**Задание 1.**

База данных «Музыкальная коллекция» должна содержать информацию о музыкальных дисках, исполнителях, стилях. Необходимо хранить следующую информацию:

■ о музыкальном диске: название диска, исполнитель, дата выпуска, стиль, издатель;

■ о стилях: названия стилей;

■ об исполнителях: название;

■ об издателях: название, страна;

■ о песнях: название песни, название диска, длительность песни, музыкальный стиль песни, исполнитель.

**create** **database** MusicCollection;

**create** **table** Styles

(

StyleId **int** **not** **null** **auto\_increment** **primary** **key**,

NameStyle **varchar**(100) **not** **null**

);

**create** **table** Musicians

(

MusicianId **int** **not** **null** **auto\_increment** **primary** **key**,

NameMusician **varchar**(100) **not** **null**

);

**create** **table** Publishers

(

PublisherId **int** **not** **null** **auto\_increment** **primary** **key**,

NamePublisher **varchar**(100) **not** **null**,

Country **varchar**(100) **not** **null**

);

**create** **table** MusicDiscs

(

MusicDiscId **int** **not** **null** **auto\_increment** **primary** **key**,

NameMusicDisc **varchar**(200) **not** **null**,

MusicianId **int** **not** **null**,

ReleaseDate **date** **not** **null**,

StyleId **int** **not** **null**,

PublisherId **int** **not** **null**,

**foreign** **key** (MusicianId) **references** musicians(MusicianId),

**foreign** **key** (StyleId) **references** styles(StyleId),

**foreign** **key** (PublisherId) **references** publishers(PublisherId)

);

**create** **table** Songs

(

SongId **int** **not** **null** **auto\_increment** **primary** **key**,

NameSong **varchar**(200) **not** **null**,

MusicDiscId **int** **not** **null**,

SongDuration **datetime** **not** **null**,

StyleId **int** **not** **null**,

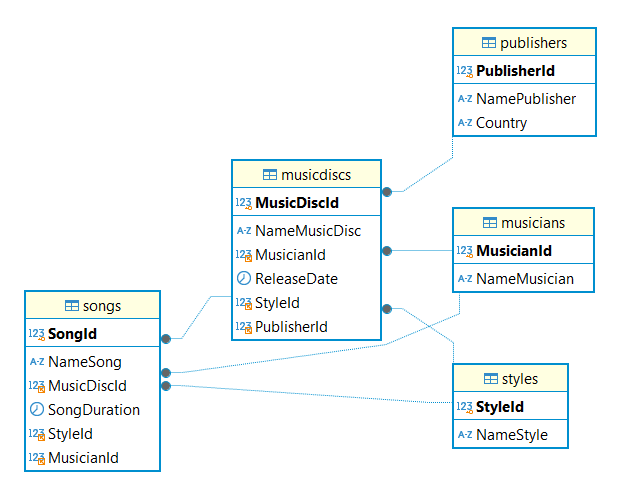
MusicianId **int** **not** **null**,

**foreign** **key** (MusicianId) **references** musicians(MusicianId),

**foreign** **key** (StyleId) **references** styles(StyleId),

**foreign** **key** (MusicDiscId) **references** musicdiscs(MusicDiscId)

);



Все задания необходимо выполнить по отношению к базе данных «Музыкальная коллекция», описанной в практическом задании для этого модуля. Создайте следующие представления:

■ Представление отображает названия всех исполнителей;

**select** \* **from** musicians;

■ Представление отображает полную информацию о всех песнях: название песни, название диска, длительность песни, музыкальный стиль песни, исполнитель;

**select** songs.SongId, songs.NameSong, musicdiscs.NameMusicDisc, songs.SongDuration, styles.NameStyle, musicians.NameMusician

**from** songs, musicdiscs, styles, musicians

**where** songs.MusicDiscId = musicdiscs.MusicDiscId

**and** songs.StyleId = styles.StyleId

**and** songs.MusicianId = musicians.MusicianId;

■ Представление отображает информацию о музыкальных дисках конкретной группы. Например, The Beatles;

**select** songs.NameSong, musicdiscs.NameMusicDisc, musicdiscs.ReleaseDate, styles.NameStyle, publishers.NamePublisher

**from** musicdiscs, songs, styles, publishers

**where** musicdiscs.MusicianId = songs.SongId

**and** musicdiscs.StyleId = styles.StyleId

**and** musicdiscs.PublisherId = publishers.PublisherId

**and** songs.NameSong = 'The Beatles';

■ Представление отображает название самого популярного в коллекции исполнителя. Популярность определяется по количеству дисков в коллекции;

**select** musicians.NameMusician

**from** musicians

**join** musicdiscs **on** musicians.MusicianId = musicdiscs.MusicianId

**group** **by** musicdiscs.MusicianId

**order** **by** **count**(\*) **desc**

**limit** 1;

■ Представление отображает топ-3 самых популярных в коллекции исполнителей. Популярность определяется по количеству дисков в коллекции;

**select** musicians.NameMusician, **count**(\*) **as** *total\_albums*

**from** musicians

**join** musicdiscs **on** musicians.MusicianId = musicdiscs.MusicianId

**group** **by** musicdiscs.MusicianId

**order** **by** *total\_albums* **desc**

**limit** 3;

■ Представление отображает самый долгий по длительности музыкальный альбом.

**select** musicdiscs.NameMusicDisc, **SUM**(songs.SongDuration) **as** *total\_duration*

**from** musicdiscs

**join** songs **on** musicdiscs.MusicDiscId = songs.MusicDiscId

**group** **by** musicdiscs.MusicDiscId

**order** **by** *total\_duration* **desc**

**limit** 1;

**Задание 2.**

Все задания необходимо выполнить по отношению к базе данных «Музыкальная коллекция», описанной в практическом задании для этого модуля:

■ Создайте обновляемое представление, которое позволит вставлять новые стили;

**insert** styles(NameStyle) **values** ('pank');

■ Создайте обновляемое представление, которое позволит вставлять новые песни;

**insert** songs(NameSong, MusicDiscId, SongDuration, StyleId, MusicianId) **values** ('Yesterday', 1, '0:02:03', 1, 1);

■ Создайте обновляемое представление, которое позволит обновлять информацию об издателе;

**select** \* **from** publishers;

■ Создайте обновляемое представление, которое позволит удалять исполнителей;

**delete** **from** musicians;

■ Создайте обновляемое представление, которое позволит обновлять информацию о конкретном исполнителе. Например, Muse.

**select** musicians.NameMusician, songs.NameSong, musicdiscs.NameMusicDisc **from** songs, musicians, musicdiscs

**where** songs.MusicianId = musicians.MusicianId

**and** songs.MusicDiscId = musicdiscs.MusicDiscId

**and** musicians.NameMusician = 'Muse';

**Задание 3.**

База данных «Продажи» должна содержать информацию о продавцах, покупателях, продажах. Необходимо хранить следующую информацию:

■ о продавцах: ФИО, email, контактный телефон;

■ о покупателях: ФИО, email, контактный телефон;

■ о продажах: покупатель, продавец, название товара, цена продажи, дата сделки.

**create** **database** sales;

**create** **table** sellers

(

SellersID **int** **not** **null** **auto\_increment** **primary** **key**,

FIO **varchar**(300) **not** **null**,

Email **varchar**(300) **not** **null**,

telephone **varchar**(300) **not** **null**

);

**create** **table** customers

(

CustomersID **int** **not** **null** **auto\_increment** **primary** **key**,

FIO **varchar**(300) **not** **null**,

Email **varchar**(300) **not** **null**,

telephone **varchar**(300) **not** **null**

);

**create** **table** sales

(

SalesID **int** **not** **null** **auto\_increment** **primary** **key**,

CustomersID **int** **not** **null**,

SellersID **int** **not** **null**,

ProductName **varchar**(100) **not** **null**,

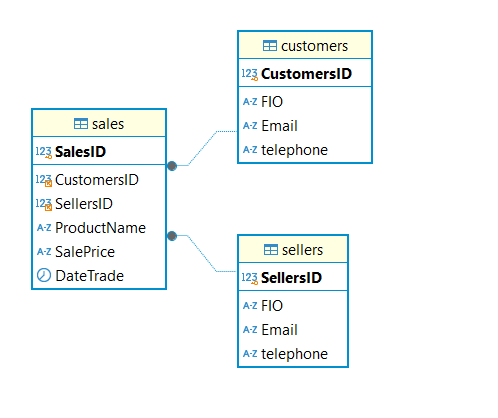
SalePrice **varchar**(300) **not** **null**,

DateTrade **date** **not** **null**,

**foreign** **key** (CustomersID) **references** customers(CustomersID),

**foreign** **key** (SellersID) **references** sellers(SellersID)

);



Все задания необходимо выполнить по отношению к базе данных «Продажи», описанной в практическом задании для этого модуля:

■ Создайте обновляемое представление, которое отображает информацию о всех продавцах;

**select** \* **from** sellers;

■ Создайте обновляемое представление, которое отображает информацию о всех покупателях;

**select** \* **from** customers;

■ Создайте обновляемое представление, которое отображает информацию о всех продажах конкретного товара. Например, яблок;

**select** \* **from** sales

**where** ProductName = 'яблоко';

■ Создайте представление, отображающее все осуществленные сделки;

**select** \* **from** sales;

■ Создайте представление, отображающее информацию о самом активном продавце. Определяем самого активного продавца по максимальной общей сумме продаж;

**select** sellers.FIO, sellers.Email, sellers.telephone **from** sellers **join** sales

**on** sellers.SellersID = sales.SellersID

**group** **by** sales.SellersID

**order** **by** **sum**(sales.SalePrice) **desc**

**limit** 1;

■ Создайте представление, отображающее информацию о самом активном покупателе. Определяем самого активного покупателя по максимальной общей сумме покупок.

**select** customers.FIO, customers.Email, customers.telephone **from** customers **join** sales

**on** customers.CustomersID = sales.CustomersID

**group** **by** sales.CustomersID

**order** **by** **sum**(sales.SalePrice) **desc**

**limit** 1;