**Задание 1.**

Для базы данных «Спортивный магазин» из практического задания создайте следующие хранимые процедуры:

■ Хранимая процедура отображает полную информацию о всех товарах;

**DELIMITER //**

**create** **procedure** GetAllProducts

**begin**

**select** \* **from** products

**end** //

**DELIMITER ;**

■ Хранимая процедура показывает полную информацию о товаре конкретного вида. Вид товара передаётся в качестве параметра. Например, если в качестве параметра указана обувь, нужно показать всю обувь, которая есть в наличии;

**DELIMITER //**

**create** **procedure** GetProductsByType (ProductType\_param **varchar**(100))

**begin**

**select** \* **from** products

**where** ProductType == ProductType\_param

**end** //

**DELIMITER ;**

■ Хранимая процедура показывает топ-3 самых старых клиентов. Топ-3 определяется по дате регистрации;

**DELIMITER //**

**create** **procedure** GetProductsByType

**begin**

**select** \* **from** clients

**order** **by** DateOfRegistration **asc**

**limit** 3

**end** //

**DELIMITER ;**

■ Хранимая процедура показывает информацию о самом успешном продавце. Успешность определяется по общей сумме продаж за всё время;

**DELIMITER //**

**create** **procedure** GetMostSuccessfulSalesperson

**begin**

**select** w.WorkerId, w.FIOWorker, **sum**(s.Quantity) **as** TotalQuantity

**from** workers w

**inner** **join** sales s **on** w.WorkerId = s.WorkerId

**group** **by** w.WorkerId, w.FIOWorker

**order** **by** TotalQuantity **desc**

**limit** 1

**end** //

**DELIMITER ;**

■ Хранимая процедура проверяет есть ли хоть один товар указанного производителя в наличии. Название производителя передаётся в качестве параметра. По итогам работы хранимая процедура должна вернуть yes в том случае, если товар есть, и no, если товара нет;

**DELIMITER //**

**create** **procedure** CheckProductByManufacturer (Manufacturer\_param **varchar**(100),

**Result** **varchar**(3) **output**)

**begin**

**if** **exists** (

**select** \* **from** products

**where** Manufacturer == Manufacturer\_param

) **then**

**set** **result** = 'yes'

**else**

**set** **result** = 'no'

**end** //

**DELIMITER ;**

■ Хранимая процедура отображает информацию о самом популярном производителе среди покупателей. Популярность среди покупателей определяется по общей сумме продаж;

**DELIMITER //**

**create** **procedure** GetMostPopularManufacturer

**begin**

**select** p.Manufacturer, **sum**(s.Quantity) **as** TotalQuantity

**from** sales s

**inner** **join** products p **on** s.ProductId = p.ProductId

**group** **by** p.Manufacturer

**order** **by** TotalQuantity **desc**

**limit** 1

**end** //

**DELIMITER ;**

■ Хранимая процедура удаляет всех клиентов, зарегистрированных после указанной даты. Дата передаётся в качестве параметра. Процедура возвращает количество удаленных записей.

**DELIMITER //**

**create** **procedure** DeleteCustomersAfterDate (DateOfRegistration\_param **date**, @

DeletedCount **int** **output**)

**begin**

**delete** **from** clients

**where** DateOfRegistration > DateOfRegistration\_param;

**set** @DeletedCount = @@rowcount

**end** //

**DELIMITER ;**

**Задание 2.**

Для базы данных «Музыкальная коллекция» из практического задания создайте следующие хранимые процедуры:

■ Хранимая процедура показывает полную информацию о музыкальных дисках;

**DELIMITER //**

**create** **procedure** GetFullDiskInformation

**begin**

**select** md.MusicDiscId, md.NameMusicDisc, m.NameMusician, md.ReleaseDate, st.NameStyle, p.NamePublisher, p.Country

**from** musicdiscs md

**inner** **join** musicians m **on** md.MusicianId = m.MusicianId

**inner** **join** styles st **on** md.StyleId = st.StyleId

**inner** **join** publishers p **on** md.PublisherId = p.PublisherId ;

**end** //

**DELIMITER ;**

■ Хранимая процедура показывает полную информацию о всех музыкальных дисках конкретного издателя. Название издателя передаётся в качестве параметра;

**DELIMITER //**

**create** **procedure** GetDiskInformationByIssuer (NamePublisher\_param **varchar**(100))

**begin**

**select** md.MusicDiscId, md.NameMusicDisc, m.NameMusician, md.ReleaseDate, st.NameStyle, p.NamePublisher, p.Country

**from** musicdiscs md

**inner** **join** musicians m **on** md.MusicianId = m.MusicianId

**inner** **join** styles st **on** md.StyleId = st.StyleId

**inner** **join** publishers p **on** md.PublisherId = p.PublisherId

**where** p.NamePublisher == NamePublisher\_param

**end** //

**DELIMITER ;**

■ Хранимая процедура показывает название самого популярного стиля**.** Популярность стиля определяется по количеству дисков в коллекции;

**DELIMITER //**

**create** **procedure** GetMostPopularStyle

**begin**

**select** styles.NameStyle **from** styles

**inner** **join** musicdiscs **on** musicdiscs.StyleId = styles.StyleId

**group** **by** styles.StyleId

**order** **by** **count**(musicdiscs.MusicDiscId) **desc**

**limit** 1

**end** //

**DELIMITER ;**

■ Хранимая процедура отображает информацию о диске конкретного стиля с наибольшим количеством песен. Название стиля передаётся в качестве параметра, если передано слово all, анализ идёт по всем стилям;

**DELIMITER //**

**create** **or** **alter** **procedure** GetDiskInfoByStyle (NameStyle\_param **varchar**(100))

**begin**

**if** NameStyle\_param == 'all' **then**

**begin**

**select** md.NameMusicDisc, m.NameMusician, st.NameStyle, **count**(\*) **as** NumOfSong

**from** musicdiscs md

**inner** **join** musicians m **on** md.MusicianId = m.MusicianId

**inner** **join** songs s **on** md.MusicianId = s.MusicDiscId

**inner** **join** styles st **on** md.StyleId = st.StyleId

**group** **by** md.NameMusicDisc, m.NameMusician, st.NameStyle

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

**limit** 1

**end**

**else**

**begin**

**select** md.NameMusicDisc, m.NameMusician, st.NameStyle, **count**(\*) **as** NumOfSong

**from** musicdiscs md

**inner** **join** musicians m **on** md.MusicianId = m.MusicianId

**inner** **join** songs s **on** md.MusicianId = s.MusicDiscId

**inner** **join** styles st **on** md.StyleId = st.StyleId

**where** st.NameStyle == NameStyle\_param

**group** **by** md.NameMusicDisc, m.NameMusician, st.NameStyle

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

**limit** 1

**end if**;

**end** //

**DELIMITER ;**

■ Хранимая процедура удаляет все диски заданного стиля. Название стиля передаётся в качестве параметра. Процедура возвращает количество удаленных альбомов;

**DELIMITER //**

**create** **procedure** DeleteDisksByStyle (NameStyle\_param **varchar**(100))

**begin**

**declare** @**Count** **int**

**set** @**Count** = (

**select** **count**(\*) **from** musicdiscs md

**inner** **join** styles st **on** md.StyleId = st.StyleId

**where** st.NameStyle == NameStyle\_param

)

**delete** **from** musicdiscs

**where** MusicDiscId **in** (

**select** musicdiscs.MusicDiscId **from** musicdiscs md

**inner** **join** styles st **on** md.StyleId = st.StyleId

**where** st.NameStyle == NameStyle\_param

)

**end** //

**DELIMITER ;**

■ Хранимая процедура отображает информацию о самом «старом» альбом и самом «молодом». Старость и молодость определяются по дате выпуска;

**DELIMITER //**

**create** **procedure** GetOldestAndNewestDisk

**begin**

**select** md.MusicDiscId, md.NameMusicDisc, m.NameMusician, md.ReleaseDate, st.NameStyle, p.NamePublisher, p.Country

**from** musicdiscs md

**inner** **join** musicians m **on** md.MusicianId = m.MusicianId

**inner** **join** styles st **on** md.StyleId = st.StyleId

**inner** **join** publishers p **on** md.PublisherId = p.PublisherId

**order** **by** **row\_number**() **over** (**partition** **by** md.ReleaseDate **order** **by** md.ReleaseDate)

**limit** 1

**end** //

**DELIMITER ;**

■ Хранимая процедура удаляет все диски, в названии которых есть заданное слово. Слово передаётся в качестве параметра. Процедура возвращает количество удаленных альбомов.

**DELIMITER //**

**create** **procedure** DeleteDisksByWord (word = **varchar**(100))

**begin**

**declare** @**Count** **int**

**set** @**Count** = (

**select** **count**(\*) **from** musicdiscs

**where** NameMusicDisc **like** '%' + word + '%'

)

**delete** **from** musicdiscs

**where** NameMusicDisc **like** '%' + word + '%'

**select** @**Count** **as** NumberOfDeletedDisks

**end** //

**DELIMITER ;**