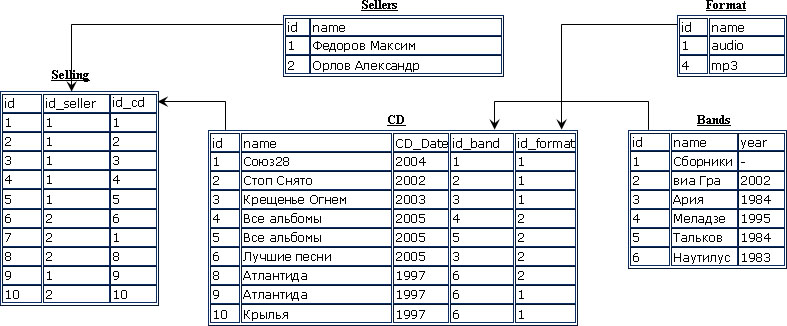
***Группа ИТ-2707. Леженко С.А.***

1. Используя запросы и ограничения создать базу данных, используя слудующую схему.



--Создаём базу

CREATE DATABASE CDShop;

USE CDShop;

--Создаём таблицу продавцов

CREATE TABLE Sellers

(id int identity(1,1) not null

Constraint PK\_Sellers primary key(id),

name varchar(255) not null

Constraint UniSellers UNIQUE(name));

--Заполняем

INSERT Sellers(name)

VALUES('Федоров Максим'),

('Орлов Александр');

--Создаём таблицу Format

CREATE TABLE Format

(id int identity(1,1) not null

Constraint PK\_Format primary key(id),

name varchar(25) not null

Constraint UniFormat UNIQUE(name));

--Наполняем таблицу Format

INSERT Format(name)

VALUES('audio'),('mp3');

--Создаём таблицу Bands

CREATE TABLE Bands

(id int identity(1,1) not null

Constraint PK\_Bands primary key(id),

name nvarchar(50) not null

Constraint UniBand UNIQUE(name),

year date);

--Заполняем таблицу Bands

INSERT Bands(name, year)

VALUES('Сборники',null),

('виа ГРА','2002'),

('Ария','1984'),

('Меладзе','1995'),

('Тальков','1984'),

('Наутилус','1983');

--Создаём таблицу CD

CREATE TABLE CD

(id int identity(1,1) not null

Constraint PK\_CD primary key(id),

name nvarchar(255) not null,

CD\_Date date not null,

id\_band int not null

Constraint FK\_Band foreign key(id\_band)

references Bands(id),

id\_format int not null

Constraint FK\_Format foreign key(id\_format)

references Format(id));

--Заполняем таблицу CD

INSERT CD

VALUES('Союз28','2004',1,1),

('Стоп снято','2002',2,1),

('Крещение огнем','2003',3,1),

('Все альбомы','2005',4,2),

('Все альбомы','2005',5,2),

('Лучшие песни','2005',3,2),

('Атлантида','1997',6,2),

('Атлантида','1997',6,1),

('Крылья','1997',6,1);

--Создаём таблицу продаж

CREATE TABLE Selling

(id int identity(1,1) not null

Constraint PK\_Selling primary key(id),

id\_seller int not null

Constraint FK\_Sellers foreign key(id\_seller)

references Sellers(id),

id\_cd int not null

Constraint FK\_CD foreign key(id\_cd)

references CD(id));

--Заполняем таблицу Selling

INSERT Selling

VALUES(1,1),

(1,2),

(1,3),

(1,4),

(1,5),

(2,6),

(2,1),

(2,7),

(1,8),

(2,9);

Заполнить ее значениями при помощи Insert и написать такие запросы

--Показать всю информацию о продажах

CREATE VIEW Selling\_View

AS

SELECT CD.name AS 'CD',

CD\_Date AS 'CD date',

Bands.name AS 'Band',

Format.name AS 'Format',

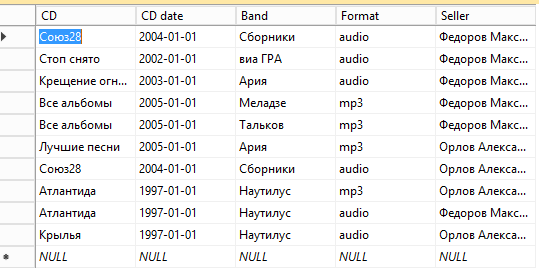
Sellers.name AS 'Seller'

FROM (((Selling LEFT JOIN CD ON Selling.id\_cd=CD.id)

LEFT JOIN Sellers ON Selling.id\_seller=Sellers.id)

LEFT JOIN Format ON CD.id\_format = Format.id)

LEFT JOIN Bands ON CD.id\_band = Bands.id;



--Показать кол-во проданных дисков по каждой из групп

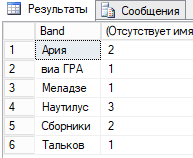
SELECT Bands.name AS 'Band',

COUNT(Selling.id\_cd)

FROM (Selling LEFT JOIN CD ON Selling.id\_cd=CD.id)

LEFT JOIN Bands ON CD.id\_band = Bands.id

GROUP BY Bands.name;



--Показать самую популярную группу

SELECT TOP(1) Bands.name AS 'Most popular Band'

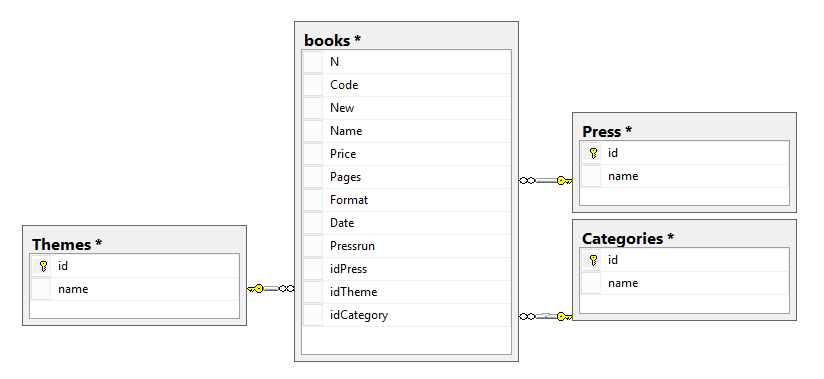
FROM (Selling LEFT JOIN CD ON Selling.id\_cd=CD.id)

LEFT JOIN Bands ON CD.id\_band = Bands.id

GROUP BY Bands.name

ORDER BY COUNT(id\_cd) DESC;





--Написать следующие запросы,

--используя базу данных Books (многотабличной)

USE books;

--Вычитать издательство, которое издало

--наибольшее кол-во книг по программированию

SELECT name AS 'Press having max(books) for Programming'

FROM Press

WHERE id IN

(SELECT TOP(1) idPress

FROM Books

WHERE idTheme =

(SELECT id

FROM Themes

WHERE name LIKE 'Программирование')

GROUP BY idPress

ORDER BY COUNT(N) DESC);



--Показать тематику,

--по которой издано наименьшее кол-во страниц

SELECT name AS 'Theme with MIN(Pages)'

FROM Themes

WHERE id IN

(SELECT TOP(1) idTheme

FROM Books

GROUP BY idTheme

ORDER BY SUM(Pages));



--Вычитать самую дорогую книгу издательства BHV

SELECT TOP(1) name AS 'Book with MAX(Price) of BHV Press'

FROM Books

WHERE idPress IN

(SELECT id

FROM Press

WHERE name LIKE '%BHV%')

ORDER BY Price DESC;



--Вычитать книги, у которых кол-во страниц больше чем среднее

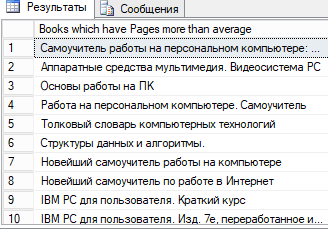
SELECT name AS 'Books which have Pages more than average'

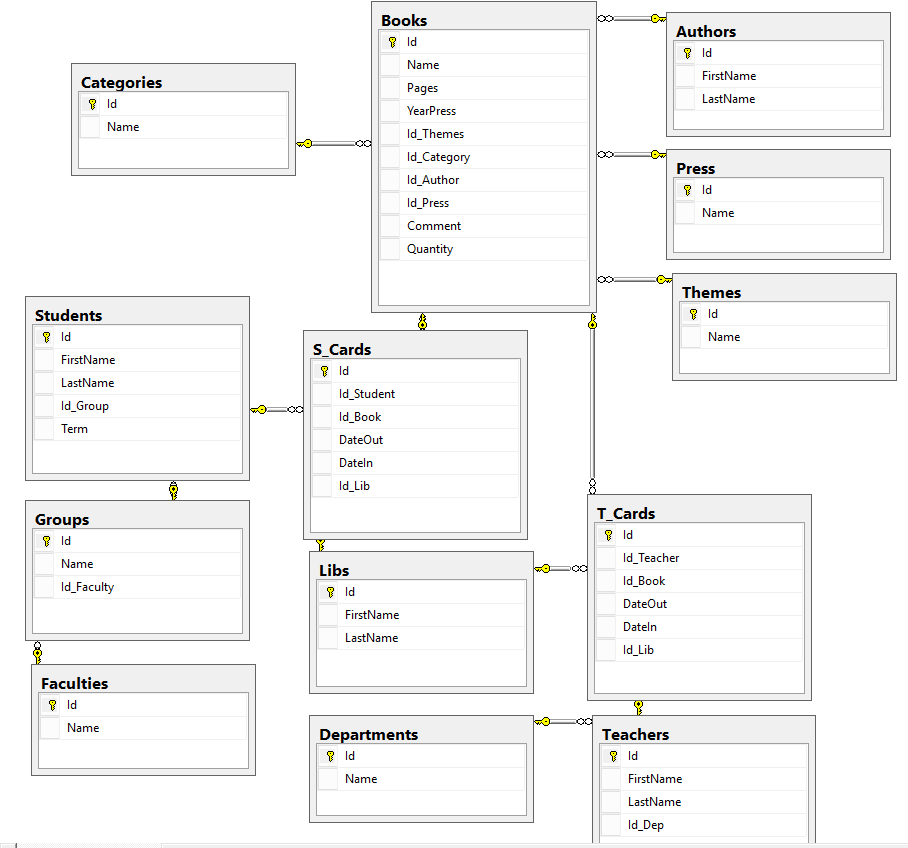
FROM Books

WHERE Pages >

(SELECT AVG(Pages)

FROM Books);





--Написать следующие запросы, используя базу данных Library

USE library;

--Показать автора самой популярной книги у студентов

SELECT FirstName+' '+LastName AS 'Most popular Author'

FROM Authors

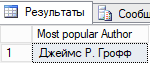
WHERE id IN

(SELECT TOP(1) id\_Author

FROM S\_Cards LEFT JOIN Books ON S\_Cards.id\_Book=Books.id

GROUP BY id\_Author, Books.id

ORDER BY COUNT(S\_Cards.id) DESC);



--Показать кафедру (department),

--которая брала наибольшее кол-во книг

SELECT Name AS 'Departmebt'

FROM Departments

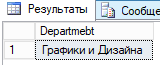
WHERE id IN

(SELECT TOP(1) id\_Dep

FROM T\_Cards LEFT JOIN Teachers ON T\_Cards.id\_Teacher=Teachers.id

GROUP BY id\_Dep

ORDER BY COUNT(T\_Cards.id) DESC);



--Показать тематику, самую популярную среди преподавателей

SELECT Name AS 'Themes are the most semilater in teachers'

FROM Themes

WHERE id IN

(SELECT TOP(1) id\_Themes

FROM T\_Cards LEFT JOIN Books ON T\_Cards.id\_Book=Books.id

GROUP BY id\_Themes

ORDER BY COUNT(T\_Cards.id) DESC);



