

**Национальный исследовательский ядерный
университет «МИФИ»**



**Институт
интеллектуальных кибернетических систем**

Кафедра кибернетики (№ 22)

**Отчёт о работе по курсу
«Базы данных (теоретические основы баз данных)»
Вариант «Яндекс Музыка»**

Выполнил	Федоров Е.А.
Группа	Б19-501
Вариант	Яндекс Музыка
Преподаватель	Петровская А.В.
Проверяющий	Старков С.В
Оценка	

Оглавление

1. Формулировка задания	3
2. Концептуальная модель базы данных	3
3. Логическое проектирование.	7
4. Физическое проектирование	8
5. Выполнение запросов	21

1. Формулировка задания

Спроектировать базу данных для интернет-сервиса прослушивания музыки «Яндекс Музыка». База должна содержать информацию о пользователях, пользующихся сервисом, о треках и сборниках треков и их исполнителях, музыкальных предпочтениях пользователей.

2. Концептуальная модель базы данных

В ходе работы над заданием была проанализирована предметная область, особенности которой отражает построенная концептуальная модель:

Концептуальная модель базы данных для предметной области «Яндекс Музыка» (Рис. 1).

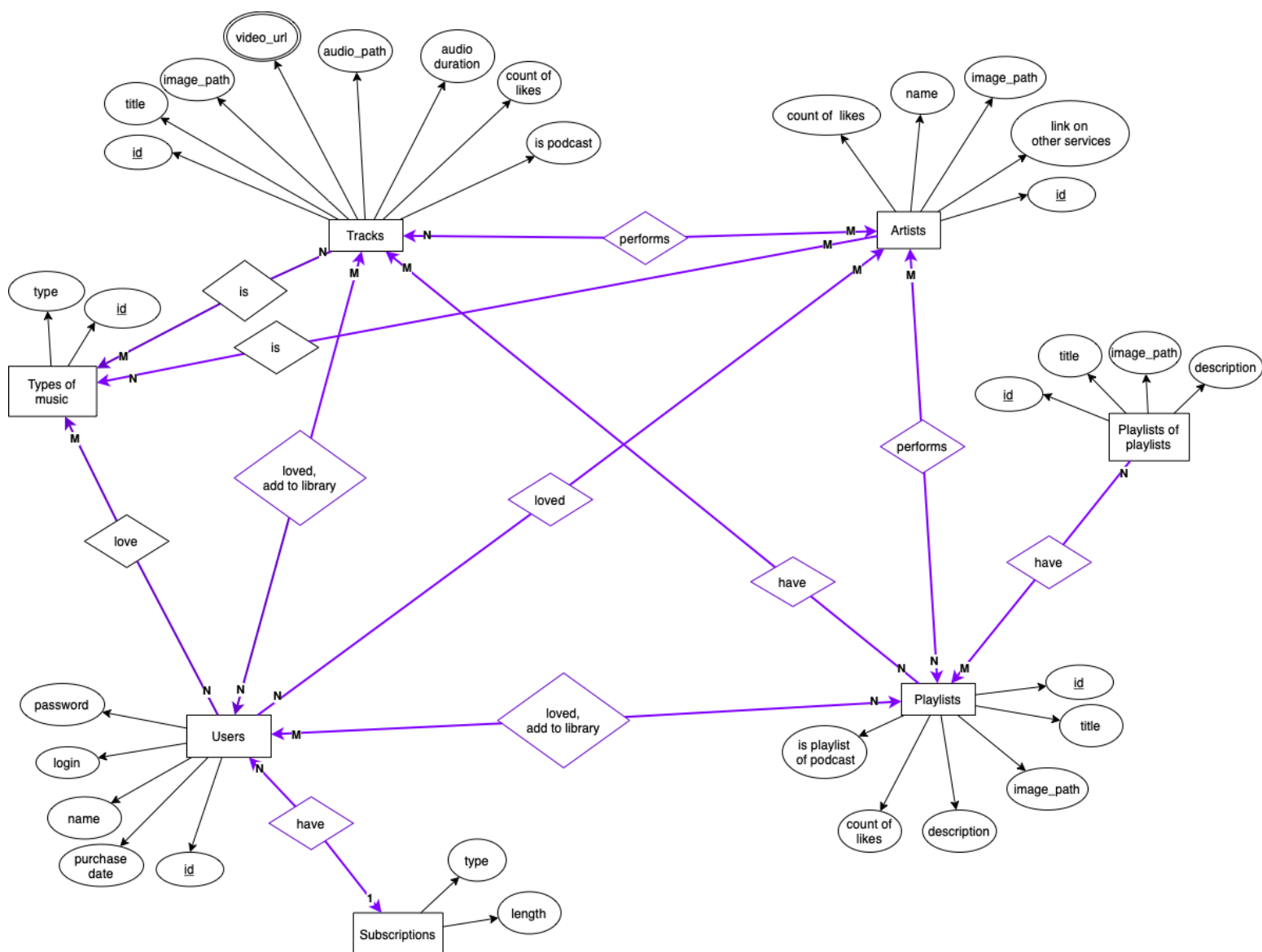


Рис. 1 - Концептуальная модель

2.1. Конкретизация предметной области

Необходимо создать систему, которая отражает основную информацию о музыке, пользователях и артистах, находящихся в интернет сервисе “Яндекс Музыка”. Система также должна хранить информацию о музыкальных предпочтениях пользователей, информацию о подписке пользователя, информацию о треках и сборниках треков и их исполнителях.

2.2. Описание предметной области

Музыкальная площадка «Яндекс Музыка» ориентирована на зарегистрированных пользователей и не предоставляет незарегистрированным пользователям возможность прослушивания музыки.

Рассмотрим далее возможности, которыми обладают пользователи:

1. Пользователь может добавить трек в избранное, так как “Яндекс Музыка” это платформа для прослушивания музыки, то этот функционал является обязательным.
2. Пользователь может добавить исполнителя в избранное, этот функционал также является очень нужным, по тем же причинам что описаны в предыдущем пункте.
3. Аналогично первым двум пунктам, пользователь может добавить в избранное плейлист.
4. Также пользователь может установить жанры музыки которые он предпочитает, это также является обязательным, так как пользователи хотят видеть в рекомендациях именно ту музыку которая им потенциально может понравиться.

Исходя из описания возможностей, предоставляемых интернет-сервисом

«Яндекс Музыка», были выделены следующие сущности:

1. Пользователь (User);
2. Трек (Track);
3. Плейлист (Playlist);
4. Плейлист плейлистов (Playlist of playlist);
5. Артист (Artist);
6. Подписка (Subscription)
7. Жанр музыки (Type of music);

2.3. Описание атрибутов

Для сущности пользователь (User):

Атрибут	Расшифровка
id	Уникальный идентификатор пользователя.
name	Имя пользователя
login	Логин пользователя
password	Пароль пользователя
type of subscription	Тип подписки пользователя
purchase date	Дата покупки подписки

Для сущности трек (Track):

Атрибут	Расшифровка
id	Уникальный идентификатор трека.
title	Название трека.
image_path	Ссылка на обложку трека
video path	Ссылка на видео(у некоторых треков есть видео клипы)
audio path	Ссылка на аудио файл трека
audio duration	Продолжительность трека
count of likes	Количество лайков на треке
is podcast	Свойство которое указывает на то, подкаст это или нет

Для сущности подписка (Subscription):

Атрибут	Расшифровка
type	Тип подписки
length	Длина подписки в днях

Для сущности жанра музыки (Type of music)

Атрибут	Расшифровка
id	Уникальный идентификатор жанра.
type	Жанр музыки

Для сущности артист (Artist):

Атрибут	Расшифровка
id	Уникальный идентификатор артиста.
name	Имя артиста
image_path	Ссылка на обложку артиста
count of likes	Количество лайков у артиста
link on other services	Ссылка на любой другой сервис исполнителя(один сервис)

Для сущности плейлист (Playlist):

Атрибут	Расшифровка
id	Уникальный идентификатор плейлиста.
title	Название плейлиста
image_path	Ссылка на обложку плейлиста
description	Описание плейлиста
count of likes	Количество лайков на плейлисте
is playlist of podcast	Логическое значение указывающее на то, плейлист подкастов это или же нет

Для сущности плейлист плейлистов (Playlist of playlist):

Атрибут	Расшифровка
id	Уникальный идентификатор плейлиста плейлистов
title	Название плейлиста плейлистов
image_path	Ссылка на обложку плейлиста плейлистов
description	Описание плейлиста плейлистов

3. Логическое проектирование.

Далее, на основе КМПО была разработана логическая модель базы данных, все связи «многие ко многим» были реализованы в виде отдельных таблиц (Рис. 2). Стоит отметить, что данная база данных приведена к 3 нормальной форме, соответственно к 1 и 2 тоже.

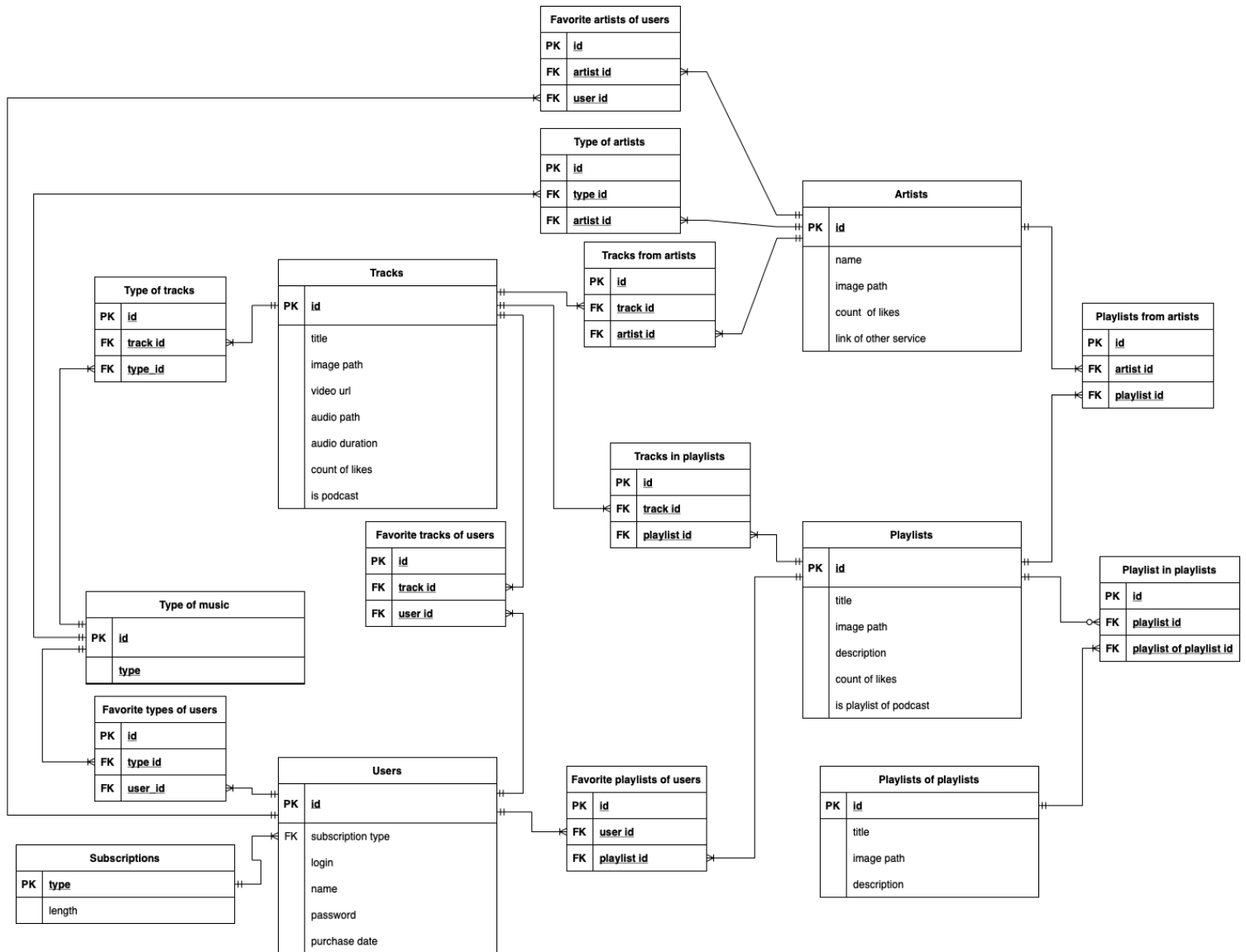


Рис. 2 - Логическая модель

4. Физическое проектирование

В качестве СУБД для реализации разработанной базы данных был выбран DataGrip. Исходя из приведенной логической модели была построена следующая физическая модель базы данных (Рис. 3):

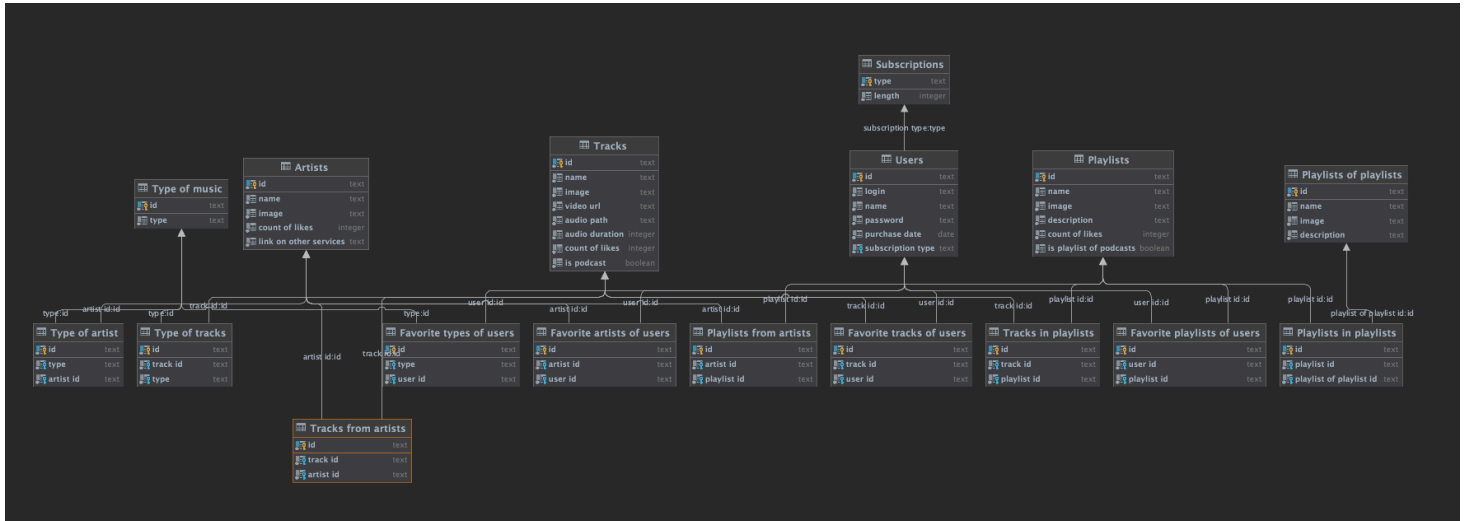


Рис. 3 - Физическая модель

На приведенном рисунке типы данных обозначаются следующими значениями:

- «integer» - целое число;
- «text» - строка;
- «boolean» - логическое значение;
- «date» - дата;
- Значок с желтым ключом - primary key;
- Значок с голубым ключом - foreign key;

4.1. Создание таблиц

Создание таблицы артистов (Artists):

```
CREATE TABLE "Artists"
(
    id INT,

    name TEXT NOT NULL,

    "image path" TEXT NOT NULL,

    "count of likes" INT NOT NULL DEFAULT
0,

    "link on other services" TEXT NOT
NULL,

    PRIMARY KEY (id)
);
```


Создание таблицы плейлистов (Playlists):

```
CREATE TABLE "Playlists"
(
    id INT,
    title TEXT NOT NULL,
    "image path" TEXT NOT NULL,
    description TEXT NOT NULL,
    "count of likes" INT NOT NULL DEFAULT
0,
    "is playlist of podcasts" BOOL NOT
NULL DEFAULT FALSE,
    PRIMARY KEY (id)
);
```

Создание таблицы плейлиста плейлистов (Playlist of playlists):

```
CREATE TABLE "Playlists of playlists"
(
    id INT,
    title TEXT NOT NULL,
    "image path" TEXT NOT NULL,
    description TEXT NOT NULL,
    PRIMARY KEY (id)
);
```

Создание таблицы подписок (Subscriptions):

```
CREATE TABLE "Subscriptions"
(
    type TEXT,
    length INT NOT NULL,
    PRIMARY KEY (type)
);
```

Создание таблицы треков (Tracks):

```
CREATE TABLE "Tracks"
(
  id INT,
  title TEXT NOT NULL,
  "image path" TEXT NOT NULL,
  "video url" TEXT,
  "audio path" TEXT NOT NULL,
  "audio duration" INT NOT NULL,
  "count of likes" INT NOT NULL DEFAULT 0,
  "is podcast" BOOL NOT NULL DEFAULT FALSE,
  PRIMARY KEY (id)
);
```

Создание таблицы типа музыки (Type of music):

```
CREATE TABLE "Type of music"
(
  id INT,
  type TEXT NOT NULL,
  PRIMARY KEY (id)
);
```

Создание таблицы пользователей (Users):

```
CREATE TABLE "Users"
(
    id INT,
    login TEXT NOT NULL,
    name TEXT NOT NULL,
    password TEXT NOT NULL,
    "purchase date" DATE NOT NULL,
    "subscription type" TEXT NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY ("subscription type") REFERENCES "Subscriptions"(type)
);
```

Создание таблицы, связывающей артистов и юзеров(Артисты которых слушает юзер)
(Favorite artists of users):

```
CREATE TABLE "Favorite artists of users"
(
    id INT,
    "artist id" INT NOT NULL,
    "user id" INT NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY ("artist id") REFERENCES "Artists"(id),
    FOREIGN KEY ("user id") REFERENCES "Users"(id)
);
```

Создание таблицы, связывающей плейлисты и юзеров (Favorite playlists of users):

```
CREATE TABLE "Favorite playlists of users"
(
    id INT,
    "user id" INT NOT NULL,
    "playlist id" INT NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY ("user id") REFERENCES "Users"(id),
    FOREIGN KEY ("playlist id") REFERENCES "Playlists"(id)
);
```

Создание таблицы, связывающей треки и юзеров (Favorite tracks of users):

```
CREATE TABLE "Favorite tracks of users"
(
    id INT,
    "track id" INT NOT NULL,
    "user id" INT NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY ("track id") REFERENCES "Tracks"(id),
    FOREIGN KEY ("user id") REFERENCES "Users"(id)
);
```

Создание таблицы, связывающей тип музыки с юзером (Favorite types of users):

```
CREATE TABLE "Favorite types of users"
(
  id INT,
  "type id" INT NOT NULL,
  "user id" INT NOT NULL,
  PRIMARY KEY (id),
  FOREIGN KEY ("type id") REFERENCES "Type of music"(id),
  FOREIGN KEY ("user id") REFERENCES "Users"(id)
);
```

Создание таблицы, связывающей плейлист с артистом (Playlists from artists):

```
CREATE TABLE "Playlists from artists"
(
  id INT,
  "artist id" INT NOT NULL,
  "playlist id" INT NOT NULL,
  PRIMARY KEY (id),
  FOREIGN KEY ("artist id") REFERENCES "Artists"(id),
  FOREIGN KEY ("playlist id") REFERENCES "Playlists"(id)
);
```

Создание таблицы, связывающей плейлист плейлистов с плейлистом(Playlists in playlists):

```
CREATE TABLE "Playlists in playlists"
(
  id INT,
  "playlist id" INT NOT NULL,
  "playlist of playlist id" INT NOT NULL,
  PRIMARY KEY (id),
  FOREIGN KEY ("playlist id") REFERENCES "Playlists"(id),
  FOREIGN KEY ("playlist of playlist id") REFERENCES "Playlists of playlists"(id)
);
```

Создание таблицы, связывающей треки с артистами(Tracks from artists):

```
CREATE TABLE "Tracks from artists"
(
  id INT,
  "track id" INT NOT NULL,
  "artist id" INT NOT NULL,
  PRIMARY KEY (id),
  FOREIGN KEY ("track id") REFERENCES "Tracks"(id),
  FOREIGN KEY ("artist id") REFERENCES "Artists"(id)
);
```

Создание таблицы, связывающей треки с плейлистами(Tracks in playlists):

```
CREATE TABLE "Tracks in playlists"
(
  id INT,
  "track id" INT NOT NULL,
  "playlist id" INT NOT NULL,
  PRIMARY KEY (id),
  FOREIGN KEY ("track id") REFERENCES "Tracks"(id),
  FOREIGN KEY ("playlist id") REFERENCES "Playlists"(id)
);
```

Создание таблицы, связывающей тип музыки с артистами(Type of artists):

```
CREATE TABLE "Type of artist"
(
  id INT,
  "type id" INT NOT NULL,
  "artist id" INT NOT NULL,
  PRIMARY KEY (id),
  FOREIGN KEY ("type id") REFERENCES "Type of music"(id),
  FOREIGN KEY ("artist id") REFERENCES "Artists"(id)
);
```

Создание таблицы, связывающей тип музыки с артистами(Type of tracks):

```
CREATE TABLE "Type of tracks"
(
  id INT,
  "track id" INT NOT NULL,
  "type id" INT NOT NULL,
  PRIMARY KEY (id),
  FOREIGN KEY ("track id") REFERENCES "Tracks"(id),
  FOREIGN KEY ("type id") REFERENCES "Type of music"(id)
);
```

4.2. Заполнение базы данных

Все базы заполнялись скриптом на языке Swift, представленном в приложении

4.3. Результаты заполнения

Таблица артистов(Artists) в ней 150 записей:

	id	name	image path	count of likes	link on other services
1	0	John Davis	https:\B1mPY0r	249720	https:\KWbypb8Mn4
2	1	Teresa Hill	https:\N3R3jrI	639879	https:\rHBu1igG1U
3	2	Louise Robinson	https:\aVQU5Kz	717934	https:\ExG9S1hMk3
4	3	Justin Webster	https:\2ubnXbG	510269	https:\eS6bc9c8QT
5	4	Walter Washington	https:\D6lX2l4	387007	https:\AYumqgNSfN
6	5	Frances Alexander	https:\YQkjKm3	494517	https:\m0Yf81uG4A
7	6	William Colon	https:\nynqF8d	188848	https:\fD68HSJYPh
8	7	Winkie J. Hill	https:\kbnjW8	25423	https:\YfY...

Таблица любимых артистов(Favorite artists of users) в ней 423 записи:

	id	artist id	user id
1	0	51	0
2	1	116	0
3	2	92	1
4	3	39	1
5	4	0	1

Таблица любимых плейлистов(Favorite playlists of users) в ней 444 записи:

	id	user id	playlist id
1	0	0	100
2	1	0	111
3	2	1	134
4	3	1	29
5	4	1	70
6	5	2	84

Таблица любимых треков(Favorite tracks of users) в ней 470 записей:

	id	track id	user id
1	0	8	0
2	1	122	0
3	2	204	0
4	3	276	1
5	4	82	1
6	5	149	2

Таблица любимых типов музыки(Favorite types of users) в ней 426 записей:

	id	type id	user id
1	0	9	0
2	1	1	0
3	2	4	0
4	3	14	0
5	4	16	0
6	5	6	0

Таблица плейлистов(Playlists) в ней 150 записей:

	id	title	image path	description	count of likes	is playlist of podcasts
1	0	Graswyn	https://8J6hnVw	some description0	197991	true
2	1	Nuandis	https://IEGb4xG	some description1	530491	false
3	2	Hugirus	https://tM2d09L	some description2	196331	false
4	3	Kelelen	https://3Q0VK4x	some description3	135091	false
5	4	Goltirg	https://wHJLIaC	some description4	546778	true
6	5	Adothis	https://gHJHhTN	some description5	396761	false
7	6	Samuzil	https://9pn0pnN	some description6	996702	false

Таблица плейлистов от артистов(Playlists from artists) в ней 500 записей:

	id	artist id	playlist id
1	0	0	47
2	1	0	65
3	2	0	93
4	3	0	25
5	4	1	82
6	5	1	149

Таблица плейлистов в плейлисте(Playlists in playlists) в ней 150 записей

	id	playlist id	playlist of playlist id
1	0	139	0
2	1	56	1
3	2	101	1
4	3	1	2
5	4	82	3

Таблица плейлистов плейлистов(Playlists of playlist) в ней 150 записей

	id	title	image path	description
1	0	Granimeena	https://DRkANn.jpg	some description0
2	1	Arashikree	https://eTCUvi.jpg	some description1
3	2	Dianagrinn	https://Zafp5L.jpg	some description2
4	3	Andromazan	https://BNLYk7.jpg	some description3
5	4	Anayardred	https://fKFLxA.jpg	some description4
6	5	Adrierdana	https://zcvqvN.jpg	some description5

Таблица подписок Subscriptions(представлена полностью)

	type	length
1	1 month	30
2	2 month	60
3	3 month	90
4	6 month	180
5	9 month	270
6	year	365

Таблица треков Tracks 300 записей

	id	title	image path	video url	audio path	audio duration	count of Likes	is podcast
1	0	Kazregis	https://7L4mXP.R.jpg	null	https://wFRxnh8.mp3	293	228198	false
2	1	ConjuLsa	https://2IA2Bwa.jpg	null	https://WvcX82A.mp3	88	786564	false
3	2	ZoLojinn	https://E12T49u.jpg	https://EQBVdhg.mp4	https://EApzCmj.mp3	133	131882	true
4	3	Bluefont	https://Jk2Zoqog.jpg	https://K1j6Zmc.mp4	https://gEN41e6.mp3	256	882976	false
5	4	Ballawyn	https://8XU23db.jpg	https://cgj81ge.mp4	https://wMvum3K.mp3	225	328151	false
6	5	Broadeye	https://pA4cFXT.jpg	https://oIr9HLI.mp4	https://9Q0hBq6.mp3	103	858774	false
7	6	Keleriel	https://zcr44eM.jpg	https://nmsw12t.mp4	https://CUQs6Z5.mp3	168	659833	false
8	7	Talrajcs	https://GaoFWc0.jpg	https://8NRua7K.mp4	https://bqfY6j2.mp3	62	781494	false
9	8	Ariughma	https://ked3MHv.jpg	https://gSvF3ne.mp4	https://dod8EIK.mp3	129	716132	false
10	9	Felhanis	https://ue6HIS5.jpg	https://XQm7NNY.mp4	https://0HAe74R.mp3	106	92345	true

Таблица треков от артистов Tracks from artists 500 записей

	id	track id	artist id
1	0	103	0
2	1	135	0
3	2	50	0
4	3	117	0
5	4	84	0
6	5	120	0
7	6	39	0
8	7	26	0

Таблица треков в плейлисте Tracks in playlists 500 значений

	id	track id	playlist id
1	0	110	0
2	1	77	0
3	2	52	0
4	3	64	0
5	4	201	1
6	5	266	1

Таблица типов музыки артиста Type of artists 190 значений

id	name	image path	count of likes	link on other services
132	John King	https:\wbAJZ6I	268457	https:\TQ1uMzR24d
133	John Miller	https:\ScNEyp2	218262	https:\H8y0IQuzn0
134	Kim McGuire	https:\okfzxGT	208256	https:\FDYCGofawp
135	Matthew Wright	https:\GGDKN8P	676354	https:\EoKV8wIOBN
136	Glen Diaz	https:\1IWj\Xz	368093	https:\c0fy14AUqw
137	Julia Phillips	https:\9UPEQWY	216413	https:\pIooPuS4JS
138	Michelle Schneider	https:\4cW0Tia	888761	https:\i2c7Q0HtMS

Таблица

типов музыки Type of music представлена полностью

id	type
0	blues
1	classical
2	country
3	dance
4	easy listening
5	electronic
6	folk
7	heavy metal
8	hip hop
9	jazz
10	Latin
11	opera
12	pop
13	rap
14	reggae
15	rock
16	techno

Таблица типа треков Type of tracks 500 значений

	id	track id	type id
1	0	0	16
2	1	1	9
3	2	1	13
4	3	2	16
5	4	2	7
6	5	2	4

Таблица юзеров Users 150 значений

	id	login	name	password	purchase date	subscription type
1	0	Samuel Ross	Dalabandis	Fautaxe	2019-05-08	6 mounth
2	1	Charlie Russell	Painwalker	Burilen	2019-06-07	9 mounth
3	2	Beatrice Wilson	Thorgaswyn	Sinfang	2019-06-20	3 mounth
4	3	Denise Garza	Sinbreaker	Aganara	2019-07-01	9 mounth
5	4	Tina Hunter	Bladestone	Taulkis	2019-07-03	mounth
6	5	Charles Carter	Dawnsinger	Keraron	2019-07-11	3 mounth

5. Выполнение запросов

В этом разделе приведены запросы, которые были написаны к разработанной базе данных в рамках курса.

- 1) Вывести id 5 треков у которых нет видео

```
SELECT id FROM "Tracks"
WHERE "video url" IS NOT NULL
ORDER BY "count of likes" DESC LIMIT(5);
```

Результат

	id
1	182
2	284
3	168
4	66
5	287

2) Вывести id и login тех пользователей у кого подписка куплена раньше 2019-10-10

```
SELECT id, login FROM "Users"  
WHERE "purchase date" <= '2019-10-10';
```

Результат(показана не вся таблица, потому что она весьма большая)

	id	login
1	0	Samuel Ross
2	1	Charlie Russell
3	2	Beatrice Wilson
4	3	Denise Garza
5	4	Tina Hunter
6	5	Charles Carter
7	6	Mary Manning
8	7	Robert Morgan
9	8	Joyce Ortiz
10	9	Joe Williamson
11	10	Ernest Jackson

3) Вывести id login name у тех людей которые слушают подкасты длинней 200 секунд

```
SELECT id, login, name FROM "Users"  
WHERE id in (  
    SELECT "user id" FROM "Favorite tracks of users"  
    WHERE "track id" in (  
        SELECT id FROM "Tracks"  
        WHERE "is podcast" = true AND "audio duration" >= 600  
    )  
);
```

Результат(показана не вся таблица)

	id	login	name
1	2	Beatrice Wilson	Thorgaswyn
2	4	Tina Hunter	Bladestone
3	5	Charles Carter	Dawnsinger
4	6	Mary Manning	Balladorin
5	7	Robert Morgan	Doombourne
6	8	Joyce Ortiz	Samushakar
7	11	Marilyn Johnson	Munillador
8	14	Mabel Johnson	Mighty Flame

4) Вывести сколько треков каждого исполнителя есть у пользователя в порядке возрастания

```
SELECT
    "artist id",
    COUNT("track id") AS countOfTracks
FROM
    "Favorite tracks of users" JOIN "Tracks from artists"
    ON "Favorite tracks of users"."track id" = "Tracks from artists"."track id"
WHERE "user id" = 11123421
GROUP BY "artist id"
ORDER BY countOfTracks;
```

Результат(только часть таблицы)

	artist id	countoftracks
20	125	1
21	26	1
22	49	2
23	2	2

5) Обновить количество лайков у исполнителей(Пробежались по всем трекам и посчитали сумму)

```
CREATE TEMP TABLE tempTable AS SELECT "artist id", SUM("Tracks"."count of likes")
AS countOfLikes FROM
"Tracks from artists" JOIN "Tracks" ON "Tracks from artists"."track id" =
"Tracks".id
GROUP BY "artist id";

UPDATE "Artists"
SET "count of likes" = tempTable.countOfLikes
FROM tempTable
WHERE "Artists".id = tempTable."artist id";
```

6) Смотрим самого прослушиваемого артиста и смотрим сколько его треки слушают без добавления (На след. странице)

```
WITH MostLikedArtists AS (  
    SELECT id as "artist id" FROM "Artists"  
    WHERE "count of likes" IN (  
        SELECT  
            MAX("count of likes")  
        FROM "Artists"  
    )  
, TrackOfMostLikedArtistsStatistics AS (  
    SELECT "track id",  
        COUNT("user id") AS count_of_love_users  
    FROM "Favorite tracks of users"  
    WHERE "track id" IN (  
        SELECT  
            "track id"  
        FROM MostLikedArtists JOIN "Tracks from artists" USING("artist id")  
    )  
    GROUP BY "track id"  
    ORDER BY count_of_love_users  
, TracksOfMostLikedFullInfoStatistics AS (  
    SELECT  
        "track id",  
        title,  
        "count of likes" - count_of_love_users as likes_without_add_track_in_library  
    FROM TrackOfMostLikedArtistsStatistics JOIN "Tracks"  
        ON TrackOfMostLikedArtistsStatistics."track id" = "Tracks".id  
    ORDER BY likes_without_add_track_in_library DESC  
)
```



```

, TracksFullInfoStatisticsWithPlaylist AS (
    SELECT
        "track id",
        TracksOfMostLikedFullInfoStatistics.title,
        likes_without_add_track_in_library,
        "playlist id"
    FROM TracksOfMostLikedFullInfoStatistics
        JOIN "Tracks in playlists" USING("track id")
), FinalStat AS (
    SELECT
        "track id",
        TracksFullInfoStatisticsWithPlaylist.title,
        likes_without_add_track_in_library,
        "playlist id",
        "Playlists".title as playlist_title
    FROM TracksFullInfoStatisticsWithPlaylist JOIN "Playlists" ON
TracksFullInfoStatisticsWithPlaylist."playlist id" = "Playlists".id
)
SELECT * FROM FinalStat

```

Результат(и исполнителя один трек)

"track id"	title	likes_without_add_track_in_library	"playlist id"	playlist_title
23	Vudojinn	829832	18	Hugiron

7 Запрос

- 1) Берем юзера у которого больше всего жанров музыки лайкнуто
- 2) Смотрим все треки которые ему понравились и смотрим на их жанр, выявляем жанр который чаще всего встречается в треках
- 3) Берем понравившиеся ему плейлисты и смотрим треки в них и выявляем опять самый частовстречающийся жанр
- 4) Таким образом мы имеем два жанра и количество треков в этом жанре
- 5) И так мы понимаем тот жанр который больше всего нравится пользователю

```

WITH MostLikedTypedUser as (

    SELECT

        "user id",

        COUNT("user id") as count_of_liked_types

    FROM "Favorite types of users"

    GROUP BY "user id"

    ORDER BY count_of_liked_types DESC

    LIMIT(1)

), MostLikedTypedUserIds AS (

    SELECT "user id" FROM MostLikedTypedUser

), TracksOfUser AS (

    SELECT

        "track id"

    FROM "Favorite tracks of users"

    WHERE "user id" in (

        SELECT * FROM MostLikedTypedUserIds

    )

), CountOfTracksWithType AS (

    SELECT "type id", COUNT("type id") as count_of_tracks_with_type

    FROM TracksOfUser

        JOIN "Type of tracks" USING ("track id")

    GROUP BY "type id"

    ORDER BY count_of_tracks_with_type DESC

    LIMIT(1)

), PlaylistOfUser AS (

    SELECT

        "playlist id"

    FROM "Favorite playlists of users"

    WHERE "user id" in (

        SELECT * FROM MostLikedTypedUserIds

    )

),



```

```

CountOfPlaylistWithType AS (
    SELECT "track id" FROM "Tracks in playlists"
    WHERE "playlist id" in (
        SELECT * FROM PlaylistOfUser
    )
), CountOfTracksInPlaylistWithType AS (
    SELECT "type id", COUNT("type id") as count_of_tracks_with_type
    FROM CountOfPlaylistWithType
        JOIN "Type of tracks" USING ("track id")
    GROUP BY "type id"
    ORDER BY count_of_tracks_with_type DESC
    LIMIT(1)
)
SELECT * FROM CountOfTracksWithType
UNION ALL
SELECT * FROM CountOfTracksInPlaylistWithType
ORDER BY count_of_tracks_with_type DESC
LIMIT(1)

```

Результат(тип который больше всего нравится пользователю)

	 "type id" ▾	 count_of_tracks_with_type ▾
1	2	2

8) Мы выбираем плейлист плейлистов смотрим в его плейлисты, а затем в их треки и понимаем сколько лайков в сумме на этом плейлисте, сколько его суммарная длительность, и сколько людей слушало хотя бы один трек

```
WITH TracksInPlaylistOfPlaylist AS (  
    SELECT "track id"  
    FROM "Tracks in playlists"  
    WHERE "playlist id" IN (  
        SELECT "playlist id"  
        FROM "Playlists in playlists"  
        WHERE "playlist of playlist id" = 4  
    )  
, TracksInPlaylistOfPlaylistWithListeners AS (  
    SELECT  
        TracksInPlaylistOfPlaylist."track id",  
        "user id"  
    FROM TracksInPlaylistOfPlaylist JOIN "Favorite tracks of users"  
        USING ("track id")  
, Stat AS (  
    SELECT  
        SUM("audio duration") AS total_duration,  
        SUM("count of likes") AS total_count_of_likes,  
        COUNT("user id") AS count_of_people  
    FROM TracksInPlaylistOfPlaylistWithListeners JOIN "Tracks" ON  
        "track id" = id  
    ORDER BY total_duration DESC  
) SELECT * FROM Stat
```

Результат

	total_duration ▾	total_count_of_likes ▾	count_of_people ▾
1	5203	11695123	25

9) Сколько юзеров используют каждую из типов подписок

```
SELECT "subscription type", COUNT("subscription type") as count_of_users FROM
"Users"

GROUP BY "subscription type"

ORDER BY count_of_users DESC
```

Результат

	"subscription type" ▾	count_of_users ▾
1	9 mounth	23
2	6 mounth	21
3	1 mounth	16

10) Средняя длина подписки

```
SELECT

    AVG(length)

FROM "Users" JOIN "Subscriptions"

ON "subscription type" = "Subscriptions".type
```




Результат

	avg ▾
1	178.35

11) Артисты которых никто не слушает

```
SELECT id FROM "Artists"
WHERE id NOT IN (
    SELECT DISTINCT "artist id" FROM "Favorite artists of users"
)
```





Результат

	  id 
1	13
2	15
3	30
4	38

12) Выведем те треки у которых больше 5 исполнителей

```
SELECT
    "track id",
    COUNT("track id") as count_of_artists
FROM "Tracks from artists"
GROUP BY "track id" HAVING COUNT("track id") > 5
```

Результат

	 "track id" 	 count_of_artists 
1	51	6
2	80	6
3	70	6
4	84	7