ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ

Государственное автономное образовательное учреждение

Высшего образования города Москвы

«Московский городской педагогический университет»

(ГАОУ ВО МГПУ)

Институт цифрового образования

Практикум по Базам данных

Лабораторная работа №2

## Тема: «Разработка ER-диаграммы для заданной предметной области. Применение нормализации к созданной модели данных.»

Выполнил: Малеванный И. Б.

Группа: ЦИБ-241

Вариант 12

Проверил: Босенко Т. М.

**Цель работы:**

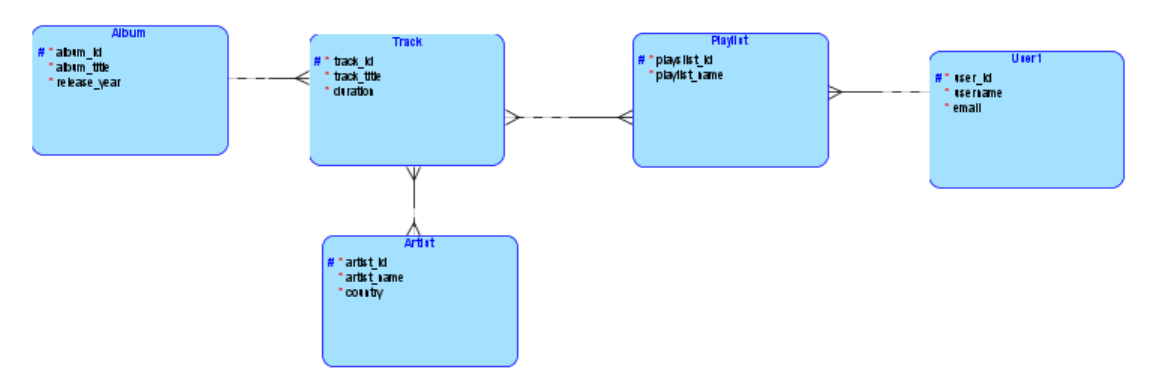
* овладеть практическими навыками анализа бизнес-требований для проектирования баз данных;
* научиться строить логическую, даталогическую (реляционную) и физическую модели данных с использованием Oracle SQL Developer Data Modeler;
* освоить принципы нормализации данных для устранения избыточности и аномалий, приводя структуру таблиц к третьей нормальной форме (3NF);
* научиться генерировать SQL-скрипт из физической модели и выполнять его на сервере MySQL с помощью MySQL Workbench.

**Предметная область:**

Музыкальный стриминг: БД Содержит артистов (Имена, жанр), треки (Название, Длительность),Альбомы и плейлисты.

**Практическая часть:**

Для начала создаем логическую модель ER-диаграмму в Oracle SQL Developer Data Modeler. (Рис.1)

****Рисунок 1 – Логическая модель , нотация Баркера

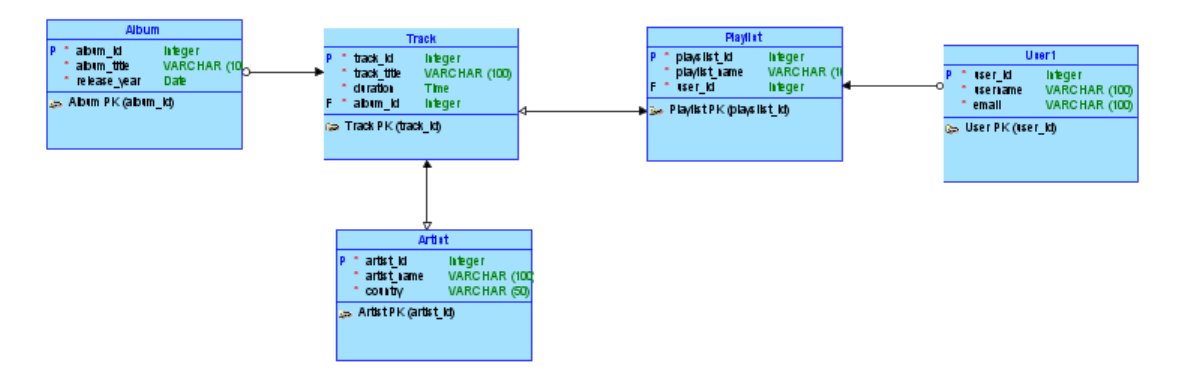


Рисунок 2 – Логическая модель , нотация Бахмана

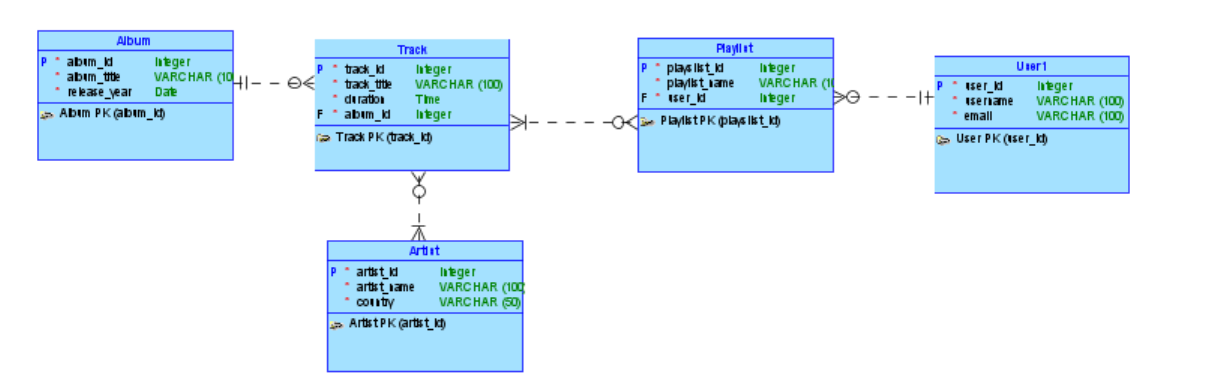


Рисунок 3 – Логическая модель , технологическая

Теперь с помощью генерации создаем реляционную модель проектирования. (рис. 4)

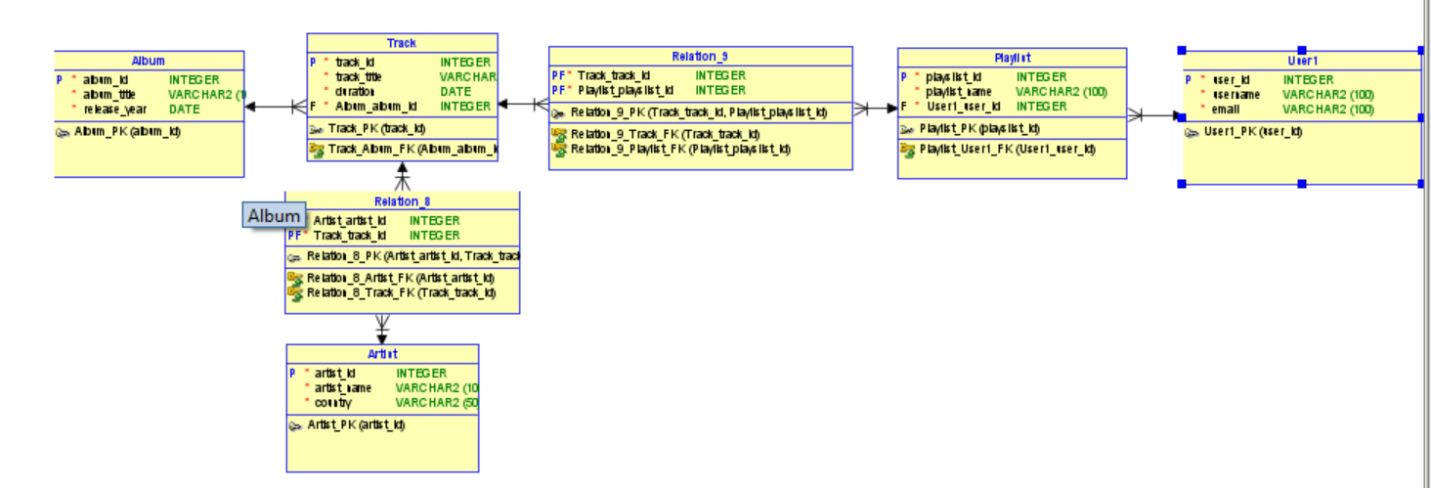


Рисунок 4 – Реляционная модель

Общее обоснование, что структура таблиц соответствует 3NF:

Отсутствие транзитивных зависимостей: Во всех таблицах неключевые атрибуты (не PK/FK) зависят исключительно от PK. FK используются только для ссылок между таблицами, но не вводят зависимости внутри.

Соответствие 1NF и 2NF: Подтверждено для всех (атомарность, полная зависимость от PK).

Финальный SQL-код. Результат успешного выполнения кода (Рисунок 5).

-- Generated by Oracle SQL Developer Data Modeler 24.3.1.351.0831

-- at: 2025-10-29 13:37:27 MSK

-- site: Oracle Database 11g

-- type: Oracle Database 11g

-- predefined type, no DDL - MDSYS.SDO\_GEOMETRY

-- predefined type, no DDL - XMLTYPE

CREATE TABLE Album

(

album\_id INTEGER NOT NULL ,

album\_title VARCHAR (100) NOT NULL ,

release\_year DATE NOT NULL

)

;

ALTER TABLE Album

ADD CONSTRAINT Album\_PK PRIMARY KEY ( album\_id ) ;

CREATE TABLE Artist

(

artist\_id INTEGER NOT NULL ,

artist\_name VARCHAR (100) NOT NULL ,

country VARCHAR (50) NOT NULL

)

;

ALTER TABLE Artist

ADD CONSTRAINT Artist\_PK PRIMARY KEY ( artist\_id ) ;

CREATE TABLE Playlist

(

playslist\_id INTEGER NOT NULL ,

playlist\_name VARCHAR (100) NOT NULL ,

User1\_user\_id INTEGER NOT NULL

)

;

ALTER TABLE Playlist

ADD CONSTRAINT Playlist\_PK PRIMARY KEY ( playslist\_id ) ;

CREATE TABLE Relation\_8

(

Artist\_artist\_id INTEGER NOT NULL ,

Track\_track\_id INTEGER NOT NULL

)

;

ALTER TABLE Relation\_8

ADD CONSTRAINT Relation\_8\_PK PRIMARY KEY ( Artist\_artist\_id, Track\_track\_id ) ;

CREATE TABLE Relation\_9

(

Track\_track\_id INTEGER NOT NULL ,

Playlist\_playslist\_id INTEGER NOT NULL

)

;

ALTER TABLE Relation\_9

ADD CONSTRAINT Relation\_9\_PK PRIMARY KEY ( Track\_track\_id, Playlist\_playslist\_id ) ;

CREATE TABLE Track

(

track\_id INTEGER NOT NULL ,

track\_title VARCHAR (100) NOT NULL ,

duration DATE NOT NULL ,

Album\_album\_id INTEGER NOT NULL

)

;

ALTER TABLE Track

ADD CONSTRAINT Track\_PK PRIMARY KEY ( track\_id ) ;

CREATE TABLE User1

(

user\_id INTEGER NOT NULL ,

username VARCHAR (100) NOT NULL ,

email VARCHAR (100) NOT NULL

)

;

ALTER TABLE User1

ADD CONSTRAINT User1\_PK PRIMARY KEY ( user\_id ) ;

ALTER TABLE Playlist

ADD CONSTRAINT Playlist\_User1\_FK FOREIGN KEY

(

User1\_user\_id

)

REFERENCES User1

(

user\_id

)

;

ALTER TABLE Relation\_8

ADD CONSTRAINT Relation\_8\_Artist\_FK FOREIGN KEY

(

Artist\_artist\_id

)

REFERENCES Artist

(

artist\_id

)

;

ALTER TABLE Relation\_8

ADD CONSTRAINT Relation\_8\_Track\_FK FOREIGN KEY

(

Track\_track\_id

)

REFERENCES Track

(

track\_id

)

;

ALTER TABLE Relation\_9

ADD CONSTRAINT Relation\_9\_Playlist\_FK FOREIGN KEY

(

Playlist\_playslist\_id

)

REFERENCES Playlist

(

playslist\_id

)

;

ALTER TABLE Relation\_9

ADD CONSTRAINT Relation\_9\_Track\_FK FOREIGN KEY

(

Track\_track\_id

)

REFERENCES Track

(

track\_id

)

;

ALTER TABLE Track

ADD CONSTRAINT Track\_Album\_FK FOREIGN KEY

(

Album\_album\_id

)

REFERENCES Album

(

album\_id

)

;

-- Oracle SQL Developer Data Modeler Summary Report:

--

-- CREATE TABLE 7

-- CREATE INDEX 0

-- ALTER TABLE 13

-- CREATE VIEW 0

-- ALTER VIEW 0

-- CREATE PACKAGE 0

-- CREATE PACKAGE BODY 0

-- CREATE PROCEDURE 0

-- CREATE FUNCTION 0

-- CREATE TRIGGER 0

-- ALTER TRIGGER 0

-- CREATE COLLECTION TYPE 0

-- CREATE STRUCTURED TYPE 0

-- CREATE STRUCTURED TYPE BODY 0

-- CREATE CLUSTER 0

-- CREATE CONTEXT 0

-- CREATE DATABASE 0

-- CREATE DIMENSION 0

-- CREATE DIRECTORY 0

-- CREATE DISK GROUP 0

-- CREATE ROLE 0

-- CREATE ROLLBACK SEGMENT 0

-- CREATE SEQUENCE 0

-- CREATE MATERIALIZED VIEW 0

-- CREATE MATERIALIZED VIEW LOG 0

-- CREATE SYNONYM 0

-- CREATE TABLESPACE 0

-- CREATE USER 0

--

-- DROP TABLESPACE 0

-- DROP DATABASE 0

--

-- REDACTION POLICY 0

--

-- ORDS DROP SCHEMA 0

-- ORDS ENABLE SCHEMA 0

-- ORDS ENABLE OBJECT 0

--

-- ERRORS 0

-- WARNINGS 0



Рисунок 5 – Успешное выполнение

Далее рассмотрим физическую модель в PhpMyAdmin (Рисунок 6).

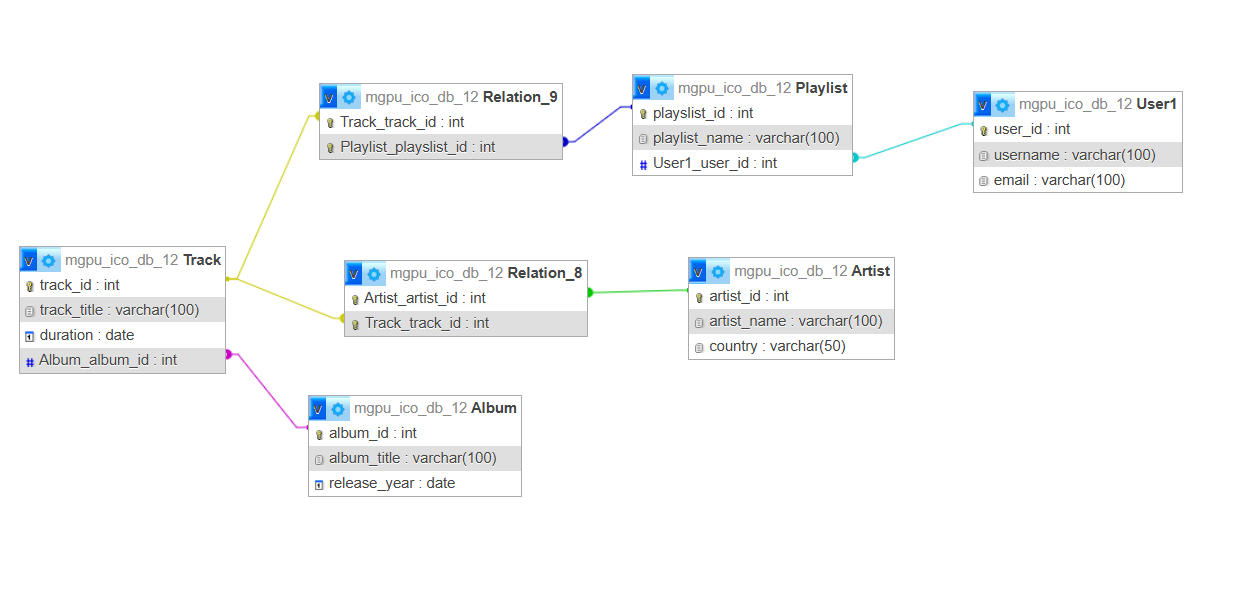


Рисунок 5 – Физическая модель в phpMyAdmin

**Выводы:**

В ходе работы были успешно достигнуты все поставленные цели по проектированию базы данных. Мы науились проводить практический анализ бизнес-требований, позволивший выявить ключевые сущности и их взаимосвязи. С помощью Oracle SQL Developer Data Modeler построили от логической до физической модели. Потом довели структуру таблиц до третьей нормальной формы, Практически закрепили навыки генерации SQL-скриптов и их выполнения через MySQL Workbench. Результатом стала оптимизированная структура базы данных.