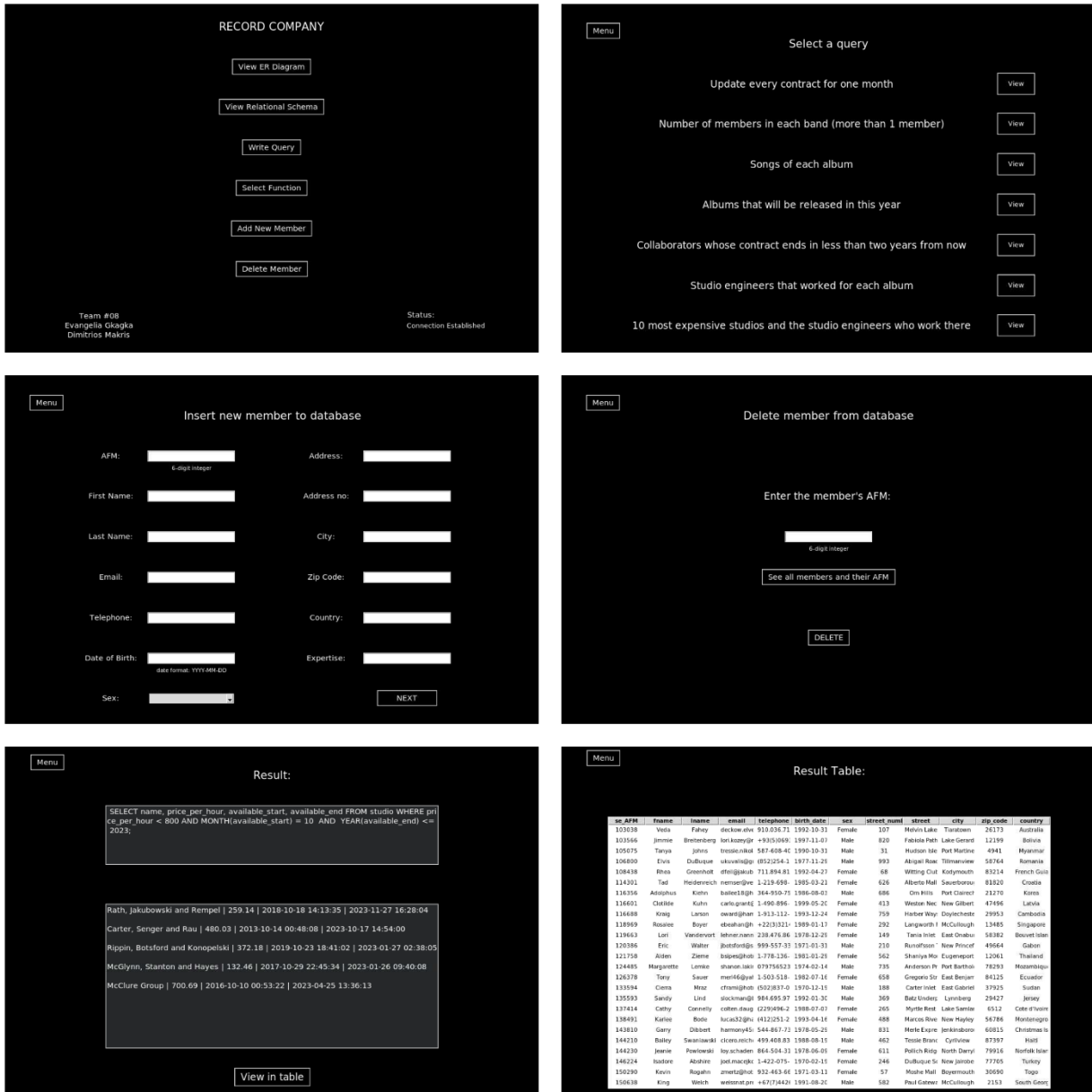
11 class DbConnection():
12     def __init__(self, dbname):
13         self.dbname = dbname
14         self.status = ''
15         try:
16             self.db = mysql.connect(host = 'localhost',
17                                     user = 'db_project',
18                                     password = 'password',
19                                     database = dbname)
20             self.cursor = self.db.cursor()
21
22             self.cursor.execute("use discography_company;")
23
24         except mysql.Error as err:
25             if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
26                 self.status = 'Connection Refused\nSomething is wrong with\nyour user name or password'
27             elif err.errno == errorcode.ER_BAD_DB_ERROR:
28                 self.status = 'Connection Refused\nDatabase does not exist'
29             else:
30                 self.status = "Connection Refused\nA connection error has occurred"
31         else:
32             self.status = 'Connection Established'
33
34     def close(self):
35         self.db.commit()
36         self.db.close()
37
38
39     def executeQuery(self, query):
40         result = ''
41         self.error = "notError"
42
43         try :
44             self.cursor.execute(query)
45             rowsAffected = self.cursor.rowcount
46             for row in self.cursor.fetchall():
47                 result += (" | ".join([str(item) for item in row]))
48                 result += '\n\n'
49             self.db.commit()
50         except mysql.Error as err:
51             self.error = err
52             result = 'An error has occurred, please check your query '
53             rowsAffected = 0
54
55         return result, rowsAffected
56

```

Εικόνα 9: Κώδικας σύνδεσης Python – MySQL (αρχείο *index.py*)

5.2 Γραφική Διεπαφή

Για την καλύτερη αλληλεπίδραση του χρήστη με τη βάση δημιουργήθηκε ένα απλό και ταυτόχρονα φιλικό γραφικό περιβάλλον. Μέσα από ένα κεντρικό Menu (Εικ. 10) ο χρήστης επιλέγει την επιθυμητή λειτουργία. Ακόμα, μπορεί να δει εάν η εφαρμογή έχει συνδεθεί επιτυχώς με τη βάση. Οι διαθέσιμες λειτουργίες βασίζονται στην εκτέλεση queries και στην προβολή των αποτελεσμάτων. Υπάρχει δυνατότητα προβολής του διαγράμματος ERD και του σχεσιακού μοντέλου. Ακόμα, ο χρήστης μπορεί είτε να επιλέξει από ένα σύνολο λειτουργιών που εκτελούν έτοιμα queries, είτε να γράψει οποιοδήποτε δικό του query. Σε κάθε περίπτωση, υπάρχει η δυνατότητα προβολής των αποτελεσμάτων σε πίνακες. Ακόμα, μπορεί να πραγματοποιήσει μια εγγραφή ενός νέου μέλους (member) στη βάση, συμπληρώνοντας μια φόρμα που αντιστοιχεί στους πίνακες member, contract και artist. Τέλος, μπορεί να γίνει διαγραφή ενός μέλους (member) από τη βάση δίνοντας το ΑΦΜ (που είναι μοναδικό όντας πρωτεύον κλειδί). Στην Εικ. 10 φαίνονται ενδεικτικά μερικές λειτουργίες της εφαρμογής.



Εικόνα 10: Με τη σειρά από αριστερά προς τα δεξιά και από πάνω προς τα κάτω φαίνονται οι λειτουργίες (α) Μενού (β) επιλογή λειτουργίας (έτοιμα queries) (γ) εγγραφή νέου μέλους (δ) διαγραφή μέλους (ε) εμφάνιση αποτελεσμάτων (στ) εμφάνιση αποτελεσμάτων σε πίνακα.

5.3 Εντολές τύπου CRUD

Ενδεικτικές εντολές τύπου Read και Update υπάρχουν στην εφαρμογή ως ενσωματωμένες λειτουργίες, δηλαδή λειτουργίες που εκτελούν έτοιμα queries ανάγνωσης - read (SELECT) και ενημέρωσης (UPDATE) δεδομένων, όπως φαίνεται στην Εικ. 11.

```

464 #Queries
465 text1 = ''' Update every contract for one month '''
466 query1 = ''' UPDATE contract SET end_date = DATE_ADD(end_date, INTERVAL 1 MONTH); '''
467 QueryDesc(text1, query1, 0.2)
468
469 text2 = ''' Number of members in each band (more than 1 member) '''
470 query2 = ''' SELECT A.name, COUNT(H.name) FROM artist AS A, Member_Has_Artist as MA, member as M WHERE A.artist_ID = MA.artist_ID AND MA.person_AFM = M.person_AFM GROUP BY A.artist_ID HAVING COUNT(H.name) > 1 ORDER
BY COUNT(H.name) DESC; '''
471 QueryDesc(text2, query2, 0.3)
472
473 text3 = ''' Songs of each album '''
474 query3 = ''' SELECT A.name, S.title FROM album AS A JOIN song AS S ON A.album_ID = S.album_ID ORDER BY A.name; '''
475 QueryDesc(text3, query3, 0.4)
476
477 text4 = ''' Albums that will be released in this year '''
478 dropIndex = ''' ALTER TABLE album DROP INDEX relDateIndex; '''
479 index = ''' CREATE INDEX relDateIndex ON album (released_date); '''
480 query4 = ''' SELECT name, released_date FROM album WHERE YEAR(released_date) = YEAR(CURDATE()) ORDER BY released_date; '''
481 QueryDesc(text4, query4, 0.5)
482
483 text5 = ''' Collaborators whose contract ends in less than two years from now '''
484 query5 = ''' SELECT Col.person_AFM, Col.fname, Col.lname, Con.end_date FROM collaborator AS Col, contract AS Con, Col_Signs_Contract AS ColCon WHERE Col.person_AFM = ColCon.person_AFM AND Con.contract_ID =
ColCon.contract_ID AND DATEDIFF(end_date, CURDATE()) <= 730 ORDER BY DATEDIFF(end_date, CURDATE()); '''
485 QueryDesc(text5, query5, 0.6)
486
487 text6 = ''' Studio engineers that worked for each album '''
488 query6 = ''' SELECT E.fname, E.lname, T.type, S.name FROM studio_engineer AS E, se_type AS T, SE_Has_SEType AS ET, studio AS S, SE_WorksAt_Studio AS ES WHERE E.se_AFM = ET.se_AFM AND T.se_type_ID = ET.se_type_ID AND
S.studio_ID = ES.studio_ID AND E.se_AFM = ES.se_AFM; '''
489 QueryDesc(text6, query6, 0.7)
490
491 text7 = ''' 10 most expensive studios and the studio engineers who work there '''
492 query7 = ''' SELECT S.name, S.price_per_hour, SE.fname, SE.lname FROM studio AS S, studio_engineer AS SE, SE_WorksAt_Studio AS SES WHERE SE.se_AFM = SES.se_AFM AND S.studio_ID = SES.studio_ID ORDER BY price_per_hour
DESC LIMIT 10; '''
493 QueryDesc(text7, query7, 0.8)
494
495 text8 = ''' All studios with price per hour less than 800 that are not available after October 2023 '''
496 query8 = ''' SELECT name, price_per_hour, available_start, available_end FROM studio WHERE price_per_hour < 800 AND MONTH(available_start) = 10 AND YEAR(available_end) <= 2023; '''
497 QueryDesc(text8, query8, 0.9)

```

Εικόνα 11: Queries που εκτελούν ανάγνωση και ενημέρωση.

Η εισαγωγή δεδομένων - create (INSERT) γίνεται μέσω της λειτουργίας Add new member που εισάγει στοιχεία μέσω μιας φόρμας, όπως αναφέρθηκε παραπάνω. Η διαγραφή δεδομένων (DELETE) υλοποιείται μέσω της λειτουργίας Delete member που επιλέγεται ένα μέλος προς διαγραφή μέσω του ΑΦΜ του.

6 ΑΞΙΟΛΟΓΗΣΗ

Η αξιολόγηση της εφαρμογής γίνεται σε τέσσερις άξονες. Πρώτον, ελέγχθηκε η απόκριση της εφαρμογής στα αιτήματα του χρήστη. Δεύτερον, αξιολογήθηκε η αντιμετώπιση σε πιθανά σφάλματα που είτε προκαλούνται ακούσια από τον χρήστη είτε προκύπτουν από την λανθασμένη σύνδεση με τη βάση δεδομένων. Έχοντας ολοκληρώσει τους παραπάνω ελέγχους επιβεβαιώνεται η ευκολία στην χρήση της γραφικής διεπαφής και η ευανάγνωση προβολή αποτελεσμάτων στα αιτήματα του χρήστη. Συμπερασματικά, η αξιολόγηση περιλαμβάνει τόσο την σωστή λειτουργία της βάσης δεδομένων όσο και την αλληλεπίδραση με τον χρήστη.

7 ΕΡΓΑΛΕΙΑ

Η κατασκευή του ERD της βάσης δεδομένων έγινε με τη χρήση της online εφαρμογής [ERDMaker](#).

Η μετάβαση στο Relational Schema με το online εργαλείο [dbDesigner](#).

Τα δεδομένα παράχθηκαν με τυχαίο τρόπο από την ιστοσελίδα [Fill Database](#).

Για όλη την υλοποίηση της βάσης δεδομένων και της εφαρμογής επιλέχθηκε το [goormIDE](#) (Web-based Cloud Programming Tool). Ο κώδικας αναπτύχθηκε σε ένα container που διευκολύνει την εξ' αποστάσεως συνεργασία.

Το DBMS (Database Management System) που χρησιμοποιήθηκε είναι η [MySQL](#).

Η εφαρμογή με την γλώσσα προγραμματισμού [Python](#) και πιο συγκεκριμένα με τη χρήση της βιβλιοθήκης [Tkinter](#) για τη γραφική διεπαφή και της βιβλιοθήκης [mysql.connector](#) για τη σύνδεση με το DBMS.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Python 3.10.2 documentation, <https://docs.python.org/3/>
- [2] MySQL documentation, <https://dev.mysql.com/doc/>
- [3] GoormIDE documentation, <https://help.goorm.io/>