



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:

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

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

Вариант 18: Аудио коллекция.

Создание БД с помощью PostgreSQL.



```
roman=# CREATE DATABASE audio_collection
roman=# ;
CREATE DATABASE
```

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)
);

```

```
~/Documents/BMSTU/базы-данных/data-base master x
> psql -U roman -d audio_collection -f data.sql
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
```

Рис 2.1. Создание таблиц SQL

```
audio_collection=# \dt
List of relations
```

Schema	Name	Type	Owner
public	albums	table	roman
public	artists	table	roman
public	audio_tracks	table	roman
public	audio_types	table	roman
public	authors	table	roman
public	authors_artists	table	roman
public	collections	table	roman
public	collections_tracks	table	roman
public	genres	table	roman
public	tracks_authors	table	roman
public	users	table	roman

(11 rows)

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);

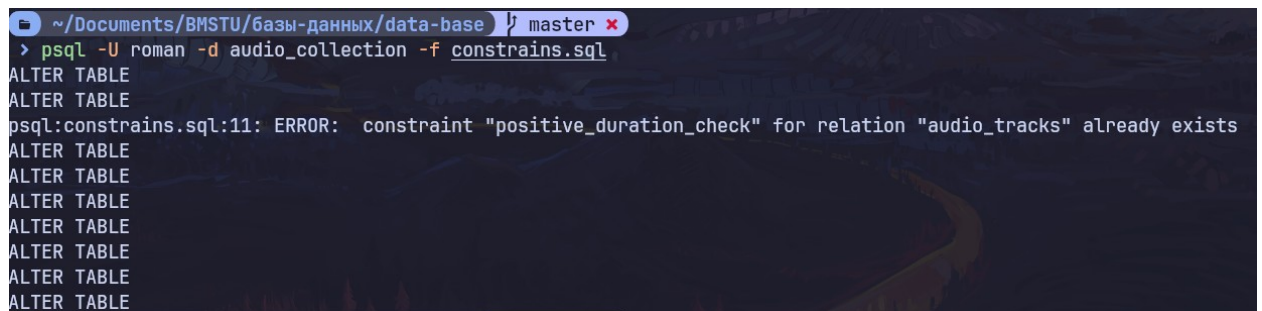
ALTER TABLE artists
ADD CONSTRAINT longer_than_0_characters_full_name_check
CHECK (LENGTH(full_name) > 0);

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);
```

A screenshot of a terminal window with a dark background. The window title is '~ / Documents / BMSTU / базы-данных / data-base master'. The prompt is '>'. The user has entered the command 'psql -U roman -d audio\_collection -f constrains.sql'. The terminal shows the output of the SQL script, which includes several 'ALTER TABLE' statements. An error message is displayed: 'psql:constrains.sql:11: ERROR: constraint "positive\_duration\_check" for relation "audio\_tracks" already exists'. The error message is highlighted in red. The terminal shows the following sequence of commands and output:

```
> psql -U roman -d audio_collection -f constrains.sql
ALTER TABLE
ALTER TABLE
psql:constrains.sql:11: ERROR: constraint "positive_duration_check" for relation "audio_tracks" already exists
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
```

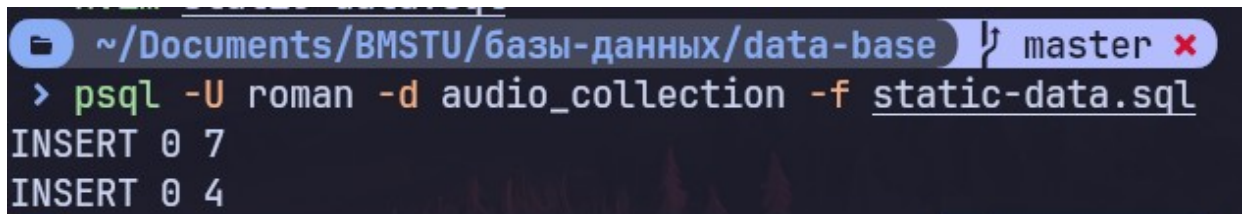
Рис 3. Ограничения целостности

## Добавление данных.

### Листинг 3.1

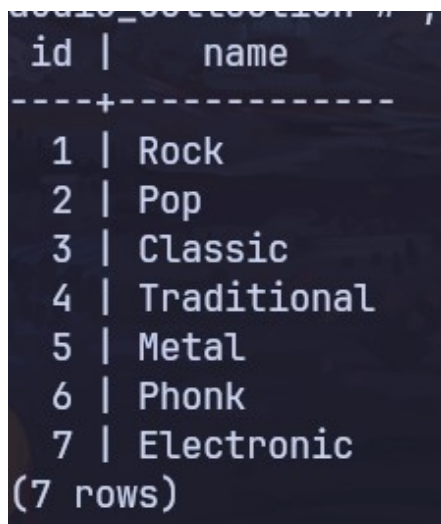
```
INSERT INTO genres (name)
VALUES ('Rock'), ('Pop'), ('Classic'), ('Traditional'), ('Metal'), ('Phonk'),
('Electronic');

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



The screenshot shows a terminal window with a dark background. The title bar indicates the path `~/Documents/BMSTU/базы-данных/data-base` and the user `master`. The command prompt shows the execution of `psql -U roman -d audio_collection -f static-data.sql`. The output shows two successful insert operations: `INSERT 0 7` and `INSERT 0 4`.

Рис 4.1 Ввод Жанров и Типов.

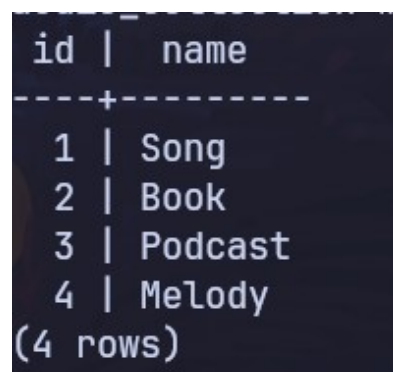


The screenshot shows the output of a SQL query, likely `\d genres`, displaying the structure and data of the 'genres' table. The table has two columns: 'id' and 'name'. The data is as follows:

id	name
1	Rock
2	Pop
3	Classic
4	Traditional
5	Metal
6	Phonk
7	Electronic

(7 rows)

Рис 4.2 Таблица Жанров



The screenshot shows the output of a SQL query, likely `\d audio_types`, displaying the structure and data of the 'audio\_types' table. The table has two columns: 'id' and 'name'. The data is as follows:

id	name
1	Song
2	Book
3	Podcast
4	Melody

(4 rows)

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
```

```
~/Documents/BMSTU/базы-данных/data-base master x
> psql -U roman -d audio_collection -f insert-Authors.sql
INSERT 0 5
INSERT 0 22
INSERT 0 22
```

Рис 5.1 Ввод Авторы и Исполнителей

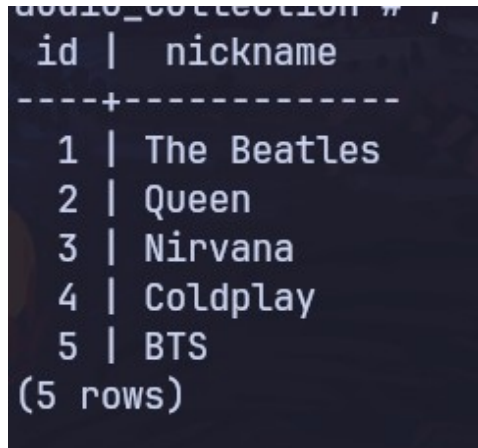
audio\_collection=# ,

id	full_name
1	John Lennon
2	Paul McCartney
3	George Harrison
4	Ringo Starr
5	Freddie Mercury
6	Brian May
7	Roger Taylor
8	John Deacon
9	Kurt Cobain
10	Krist Novoselic
11	Dave Grohl
12	Chris Martin
13	Jonny Buckland
14	Guy Berryman
15	Will Champion
16	RM
17	Jin
18	Suga
19	J-Hope
20	Jimin
21	V
22	Jungkook

(22 rows)

Рис 5.2 Таблица Исполнителей





id	nickname
1	The Beatles
2	Queen
3	Nirvana
4	Coldplay
5	BTS

(5 rows)

Рис 5.3 Таблица Авторы

### 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

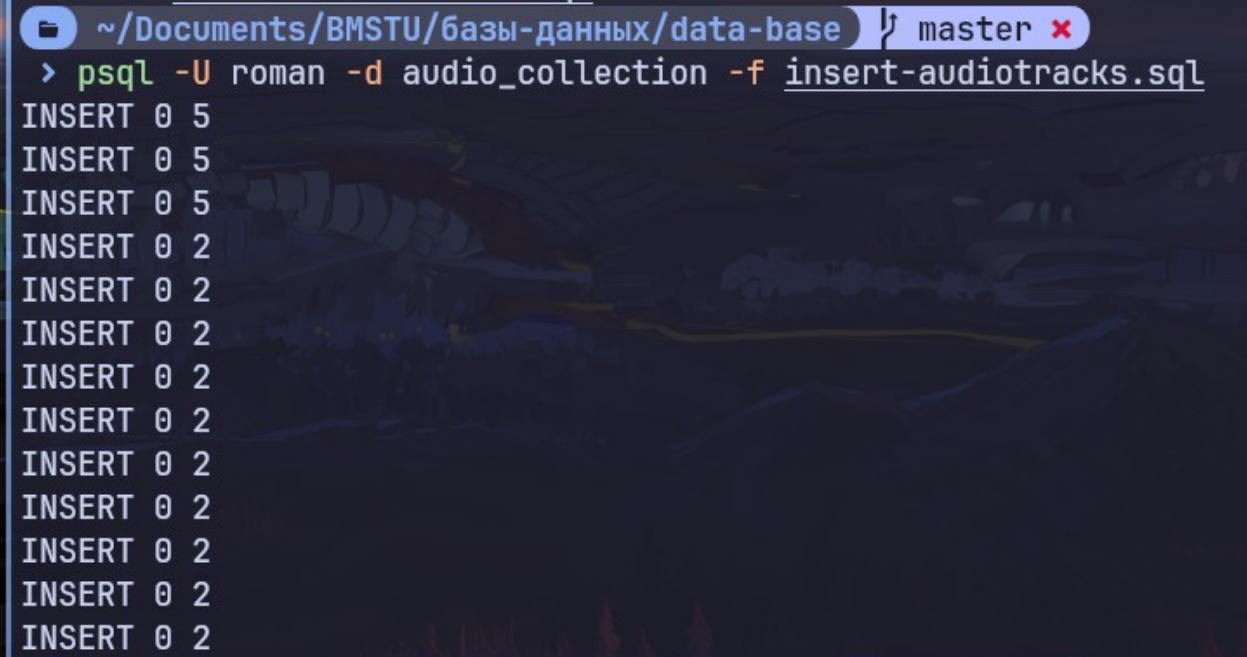
-- Queen's tracks
INSERT INTO tracks_authors (track_id, author_id) VALUES
(8, 2), -- We Will Rock You: Queen
(9, 2); -- We Are the Champions: Queen

-- 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

```



```

~/.Documents/BMSTU/базы-данных/data-base master x
> psql -U roman -d audio_collection -f insert-audiotracks.sql
INSERT 0 5
INSERT 0 5
INSERT 0 5
INSERT 0 2
INSERT 0 2
INSERT 0 2
INSERT 0 2
INSERT 0 2
INSERT 0 2
INSERT 0 2
INSERT 0 2
INSERT 0 2
INSERT 0 2
INSERT 0 2

```

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);

```

```

~/Documents/ВМSTU/базы-данных/data-base master x
> psql -U roman -d audio_collection -f insert-user.sql
INSERT 0 1
INSERT 0 1
  id
----
   1
(1 row)

INSERT 0 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.