2024-2025

Εργαστήριο Βάσεων Δεδομένων

2024 – 2025

Η ομάδα αποτελέιται από τους εξής φοιτητές:

Σπυρίδων Μανταδάκης

Παναγιώτης Παπανικολάου

Αλέξανδρος Γεώργιος Χαλαμπάκης

**Αναλυτικότερες Πληροφορίες:**

Αλέξανδρος Γεώργιος Χαλαμπάκης 1100754

[up1100754@ac.upatras.gr](mailto:up1100754@ac.upatras.gr)

Φοιτητής 3ου έτους

Σπυρίδων Μανταδάκης 1100613

[up1100613@ac.upatras.gr](mailto:up1100613@ac.upatras.gr)

Φοιτητής 3ου έτους

Παναγιώτης Παπανικολάου 1104804

[up1104804@ac.upatras.gr](mailto:up1104804@ac.upatras.gr)

Φοιτητής 3ου έτους

# **ΠΕΡΙΕΧΟΜΕΝΑ**

## **Μέρος Α:**

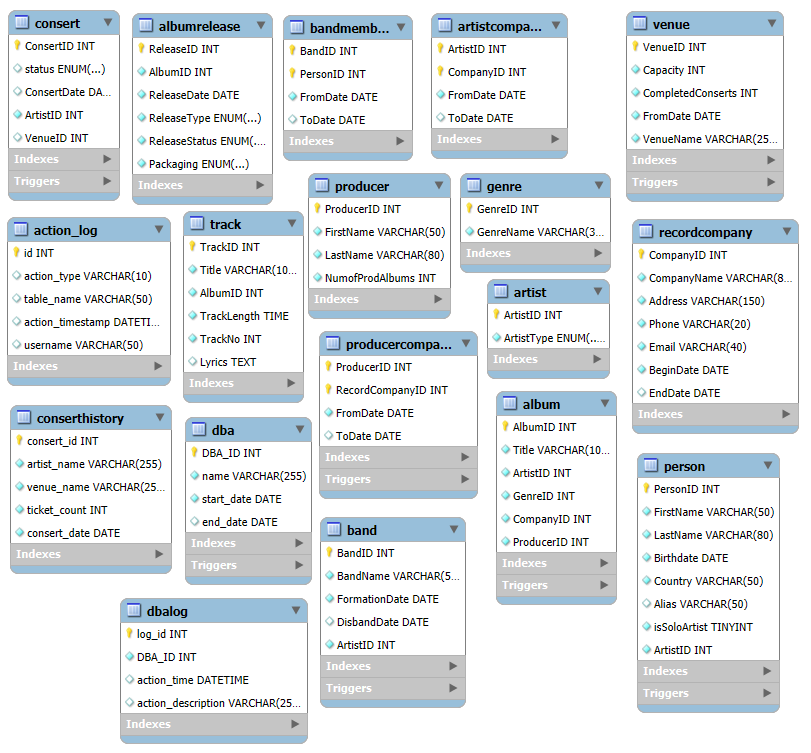
1. Περιγραφή Βάσης δεδομένων................................................3

**Μέρος Β:**

# Περιγραφή Βάσης δεδομένων

Παρακάτω φαίνεται το σχεσιακό διάγραμμα της συνολοκής αναθεωρημένης Βάσης δεδομένων

με ολες τις επεκτάσεις που χρειαστηκαν να γίνουν κατα την υλοποίηση των ζητούμενων



Οι κύριες αλλαγές στο διάγραμμα είναι η προσθήκες νέων πινάκων consert και venue, αρχικά για την υλοποιήση των αρχικών απαιτήσεων, και action\_log, dba, conserthistory, dbalog είναι οι υπόλοιποι πίνακες που χρειάστηκαν για την ανάπτυξη του Α μέρους της εργασίας. Δεν έγινε κάποια αλλαγή στους πίνακες της προκατασκευαστικής φάσης

Ακολουθεί μια σύντομη περιγραφή των πινάκων της βάσης δεδομένων της προκατασκευαστικής φάσης:

1. **person**: Πληροφορίες ατόμων (καλλιτεχνών), με δυνατότητα να είναι solo καλλιτέχνες ή μέλη συγκροτημάτων.
2. **bandmember**: Συσχέτιση ατόμων με συγκροτήματα, με χρονικό διάστημα συμμετοχής.
3. **band**: Πληροφορίες για συγκροτήματα, όπως όνομα, ημερομηνία ίδρυσης/διάλυσης.
4. **genre**: Είδη μουσικής (π.χ., Rock, Pop).
5. **artist**: Καλλιτέχνες γενικά, με κατηγοριοποίηση (π.χ., άτομο, συγκρότημα).
6. **albumrelease**: Κυκλοφορίες άλμπουμ (μορφή, κατάσταση, συσκευασία).
7. **artistcompany**: Συνεργασίες καλλιτεχνών με δισκογραφικές εταιρείες.
8. **recordcompany**: Πληροφορίες δισκογραφικών εταιρειών (όνομα, στοιχεία επικοινωνίας).
9. **album**: Πληροφορίες άλμπουμ (τίτλος, καλλιτέχνης, είδος, παραγωγός).
10. **producercompany**: Συνεργασίες παραγωγών με δισκογραφικές εταιρείες.
11. **producer**: Πληροφορίες παραγωγών (όνομα, αριθμός άλμπουμ παραγωγής).
12. **track**: Πληροφορίες τραγουδιών (τίτλος, διάρκεια, αριθμός κομματιού σε άλμπουμ).

**Αναλυτική περιγραφή των επιλεγμένων πινάκων και των πεδίων τους:**

**1. consert (Πίνακας Συναυλιών)**

Περιγράφει τις συναυλίες που έχουν προγραμματιστεί ή έχουν πραγματοποιηθεί.

* **ConsertID**: Μοναδικό αναγνωριστικό για κάθε συναυλία (Πρωτεύον Κλειδί).
* **status**: Κατάσταση της συναυλίας (π.χ., Scheduled, Completed, Cancelled).
* **ConsertDate**: Ημερομηνία διεξαγωγής της συναυλίας.
* **ArtistID**: Αναφορά στον καλλιτέχνη που συμμετέχει (Ξένο Κλειδί που συνδέεται με τον πίνακα artist).
* **VenueID**: Αναφορά στον χώρο όπου λαμβάνει χώρα η συναυλία (Ξένο Κλειδί που συνδέεται με τον πίνακα venue).

**2. venue (Πίνακας Χώρων Συναυλιών)**

Καταγράφει πληροφορίες για τους χώρους όπου διεξάγονται συναυλίες.

* **VenueID**: Μοναδικό αναγνωριστικό για κάθε χώρο (Πρωτεύον Κλειδί).
* **VenueName**: Όνομα του χώρου (υποχρεωτικό).
* **Capacity**: Χωρητικότητα του χώρου σε αριθμό θεατών.
* **CompletedConserts**: Αριθμός συναυλιών που έχουν ολοκληρωθεί στον συγκεκριμένο χώρο.
* **FromDate**: Ημερομηνία έναρξης λειτουργίας του χώρου.

**3. consertHistory (Ιστορικό Συναυλιών)**

Καταγράφει πληροφορίες για παρελθοντικές συναυλίες.

* **consert\_id**: Μοναδικό αναγνωριστικό της συναυλίας (Πρωτεύον Κλειδί).
* **artist\_name**: Όνομα του καλλιτέχνη που συμμετείχε στη συναυλία.
* **venue\_name**: Όνομα του χώρου όπου πραγματοποιήθηκε η συναυλία.
* **ticket\_count**: Αριθμός εισιτηρίων που πουλήθηκαν για τη συναυλία.
* **consert\_date**: Ημερομηνία διεξαγωγής της συναυλίας.

**4. DBA (Πίνακας Διαχειριστών της Βάσης Δεδομένων)**

Καταγράφει πληροφορίες για τους διαχειριστές της βάσης δεδομένων.

* **DBA\_ID**: Μοναδικό αναγνωριστικό για κάθε διαχειριστή (Πρωτεύον Κλειδί).
* **name**: Όνομα του διαχειριστή.
* **start\_date**: Ημερομηνία έναρξης των καθηκόντων του διαχειριστή.
* **end\_date**: Ημερομηνία λήξης των καθηκόντων του διαχειριστή (αν υπάρχει).

**5. DBALog (Καταγραφή Ενεργειών Διαχειριστών)**

Καταγράφει τις ενέργειες που πραγματοποιούν οι διαχειριστές στη βάση δεδομένων.

* **log\_id**: Μοναδικό αναγνωριστικό για κάθε καταγραφή (Πρωτεύον Κλειδί).
* **DBA\_ID**: Αναφορά στον διαχειριστή που πραγματοποίησε την ενέργεια (Ξένο Κλειδί που συνδέεται με τον πίνακα DBA).
* **action\_time**: Χρονική στιγμή που πραγματοποιήθηκε η ενέργεια (προκαθορισμένη τιμή το τρέχον timestamp).
* **action\_description**: Περιγραφή της ενέργειας (π.χ., "Διαγραφή εγγραφής", "Ενημέρωση δεδομένων").

**6. action\_log (Γενική Καταγραφή Ενεργειών)**

Καταγράφει ενέργειες χρηστών ή διαχειριστών στη βάση δεδομένων.

* **id**: Μοναδικό αναγνωριστικό για κάθε καταγραφή (Πρωτεύον Κλειδί).
* **action\_type**: Τύπος ενέργειας (π.χ., INSERT, UPDATE, DELETE).
* **table\_name**: Όνομα του πίνακα όπου πραγματοποιήθηκε η ενέργεια.
* **action\_timestamp**: Χρονική στιγμή της ενέργειας.
* **username**: Όνομα χρήστη που εκτέλεσε την ενέργεια.

Εντολές sql για τη δημιουργία της ΒΔ:

Πίνακες :

create table person(

PersonID int(11) not null auto\_increment,

FirstName varchar(50) not null,

LastName varchar(80) not null,

Birthdate date not null,

Country varchar(50) not null,

Alias varchar(50),

isSoloArtist tinyint(4) not null,

ArtistID int(11) not null,

primary key(PersonID),

constraint isartist

foreign key(ArtistID) references artist(ArtistID)

on delete cascade on update cascade);

create table bandmember(

BandID int(11) not null,

PersonID int(11) not null,

FromDate date not null,

ToDate date,

primary key(BandID,PersonID),

unique(FromDate),

constraint ismember

foreign key(PersonID) references person(PersonID)

on delete cascade on update cascade,

constraint fromband

foreign key(BandID) references band(BandID)

on delete cascade on update cascade);

create table band(

BandID int(11) not null auto\_increment,

BandName varchar(50) not null,

FormationDate date not null,

DisbandDate date,

ArtistID int(11) not null,

primary key(BandID),

constraint isartist

foreign key(ArtistID) references artist(ArtistID)

on delete cascade on update cascade);

create table genre(

GenreID int(11) not null auto\_increment,

GenreName varchar(30) not null,

primary key(GenreID));

create table artist(

ArtistID int(11) not null auto\_increment,

ArtistType enum('PERSON','BAND','ORCHESTRA','CHOIR','VIRTUAL','OTHER') not null,

primary key(ArtistID));

create table albumrelease(

ReleaseID int(11) not null auto\_increment,

AlbumID int(11) not null,

ReleaseDate date not null,

ReleaseType enum('LP','CD','MP3') not null,

ReleaseStatus enum('OFFICIAL','PROMOTION','BOOTLEG','WITHDRAWN','CANCELED') not null,

Packaging enum ('BOOK','CARDBOARD SLEEVE','DIGIPAK','JEWEL CASE','NA') not null,

primary key(ReleaseID),

constraint isalbum

foreign key(AlbumID) references album(AlbumID)

on delete cascade on update cascade);

create table artistcompany(

ArtistID int(11) not null,

CompanyID int(11) not null,

FromDate date not null,

ToDate date,

primary key(ArtistID,CompanyID),

unique(FromDate),

constraint isartist

foreign key(ArtistID) references artist(ArtistID)

on delete cascade on update cascade,

constraint iscompany

foreign key(CompanyID) references recordcompany(CompanyID)

on delete cascade on update cascade);

create table recordcompany(

CompanyID int(11) not null auto\_increment,

CompanyName varchar(80) not null,

Address varchar(150) not null,

Phone varchar(20) not null,

Email varchar(40) not null,

BeginDate date not null,

EndDate date,

primary key(CompanyID));

create table album(

AlbumID int(11) not null auto\_increment,

Title varchar(100) not null,

ArtistID int(11) not null,

GenreID int(11) not null,

CompanyID int(11) not null,

ProducerID int(11) not null,

primary key(AlbumID),

unique(Title),

constraint fromartist

foreign key(ArtistID) references artist(ArtistID)

on delete cascade on update cascade,

constraint isgenre

foreign key(GenreID) references genre(GenreID)

on delete cascade on update cascade,

constraint isartist

foreign key(CompanyID) references recordcompany(CompanyID)

on delete cascade on update cascade,

constraint produced

foreign key(ProducerID) references producer(ProducerID)

on delete cascade on update cascade);

create table producercompany(

ProducerID int(11) not null,

RecordCompanyID int(11) not null,

FromDate date not null,

ToDate date,

primary key(ProducerID,RecordCompanyID),

unique(FromDate),

constraint isproducer

foreign key(ProducerID) references producer(ProducerID)

on delete cascade on update cascade,

constraint reccompany

foreign key(RecordCompanyID) references recordcompany(RecordCompanyID)

on delete cascade on update cascade);

create table producer(

ProducerID int(11) not null auto\_increment,

FirstName varchar(50) not null,

LastName varchar(80) not null,

NumofProdAlbums int(11) not null,

primary key(ProducerID));

create table track(

TrackID int(11) not null auto\_increment,

Title varchar(100) not null,

AlbumID int(11) not null,

TrackLength time not null,

TrackNo int(11) not null,

Lyrics text,

primary key(TrackID),

constraint fromalbum

foreign key(AlbumID) references album(AlbumID)

on delete cascade on update cascade);

create table consert(

ConsertID int(11) not null auto\_increment,

status Enum('Scheduled','Completed','Cancelled') default 'scheduled',

ConsertDate date,

ArtistID int(11) not null,

VenueID int(11) not null,

primary key(ConsertID),

constraint Artisthasconsert

foreign key(ArtistID) references artist(ArtistID)

on delete cascade on update cascade,

constraint Takesplace

foreign key(VenueID) references venue(VenueID)

on delete cascade on update cascade

);

create table venue(

VenueID int(11) not null auto\_increment,

VenueName varchar not null,

Capacity int not null,

CompletedConserts int not null,

FromDate date,

primary key(VenueID)

);

CREATE TABLE consertHistory (

consert\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

artist\_name VARCHAR(255) NOT NULL,

venue\_name VARCHAR(255) NOT NULL,

ticket\_count INT NOT NULL,

consert\_date DATE NOT NULL

);

CREATE INDEX idx\_tickets\_count ON consertHistory(ticket\_count);

CREATE INDEX idx\_venue\_name ON consertHistory(venue\_name);

CREATE TABLE DBA (

DBA\_ID INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

start\_date DATE NOT NULL,

end\_date DATE NULL

);

CREATE TABLE DBALog (

log\_id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

DBA\_ID INT NOT NULL,

action\_time DATETIME DEFAULT CURRENT\_TIMESTAMP,

action\_description VARCHAR(255),

FOREIGN KEY (DBA\_ID) REFERENCES DBA(DBA\_ID)

);

CREATE TABLE action\_log (

id INT PRIMARY KEY AUTO\_INCREMENT,

action\_type VARCHAR(10),

table\_name VARCHAR(50),

action\_timestamp DATETIME,

username VARCHAR(50)

);

Εντολές insert:

Σημείωση: οι εγγραφές των πινάκων ενδέχεται να μην είναι όλες ίδιες καθώς χρειάστηκε να γίνουν προσθήκες και τροποποιήσεις στους πίνακες για τον έλεγχο λειτουργίας των υπόλοιπων ζητούμενων.

insert into artist(ArtistType) values

('PERSON'),

('PERSON'),

('BAND'),

('CHOIR'),

('ORCHESTRA'),

('OTHER'),

('PERSON'),

('BAND'),

('BAND'),

('VIRTUAL');

insert into person(FirstName,LastName,BirthDate,Country,Alias,isSoloArtist,ArtistID) values

('James','Woods','1995-07-24','USA',null,1,1),

('Michael','Stevens','1980-11-05','Canada','Macklemore',1,2),

('Elizabeth','Owens','1983-04-14','UK',null,0,3),

('Howard','Nelson','1979-03-14','Mexico',null,0,3),

('David','Ackerland','1995-12-30','Belgium',null,1,7),

('Maria','Fockel','2000-09-28','France','Rosewalk',1,5),

('Anna','Larston','1959-03-31','Dennmark',null,0,8);

insert into recordcompany values

(null,'StarRecords','St. Birkins','424536546', 'starrrecords@gmail.com','1970-12-02',null),

(null,'PanikRecords','Vouliagmenis 77','2108943935','info@panikrecords.gr','2011-09-28',null),

(null,'Sony Music Entertainment','Mesogeion 311','2106722150','reception.enquiries@sonymusic.gr','1929-09-09',null),

(null,'SoundWave Records','123 Melody Lane','3105551234','info@soundwave.com','1995-06-15',null),

(null,'Harmony Studios','456 Rhythm Ave','6155555678','contact@harmonystudios.com','1987-03-10',null);

insert into producer values

(null,'Sophia', 'Taylor', 10),

(null,'Liam', 'Walker', 35),

(null,'Isabella', 'Clark', 16),

(null,'Ethan', 'Allen', 21),

(null,'Mia', 'Hernandez', 14),

(null,'Noah', 'White', 40);

insert into band values

(null,'Neon Horizon', '2012-05-14',null,3),

(null,'Twilight Rhythm', '1985-09-20',null,8),

(null,'Echoes of Eternity', '2007-02-28',null,9);

insert into bandmember values

(3,3,'2015-05-14',null),

(1,4,'2018-08-17',null),

(2,7,'2007-02-28',null),

(2,4,'1985-09-20',null);

insert into genre values

(null,'rock'),

(null,'jazz'),

(null,'pop'),

(null,'metal'),

(null,'rock'),

(null,'pop');

insert into artistcompany values

(1,1,"1990-04-23",null),

(2,1,"2000-03-14",null),

(3,3,"1999-11-30",null),

(7,4,"2010-08-10",null),

(8,5,"2015-12-01",null),

(9,2,"2016-03-09",null),

(1,4,"1998-12-08",null),

(2,3,"2022-01-23",null),

(7,2,"2004-07-17",null),

(8,4,"2017-04-18",null),

(9,5,"2019-02-20",null);

insert into producercompany values

(1,5,'1987-03-10',null),

(2,4,'1995-06-15',null),

(3,3,'1929-09-09',null),

(4,2,'2011-09-28',null),

(5,1,'1970-12-02',null),

(3,2,'1938-04-13',null),

(1,4,'2000-01-19',null),

(2,1,'2010-11-30',null);

insert into album values

(null,'Days Together',1,3,1,1),

(null,'Generations',2,5,4,2),

(null,'In the Sun',3,2,5,3),

(null,'Better Together',7,6,3,5),

(null,'Wanderlust Chronicles',8,1,4,3),

(null,'Over the Years',1,2,3,4),

(null,'This is us',8,6,2,6),

(null,'Smiling Ocean',9,6,1,2),

(null,'Teen Pain',1,1,2,4),

(null,'think about Promise',2,6,3,4);

insert into track values

(null,'All in Moon',1,'00:05:23',1,null),

(null,'The Unbeat',1,'00:05:20',2,null),

(null,'First Wave',1,'00:04:53',3,null),

(null,'Imagine Scars',1,'00:04:00',4,null),

(null,'Soft Horizon',1,'00:05:20',5,null),

(null,'The Passion',1,'00:04:28',6,null),

(null,'Inspired Summer',1,'00:05:21',7,null),

(null,'Running Out of Time',1,'00:05:22',8,null),

(null,'Soul of Miles',2,'00:05:23',1,null),

(null,'Catle of Secrets',2,'00:04:03',2,null),

(null,'Cheery Lover',2,'00:05:40',3,null),

(null,'A Time of Today',2,'00:03:43',4,null),

(null,'Hot Touch',2,'00:04:50',5,null),

(null,'Sky of Blue',2,'00:05:32',6,null),

(null,'Foul Soul',2,'00:04:18',7,null),

(null,'Earning Break',2,'00:03:57',8,null),

(null,'Story of my Life',3,'00:05:15',1,null),

(null,'Dreamy Future',3,'00:03:50',2,null),

(null,'Break of Dawn',3,'00:04:13',3,null),

(null,'Hold the Line',3,'00:05:33',4,null),

(null,'Late Club',3,'00:02:59',5,null),

(null,'Stairway to Heaven',3,'00:04:10',6,null),

(null,'Thank you',3,'00:03:57',7,null),

(null,'Tortured Soul',3,'00:05:27',8,null),

(null,'Light Breeze',3,'00:03:45',9,null);

insert into albumrelease values

(null,1,'2000-03-13','CD','OFFICIAL','BOOK'),

(null,2,'2005-01-23','MP3','PROMOTION','CARDBOARD SLEEVE'),

(null,3,'2013-07-30','LP','OFFICIAL','BOOK'),

(null,1,'2016-03-31','CD','PROMOTION','BOOK'),

(null,2,'1998-03-13','MP3','BOOTLEG','NA'),

(null,3,'1996-11-25','CD','WITHDRAWN','DIGIPACK'),

(null,1,'2004-07-17','LP','CANCELED','BOOK'),

(null,2,'1980-03-23','CD','OFFICIAL','BOOK'),

(null,3,'2014-05-13','MP3','OFFICIAL','NA'),

(null,1,'1997-02-25','LP','PROMOTION','CARDBOARD SLEEVE'),

(null,2,'2019-01-29','MP3','OFFICIAL','BOOK'),

(null,1,'2001-01-31','CD','BOOTLEG','NA');

INSERT INTO DBA (name, start\_date) VALUES

('John Doe', '2023-01-01'),

('Jane Smith', '2023-03-15'),

('Alice Johnson', '2023-06-01');

# 2. Stored Procedures

# 3. Triggers