ОГБПОУ «ТОМСКИЙ ТЕХНИКУМ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»

Итоговый отчет по учебной практике   
*УП.07 Управление и автоматизация баз данных*

Выполнил:

студент 691-ой группы

Маметьев С. И.

Проверил:

преподаватель

Бугаев А. С.

г. Томск – 2021 г

1 неделя

Задание 1. [Проектирование структуры баз данных. Построение логической и физической модели данных](https://moodle.tomtit.tomsk.ru/mod/assign/view.php?id=43775).

Пусть в некоторой таблице "Товары" хранится следующая информация об имеющейся на складе продукции:

* наименование товара;
* тип товара;
* дата производства;
* срок годности;
* дата поставки;
* цена единицы товара;
* склад, на который товар поставлен;
* сведения о поставщике товара (наименование организации, юридический адрес, контактный телефон);
* количество товаров.

Задание 1: Построить логическую модель базы данных. Сохранить результат в файле Задание\_1.pdf

Задание 2: Построить физическую модель базы данных. Сохранить результат в файле Задание\_2.pdf

Примечание:

Физическая модель – логическая модель базы данных, выраженная в терминах языка описания данных конкретной СУБД.

Физическая модель базы данных содержит все детали, необходимые конкретной СУБД для создания базы: наименования таблиц и столбцов, типы данных, определения первичных и внешних ключей, ограничения уникальности и т.д..

Используемые инструменты: MS Visio, draw.io.

Результат работы:

Задание 1 (рис. 1):

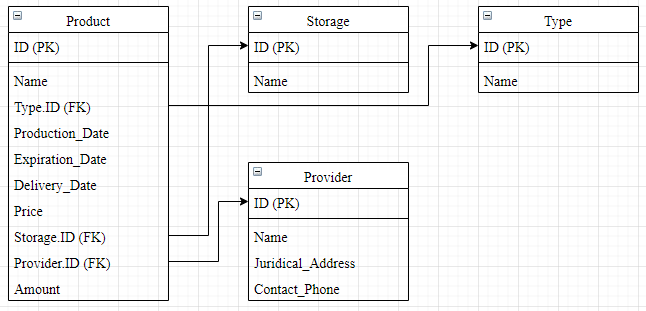


Рис. 1. Логическая модель базы данных

Задание 2 (рис. 2):

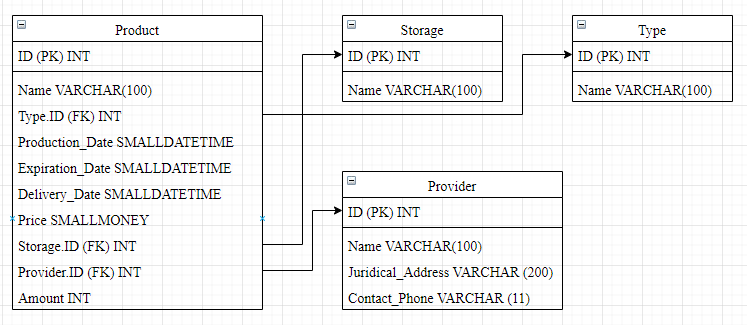


Рис. 2. Физическая модель базы данных

Задание 2. Создание объектов баз данных в различных СУБД

* Создать скрипт, позволяющий реализовать нормализованную схему базы данных из задания 1;
* Сохранить его в файле Structure.sql;
* Создать базу данных Store (если нет возможности, то на своей БД);
* Наполнить базу тестовыми данными (не менее 20 записей в основной таблице и не менее 10 в остальных);
* Сохранить скрипт, наполняющий базу данных в файле Insert.sql.

ВЫСЛАТЬ НА ПРОВЕРКУ 2 ФАЙЛА: Structure.sql, Insert.sql.

Файл “Structure.sql” содержит в себе:

USE gr691\_msi

GO

CREATE TABLE Storage

(

ID INT IDENTITY PRIMARY KEY,

Name VARCHAR(100) NOT NULL

);

CREATE TABLE Type

(

ID INT IDENTITY PRIMARY KEY,

Name VARCHAR(100) NOT NULL

);

CREATE TABLE Provider

(

ID INT IDENTITY PRIMARY KEY,

Name VARCHAR(100) NOT NULL,

Juridical\_Address VARCHAR(200) NOT NULL,

Contact\_Phone VARCHAR(11) NOT NULL

);

CREATE TABLE Product

(

ID INT IDENTITY PRIMARY KEY,

Name VARCHAR(100) NOT NULL,

TypeID INT FOREIGN KEY REFERENCES Type (ID),

Production\_Date SMALLDATETIME,

Expiration\_Date SMALLDATETIME,

Delivery\_Date SMALLDATETIME,

Price SMALLMONEY,

StorageID INT FOREIGN KEY REFERENCES Storage (ID),

ProviderID INT FOREIGN KEY REFERENCES Provider (ID),

Amount INT

);

Результат скрипта показан на рис. 3.

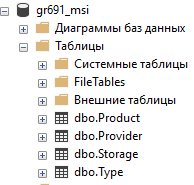


Рис. 3. Результат скрипта “Structure.sql”

Файл “Insert.sql” содержит в себе:

USE gr691\_msi

GO

INSERT INTO Provider VALUES

('Device', 'г. Верхняя Пышма, ул. Кривоусова, 18д', '89122390670'),

('Бизон-Сибирь', 'г. Новосибирск, ул. Каменская, 54', '89137883152'),

('diHouse', 'г. Москва, ул. Цирка, 2', '84956516290'),

('Флексмар', 'г. Москва, ул. Иловайская, 2б', '89067770868'),

('ОптиТрейд', 'г. Самара, ул. Авроры, 63', '88462074575'),

('Бука', 'г. Екатеринбург, ул. Шакалов, 10', '89122062207'),

('Сетевид', 'г. Москва, ул. 2-ая Карачаровская, 1', '89260296818'),

('Лепрекон Северо-Запад', 'г. Санкт-Петербург, ул. Нереальных, 4', '88125792840'),

('Arkonada', 'г. Екатеринбург, ул. Чайковского, 62', '83434887406'),

('ПозиТроника', 'г. Москва, ул. Строителей, 4', '84959818484');

INSERT INTO Type VALUES

('Компьютерный монитор'),

('Компьютерная мышь'),

('Компьютерная клавиатура'),

('Корпус'),

('Видеокарта'),

('Процессор'),

('Материнская плата'),

('Wi-Fi оборудование'),

('Веб-камера'),

('Наушники с микрофоном');

INSERT INTO Storage VALUES

('Для мониторов'),

('Для мышей'),

('Для клавиатуры'),

('Для корпусов'),

('Для видеокарт'),

('Для процессоров'),

('Для материнских плат'),

('Для Wi-Fi оборудования'),

('Для веб-камер'),

('Для наушников с микрофоном');

INSERT INTO Product VALUES

('18.5" Монитор AOC e970Swn/01', '1', '2010-04-23 12:00:00', '2025-04-23 12:00:00', '2011-01-21 15:43:48', '5299', '1', '1', '5'),

('37.5" Монитор Acer Predator X38P', '1', '2020-06-10 16:16:16', '2040-06-10 16:16:16', '2020-07-11 11:24:58', '179999', '1', '1', '2'),

('Мышь беспроводная Oklick 565MW белый', '2', '2015-03-05 08:30:20', '2018-01-05 08:30:20', '2015-04-01 10:15:20', '199', '2', '2', '4'),

('Мышь беспроводная/проводная Razer Viper Ultimate Cyberpunk 2077 черный', '2', '2021-01-25 21:00:00', '2029-01-25 21:00:00', '2021-02-01 11:35:23', '14999', '2', '2', '1'),

('Клавиатура RITMIX RKB-100', '3', '2017-07-27 15:15:20', '2021-07-27 15:15:20', '2017-08-01 17:54:34', '299', '3', '3', '3'),

('Клавиатура Logitech G915 TKL', '3', '2019-02-27 10:40:20', '2023-02-27 10:40:20', '2019-03-03 13:13:13', '17999', '3', '3', '5'),

('Корпус DEXP DC-201M черный', '4', '2011-04-23 13:00:00', '2025-04-23 12:00:00', '2012-01-21 15:43:48', '1699', '4', '4', '5'),

('Корпус Thermaltake DistroCase 350P', '4', '2011-04-23 14:01:10', '2015-01-23 07:10:01', '2012-01-21 16:43:30', '45999', '4', '4', '6'),

('Видеокарта KFA2 GeForce 210', '5', '2012-04-23 15:02:20', '2016-02-24 08:20:02', '2013-01-21 17:42:31', '3099', '5', '5', '7'),

('Видеокарта PNY Quadro RTX 6000', '5', '2012-04-23 16:05:30', '2017-03-25 09:30:03', '2013-01-21 18:41:32', '150000', '5', '5', '8'),

('Процессор AMD Athlon X4 840 OEM', '6', '2013-04-23 17:06:40', '2018-04-26 10:40:04', '2014-01-21 19:40:33', '1950', '6', '6', '9'),

('Процессор AMD Ryzen Threadripper', '6', '2013-04-23 18:07:50', '2019-05-27 11:50:05', '2014-01-21 20:44:34', '143999', '6', '6', '8'),

('Материнская плата ASRock A320M-DVS R4.0', '7', '2014-04-23 19:10:15', '2020-06-11 12:55:06', '2015-01-21 21:45:35', '3799', '7', '7', '7'),

('Материнская плата GIGABYTE X299X AORUS XTREME WATERFORCE', '7', '2014-04-23 20:20:25', '2021-07-13 13:56:07', '2015-01-21 22:46:36', '129999', '7', '7', '6'),

('Точка доступа Ubiquiti UWB‑XG', '8', '2015-04-23 21:30:35', '2022-08-15 14:57:08', '2016-01-21 15:47:37', '142499', '8', '8', '5'),

('Усилитель беспроводного сигнала EDIMAX RE11', '8', '2015-04-23 22:40:45', '2023-09-17 15:58:09', '2016-01-21 14:48:38', '8750', '8', '8', '4'),

('Веб-камера Defender C-090', '9', '2016-04-23 13:50:55', '2024-10-19 16:59:10', '2017-01-21 13:49:39', '1499', '9', '9', '3'),

('Веб-камера Logitech BRIO', '9', '2017-04-23 12:55:05', '2025-11-21 17:54:11', '2017-01-21 12:50:40', '18199', '9', '9', '2'),

('Проводная гарнитура Sven AP-310M черный', '10', '2018-04-23 11:57:00', '2026-12-23 18:53:12', '2019-01-21 11:51:29', '599', '10', '10', '2'),

('Проводные наушники Fostex TH610 черный', '10', '2019-04-23 10:45:17', '2027-04-25 19:52:13', '2020-01-21 10:52:28', '48499', '10', '10', '1');

Результат скрипта показан на рис. 4.

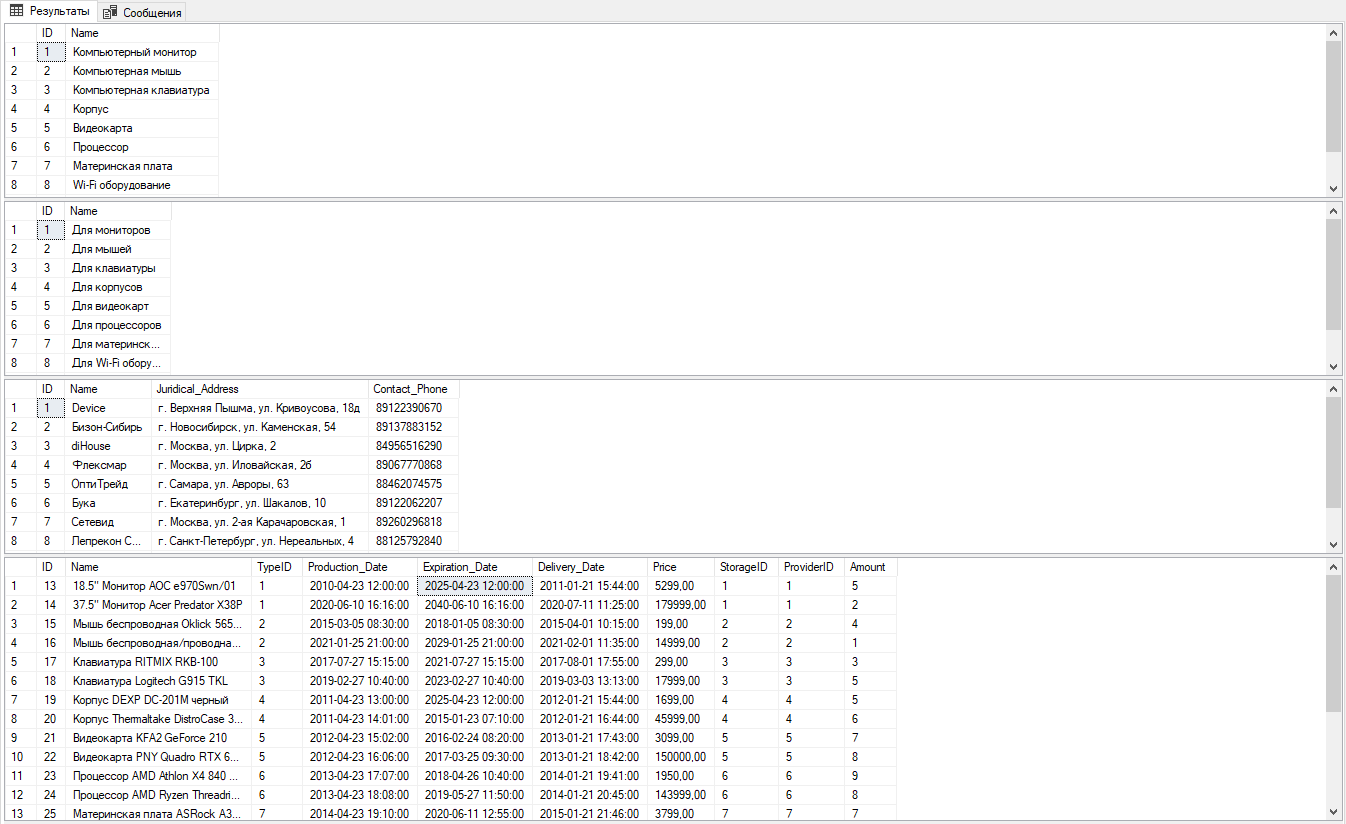


Рис. 4. Результат скрипта “Insert.sql”

Задание 3. Обработка данных с использование языка SQL.

Для базы данных реализовать указанные 10 запросов:

* Вывести все товары в ценовом диапазоне от 850 до 5000, которые поставляет фирма "Алиса";
* Найти товары, которые поставленные более полугода назад;
* Вывести список типов товаров и количество товаров данного типа;
* Вывести название товара, поставщика и стоимость тех товаров, у которых истек срок годности;
* Вывести все товары поставщиков из города Новосибирск, если дата производства у них позже 10.07.2018;
* Вывести максимальную стоимость товаров определенного типа;
* Вывести товары типа "молоко", у которого стоимость ниже среднего;
* Вывести название товара и его тип для тех товаров, у которых между датой производства и датой поставки не прошло 10 дней;
* Вывести список товаров, поставленных осенью 2018 года, у которых название состоит из 5 букв;
* Вывести первые 3 буквы названия товара, цену товара и день недели, в которую этот товар был поставлен.

Примечание: во всех запросах должна выводится результирующая таблица если искомые данные имеются в бд, либо сообщение, о том, что искомых данных нет (EXIST: http://www.sql-tutorial.ru/ru/book\_exists\_predicate.html).

Если у вас в таблицах отсутствуют искомые параметры, как в задач, то используйте свои искомые значения.

Сохранить запросы в файл Query.sql.

НА ПРОВЕРКУ ОТПРАВИТЬ ФАЙЛ Query.sql.

Файл “Query.sql” содержит в себе:

USE gr691\_msi

GO

--1-ый запрос

IF EXISTS (SELECT \* FROM Provider

WHERE Provider.Name = 'Алиса')

SELECT DISTINCT Product.Name AS 'Товар' FROM Product

JOIN Provider ON Provider.ID = Product.ProviderID

WHERE Product.Price BETWEEN '850' AND '5000' AND Provider.Name = 'Алиса'

ELSE

PRINT N'Искомых данных нет.'

--2-ой запрос

IF EXISTS (SELECT \* FROM Product

WHERE DATEDIFF(month, Product.Delivery\_Date, SYSDATETIME()) >= 6)

SELECT DISTINCT Product.Name AS 'Товар' FROM Product

WHERE DATEDIFF(month, Product.Delivery\_Date, SYSDATETIME()) >= 6

ELSE

PRINT N'Искомых данных нет.'

--3-ий запрос

SELECT DISTINCT Type.Name AS 'Тип', SUM(Product.Amount) AS 'Количество' FROM Type

JOIN Product ON Product.TypeID = Type.ID

GROUP BY Type.Name

--4-ый запрос

SELECT DISTINCT Product.Name AS 'Название', Provider.Name AS 'Поставщик', Product.Price AS 'Стоимость' FROM Product

JOIN Provider ON Provider.ID = Product.ProviderID

WHERE DATEDIFF(day, Product.Expiration\_Date, SYSDATETIME()) >= 1

--5-ый запрос

IF EXISTS (SELECT \* FROM Product

JOIN Provider ON Provider.ID = Product.ProviderID

WHERE Product.Production\_Date > '2018-07-10' AND Provider.Juridical\_Address LIKE '%Новосибирск%')

SELECT Product.Name AS 'Товар' FROM Product

JOIN Provider ON Provider.ID = Product.ProviderID

WHERE Product.Production\_Date > '2018-07-10' AND Provider.Juridical\_Address LIKE '%Новосибирск%'

ELSE

PRINT N'Искомых данных нет.'

--6-ой запрос

SELECT DISTINCT Type.Name AS 'Тип товара', MAX(Product.Price) AS 'Максимальная стоимость' FROM Type

JOIN Product ON Product.TypeID = Type.ID

WHERE Product.TypeID = '9'

GROUP BY Type.Name

--7-ой запрос

IF EXISTS (SELECT \* FROM Product

JOIN Type ON Type.ID = Product.TypeID

WHERE Type.Name LIKE '%видеокарта%'

AND Product.Price < (SELECT AVG(Product.Price) FROM Product))

SELECT Product.Name AS 'Товар' FROM Product

JOIN Type ON Type.ID = Product.TypeID

WHERE Type.Name LIKE '%видеокарта%'

AND Product.Price < (SELECT AVG(Product.Price) FROM Product)

ELSE

PRINT N'Искомых данных нет.'

--8-ой запрос

IF EXISTS (SELECT \* FROM Product

JOIN Type ON Type.ID = Product.TypeID

WHERE DATEDIFF(DAY, Product.Production\_Date, Product.Delivery\_Date) >= 10)

SELECT Product.Name AS 'Товар', Type.Name AS 'Тип товара' FROM Product

JOIN Type ON Type.ID = Product.TypeID

WHERE DATEDIFF(DAY, Product.Production\_Date, Product.Delivery\_Date) >= 10

ELSE

PRINT N'Искомых данных нет.'

--9-ый запрос

IF EXISTS (SELECT \* FROM Product

WHERE Product.Delivery\_Date > '2018-09-01' AND Product.Name LIKE '\_\_\_\_\_')

SELECT Product.Name AS 'Товар' FROM Product

WHERE Product.Delivery\_Date > '2018-09-01' AND Product.Name LIKE '\_\_\_\_\_'

ELSE

PRINT N'Искомых данных нет.'

--10-ый запрос

SELECT DISTINCT SUBSTRING(Product.Name, 1, 3) AS 'Название товара', Product.Price AS 'Стоимость',

DATENAME(WEEKDAY, Product.Delivery\_Date) AS 'День недели поставки товара' FROM Product

Результат 1-го запроса представлен на рисунке 5.

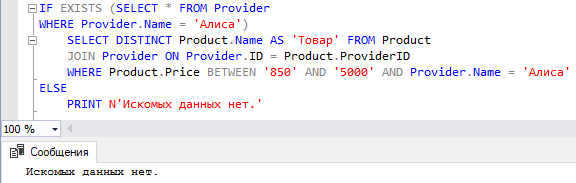


Рис. 5. Результат первого запроса

Результат 2-го запроса представлен на рисунке 6.

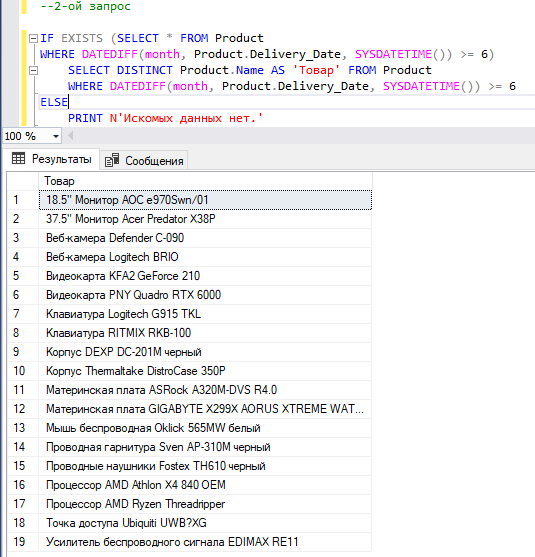


Рис. 6. Результат второго запроса

Результат 3-го запроса представлен на рисунке 7.

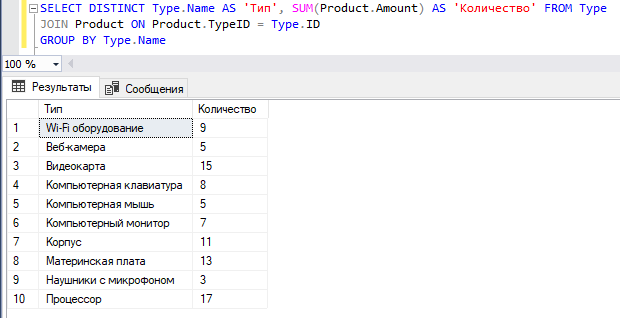


Рис. 7. Результат третьего запроса

Результат 4-го запроса представлен на рисунке 8.

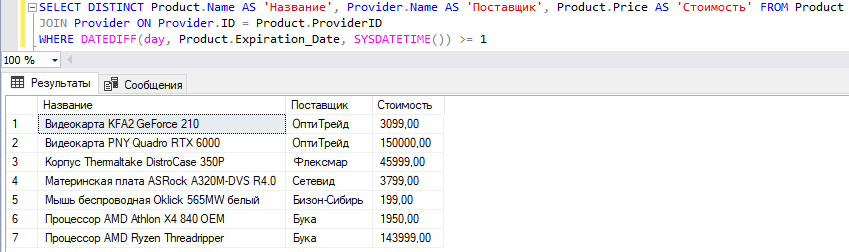


Рис. 8. Результат четвертого запроса

Результат 5-го запроса представлен на рисунке 9.

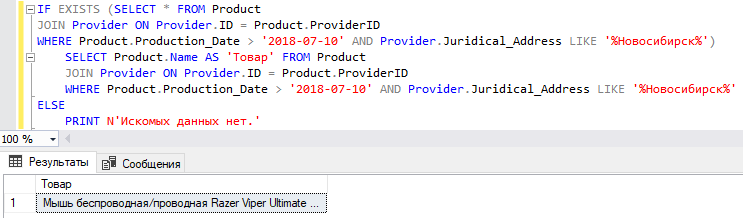


Рис. 9. Результат пятого запроса

Результат 6-го запроса представлен на рисунке 10.

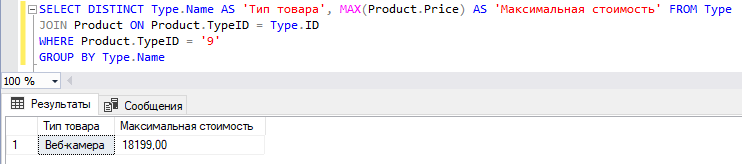


Рис. 10. Результат шестого запроса

Результат 7-го запроса представлен на рисунке 11.

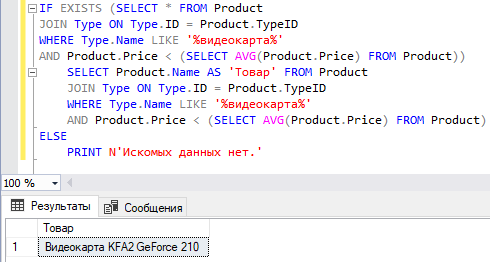


Рис. 11. Результат седьмого запроса

Результат 8-го запроса представлен на рисунке 12.

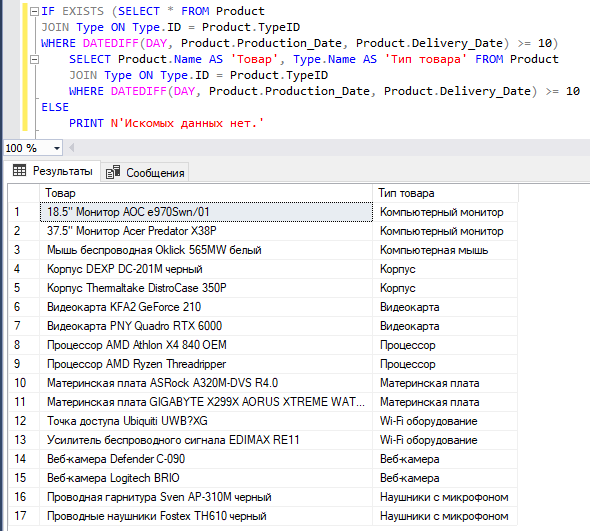


Рис. 12. Результат восьмого запроса

Результат 9-го запроса представлен на рисунке 13.

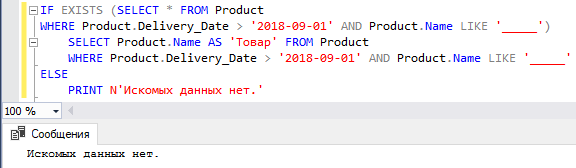


Рис. 13. Результат девятого запроса

Результат 10-го запроса представлен на рисунке 14.

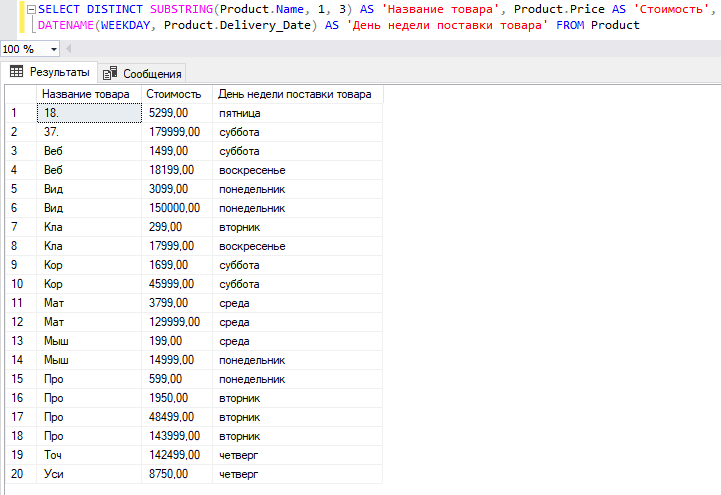


Рис. 14. Результат десятого запроса

Задание 4. Обработка данных с использование языка SQL

1. Дан скрипт (прикреплен ниже) на создание таблиц внутри БД (БД Store), необходимо его выполнить на своих экземплярах БД (рис. 16 и 17);

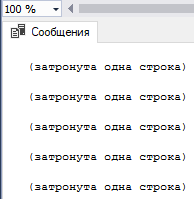


Рис. 16. Выполнен скрипт

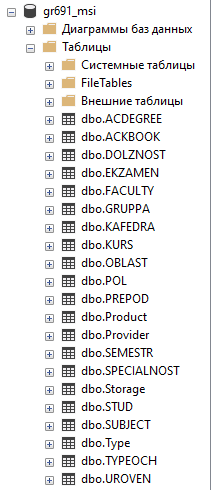


Рис. 17. Добавленные таблицы скриптом

2. Проверить целостность связей по схеме (Изображение прикреплено ниже) (если необходимо восстановить связи через запросы изменение таблиц, ЗАПРОСАМИ!!!!) (рис. 18 и 19);

Код запроса:

--Добавление и удаление ключей

ALTER TABLE Exam ADD FOREIGN KEY (FK\_Specialty\_ID) REFERENCES Specialty (ID) ON DELETE NO ACTION ON UPDATE NO ACTION

ALTER TABLE Exam ADD FOREIGN KEY (FK\_Subject\_ID) REFERENCES Subject (ID) ON DELETE NO ACTION ON UPDATE NO ACTION

ALTER TABLE Exam ADD FOREIGN KEY (FK\_Teacher\_ID) REFERENCES Teacher (ID) ON DELETE NO ACTION ON UPDATE NO ACTION

ALTER TABLE Exam DROP CONSTRAINT FK\_\_EKZAMEN\_\_FK\_SEME\_\_373B3228

ALTER TABLE [Group] ADD FOREIGN KEY (FK\_Faculty\_ID) REFERENCES Faculty (ID)

ALTER TABLE [Group] ADD FOREIGN KEY (FK\_Specialty\_ID) REFERENCES Specialty (ID)

ALTER TABLE [Group] DROP CONSTRAINT FK\_\_GRUPPA\_\_FK\_KURS\_\_\_39237A9A

ALTER TABLE Student DROP CONSTRAINT FK\_\_STUD\_\_FK\_GRUPPA\_\_\_42ACE4D4

ALTER TABLE Student DROP CONSTRAINT FK\_\_STUD\_\_FK\_POL\_ID\_\_43A1090D

ALTER TABLE Semester DROP CONSTRAINT FK\_\_SEMESTR\_\_FK\_KURS\_\_40C49C62

ALTER TABLE Record\_Book ADD FOREIGN KEY (FK\_Type\_Of\_Evaluation\_ID) REFERENCES Type\_Of\_Evaluation (ID) ON DELETE NO ACTION ON UPDATE NO ACTION

ALTER TABLE Record\_Book DROP CONSTRAINT FK\_\_ACKBOOK\_\_FK\_EKZA\_\_3552E9B6

ALTER TABLE Teacher DROP CONSTRAINT FK\_\_PREPOD\_\_FK\_ACDEG\_\_3BFFE745

ALTER TABLE Teacher DROP CONSTRAINT FK\_\_PREPOD\_\_FK\_DOLZN\_\_3CF40B7E

ALTER TABLE Teacher DROP CONSTRAINT FK\_\_PREPOD\_\_FK\_KAFED\_\_3B0BC30C

ALTER TABLE Academic\_Degree DROP CONSTRAINT FK\_\_ACDEGREE\_\_FK\_OBL\_\_318258D2

ALTER TABLE Academic\_Degree DROP CONSTRAINT FK\_\_ACDEGREE\_\_FK\_URO\_\_32767D0B

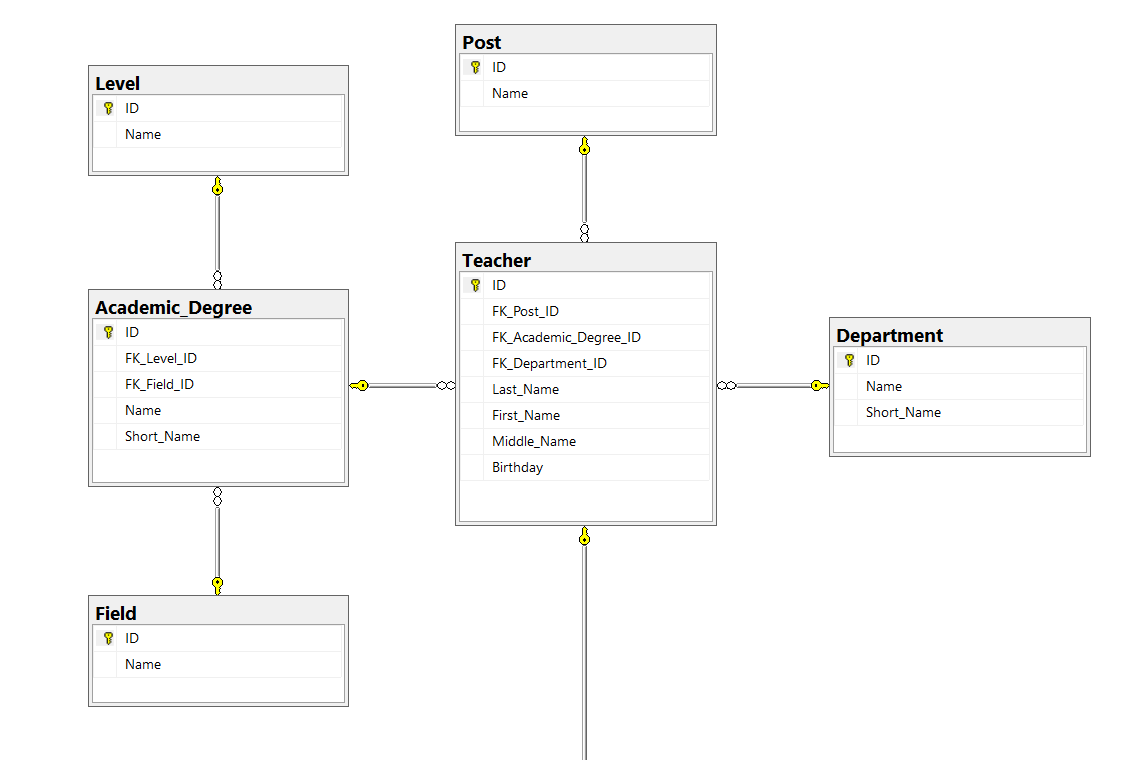


Рис. 18. Верхняя часть диаграммы баз данных

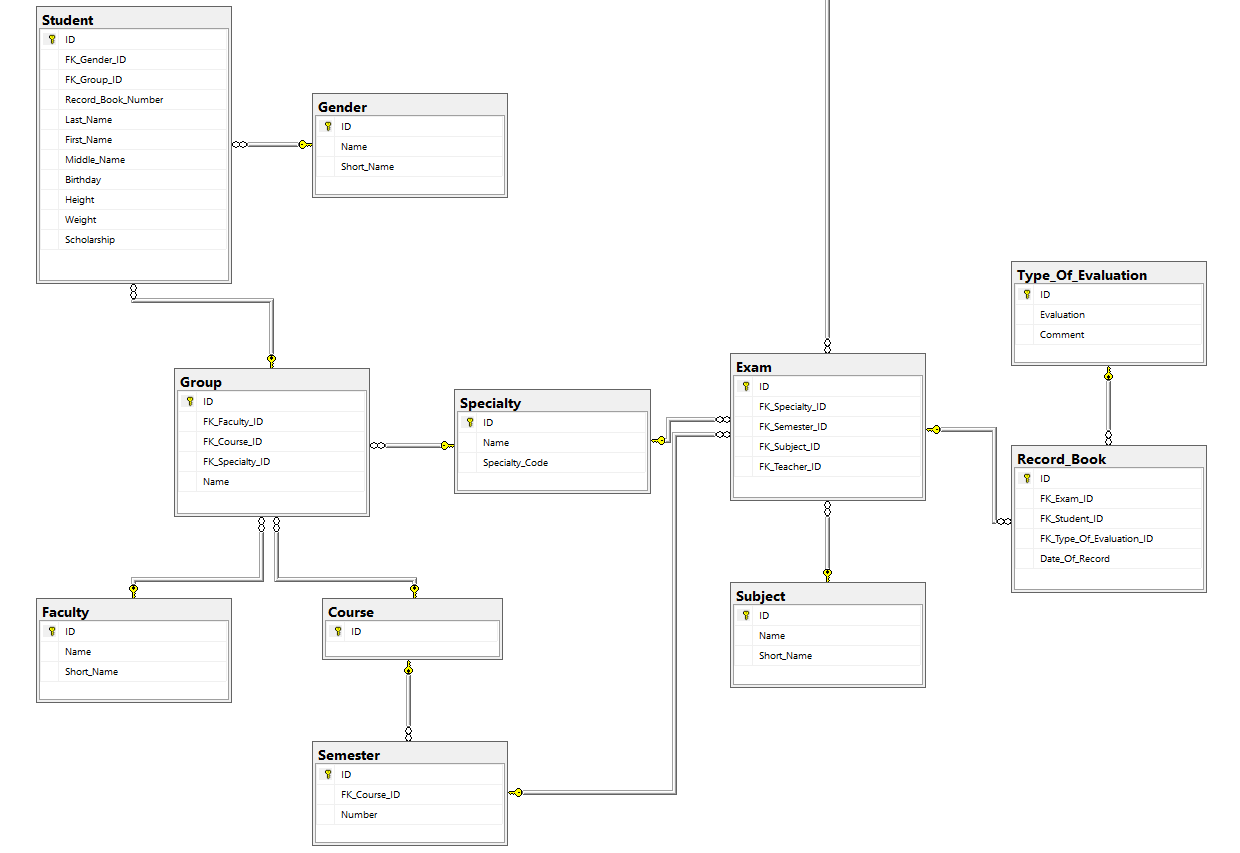


Рис. 19. Нижняя часть диаграммы баз данных

3. Привести наименование таблиц и полей к стандартному виду (исключить транслит) (ЗАПРОСАМИ) (рис. 20 и 21);

--Изменение названия таблиц с транслита на английский язык без транслита

EXEC sp\_rename 'ACDEGREE', 'Academic\_Degree'

EXEC sp\_rename 'ACKBOOK', 'Record\_Book'

EXEC sp\_rename 'DOLZNOST', 'Post'

EXEC sp\_rename 'EKZAMEN', 'Exam'

EXEC sp\_rename 'FACULTY', 'Faculty'

EXEC sp\_rename 'GRUPPA', 'Group'

EXEC sp\_rename 'KAFEDRA', 'Department'

EXEC sp\_rename 'KURS', 'Course'

EXEC sp\_rename 'OBLAST', 'Field'

EXEC sp\_rename 'POL', 'Gender'

EXEC sp\_rename 'PREPOD', 'Teacher'

EXEC sp\_rename 'SEMESTR', 'Semester'

EXEC sp\_rename 'SPECIALNOST', 'Specialty'

EXEC sp\_rename 'STUD', 'Student'

EXEC sp\_rename 'SUBJECT', 'Subject'

EXEC sp\_rename 'TYPEOCH', 'Type\_Of\_Evaluation'

EXEC sp\_rename 'UROVEN', 'Level'

--Изменение названия полей с транслита на английский язык без транслита

EXEC sp\_rename 'Academic\_Degree.ACDEGREE\_ID', 'ID', 'COLUMN'

EXEC sp\_rename 'Academic\_Degree.FK\_UROVEN\_ID', 'FK\_Level\_ID', 'COLUMN'

EXEC sp\_rename 'Academic\_Degree.FK\_OBLAST\_ID', 'FK\_Field\_ID', 'COLUMN'

EXEC sp\_rename 'Academic\_Degree.ACDEGREE', 'Name', 'COLUMN'

EXEC sp\_rename 'Academic\_Degree.ACDEGREESHORT', 'Short\_Name', 'COLUMN'

EXEC sp\_rename 'Course.KURS\_ID', 'ID', 'COLUMN'

EXEC sp\_rename 'Department.KAFEDRA\_ID', 'ID', 'COLUMN'

EXEC sp\_rename 'Department.KAFEDRA', 'Name', 'COLUMN'

EXEC sp\_rename 'Department.KAFEDRASHORT', 'Short\_Name', 'COLUMN'

EXEC sp\_rename 'Exam.EKZAMEN\_ID', 'ID', 'COLUMN'

EXEC sp\_rename 'Exam.FK\_SPECIALNOST\_ID', 'FK\_Specialty\_ID', 'COLUMN'

EXEC sp\_rename 'Exam.FK\_SEMESTR\_ID', 'FK\_Semester\_ID', 'COLUMN'

EXEC sp\_rename 'Exam.FK\_SUBJECT\_ID', 'FK\_Subject\_ID', 'COLUMN'

EXEC sp\_rename 'Exam.FK\_PREPOD\_ID', 'FK\_Teacher\_ID', 'COLUMN'

EXEC sp\_rename 'Faculty.FACULTY\_ID', 'ID', 'COLUMN'

EXEC sp\_rename 'Faculty.FACULTY', 'Name', 'COLUMN'

EXEC sp\_rename 'Faculty.FACSHORT', 'Short\_Name', 'COLUMN'

EXEC sp\_rename 'Field.OBLAST\_ID', 'ID', 'COLUMN'

EXEC sp\_rename 'Field.OBLAST', 'Name', 'COLUMN'

EXEC sp\_rename 'Gender.POL\_ID', 'ID', 'COLUMN'

EXEC sp\_rename 'Gender.POL', 'Name', 'COLUMN'

EXEC sp\_rename 'Gender.POL\_SHORT', 'Short\_Name', 'COLUMN'

EXEC sp\_rename 'Group.GRUPPA\_ID', 'ID', 'COLUMN'

EXEC sp\_rename 'Group.FK\_FACULTY\_ID', 'FK\_Faculty\_ID', 'COLUMN'

EXEC sp\_rename 'Group.FK\_KURS\_ID', 'FK\_Course\_ID', 'COLUMN'

EXEC sp\_rename 'Group.FK\_SPECIALNOST\_ID', 'FK\_Specialty\_ID', 'COLUMN'

EXEC sp\_rename 'Group.GRUPPA', 'Name', 'COLUMN'

EXEC sp\_rename 'Level.UROVEN\_ID', 'ID', 'COLUMN'

EXEC sp\_rename 'Level.UROVEN', 'Name', 'COLUMN'

EXEC sp\_rename 'Post.DOLZNOST\_ID', 'ID', 'COLUMN'

EXEC sp\_rename 'Post.DOLZNOST', 'Name', 'COLUMN'

EXEC sp\_rename 'Record\_Book.ACKBOOK\_ID', 'ID', 'COLUMN'

EXEC sp\_rename 'Record\_Book.FK\_EKZAMEN\_ID', 'FK\_Exam\_ID', 'COLUMN'

EXEC sp\_rename 'Record\_Book.FK\_STUD\_ID', 'FK\_Student\_ID', 'COLUMN'

EXEC sp\_rename 'Record\_Book.FK\_TYPEOCH\_ID', 'FK\_Type\_Of\_Evaluation\_ID', 'COLUMN'

EXEC sp\_rename 'Record\_Book.DATAREC', 'Date\_Of\_Record', 'COLUMN'

EXEC sp\_rename 'Semester.SEMESTR\_ID', 'ID', 'COLUMN'

EXEC sp\_rename 'Semester.FK\_KURS\_ID', 'FK\_Course\_ID', 'COLUMN'

EXEC sp\_rename 'Semester.SEMESTR', 'Number', 'COLUMN'

EXEC sp\_rename 'Specialty.SPECIALNOST\_ID', 'ID', 'COLUMN'

EXEC sp\_rename 'Specialty.SPECIALNOST', 'Name', 'COLUMN'

EXEC sp\_rename 'Specialty.CODOCSO', 'Specialty\_Code', 'COLUMN'

EXEC sp\_rename 'Student.STUD\_ID', 'ID', 'COLUMN'

EXEC sp\_rename 'Student.FK\_POL\_ID', 'FK\_Gender\_ID', 'COLUMN'

EXEC sp\_rename 'Student.FK\_GRUPPA\_ID', 'FK\_Group\_ID', 'COLUMN'

EXEC sp\_rename 'Student.NOZACH', 'Record\_Book\_Number', 'COLUMN'

EXEC sp\_rename 'Student.LNAME', 'Last\_Name', 'COLUMN'

EXEC sp\_rename 'Student.FNAME', 'First\_Name', 'COLUMN'

EXEC sp\_rename 'Student.MNAME', 'Middle\_Name', 'COLUMN'

EXEC sp\_rename 'Student.BIRTHDAY', 'Birthday', 'COLUMN'

EXEC sp\_rename 'Student.ROST', 'Height', 'COLUMN'

EXEC sp\_rename 'Student.VES', 'Weight', 'COLUMN'

EXEC sp\_rename 'Student.STIPENDIA', 'Scholarship', 'COLUMN'

EXEC sp\_rename 'Subject.SUBJECT\_ID', 'ID', 'COLUMN'

EXEC sp\_rename 'Subject.SUBJECT', 'Name', 'COLUMN'

EXEC sp\_rename 'Subject.SHORTSUBJECT', 'Short\_Name', 'COLUMN'

EXEC sp\_rename 'Teacher.PREPOD\_ID', 'ID', 'COLUMN'

EXEC sp\_rename 'Teacher.FK\_DOLZNOST\_ID', 'FK\_Post\_ID', 'COLUMN'

EXEC sp\_rename 'Teacher.FK\_ACDEGREE\_ID', 'FK\_Academic\_Degree\_ID', 'COLUMN'

EXEC sp\_rename 'Teacher.FK\_KAFEDRA\_ID', 'FK\_Department\_ID', 'COLUMN'

EXEC sp\_rename 'Teacher.LNAME', 'Last\_Name', 'COLUMN'

EXEC sp\_rename 'Teacher.FNAME', 'First\_Name', 'COLUMN'

EXEC sp\_rename 'Teacher.MNAME', 'Middle\_Name', 'COLUMN'

EXEC sp\_rename 'Teacher.BIRTHDAY', 'Birthday', 'COLUMN'

EXEC sp\_rename 'Type\_Of\_Evaluation.TYPEOCH\_ID', 'ID', 'COLUMN'

EXEC sp\_rename 'Type\_Of\_Evaluation.TYPEOCH', 'Evaluation', 'COLUMN'

EXEC sp\_rename 'Type\_Of\_Evaluation.COMENT', 'Comment', 'COLUMN'

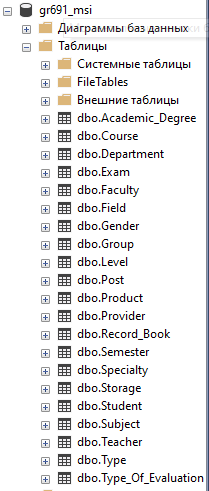


Рис. 20. Правильные названия таблиц

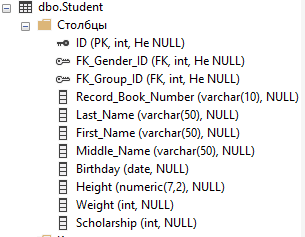


Рис. 21. Правильные названия полей

4. Выполнить запросы 1-10;

5. Отправить на проверку результат выполнения пунктов 2-5 в файле с именем Фамилия.sql.

Запросы:

ПРИМЕЧАНИЕ:

В тексте задач применяются следующие обозначения

ХХХ – номер группы

NNNN – номер зачетки

ФИО – первые буквы фамилии, имени и отчества

ДД – день в дате

ММ – месяц в дате

ГГГГ– год в дате

YY – число полных лет

БОЛЬШИЕ и маленькие буквы, подчеркивания и символы № и т.д. так же как в примере

« » обозначают, что текст который заключен в них должен выводиться пользователю как один столбец

1. Вывести данные о студентах в предоставленном формате (рис. 22):

«Фамилия И.О.», «группа№ XXX»

Код запроса:

SELECT Student.Last\_Name + ' ' + (SELECT SUBSTRING(Student.First\_Name, 1, 1)) + '.' + (SELECT SUBSTRING(Student.Middle\_Name, 1, 1)) + '.'

AS 'Фамилия И.О.', 'Группа № ' + [Group].Name AS 'Группа № XXX'

FROM [Group]

JOIN Student ON Student.FK\_Group\_ID = [Group].ID

Результат первого запроса представлен на рисунке 22:

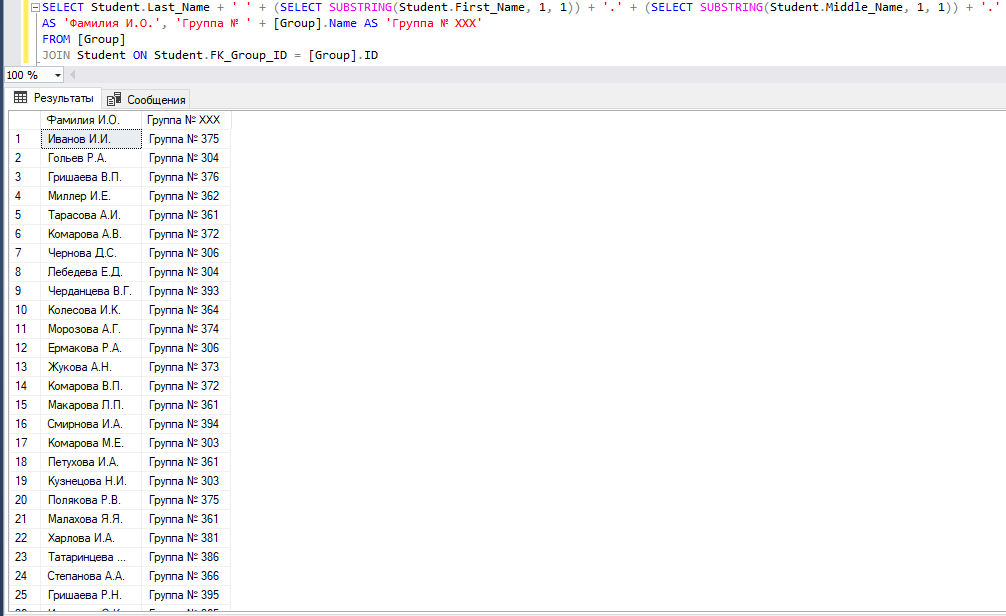


Рис. 22. Результат первого запроса

2. Выведите список студентов и их полных возраст (рис. 23):

«Фамилия (гр. XXX)», «дата рождения», «<число полных лет> лет»

Код запроса:

SELECT Student.Last\_Name + ' (гр. ' + [Group].Name + ')' AS 'Фамилия (гр. XXX)', Student.Birthday AS 'Дата рождения',

CONVERT(NVARCHAR, DATEDIFF(YEAR, Student.Birthday, GETDATE())) + ' лет' AS 'Число полных лет'

FROM Student

JOIN [Group] ON Student.FK\_Group\_ID = [Group].ID

Результат второго запроса представлен на рисунке 23:

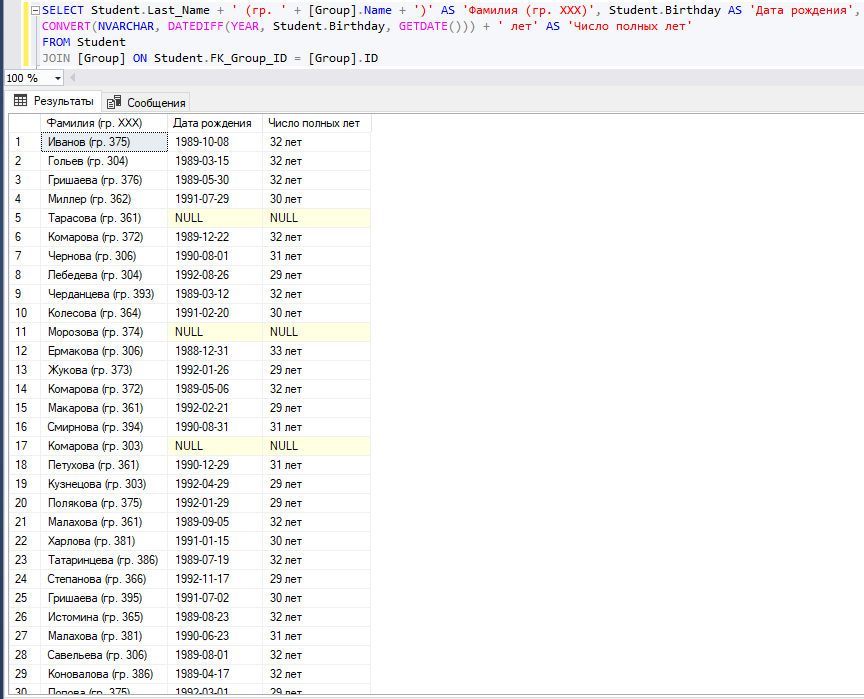


Рис. 23. Результат второго запроса

1. Выведите список уникальных фамилий студентов (рис. 24).

Код запроса:

SELECT DISTINCT Student.Last\_Name FROM Student

Результат третьего запроса представлен на рисунке 24:



Рис. 24. Результат третьего запроса

4. Выведите студентов XXX группы (рис. 25):

«ФАМИЛИЯ Имя Отчество», «XXX гр .»

Код запроса:

SELECT UPPER(Student.Last\_Name) + ' ' + Student.First\_Name + ' ' + Student.Middle\_Name

AS 'ФИО', [Group].Name + ' группа' AS 'XXX гр.'

FROM [Group]

JOIN Student ON Student.FK\_Group\_ID = [Group].ID

WHERE Student.FK\_Group\_ID = 394

Результат четвертого запроса представлен на рисунке 25:

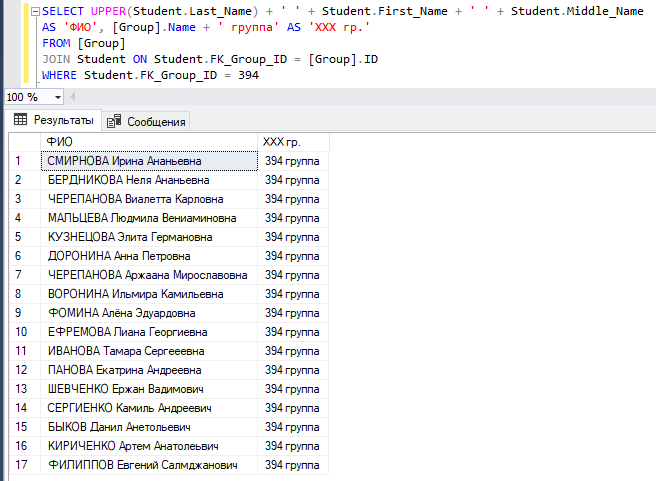


Рис. 25. Результат четвертого запроса

5. Вывести список студентов мужского пола, у которых имя и фамилия начинаются на одну букву (рис. 26)

«NNNN\_Фамилия Имя»

SELECT Student.Record\_Book\_Number + '\_' + Student.Last\_Name + ' ' + Student.First\_Name AS 'Список студентов мужского пола' FROM Student

JOIN Gender ON Gender.ID = Student.FK\_Gender\_ID

WHERE Gender.ID = 1 AND SUBSTRING(Student.Last\_Name, 1, 1) = SUBSTRING(Student.First\_Name, 1, 1)

Результат пятого запроса представлен на рисунке 26:

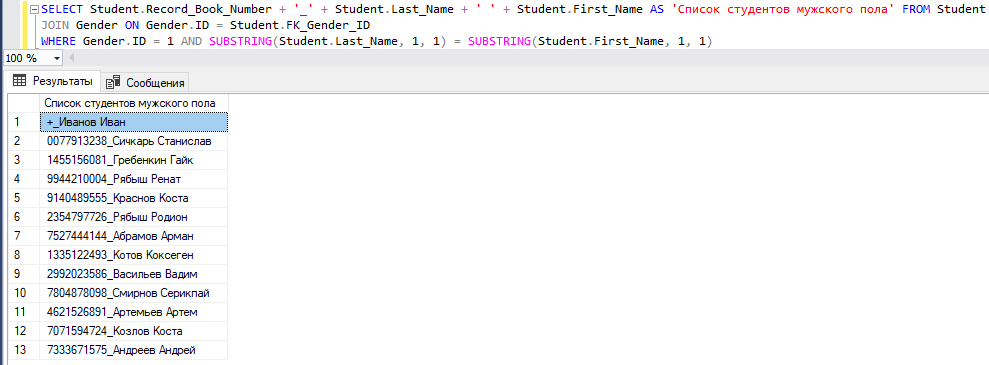


Рис. 26. Результат пятого запроса

6. Вывести информацию о студентах и их росте в формате «Фамилия И.О.», «<рост> в см.», «<категория роста>» (рис. 27)

Категория роста определяется по следующим правилам:

меньше 155 см – «низкий»

от 155 до 185 см – «средний»

больше 185 см – «высокий»

<null> – «нет данных»

Код запроса:

SELECT Student.Last\_Name + ' ' + (SELECT SUBSTRING(Student.First\_Name, 1, 1))

+ '. ' + (SELECT SUBSTRING(Student.Middle\_Name, 1, 1)) + '.' AS 'Фамилия И.О.',

CONVERT(NVARCHAR, Student.Height) + ' м.' AS 'Рост',

CASE

WHEN Student.Height < 1.55 THEN 'Низкий'

WHEN Student.Height >= 1.55 AND Student.Height <= 1.85 THEN 'Средний'

WHEN Student.Height > 1.85 THEN 'Высокий'

ELSE 'Нет данных.'

END AS 'Категория роста'

FROM Student

Результат шестого запроса представлен на рисунке 27:

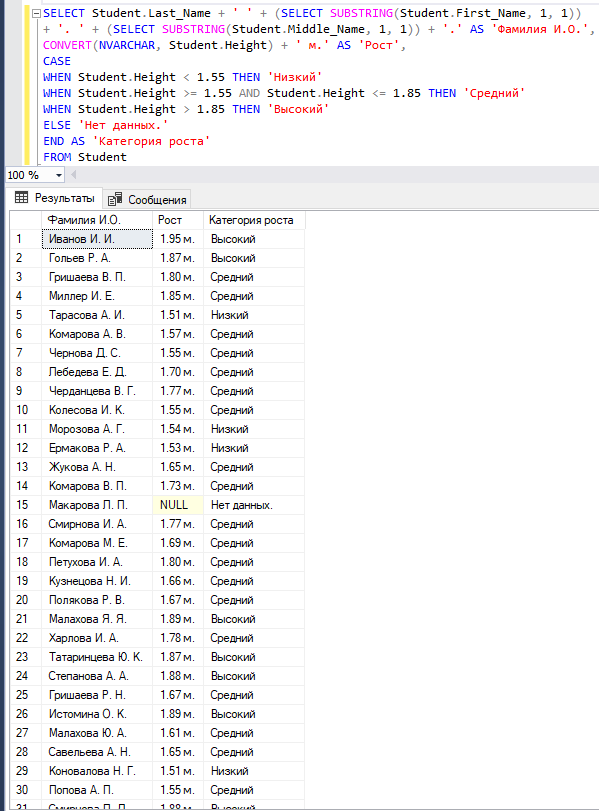


Рис. 27. Результат шестого запроса

7. Выведите количество человек и средний рост по всем студенческим группам. Назовите самую рослую группу (рис. 28).

«группа№ XXX», «<кол-во студентов>», «<средний рост>»

Код запроса:

SELECT 'Группа № ' + [Group].Name AS 'Группа № XXX',

COUNT(\*) AS 'Количество студентов',

AVG(Student.Height) AS 'Средний рост'

FROM Student

JOIN [Group]

ON Student.FK\_Group\_ID = [Group].ID

GROUP BY [Group].Name ORDER BY 'Средний рост' DESC

SELECT TOP 1 'Группа № ' + [Group].Name AS 'Группа № XXX',

COUNT(\*) AS 'Количество студентов',

AVG(Student.Height) AS 'Средний рост'

FROM Student

JOIN [Group]

ON Student.FK\_Group\_ID = [Group].ID

GROUP BY [Group].Name ORDER BY 'Средний рост' DESC

Результат седьмого запроса представлен на рисунке 28:

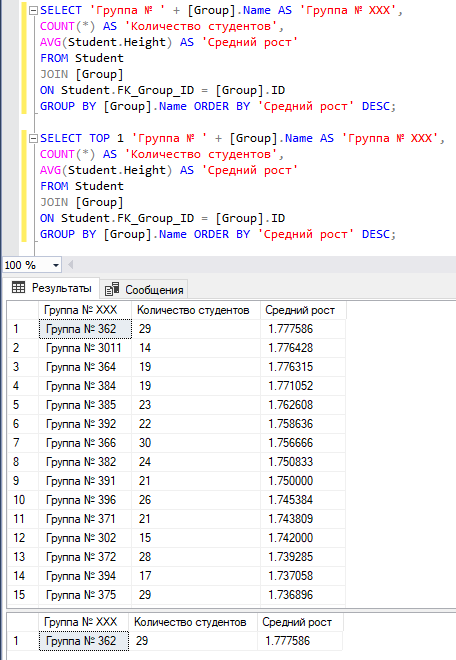


Рис. 28. Результат седьмого запроса

8. Выведите количество студентов мужского пола родившихся в тот или иной день недели, определите их средний рост и вес. Назовите самый низкорослый день недели (рис. 29).

«<день недели>», «<кол-во студентов>», «<средний рост>», «<средний вес>»

Код запроса:

SELECT DATENAME(WEEKDAY, DATEPART(WEEKDAY, Student.Birthday)-1) AS 'День недели',

COUNT(Student.ID) AS 'Количество студентов',

CAST(AVG(Student.Height) AS NVARCHAR) + ' м.' AS 'Средний рост',

AVG(Student.Weight) AS 'Средний вес'

FROM Student

WHERE Student.FK\_Gender\_ID = 1

GROUP BY DATEPART(WEEKDAY, Student.Birthday) ORDER BY DATEPART(WEEKDAY, Student.Birthday)

SELECT TOP 1 DATENAME(WEEKDAY, DATEPART(WEEKDAY, Student.Birthday)-1) AS 'День недели',

COUNT(Student.ID) AS 'Количество студентов',

CAST(AVG(Student.Height) AS NVARCHAR) + ' м.' AS 'Средний рост',

AVG(Student.Weight) AS 'Средний вес'

FROM Student

WHERE Student.FK\_Gender\_ID = 1

GROUP BY DATEPART(WEEKDAY, Student.Birthday) ORDER BY AVG(Student.Height) ASC

Результат восьмого запроса представлен на рисунке 29:

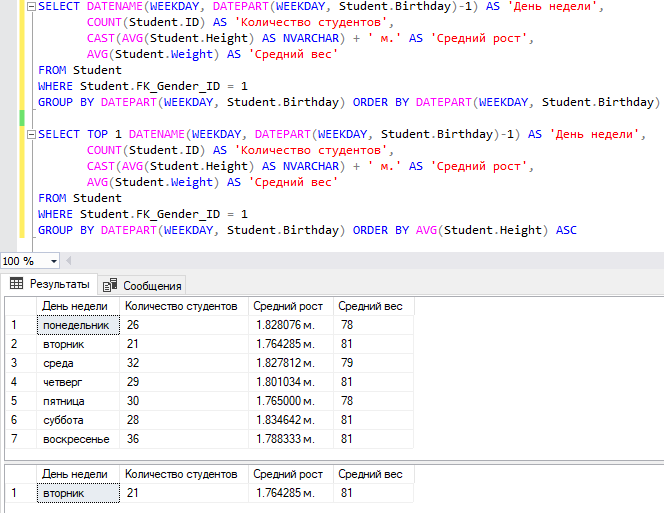


Рис. 29. Результат восьмого запроса

9. Выведите распределение студентов по полу и весу в десятках кг (рис. 30).

Код запроса:

SELECT Student.Last\_Name + ' ' + LEFT(Student.First\_Name, 1) + '.' + LEFT(Student.Middle\_Name, 1) + '.' AS 'Фамилия И.О.',

Gender.Name AS 'Пол', Student.Weight AS 'Вес' FROM Student

JOIN Gender ON Gender.ID = Student.FK\_Gender\_ID

ORDER BY Gender.Name, Student.Weight

Результат девятого запроса представлен на рисунке 30:

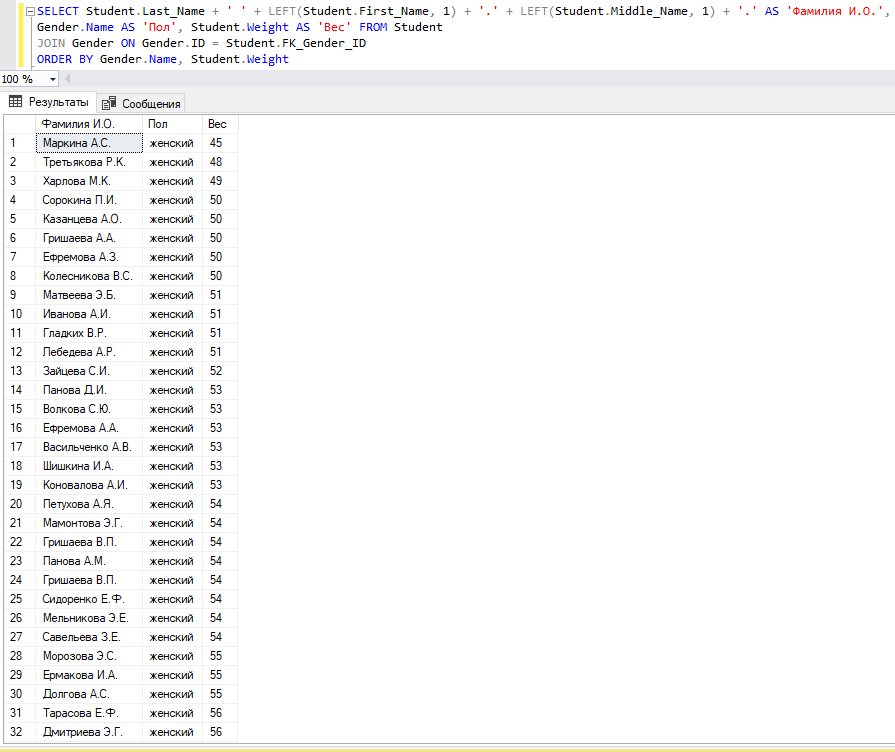


Рис. 30. Результат девятого запроса

10. Вывести список студентов 2-го курса в предоставленном формате (рис. 31).

«Фамилия И.О.»,

«<номер курса>»,

«(<код специальности>) <название специальности>»,

«<число полных лет> лет»

Код запроса:

SELECT Student.Last\_Name + ' ' + LEFT(Student.First\_Name, 1) + '.' + LEFT(Student.Middle\_Name, 1) + '.' AS 'Студент',

[Group].FK\_Course\_ID AS 'Курс', '(' + CONVERT(NVARCHAR, Specialty.ID) + ') ' + Specialty.Name AS 'Специальность',

CAST(DATEDIFF(YEAR, Student.Birthday, GETDATE()) AS NVARCHAR) + ' лет' AS 'Число полных лет'

FROM Student

JOIN [Group] ON [Group].ID = Student.FK\_Group\_ID

JOIN Specialty ON Specialty.ID = [Group].FK\_Specialty\_ID

WHERE FK\_Course\_ID = 2

Результат десятого запроса представлен на рисунке 31:

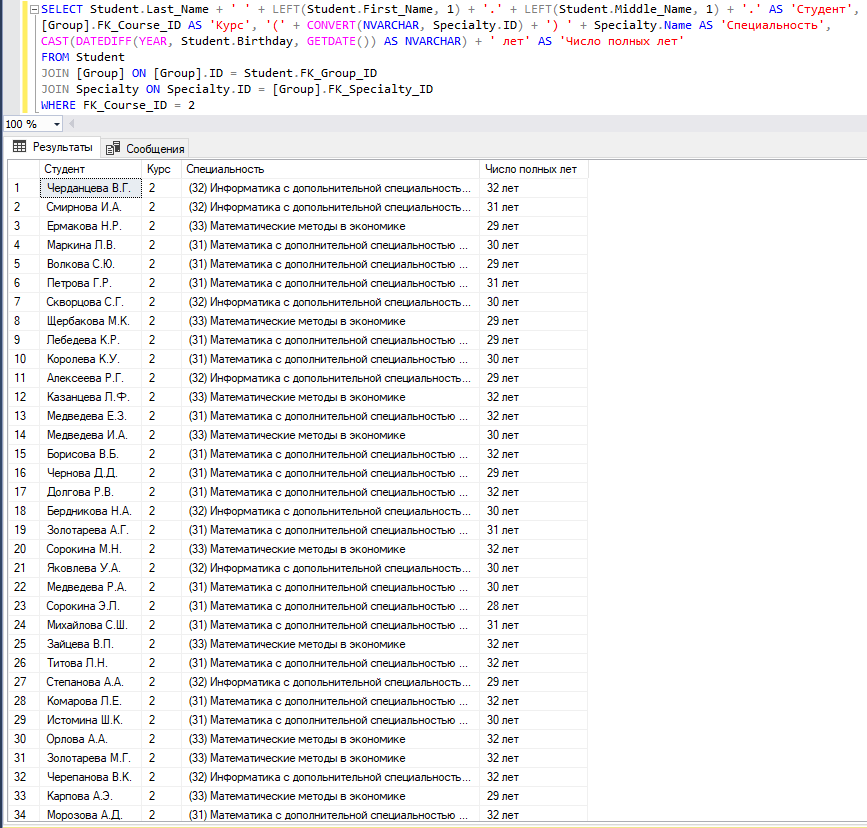


Рис. 31. Результат десятого запроса

2 неделя

Задание 1. Анализ предметной области

В соответствии с тематикой курсового проекта я провёл анализ предметной области, в соответствии со следующими пунктами:

1. Описал организацию, особенности вида деятельности, которая автоматизируется:

ИС “Электронный дневник” – это система учета успеваемости учащихся общеобразовательных учреждений, которая предоставляет информацию о предмете, оценке, домашнем задании, группе, курсе, факультете, специальности, студенте и преподавателе. ИС предоставляет возможность преподавателям выставлять оценки и задавать домашнее задание. ИС предоставляет учащимся просматривать свою группу, свои домашние задания, свои оценки, свои предметы, своих преподавателей.

Вся информация, связанная с ИС, должна храниться в базе данных.

Таблица “Предмет” хранит в себе:

* Идентификатор предмета;
* Название предмета;
* Идентификатор преподавателя.

Таблица “Оценка” хранит в себе:

* Идентификатор оценки;
* Идентификатор предмета;
* Идентификатор студента;
* Дата оценки;
* Оценка.

Таблица “Домашнее задание” хранит в себе:

* Идентификатор домашнего задания;
* Идентификатор предмета;
* Идентификатор группы;
* Дата выдачи домашнего задания;
* Описание домашнего задания;
* Статус.

Таблица “Группа” хранит в себе:

* Идентификатор группы;
* Название группы;
* Идентификатор курса;
* Идентификатор факультета;
* Идентификатор специальности;
* Идентификатор куратора (преподавателя);
* Идентификатор студента.

Таблица “Курс” хранит в себе:

* Идентификатор курса.

Таблица “Факультет” хранит в себе:

* Идентификатор факультета;
* Название факультета.

Таблица “Специальность” хранит в себе:

* Идентификатор специальности;
* Название специальности;
* Код специальности.

Таблица “Студент” хранит в себе:

* Идентификатор студента;
* Идентификатор группы;
* Фамилия;
* Имя;
* Отчество;
* Дата рождения;
* Логин;
* Пароль.

Таблица “Преподаватель” хранит в себе:

* Идентификатор преподавателя;
* Фамилия;
* Имя;
* Отчество;
* Дата рождения;
* Логин;
* Пароль.

Таблица “Предмет\_Группа” хранит в себе:

* Идентификатор предмета;
* Идентификатор группы.

1. Проанализировал возможности автоматизации данного вида деятельности:

Базу данных можно реализовать при помощи Microsoft SQL Server Management Studio. SSMS – это интегрированная среда для управления любой инфраструктурой SQL. Приложение, реализующее основные функции управления базой данных, можно сделать при помощи Microsoft Visual Studio на языке программирования C#, который имеет библиотеку Entity Framework для работы с базой данных, и WPF, который позволит создать интерфейс ИС.

1. Рассмотрел существующие варианты информационных систем, автоматизирующих данный вид деятельности (наименование, интерфейс, функциональные возможности):
2. Дневник.ру

Интерфейс выглядит простым и удобным.

Функциональные возможности:

* Предоставляет возможность просматривать свою группу;
* Предоставляет возможность проставлять оценки;
* Предоставляет возможность оставлять домашнее задание;
* Предоставляет возможность просматривать преподавателей;
* Предоставляет возможность просматривать текущие и итоговые оценки.

1. ЭлЖур

Интерфейс подходит новым преподавателям и преподавателям со стажем, потому что выглядит как обычный физический журнал.

Функциональные возможности:

* Предоставляет возможность составлять отчёт по успеваемости;
* Предоставляет возможность оставлять сообщение всему классу / группе;
* Предоставляет возможность оставлять новость на главной странице журнала;
* Предоставляет возможность оставить сообщение учителю;
* Предоставляет возможность описать причину отсутствия.

1. Определить основные технические и функциональные требования к разрабатываемой системе.
2. Добавление, изменение и удаление преподавателя;
3. Добавление, изменение и удаление студента;
4. Добавление и удаление группы;
5. Добавление, изменение и удаление оценки;
6. Добавление и удаление курса;
7. Добавление и удаление факультета;
8. Добавление и удаление специальности;
9. Добавление, изменение и удаление домашнего задания;
10. Добавление и удаление предмета.

Задание 2. Проектирование структуры баз данных.

Построение логической и физической модели данных.

Для концептуальной модели, определенной в предметной области индивидуального задания №1.

1. Построить логическую модель базы данных. Сохранить результат в файле Логическая модель.pdf;

2. Построить физическую модель базы данных. Сохранить результат в файле Физическая\_модель.pdf;

3. Подготовить словарь данных. Сохранить результат в файле Словарь данных.docx.

Результат работы:

1. Построил логическую модель базы данных (рис. 1):

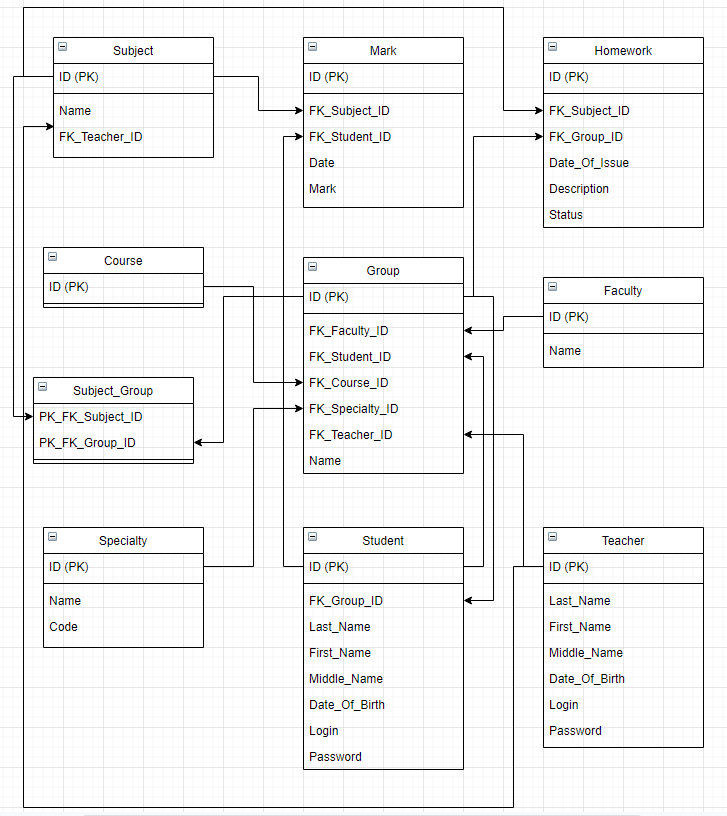


Рис. 1. Логическая модель базы данных

1. Построил физическую модель базы данных (рис. 2):

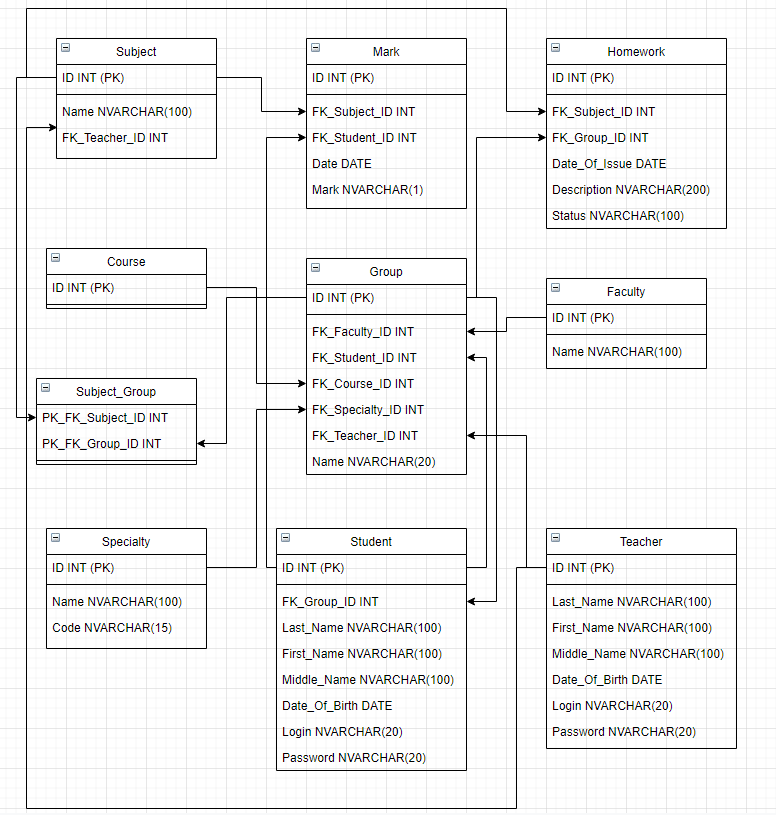


Рис. 2. Физическая модель базы данных

1. Подготовил словарь данных (таблицы приведены ниже).

Таблица 1 – Студент

|  |  |  |  |
| --- | --- | --- | --- |
| № п-п | Названия поля | Тип данных | Описание |
| 1 | ID | INT (PK) | Идентификатор студента |
| 2 | FK\_Group\_ID | INT | Внешний ключ группы |
| 3 | Last\_Name | NVARCHAR(100) | Фамилия студента |
| 4 | First\_Name | NVARCHAR(100) | Имя студента |
| 5 | Middle\_Name | NVARCHAR(100) | Отчество студента |
| 6 | Date\_Of\_Birth | DATE | Дата рождения студента |
| 7 | Login | NVARCHAR(20) | Логин студента |
| 8 | Password | NVARCHAR(20) | Пароль студента |

Таблица 2 – Преподаватель

|  |  |  |  |
| --- | --- | --- | --- |
| № п-п | Названия поля | Тип данных | Описание |
| 1 | ID | INT (PK) | Идентификатор преподавателя |
| 2 | Last\_Name | NVARCHAR(100) | Фамилия преподавателя |
| 3 | First\_Name | NVARCHAR(100) | Имя преподавателя |
| 4 | Middle\_Name | NVARCHAR(100) | Отчество преподавателя |
| 5 | Date\_Of\_Birth | DATE | Дата рождения преподавателя |
| 6 | Login | NVARCHAR(20) | Логин преподавателя |
| 7 | Password | NVARCHAR(20) | Пароль преподавателя |

Таблица 3 – Специальность

|  |  |  |  |
| --- | --- | --- | --- |
| № п-п | Названия поля | Тип данных | Описание |
| 1 | ID | INT (PK) | Идентификатор специальности |
| 2 | Name | NVARCHAR(100) | Название специальности |
| 3 | Code | NVARCHAR(15) | Код специальности |

Таблица 4 – Группа

|  |  |  |  |
| --- | --- | --- | --- |
| № п-п | Названия поля | Тип данных | Описание |
| 1 | ID | INT (PK) | Идентификатор группы |
| 2 | FK\_Faculty\_ID | INT | Внешний ключ факультета |
| 3 | FK\_Student\_ID | INT | Внешний ключ студента |
| 4 | FK\_Course\_ID | INT | Внешний ключ курса |
| 5 | FK\_Specialty\_ID | INT | Внешний ключ специальности |
| 6 | FK\_Teacher\_ID | INT | Внешний ключ преподавателя |
| 7 | Name | NVARCHAR(20) | Название группы |
| 8 | FK\_Subject\_Group | INT | Внешний ключ составного ключа |

Таблица 5 – Факультет

|  |  |  |  |
| --- | --- | --- | --- |
| № п-п | Названия поля | Тип данных | Описание |
| 1 | ID | INT (PK) | Идентификатор факультет |
| 2 | Name | NVARCHAR(100) | Название факультета |

Таблица 6 – Курс

|  |  |  |  |
| --- | --- | --- | --- |
| № п-п | Названия поля | Тип данных | Описание |
| 1 | ID | INT (PK) | Идентификатор курса |

Таблица 7 – Предмет

|  |  |  |  |
| --- | --- | --- | --- |
| № п-п | Названия поля | Тип данных | Описание |
| 1 | ID | INT (PK) | Идентификатор предмета |
| 2 | Name | NVARCHAR(100) | Название предмета |
| 3 | FK\_Teacher\_ID | INT | Внешний ключ преподавателя |

Таблица 8 – Оценка

|  |  |  |  |
| --- | --- | --- | --- |
| № п-п | Названия поля | Тип данных | Описание |
| 1 | ID | INT (PK) | Идентификатор оценки |
| 2 | FK\_Subject\_ID | INT | Внешний ключ предмета |
| 3 | FK\_Student\_ID | INT | Внешний ключ студента |
| 4 | Date | DATE | Дата оценки |
| 5 | Mark | NVARCHAR(1) | Оценка |

Таблица 9 – Домашнее задание

|  |  |  |  |
| --- | --- | --- | --- |
| № п-п | Названия поля | Тип данных | Описание |
| 1 | ID | INT (PK) | Идентификатор домашнее задание |
| 2 | FK\_Subject\_ID | INT | Внешний ключ предмета |
| 3 | FK\_Group\_ID | INT | Внешний ключ группа |
| 4 | Date\_Of\_Issue | DATE | Дата выдачи |
| 5 | Description | NVARCHAR(200) | Описание |
| 6 | Status | NVARCHAR(100) | Статус задания |

Таблица 10 – Предметы группы

|  |  |  |  |
| --- | --- | --- | --- |
| № п-п | Названия поля | Тип данных | Описание |
| 1 | PK\_FK\_Subject\_ID | INT (PK) (FK) | Идентификатор предмета |
| 2 | PK\_FK\_Group\_ID | INT (PK) (FK) | Идентификатор группы |

Задание 3. Создание объектов баз данных в СУБД

* Создать скрипт, позволяющий реализовать нормализованную схему базы данных из задания 2 (2 неделя);

Код запроса “Structure.sql”:

CREATE TABLE Student

(

ID INT PRIMARY KEY IDENTITY(1,1),

FK\_Group\_ID INT,

Last\_Name NVARCHAR(100),

First\_Name NVARCHAR(100),

Middle\_Name NVARCHAR(100),

Date\_Of\_Birth DATE,

Login NVARCHAR(20),

Password NVARCHAR(20)

)

CREATE TABLE Teacher

(

ID INT PRIMARY KEY IDENTITY(1,1),

Last\_Name NVARCHAR(100),

First\_Name NVARCHAR(100),

Middle\_Name NVARCHAR(100),

Date\_Of\_Birth DATE,

Login NVARCHAR(20),

Password NVARCHAR(20)

)

CREATE TABLE Specialty

(

ID INT PRIMARY KEY IDENTITY(1,1),

Name NVARCHAR(100),

Code NVARCHAR(15)

)

CREATE TABLE [Group]

(

ID INT PRIMARY KEY IDENTITY(1,1),

FK\_Faculty\_ID INT,

FK\_Student\_ID INT,

FK\_Course\_ID INT,

FK\_Specialty\_ID INT,

FK\_Teacher\_ID INT,

Name NVARCHAR(20)

)

CREATE TABLE Faculty

(

ID INT PRIMARY KEY IDENTITY(1,1),

Name NVARCHAR(100)

)

CREATE TABLE Course

(

ID INT PRIMARY KEY

)

CREATE TABLE Subject

(

ID INT PRIMARY KEY IDENTITY(1,1),

Name NVARCHAR(100),

FK\_Teacher\_ID INT

)

CREATE TABLE Mark

(

ID INT PRIMARY KEY IDENTITY(1,1),

FK\_Subject\_ID INT,

FK\_Student\_ID INT,

Date DATE,

Mark NVARCHAR(1)

)

CREATE TABLE Homework

(

ID INT PRIMARY KEY IDENTITY(1,1),

FK\_Subject\_ID INT,

FK\_Group\_ID INT,

Date\_Of\_Issue DATE,

Description NVARCHAR(200),

Status NVARCHAR(100)

)

CREATE TABLE Subject\_Group

(

PK\_FK\_Subject\_ID INT,

PK\_FK\_Group\_ID INT

PRIMARY KEY (PK\_FK\_Subject\_ID, PK\_FK\_Group\_ID)

)

ALTER TABLE Student ADD FOREIGN KEY (FK\_Group\_ID) REFERENCES [Group] (ID)

ALTER TABLE Subject ADD FOREIGN KEY (FK\_Teacher\_ID) REFERENCES Teacher (ID)

ALTER TABLE Subject\_Group ADD FOREIGN KEY (PK\_FK\_Subject\_ID) REFERENCES Subject (ID)

ALTER TABLE Subject\_Group ADD FOREIGN KEY (PK\_FK\_Group\_ID) REFERENCES [Group] (ID)

ALTER TABLE Mark ADD FOREIGN KEY (FK\_Subject\_ID) REFERENCES Subject (ID)

ALTER TABLE Mark ADD FOREIGN KEY (FK\_Student\_ID) REFERENCES Student (ID)

ALTER TABLE Homework ADD FOREIGN KEY (FK\_Subject\_ID) REFERENCES Subject (ID)

ALTER TABLE Homework ADD FOREIGN KEY (FK\_Group\_ID) REFERENCES [Group] (ID)

ALTER TABLE [Group] ADD FOREIGN KEY (FK\_Faculty\_ID) REFERENCES Faculty (ID)

ALTER TABLE [Group] ADD FOREIGN KEY (FK\_Student\_ID) REFERENCES Student (ID)

ALTER TABLE [Group] ADD FOREIGN KEY (FK\_Course\_ID) REFERENCES Course (ID)

ALTER TABLE [Group] ADD FOREIGN KEY (FK\_Specialty\_ID) REFERENCES Specialty (ID)

ALTER TABLE [Group] ADD FOREIGN KEY (FK\_Teacher\_ID) REFERENCES Teacher (ID)

* Наполнить созданные реляции тестовыми данными (не менее 20 записей в основной таблице и не менее 10 в остальных).

Код запроса “Insert.sql”:

INSERT INTO Course (ID) VALUES (1),

(2),

(3),

(4),

(5),

(6)

INSERT INTO Teacher VALUES ('Данькина', 'Алена', 'Анатольевна', '1980-11-27', 'Dankina', 'Alena'),

('Степанов', 'Тимофей', 'Романович', '1963-03-03', 'Stepanof', '228'),

('Титова', 'Яна', 'Нестеровна', '1970-11-30', 'TitovaYana', '19701130'),

('Ермакова', 'Дарина', 'Олеговна', '1974-05-30', 'Ermak', 'GETOVERHERE'),

('Чернышова', 'Людмила', 'Артуровна', '1979-11-24', 'Chernishova', 'HeroIsMyth'),

('Бочаров', 'Тихон', 'Владимирович', '1964-11-11', 'Bocharov', 'ILoveYou'),

('Смирнов', 'Ярослав', 'Дмитриевич', '1974-12-15', 'Smirnov', 'YarikBlin'),

('Кузнецов', 'Роман', 'Азимович', '1973-07-21', 'Kuznetsov', 'Romanchek1337'),

('Кириченко', 'Эдуард', 'Омариевич', '1950-01-06', 'Kirichenko', 'EdikMedik'),

('Одинцова', 'Любовь', 'Алексеевна', '1980-11-27', 'Odintsova', 'Kekekekech')

INSERT INTO SUBJECT VALUES ('Информационные системы', 1),

('Программирование', 2),

('Математический анализ', 3),

('Информатика', 4),

('Аналитическая геометрия', 5),

('Отечественная история', 6),

('Русский язык', 7),

('Химия', 8),

('Алгебра', 9),

('Геометрия', 10)

INSERT INTO Faculty VALUES ('Лингвистический институт'),

('Филологический факультет'),

('Факультет физической культуры'),

('Исторический факультет'),

('Институт психологии и педагогики'),

('Институт физ-мат образования'),

('Юридический институт'),

('Механико-математический факультет'),

('Факультет инновационных технологий'),

('Физико-технический факультет')

INSERT INTO Specialty VALUES ('Математика с информатикой', '111-222-333333'),

('Математические методы в экономике', '222-333-444444'),

('Химия', '333-444-555666'),

('Русский язык и литература', '456-789-000000'),

('Прикладная математика', '123-456-789123'),

('Педагогика и психология', '228-322-133728'),

('Педагогика и методика дошкольного образования', '322-228-148828'),

('Народное художественное творчество', '133-722-814883'),

('Физика с доп. спец. информатика', '548-547-345985'),

('Социально-культурный сервис и туризм', '475-584-278465')

INSERT INTO Student VALUES ('1', 'Кузнецова', 'Нинель', 'Игнатьевна', '1992-04-29', 'Kuznechik', 'Nina0'),

('1', 'Гладких', 'Лидия', 'Эльмаровна', '1991-08-20', 'Gladkaya', 'Lidiya'),

('2', 'Хохлова', 'Серафима', 'Илдаровна', '1992-04-27', 'Hohol', 'LYALYA'),

('2', 'Черданцева', 'Мальвина', 'Зотеевна', '1991-02-04', 'Cherdak', 'Malvina'),

('3', 'Савченко', 'Эмма', 'Валентиновна', '1991-12-25', 'Savchenko', 'Emmo4ka'),

('3', 'Степанова', 'Валерия', 'Прокопьевна', '1991-07-23', 'Stepanova', 'Valeriya'),

('4', 'Белкина', 'Эвелина', 'Геннадьевна', '1991-12-07', 'Belka', 'Eveline'),

('4', 'Морозова', 'Карина', 'Сергеевна', '1989-11-21', 'Morozova', 'Karina'),

('5', 'Кудашова', 'Ефросинья', 'Святославовна', '1989-03-19', 'Kuda', 'Tuda'),

('5', 'Колесникова', 'Эгине', 'Мисирхановна', '1991-09-08', 'Koleso', 'Egine'),

('6', 'Чернова', 'Варвара', 'Евгеньевна', '1992-06-30', 'Black', 'Warwara'),

('6', 'Ушаков', 'Альберт', 'Олегович', '1989-05-21', 'Ushakov', 'Allo'),

('7', 'Шевченко', 'Руслан', 'Акылбекович', '1989-01-03', 'Shevchenko', 'Rusich'),

('7', 'Фомин', 'Станислав', 'Львович', '1992-07-02', 'Fomka', 'Stasik'),

('8', 'Кириченко', 'Роман', 'Атомшаевич', '1989-01-27', 'Kirill', 'Roman'),

('8', 'Площаднов', 'Егор', 'Текпаевич', '1989-01-17', 'Plochadka', 'Egorka'),

('9', 'Злобин', 'Пётр', 'Леонидович', '1992-07-22', 'Zlobin', 'Shutka'),

('9', 'Щербинин', 'Алексей', 'Георгиевич', '1992-02-14', 'Scherbinin', 'LEHA'),

('10', 'Коваленко', 'Илья', 'Рафаэльевич', '1990-12-31', 'Kovalski', 'Ilya'),

('10', 'Иванов', 'Арсений', 'Анатольевич', '1989-07-20', 'Ivanov', 'Ivan')

INSERT INTO Mark VALUES ('3', '1', '14-01-2021', '5'),

('2', '2', '25-03-2020', '4'),

('1', '3', '05-05-2019', '3'),

('4', '4', '10-07-2018', '2'),

('5', '7', '20-09-2017', 'Н'),

('7', '9', '27-02-2016', 'З'),

('9', '15', '15-06-2022', '5'),

('10', '19', '23-02-2015', '2'),

('6', '20', '20-03-2014', '3'),

('8', '10', '10-10-2010', 'Н')

INSERT INTO [Group] VALUES ('1', '10', '1', '1', '10', '602'),

('3', '19', '2', '3', '2', '603'),

('4', '9', '2', '3', '4', '605'),

('2', '6', '4', '5', '6', '607'),

('7', '3', '4', '4', '8', '606'),

('6', '4', '5', '2', '1', '604'),

('9', '15', '5', '7', '3', '608'),

('5', '17', '6', '7', '5', '609'),

('10', '13', '6', '9', '7', '601'),

('8', '12', '6', '8', '8', '600')

INSERT INTO Homework VALUES ('3', '2', '25-02-2021', 'Прочитать стр. 45-60', 'Выдано'),

('5', '1', '01-01-2020', 'Выполнить гимнастику', 'Не выдано'),

('7', '5', '03-05-2021', 'Нет задания', 'Выдано'),

('9', '7', '07-07-2019', 'Выучить песню', 'Выдано'),

('10', '10', '10-09-2020', 'Подготовиться к сам. работе', 'Выдано'),

('2', '9', '12-12-2018', 'Отдохнуть', 'Выдано'),

('4', '9', '10-10-2017', 'Рассказать стих', 'Не выдано'),

('6', '3', '06-08-2020', 'Летние каникулы', 'Выдано'),

('8', '6', '20-05-2020', 'ОГЭ', 'Выдано'),

('1', '8', '15-03-2017', 'лол каво', 'Тест')

INSERT INTO Subject\_Group VALUES (1, 2),

(1, 3),

(2, 4),

(5, 10),

(6, 5),

(7, 6),

(8, 9),

(1, 10),

(3, 1),

(6, 6)

Задание 4. Обработка данных с использование языка SQL

Для созданной базы данных в рамках курсовой работы определить перечень основных запросов, отражающих потребности пользователей системы в получении информации (НЕ менее 15).

Запросы:

1. Вывести список студентов (результат на рис. 3).

Код запроса:

SELECT Last\_Name AS 'Фамилия', First\_Name AS 'Имя', Middle\_Name AS 'Отчество' FROM Student



Рис. 3. Результат первого запроса

1. Вывести список преподавателей, которые ведут программирование (результат на рис. 4).

Код запроса:

SELECT Teacher.Last\_Name AS 'Фамилия', Subject.Name FROM Teacher

JOIN Subject ON Subject.FK\_Teacher\_ID = Teacher.ID

WHERE Subject.Name = 'Программирование'

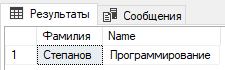


Рис. 4. Результат второго запроса.

1. Вывести список выданных домашних заданий (результат на рис. 5).

Код запроса:

SELECT Homework.Description FROM Homework

WHERE Homework.Status = 'Выдано'

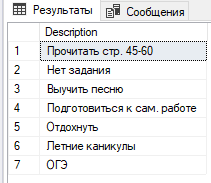


Рис. 5. Результат третьего запроса

1. Вывести всех студентов первого курса (результат на рис. 6).

Код запроса:

SELECT Student.Last\_Name AS 'Фамилия',

Student.First\_Name AS 'Имя',

Student.Middle\_Name AS 'Отчество',

FK\_Course\_ID AS 'Курс'

FROM Student

JOIN [Group] ON FK\_Student\_ID = Student.ID

WHERE FK\_Course\_ID = '1'

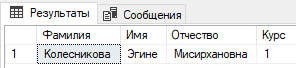


Рис. 6. Результат четвертого запроса

1. Вывести всех студентов филологического факультета, которые находятся на четвертом курсе (результат на рис. 7).

Код запроса:

SELECT Student.Last\_Name AS 'Фамилия',

Student.First\_Name AS 'Имя',

Student.Middle\_Name AS 'Отчество',

Faculty.Name AS 'Факультет',

FK\_Course\_ID AS 'Курс'

FROM Student

JOIN [Group] ON FK\_Student\_ID = Student.ID

JOIN Faculty ON Faculty.ID = FK\_Faculty\_ID

WHERE Faculty.Name = 'Филологический факультет' AND FK\_Course\_ID = '4'

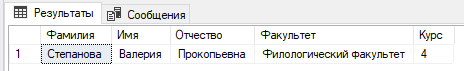


Рис. 7. Результат пятого запроса

1. Вывести всех студентов, кому больше 30 лет (результат на рис. 8).

Код запроса:

SELECT Student.Last\_Name AS 'Фамилия',

Student.First\_Name AS 'Имя',

Student.Middle\_Name AS 'Отчество',

DATEDIFF(YEAR, Student.Date\_Of\_Birth, GETDATE()) AS 'Количество полных лет'

FROM Student

WHERE DATEDIFF(YEAR, Student.Date\_Of\_Birth, GETDATE()) > 30

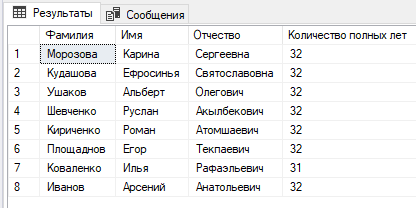


Рис. 8. Результат шестого запроса

1. Вывести студентов 602, 603 и 609 группы (результат на рис. 9).

Код запроса:

SELECT Student.Last\_Name AS 'Фамилия',

Student.First\_Name AS 'Имя',

Student.Middle\_Name AS 'Отчество',

[Group].Name AS 'Группа'

FROM Student

JOIN [Group] ON FK\_Student\_ID = Student.ID

WHERE [Group].Name = '602' OR [Group].Name = '603' OR [Group].Name = '609'

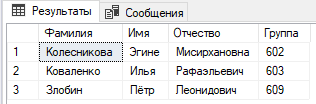


Рис. 9. Результат седьмого запроса

1. Вывести группы и предметы, которые они изучают (результат на рис. 10).

Код запроса:

SELECT [Group].Name, Subject.Name

FROM [Group]

JOIN Subject\_Group ON Subject\_Group.PK\_FK\_Group\_ID = [Group].ID

JOIN Subject ON Subject\_Group.PK\_FK\_Subject\_ID = Subject.ID



Рис. 10. Результат восьмого запроса

1. Вывести самого успевающего студента (результат на рис. 11).

Код запроса:

SELECT TOP 1 Student.Last\_Name AS 'Фамилия',

AVG(CAST(Mark.Mark AS INT)) AS 'Средняя оценка'

FROM Student

JOIN Mark

ON Mark.FK\_Student\_ID = Student.ID

GROUP BY Student.Last\_Name ORDER BY 'Средняя оценка' DESC

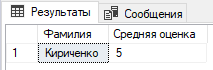


Рис. 11. Результат девятого запроса

1. Вывести ФИО преподавателя и количество предметов, которые он преподаёт (результат на рис. 12).

Код запроса:

SELECT Teacher.Last\_Name AS 'Фамилия',

Teacher.First\_Name AS 'Имя',

Teacher.Middle\_Name AS 'Отчество',

COUNT(FK\_TEACHER\_ID) AS 'Количество предметов'

FROM Teacher

JOIN Subject ON Teacher.ID = Subject.FK\_Teacher\_ID

GROUP BY Teacher.Last\_Name, Teacher.First\_Name, Teacher.Middle\_Name

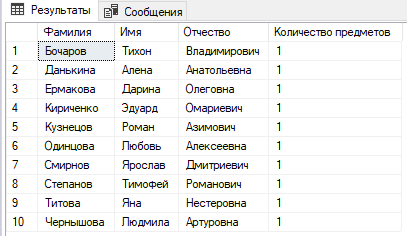


Рис. 12. Результат десятого запроса

1. Вывести студентов по курсам (результат на рис. 13).

Код запроса:

SELECT Student.Last\_Name AS 'Фамилия',

Student.First\_Name AS 'Имя',

Student.Middle\_Name AS 'Отчество',

[Group].FK\_Course\_ID AS 'Курс'

FROM Student

JOIN [Group] ON [Group].FK\_Student\_ID = Student.ID

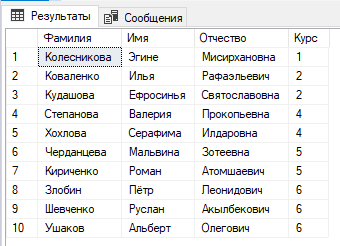


Рис. 13. Результат одиннадцатого запроса

1. Вывести специальности и предметы, которые изучаются на этой специальности (результат на рис. 14).

Код запроса:

SELECT Specialty.Name AS 'Название специальности', Subject.NAME AS 'Название предмета'

FROM Specialty

JOIN [Group] ON Specialty.ID = [Group].FK\_Specialty\_ID

JOIN Subject\_Group ON Subject\_Group.PK\_FK\_Group\_ID = [Group].ID

JOIN Subject ON Subject.ID = Subject\_Group.PK\_FK\_Subject\_ID

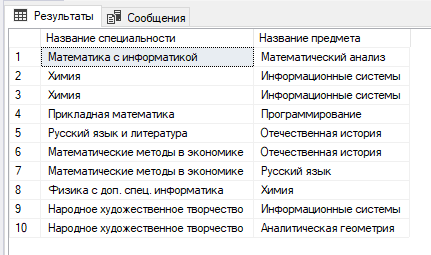


Рис. 14. Результат двенадцатого запроса

1. Вывести оценки по химии за март 2020 года группы 602 (результат на рис. 15).

Код запроса:

SELECT [Group].Name, Student.Last\_Name, Subject.Name, Mark.Date

FROM Student

JOIN [Group] ON Student.FK\_Group\_ID = [Group].ID

JOIN Mark ON Mark.FK\_Student\_ID = Student.ID

JOIN Subject ON Subject.ID = Mark.FK\_Subject\_ID

WHERE Subject.Name = 'Программирование' AND [Group].Name= '602' AND MONTH(MARK.DATE) = 3 AND YEAR(Mark.Date) = 2020

ORDER BY FK\_GROUP\_ID

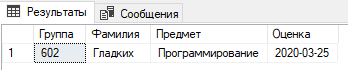


Рис. 15. Результат тринадцатого запроса

1. Вывести самого неуспевающего студента (результат на рис. 16)

Код запроса:

SELECT TOP 1 Student.Last\_Name AS 'Фамилия',

AVG(CAST(Mark.Mark AS INT)) AS 'Средняя оценка'

FROM Student

JOIN Mark

ON Mark.FK\_Student\_ID = Student.ID

GROUP BY Student.Last\_Name ORDER BY 'Средняя оценка' ASC

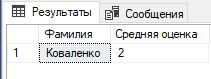


Рис. 16. Результат четырнадцатого запроса

1. Вывести все домашние задания, которые не имеют статус “Выдано” (результат на рис. 17)

Код запроса:

SELECT Homework.Description FROM Homework

WHERE Homework.Status != 'Выдано'

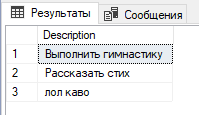


Рис. 17. Результат пятнадцатого запроса