Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ***ИУК «Информатика и управление»***

КАФЕДРА ***ИУК-2 «Информационные системы и сети»***

Домашняя работа №1

Организация взаимодействия с базой данных

ДИСЦИПЛИНА: «Базы данных»

Выполнил: студент гр. ИУК5-32Б _____ (Подпись) Ли Р. В. (Ф.И.О.)

Проверил: _____ (Подпись) _____ (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:
- Оценка:

Калуга , 2023

Цель: получение практических навыков физического проектирования базыданных в СУБД SQL Server.

Задачи: реализовать базу данных в СУБД SQL Server посредством SQL.

Вариант 18: Аудио коллекция.

Создание БД с помощью PostgreSQL.

Рис 1. Создание Базы данных.

Листинг 1. Создание таблиц.

```
create table genres (
    id serial primary key,
    name varchar not null
);

create table audio_types (
    id serial primary key,
    name varchar not null
);


create table authors (
    id serial primary key,
    nickname varchar(80) not null
);

create table artists (
```

```
    id serial primary key,
    full_name varchar(80) not null
);

create table authors_artists (
    artist_id int,
    author_id int,
    primary key (artist_id, author_id),
    foreign key (artist_id) references artists(id),
    foreign key (author_id) references authors(id)
);

create table albums (
    id serial primary key,
    name varchar(80) not null,
    release_date date not null,
    number_of_tracks int not null,
    genre_id int not null,
    author_id int not null,
    foreign key (genre_id) references genres(id),
    foreign key (author_id) references authors(id)
);

create table audio_tracks (
    id serial primary key,
    name varchar(80) not null,
    duration int not null,
    release_date date not null,
    genre_id int not null,
    type_id int not null,
    album_id int not null,
    foreign key (album_id) references albums(id),
    foreign key (genre_id) references genres(id),
    foreign key (type_id) references audio_types(id)
);

create table tracks_authors (
    track_id int,
    author_id int,
    primary key (track_id, author_id),
    foreign key (track_id) references audio_tracks(id),
    foreign key (author_id) references authors(id)
);

create table users (
    id serial primary key,
    login varchar(80) not null,
    password varchar(80) not null
);

create table collections (
    id serial primary key,
    name varchar(80) default 'unnamed',
    number_of_tracks int not null,
    total_duration int not null
);

create table collections_tracks (
    collection_id int,
    track_id int,
    primary key (collection_id, track_id),
    foreign key (collection_id) references collections(id),
    foreign key (track_id) references audio_tracks(id)
);
```

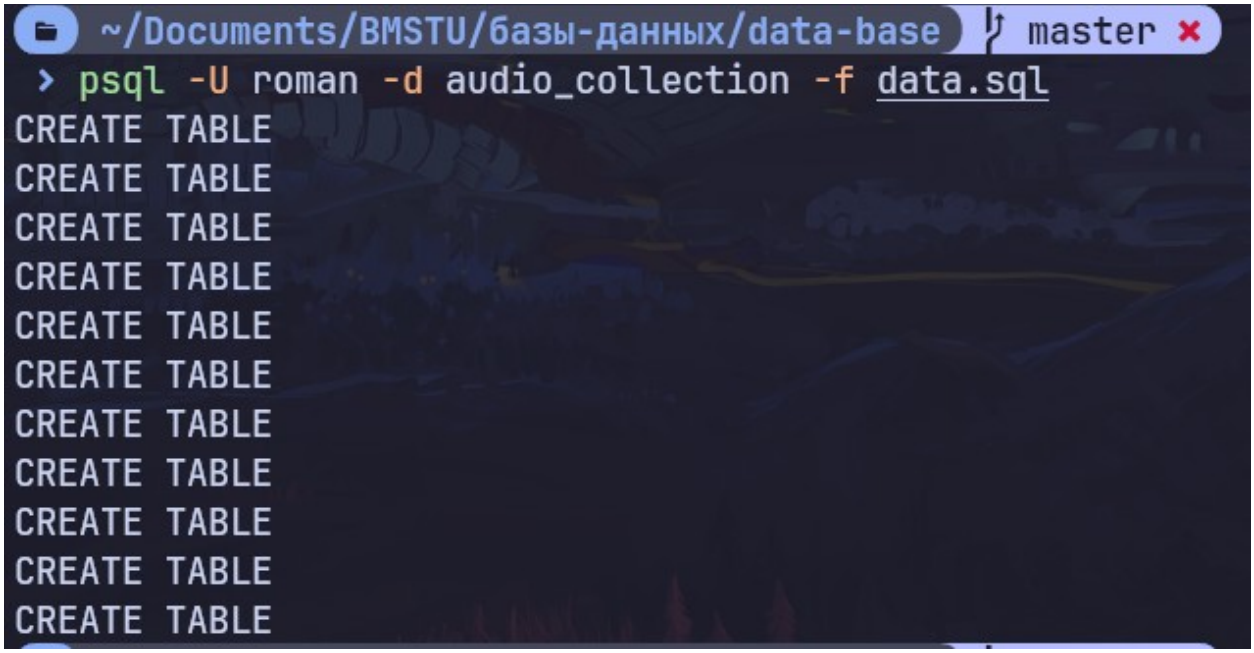


Рис 2.1. Создание таблиц SQL

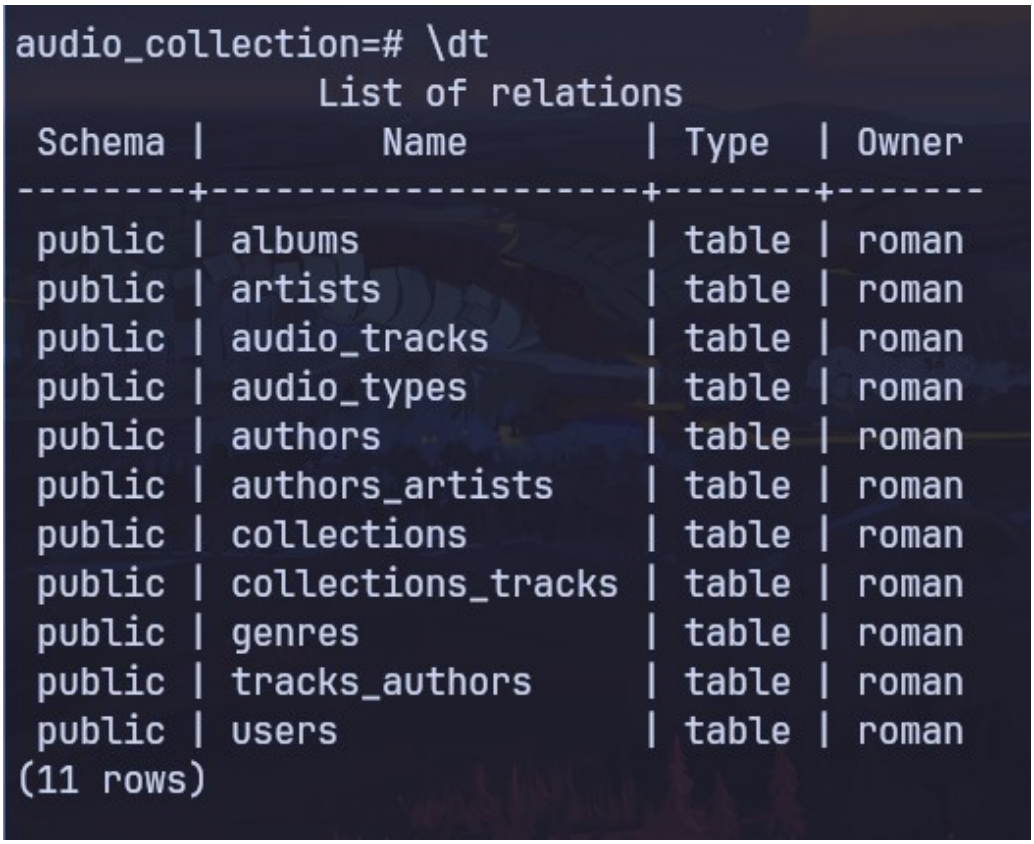


Рис 2.2. Таблицы SQL

Листинг 2. Добавление ограничений целосостности.

```
ALTER TABLE audio_tracks
ADD CONSTRAINT positive_duration_check
CHECK (duration >= 0);

ALTER TABLE albums
ADD CONSTRAINT positive_number_of_tracks_check
CHECK (number_of_tracks >= 0);

ALTER TABLE audio_tracks
ADD CONSTRAINT positive_duration_check
CHECK (duration >= 0);

ALTER TABLE collections
ADD CONSTRAINT positive_number_of_tracks_check
CHECK (number_of_tracks >= 0),
ADD CONSTRAINT positive_total_duration_check
CHECK (total_duration >= 0);

ALTER TABLE authors
ADD CONSTRAINT longer_than_0_characters_nickname_check
CHECK (LENGTH(nickname) > 0);

ALTER TABLE artists
ADD CONSTRAINT longer_than_0_characters_full_name_check
CHECK (LENGTH(full_name) > 0);


ALTER TABLE collections
ADD CONSTRAINT longer_than_0_characters_name_check
CHECK (LENGTH(name) > 0);

ALTER TABLE users
ADD CONSTRAINT longer_than_0_characters_login_check
CHECK (LENGTH(login) > 0);

ALTER TABLE genres
ADD CONSTRAINT longer_than_0_characters_name_check
CHECK (LENGTH(name) > 0);

ALTER TABLE audio_types
ADD CONSTRAINT longer_than_0_characters_name_check
CHECK (LENGTH(name) > 0);
```

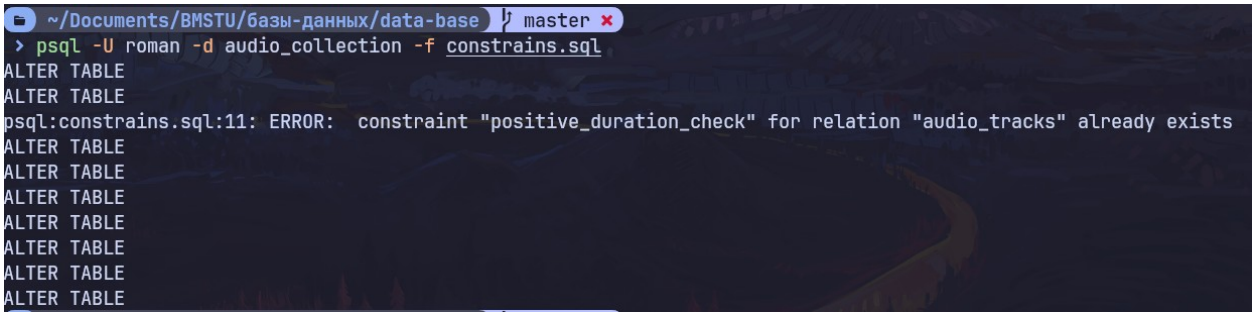


Рис 3. Ограничения целостности

Добавление данных.

Листинг 3.1

```
INSERT INTO genres (name)
VALUES ('Rock'), ('Pop'), ('Classic'), ('Traditional'), ('Metal'), ('Phonk'),
('Electronic');

INSERT INTO audio_types (name)
VALUES ('Song'), ('Book'), ('Podcast'), ('Melody');
```

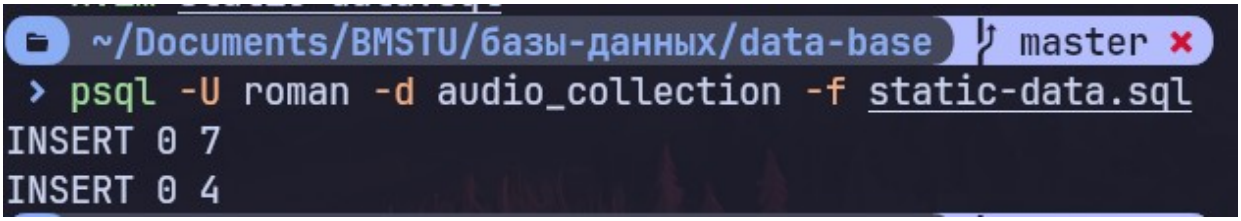


Рис 4.1 Ввод Жанров и Типов.

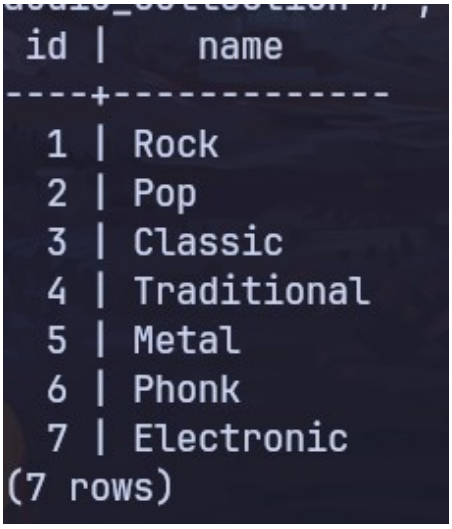


Рис 4.2 Таблица Жанров

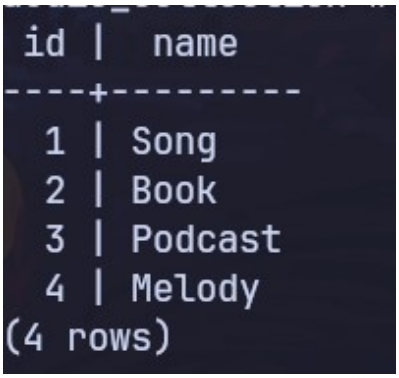


Рис 4.3 Таблица Типов аудио

Листинг 3.2

```
INSERT INTO authors (nickname) VALUES
('The Beatles'),
('Queen'),
('Nirvana'),
('Coldplay'),
('BTS');

-- Insert artists into the table
INSERT INTO artists (full_name) VALUES
('John Lennon'),
('Paul McCartney'),
('George Harrison'),
('Ringo Starr'),
('Freddie Mercury'),
('Brian May'),
('Roger Taylor'),
('John Deacon'),
('Kurt Cobain'),
('Krist Novoselic'),
('Dave Grohl'),
('Chris Martin'),
('Jonny Buckland'),
('Guy Berryman'),
('Will Champion'),
('RM'),
('Jin'),
('Suga'),
('J-Hope'),
('Jimin'),
('V'),
('Jungkook');

INSERT INTO authors_artists (author_id, artist_id) VALUES
(1, 1),  -- The Beatles: John Lennon
(1, 2),  -- The Beatles: Paul McCartney
(1, 3),  -- The Beatles: George Harrison
(1, 4),  -- The Beatles: Ringo Starr
(2, 5),  -- Queen: Freddie Mercury
(2, 6),  -- Queen: Brian May
(2, 7),  -- Queen: Roger Taylor
(2, 8),  -- Queen: John Deacon
(3, 9),  -- Nirvana: Kurt Cobain
(3, 10),-- Nirvana: Krist Novoselic
(3, 11),-- Nirvana: Dave Grohl
(4, 12),-- Coldplay: Chris Martin
(4, 13),-- Coldplay: Jonny Buckland
(4, 14),-- Coldplay: Guy Berryman
(4, 15),-- Coldplay: Will Champion
(5, 16),-- BTS: RM
(5, 17),-- BTS: Jin
(5, 18),-- BTS: Suga
(5, 19),-- BTS: J-Hope
(5, 20),-- BTS: Jimin
(5, 21),-- BTS: V
(5, 22);-- BTS: Jungkook
```

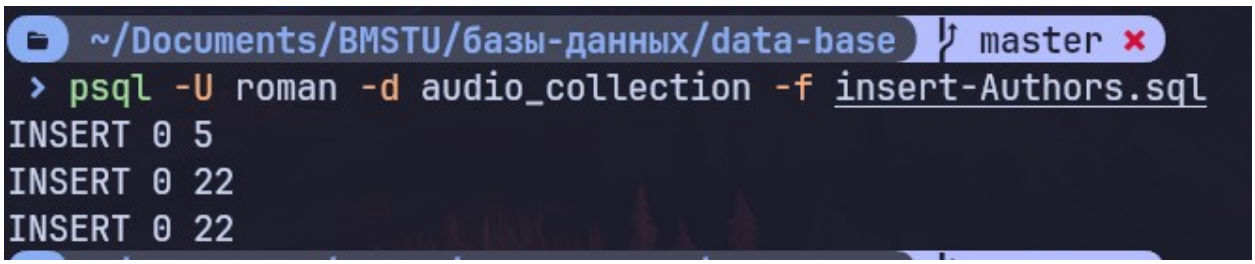


Рис 5.1 Ввод Авторов и Исполнителей

Рис 5.2 Таблица Исполнителей

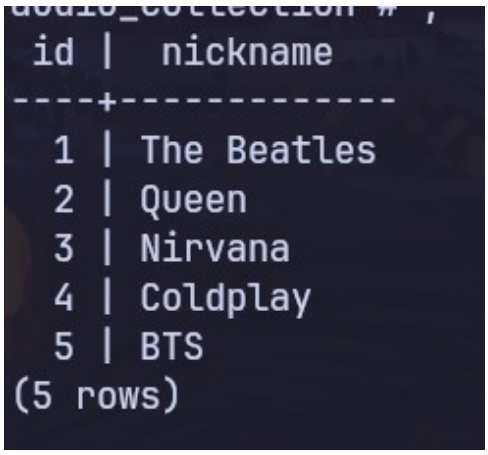


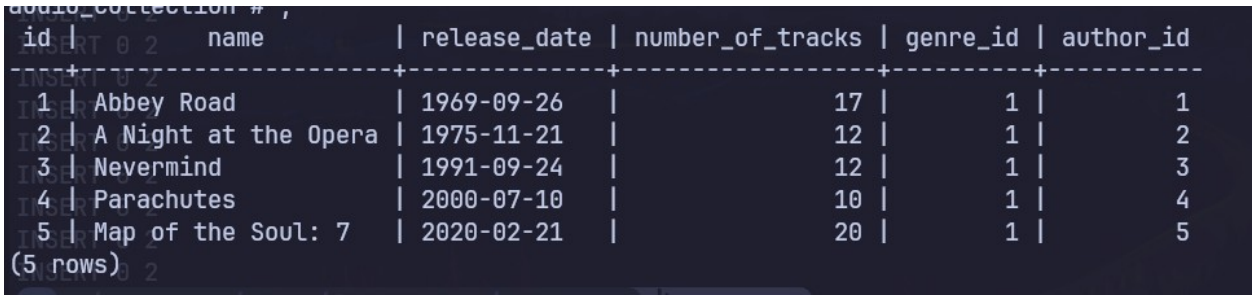
Рис 5.3 Таблица Авторов

Листинг 3.3

```
-- Insert albums for The Beatles
INSERT INTO albums (name, release_date, number_of_tracks, genre_id,
author_id) VALUES
('Abbey Road', '1969-09-26', 17, 1, 1),  -- The Beatles
('A Night at the Opera', '1975-11-21', 12, 1, 2),  -- Queen
('Nevermind', '1991-09-24', 12, 1, 3),  -- Nirvana
('Parachutes', '2000-07-10', 10, 1, 4),  -- Coldplay
('Map of the Soul: 7', '2020-02-21', 20, 1, 5);  -- BTS

-- Insert audio tracks for The Beatles
INSERT INTO audio_tracks (name, duration, release_date, genre_id, type_id,
album_id) VALUES
('Come Together', 259, '1969-09-26', 1, 1, 1),
('Bohemian Rhapsody', 354, '1975-11-21', 1, 1, 2),
('Smells Like Teen Spirit', 302, '1991-09-24', 1, 1, 3),
('Yellow', 266, '2000-07-10', 1, 1, 4),
('ON', 228, '2020-02-21', 1, 1, 5);

-- Insert tracks_authors for The Beatles
INSERT INTO tracks_authors (track_id, author_id) VALUES
(1, 1),  -- Come Together: The Beatles
(2, 2),  -- Bohemian Rhapsody: Queen
(3, 3),  -- Smells Like Teen Spirit: Nirvana
(4, 4),  -- Yellow: Coldplay
(5, 5);  -- ON: BTS
-- Insert additional tracks for The Beatles' album
INSERT INTO audio_tracks (name, duration, release_date, genre_id, type_id,
album_id) VALUES
('Here Comes the Sun', 185, '1969-09-26', 1, 1, 1), -- The Beatles
('Let It Be', 243, '1969-09-26', 1, 1, 1);        -- The Beatles

-- Insert additional tracks for Queen's album
INSERT INTO audio_tracks (name, duration, release_date, genre_id, type_id,
album_id) VALUES
('We Will Rock You', 122, '1975-11-21', 1, 1, 2), -- Queen
('We Are the Champions', 189, '1975-11-21', 1, 1, 2); -- Queen

-- Insert additional tracks for Nirvana's album
INSERT INTO audio_tracks (name, duration, release_date, genre_id, type_id,
album_id) VALUES
('Lithium', 257, '1991-09-24', 1, 1, 3),        -- Nirvana
('In Bloom', 254, '1991-09-24', 1, 1, 3);       -- Nirvana

-- Insert additional tracks for Coldplay's album
```

```
INSERT INTO audio_tracks (name, duration, release_date, genre_id, type_id,
album_id) VALUES
('Clocks', 307, '2000-07-10', 1, 1, 4),         -- Coldplay
('Fix You', 296, '2000-07-10', 1, 1, 4);        -- Coldplay

-- Insert additional tracks for BTS's album
INSERT INTO audio_tracks (name, duration, release_date, genre_id, type_id,
album_id) VALUES
('Dynamite', 199, '2020-02-21', 1, 1, 5),       -- BTS
('Boy with Luv', 229, '2020-02-21', 1, 1, 5);  -- BTS

-- Link authors with the new tracks

-- The Beatles' tracks
INSERT INTO tracks_authors (track_id, author_id) VALUES
(6, 1), -- Here Comes the Sun: The Beatles
(7, 1); -- Let It Be: The Beatles

-- Queen's tracks
INSERT INTO tracks_authors (track_id, author_id) VALUES
(8, 2), -- We Will Rock You: Queen
(9, 2); -- We Are the Champions: Queen

-- Nirvana's tracks
INSERT INTO tracks_authors (track_id, author_id) VALUES
(10, 3), -- Lithium: Nirvana
(11, 3); -- In Bloom: Nirvana

-- Coldplay's tracks
INSERT INTO tracks_authors (track_id, author_id) VALUES
(12, 4), -- Clocks: Coldplay
(13, 4); -- Fix You: Coldplay

-- BTS's tracks
INSERT INTO tracks_authors (track_id, author_id) VALUES
(14, 5), -- Dynamite: BTS
(15, 5); -- Boy with Luv: BTS
```

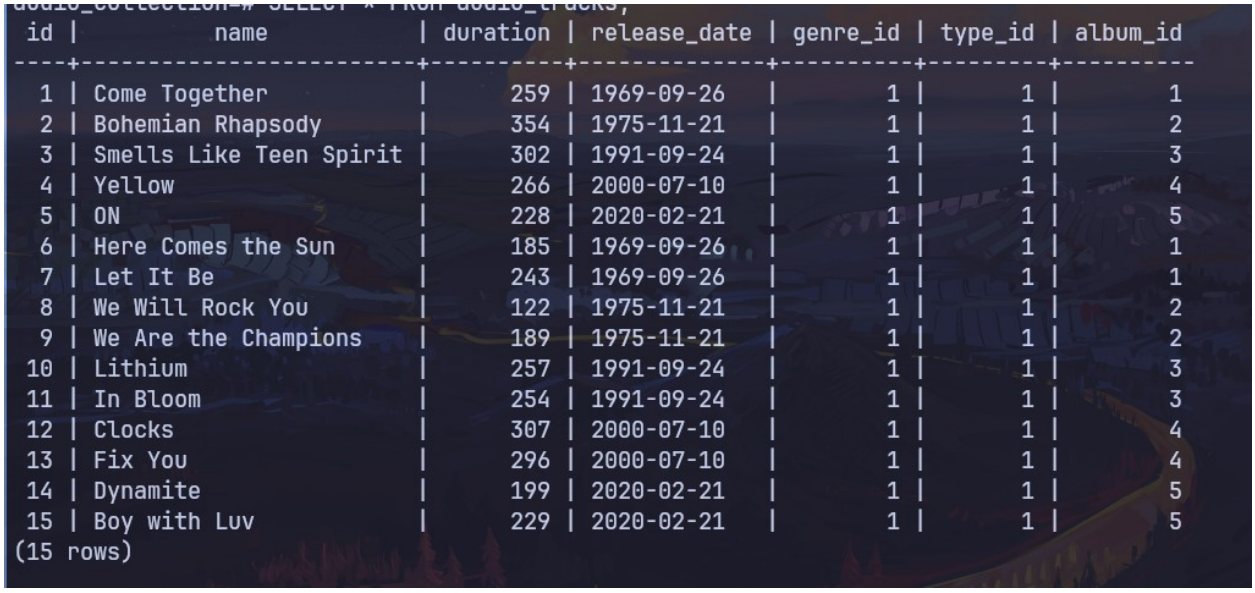
Рис 6.1 Ввод Альбомов и Аудио треков



id	name	release_date	number_of_tracks	genre_id	author_id
1	Abbey Road	1969-09-26	17	1	1
2	A Night at the Opera	1975-11-21	12	1	2
3	Nevermind	1991-09-24	12	1	3
4	Parachutes	2000-07-10	10	1	4
5	Map of the Soul: 7	2020-02-21	20	1	5

(5 rows)

Рис 6.2 Таблица Альбомы



id	name	duration	release_date	genre_id	type_id	album_id
1	Come Together	259	1969-09-26	1	1	1
2	Bohemian Rhapsody	354	1975-11-21	1	1	2
3	Smells Like Teen Spirit	302	1991-09-24	1	1	3
4	Yellow	266	2000-07-10	1	1	4
5	ON	228	2020-02-21	1	1	5
6	Here Comes the Sun	185	1969-09-26	1	1	1
7	Let It Be	243	1969-09-26	1	1	1
8	We Will Rock You	122	1975-11-21	1	1	2
9	We Are the Champions	189	1975-11-21	1	1	2
10	Lithium	257	1991-09-24	1	1	3
11	In Bloom	254	1991-09-24	1	1	3
12	Clocks	307	2000-07-10	1	1	4
13	Fix You	296	2000-07-10	1	1	4
14	Dynamite	199	2020-02-21	1	1	5
15	Boy with Luv	229	2020-02-21	1	1	5

(15 rows)

Рис 6.3 Таблица Аудио треки

Листинг 3.4

```
INSERT INTO users (login, password) VALUES ('roman', '123');

INSERT INTO collections (name, number_of_tracks, total_duration) VALUES
('Romans Collection', 0, 0);

SELECT id FROM collections WHERE name = 'Romans Collection';

INSERT INTO collections_tracks (collection_id, track_id) VALUES
(1, 1),
(1, 2),
(1, 3),
(1, 4),
(1, 5),
(1 ,6),
(1, 7);
```

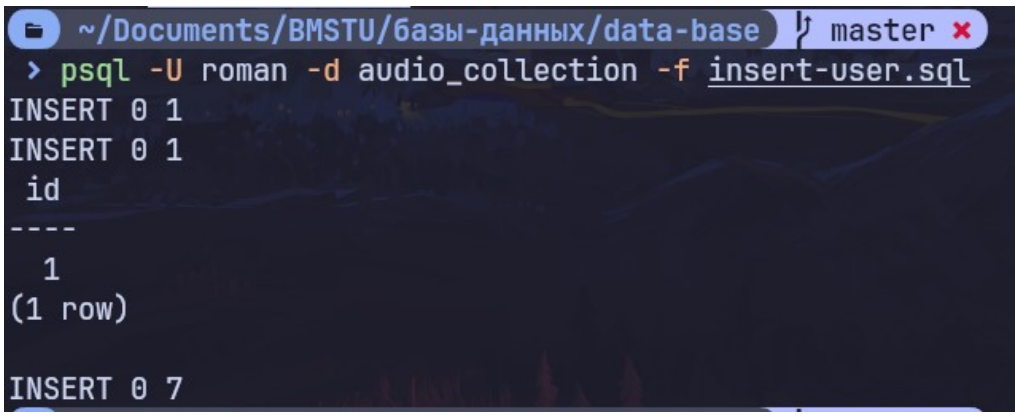
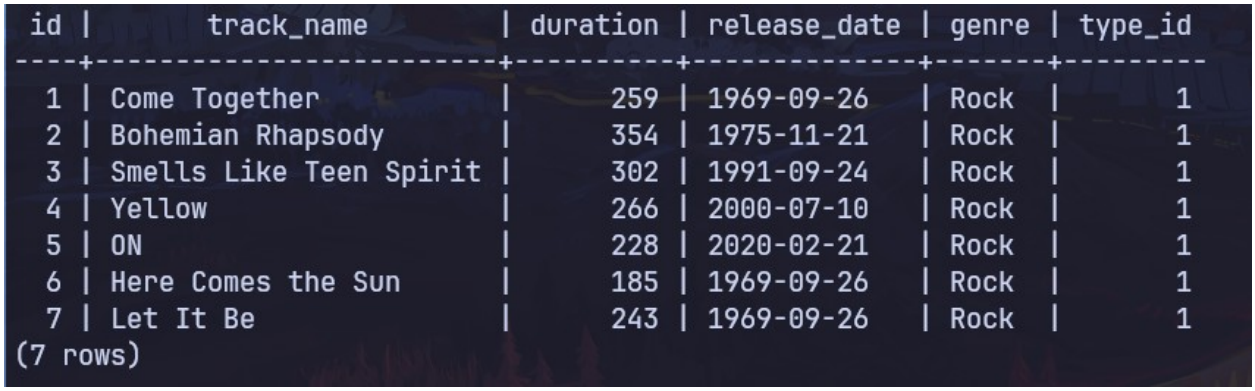


Рис 7.1 Ввод Пользователя и его Коллекции.

Просмотр данных.

Листинг 4.1

```
-- look up collection
SELECT at.id, at.name AS track_name, at.duration, at.release_date, g.name AS
genre, at.type_id
FROM collections_tracks ct
JOIN audio_tracks at ON ct.track_id = at.id
JOIN albums alb ON at.album_id = alb.id
JOIN genres g ON at.genre_id = g.id
JOIN collections c ON ct.collection_id = c.id
WHERE c.name = 'Romans Collection';
```



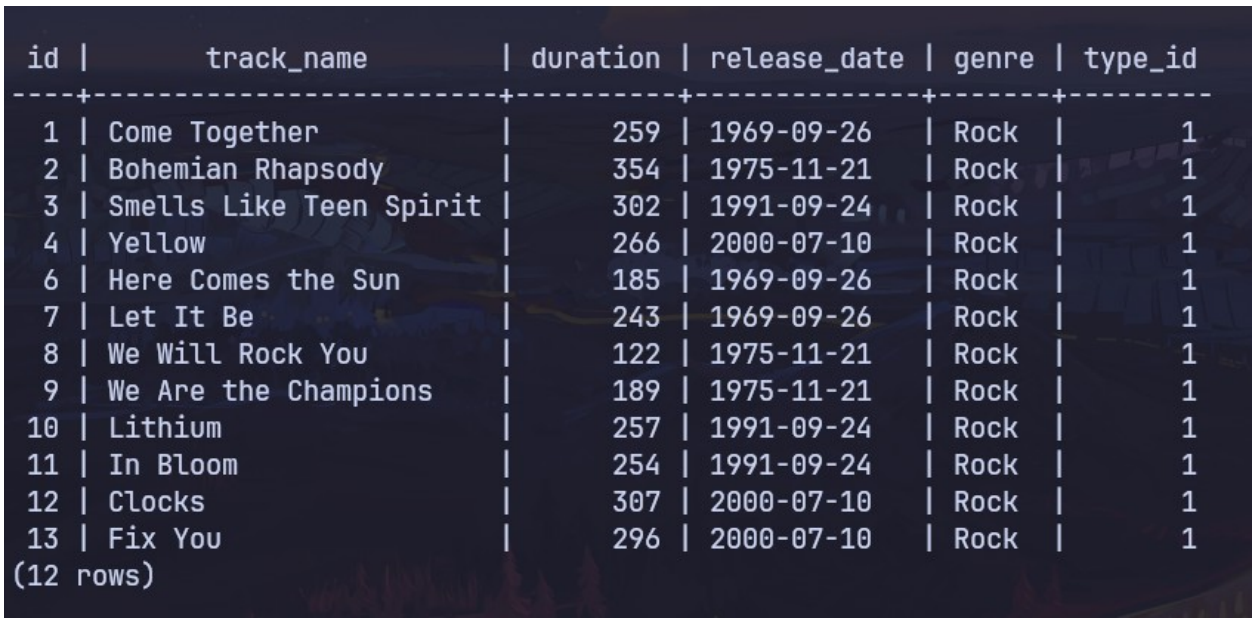
id	track_name	duration	release_date	genre	type_id
1	Come Together	259	1969-09-26	Rock	1
2	Bohemian Rhapsody	354	1975-11-21	Rock	1
3	Smells Like Teen Spirit	302	1991-09-24	Rock	1
4	Yellow	266	2000-07-10	Rock	1
5	ON	228	2020-02-21	Rock	1
6	Here Comes the Sun	185	1969-09-26	Rock	1
7	Let It Be	243	1969-09-26	Rock	1

(7 rows)

Рис 8.1 Коллекция пользователя

Листинг 4.1

```
-- look up tracks by The Beatles
SELECT at.id, at.name AS track_name, at.duration, at.release_date, g.name AS
genre, at.type_id
FROM audio_tracks at
JOIN albums alb ON at.album_id = alb.id
JOIN authors_artists aa ON aa.artist_id = alb.author_id
JOIN authors a ON aa.author_id = a.id
JOIN genres g ON at.genre_id = g.id
WHERE a.nickname = 'The Beatles';
```



id	track_name	duration	release_date	genre	type_id
1	Come Together	259	1969-09-26	Rock	1
2	Bohemian Rhapsody	354	1975-11-21	Rock	1
3	Smells Like Teen Spirit	302	1991-09-24	Rock	1
4	Yellow	266	2000-07-10	Rock	1
6	Here Comes the Sun	185	1969-09-26	Rock	1
7	Let It Be	243	1969-09-26	Rock	1
8	We Will Rock You	122	1975-11-21	Rock	1
9	We Are the Champions	189	1975-11-21	Rock	1
10	Lithium	257	1991-09-24	Rock	1
11	In Bloom	254	1991-09-24	Rock	1
12	Clocks	307	2000-07-10	Rock	1
13	Fix You	296	2000-07-10	Rock	1

(12 rows)

Рис 8.2 Песни группы The Beatles

Вывод: были получены практические навыки физического проектирования базы данных в СУБД SQL Server.