*ТОГАОУ «Мичуринский лицей»*

Проект на тему

«Разработка базы данных общеобразовательного учреждения»

Выполнил: ученик 10 Б класса

Мордасов Дмитрий

Руководитель: учитель информатики

Десятник А.А.

*г. Мичуринск-наукоград, 2021-2022*

Содержание

1.Теория……………………………………………………...……………. 3

1.1 Определение…………………………………………………………... 3

1.2 Разновидности………………………………………………………… 3

1.3 Составляющие части…………………………………………………. 3

2. Введение……………………………………………………………....... 4

2.4 Актуальность…………………………………………………………. 4

2.5 Проблематика………………………………………………………… 4

2.6 Цель……………………………………………………………………. 4

2.7 План реализации……………………………………………………… 4

3. Проектирование базы данных (1часть) ………………………………. 5

3.1 Описание предметной области………………………………………. 5

3.2 Концептуальная модель предметной области……………………… 6

3.3 Даталогическая модель базы данных……………………………….. 11

4. Проектирование базы данных (2часть) ……………………………… 14

4.1 Сценарий создания объектов базы данных………………………… 14

4.2 Заполнение базы данных……………………………………………. 15

4.3 Проверка запросами…………………………………………………. 20

5. Итог …………………………………………………….……………. .. 25

**Теория**

Определение

База данных – совокупность определенным образом организованной информации на какую-то тему (в рамках некоторой предметной области). База данных позволят работать с большими объёмами материалов, упорядочивать их и находить при необходимости. В мире они используются постоянно. Практически во все области проникли БД, например: любые приложения на телефонах, компьютерах или любых других электронных устройствах, WEB-технологии, управление крупными компаниями (бухгалтерия, кадровый состав, количество товаров/услуг), в сферу общеобразовательных учреждений.

Разновидности баз данных

* Иерархические
* Объектно-ориентированные
* Реляционные
* Сетевые
* Функциональные

Для своего проекта я выбрал реляционную базу данных, созданную на основе общеобразовательного учреждения.

Составляющие части базы данных

Таблица – объект, предназначенный для хранения данных в виде записей (строк) и полей (столбцов). Таблицы могут быть связаны между собой. Таблица – базовый объект БД, все остальные объекты создаются на основе существующих таблиц.

Запрос – объект, позволяющий получить необходимые данные из одной или нескольких таблиц. С помощью запроса можно отбирать записи или поля, удовлетворяющие критериям отбора, можно вводить изменения в таблицы, можно производить вычисления. Фактически запросы являются важнейшим инструментом БД.

Форма – объект, предназначенный для отображения или ввода данных в таблицы. Также форма является удобным средством для поиска и коррекции информации в таблицах. Часто форма представляет собой бланк, выводящий содержимое одной записи таблицы. Фактически с помощью формы создается графический интерфейс доступа к базе данных.

**Введение**

Мы живем в XXI веке, когда информация становится основным ресурсом, определяющим эффективность деятельности тех, кто ей обладает. Даже Интернет – это всего лишь коммуникационная система, ценность которой, в конечном счете, определяется информацией и данными, передаваемыми от базы данных к пользователю и обратно.

По мере накопления человечеством информации возрастает и значимость баз данных как основного ее хранилища. Это относится и к персональным базам данных (БД) на настольных компьютерах, и к большим распределенным базам, хранящимся на разбросанных по всему миру серверах. Как я уже говорил, сегодня базы данных используются в науке, образовании, торговле, маркетинге, производстве, финансах, бухгалтерском учете, менеджменте и других сферах деятельности.

Актуальность

Из-за частого изменения информации необходимо её постоянно изменять, добавлять или убирать. База данных позволяет максимально автоматизировать эти процессы.

Проблема

Без базы данных сложно обрабатывать всё-то количество информации, что мы имеем.

Цель

Создать базу данных на основе общеобразовательного учреждения ТОГАОУ «Мичуринский лицей».

План реализации

1. Описание предметной области
2. Определение сущностей
3. Определение связей
4. Определение атрибутов сущностей
5. Построение диаграммы сущность-связь
6. Построение даталогической модели БД
7. Заполнение БД
8. Тестирование и итог

**Проектирование базы данных l**

Описание предметной области

Каждый человек часть своей жизни обязан проводить в общеобразовательном учреждении. Это может быть обычная школа, лицей, гимназия, вечерняя школа, кадетская школа или школа-интернат. Поэтому общее представление о структуре таких учреждений имеется. Прежде всего, они ассоциируются с учениками, учителями и их взаимодействием во время учебного процесса.

Как правило, в сравнении с учреждениями высшего образования, общеобразовательные организации значительно уступают в размерах. Тем не менее, они имеют не менее сложную и периодически изменяющуюся структуру. Например, ТОГАОУ «Мичуринский лицей-интернат» несколько раз глобально реорганизовал свою структуру. Помимо таких всеобъемлющих изменений есть и другие: поступил/выпустился очередной класс/ученик, был принят/уволен учитель. Все это нужно как-то отслеживать, чтобы школа работала в штатном режиме. И естественно данный процесс можно оптимизировать и автоматизировать. Одним из вариантов является создание базы данных.

Помимо уже перечисленных функций, через базу данных можно отслеживать успеваемость учеников/классов, эффективность работы учителей, достижения учеников/учителей в олимпиадах/конференциях. Подобные вещи необходимо отслеживать, поскольку они напрямую влияют на рейтинг образовательного учреждения.

Структура БД

1. при приеме/отчислении ученика будет заноситься/удаляться соответствующая запись в таблицу «Ученик», которая связана с конкретным «Классом»;

2. для каждого «Ученика» будут вноситься записи в таблицу его «Успеваемости» по различным предметам (связь с «Предметами»);

3. класс, в свою очередь, будет связан с таблицей «Расписания», а она с таблицами «Предмета», «Кабинета» и «Учителя», в «Расписание» можно будет добавлять/изменять/удалять уроки.

4. при приеме/увольнении учителя будет заноситься/удаляться соответствующая запись в таблицу «Учитель», которая связана с таблицами «Профиля», «Заработной платы» и «Предмета».

**Концептуальная модель предметной области**

Проектируемая БД будет состоять из следующих 9 сущностей “pupils”, “teachers”, “classes”, “marks”, “timetable”, “cabinets”, “subjects”, “direction”, “salary”.

**classes – timetable 1:М, обязательная, идентифицирующая**

Одному классу может соответствовать множество записей в

расписании, а конкретная запись в расписании не может одновременно

принадлежать двум классам; при этом у класса не может не быть хотя бы

одной записи в расписании, а в расписании не может не быть классов; связь

идентифицирующая, поскольку первичный ключ класса мигрирует в

первичный ключ расписания;

**subject – timetable 1:М, необязательная, неидентифицирующая**

один предмет может появляться во многих записях таблицы расписания,

но одна запись расписания не может соответствовать нескольким предметам (в одно и тоже время нельзя провести два предмета у класса); при этом предмет может ни разу не появиться в расписании; первичный ключ предмета не мигрирует в первичный ключ расписания;

**teachers – timetable 1:М, необязательная, неидентифицирующая**

один учитель может появляться во многих записях таблицы расписания,

но одна запись расписания не может соответствовать нескольким учителям

(одно и тоже занятие у одного класса в одно время не может проводиться двумя людьми одновременно); при этом учитель может ни разу не появиться в расписании; первичный ключ учителя не мигрирует в первичный ключ

расписания;

**cabinets – timetable 1:М, необязательная, неидентифицирующая**

один кабинет может появляться во многих записях таблицы расписания,

но одна запись расписания не может соответствовать нескольким кабинетам

(одно и тоже занятие у одного класса в одно время не может проводиться в двух местах одновременно); при этом кабинет может ни разу не появиться в

расписании; первичный ключ кабинета не мигрирует в первичный ключ

расписания;

**teachers – subject М:М, обязательная**

один учитель может вести несколько предметов и один предмет может

вестись несколькими учителями; при этом у учителя всегда есть предмет, а у

предмета – учитель;

**direction – teachers 1:М, обязательная, неидентифицирующая**

учитель может состоять только в одном профиле, а профиль может

содержать нескольких учителей; при этом не существует профиля без

учителей и учителя без профиля; первичный ключ профиля не мигрирует в

первичный ключ учителя;

**salary – teachers 1:М, необязательная, неидентифицирующая**

учитель может получать только один вариант заработной платы, а один

тип заработной платы может назначаться нескольким учителям; при этом

может существовать тип зарплаты, который не назначается никому;

первичный ключ зарплаты не мигрирует в первичный ключ учителя;

**сlasses – pupils 1:М, обязательная, неидентифицирующая**

в одном классе может быть множество учеников, а каждый ученик

может принадлежать только одному классу; при этом не существует класса

без учеников и ученика без класса; первичный ключ класса не мигрирует в

первичный ключ ученика;

**pupils – marks 1:М, необязательная, идентифицирующая**

у одного ученика может быть несколько записей в таблице оценок

(например, по разным предметам), но каждой записи соответствует только

один ученик, при этом ученик может и не иметь записей в таблице оценок;

связь идентифицирующая, поскольку первичный ключ ученика мигрирует в

первичный ключ оценок;

**subject – marks 1:M, обязательная, идентифицирующая**

одному предмету может соответствовать несколько записей-оценок

(много учеников получают по одному и тому же предмету оценки), но одна

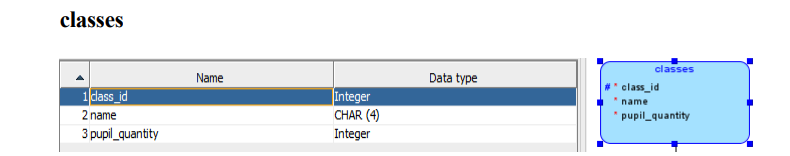
запись-оценка соответствует только одному предмету, при этом оценка не

может существовать без предмета; связь идентифицирующая поскольку

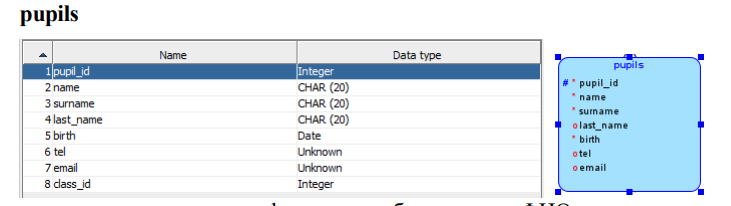
первичный ключ предмета мигрирует в первичный ключ оценок.

Определение атрибутов

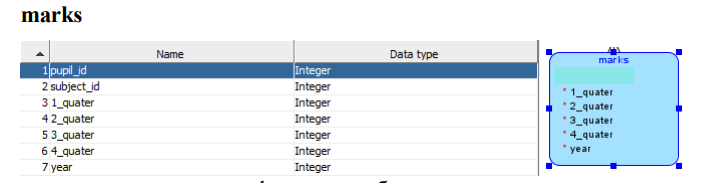
Теперь перейду к описанию хранимой в сущностях информации. Домены атрибутов указаны на рисунках.



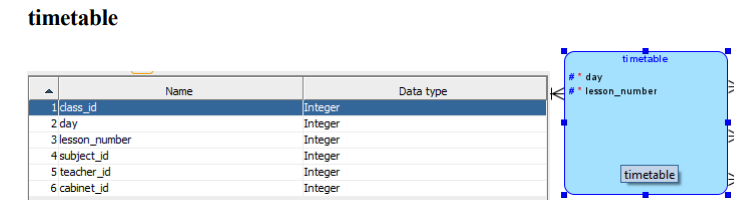
данная сущность содержит информацию о названии класса – сочетание букв и цифр (например, 1а, 8б) и количестве учеников. Уникальным идентификатором было решено сделать class\_id – искусственный ключ, поскольку название класса нельзя брать в качестве первичного ключа из-за типа данных.



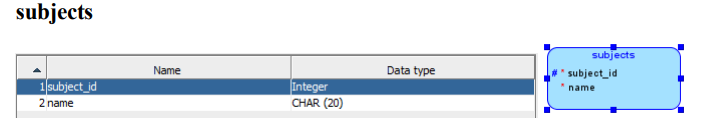
данная сущность содержит информацию об учениках: ФИО, дата рождения, класс обучения (внешний ключ, наследуется от classes), телефон и почта. Уникальный идентификатор – искусственный ключ pupil\_id.



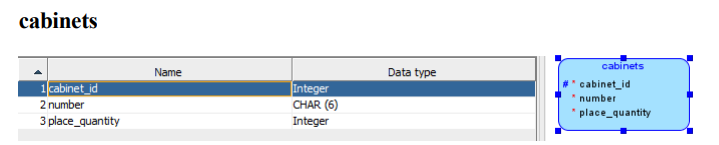
данная сущность содержит информацию об оценках учеников за все четверти и год по какому-либо предмету. Первичный ключ будет составным и включать в себя pupil\_id (внешний ключ – наследуется от pupils) и subject\_id (внешний ключ – наследуется от subject).



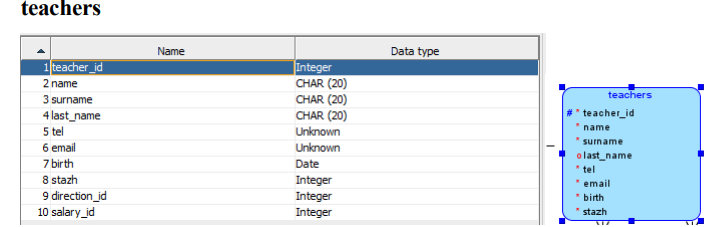
сущность расписания для различных классов. Содержит информацию о предмете, месте проведения и учителе – внешние ключи от одноименных сущностей. Первичный ключ – составной, включает день, номер урока в этот день и class\_id (внешний ключ).



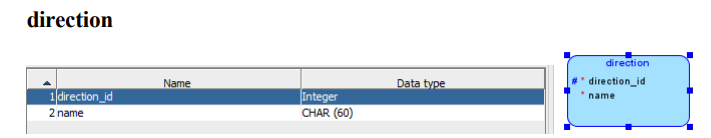
сущность с названием предмета, уникальный идентификатор – искусственный ключ subject\_id



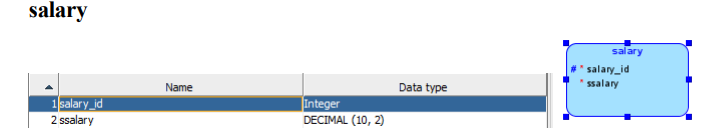
сущность с обозначениями кабинетов и их вместимостью, уникальный идентификатор – искусственный ключ cabinet\_id.



данная сущность содержит информацию об учителях: ФИО, дата рождения, стаж, телефон, почта, профиль преподавания и заработная плата. Последние два атрибута – внешние ключи, наследуется от одноименных сущностей. Уникальный идентификатор – искусственный ключ teacher\_id.

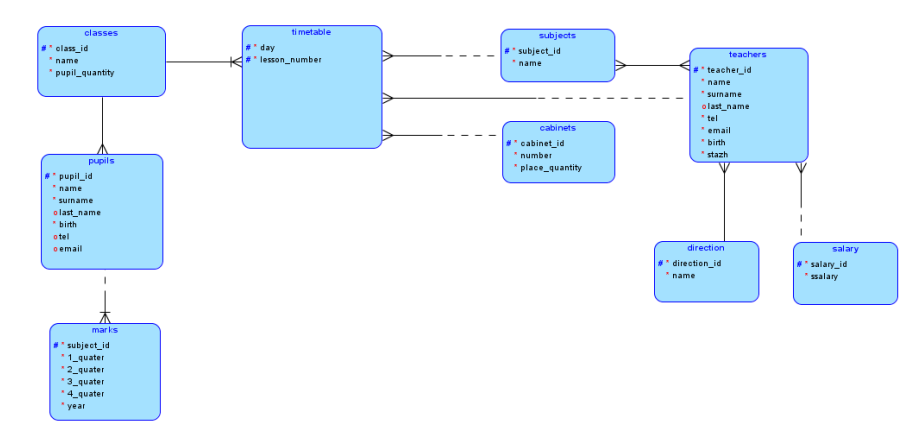


сущность с названием профиля преподавания, уникальный идентификатор – искусственный ключ direction\_id.



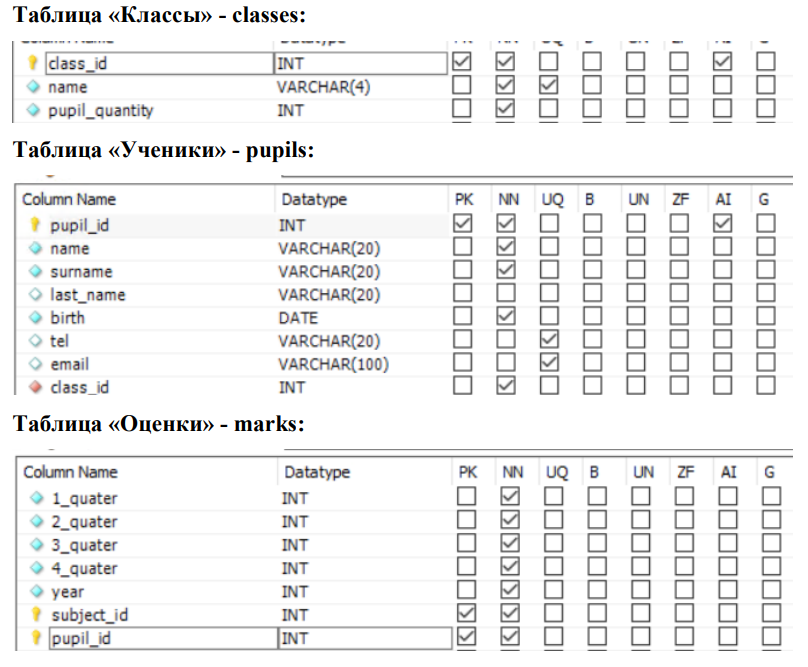
сущность с размером заработной платы, уникальный идентификатор – искусственный ключ salary\_id.

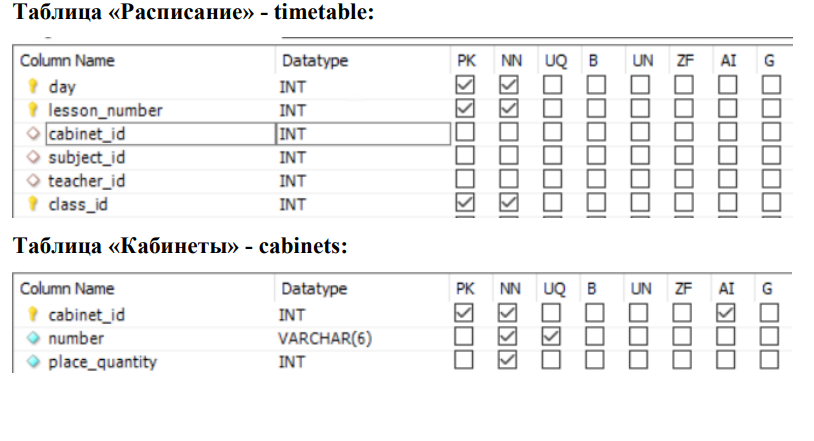
**Концептуальная модель предметной области**

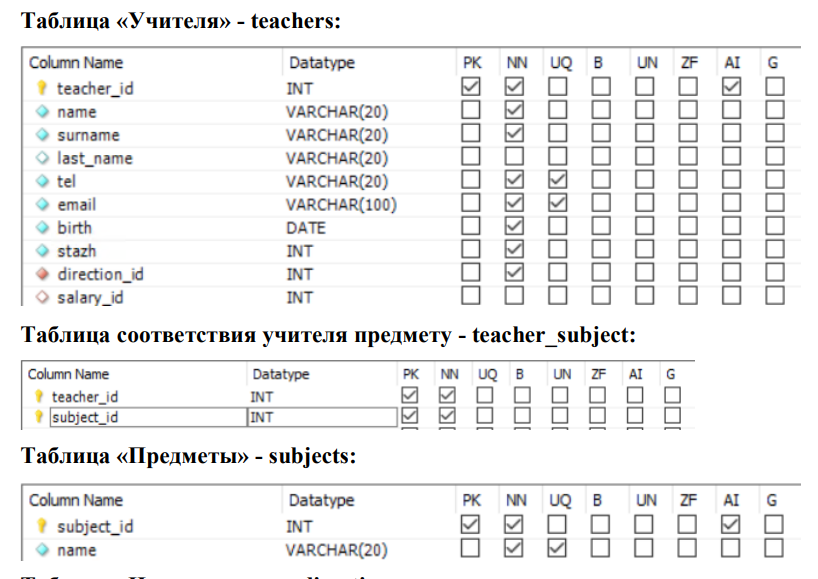
****

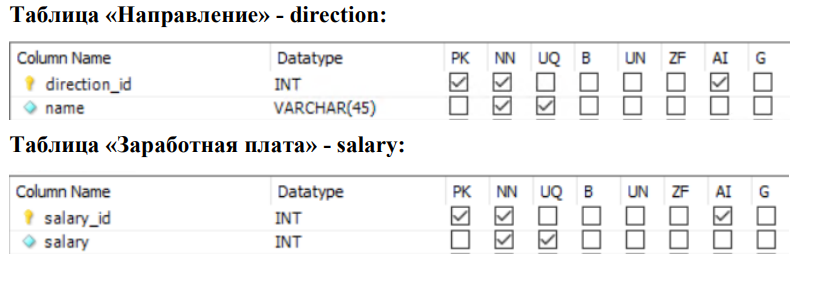
**Даталогическая модель базы данных**

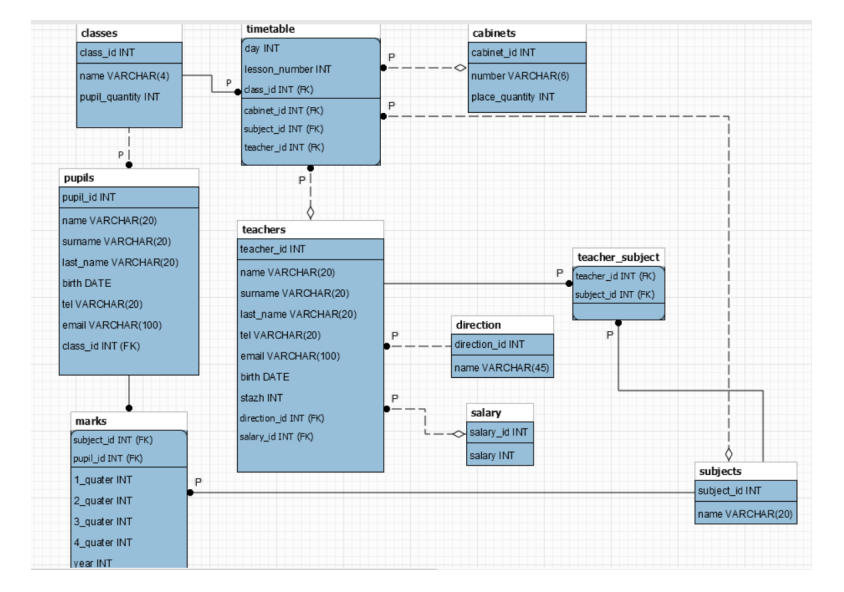
Ниже приведены свойства полей таблиц, которые будут созданы.





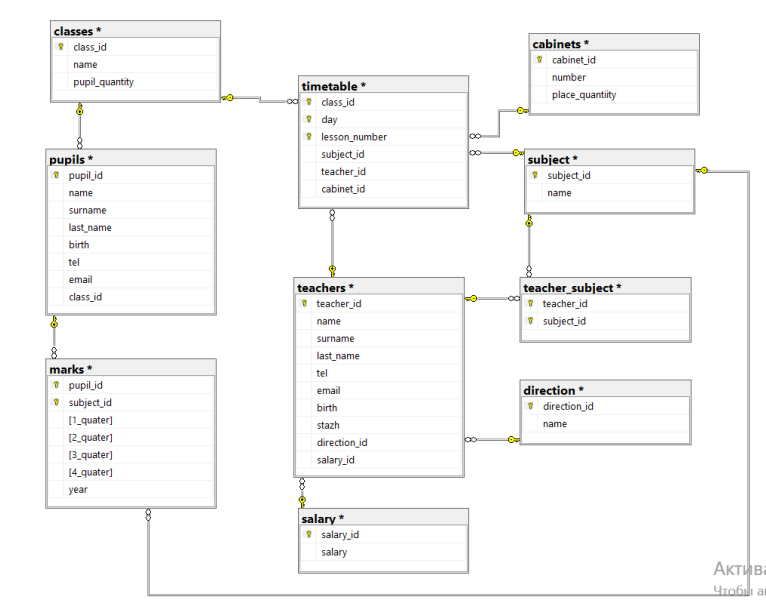






**Проектирование базы данных ll**

Сценарий создания объектов базы данных



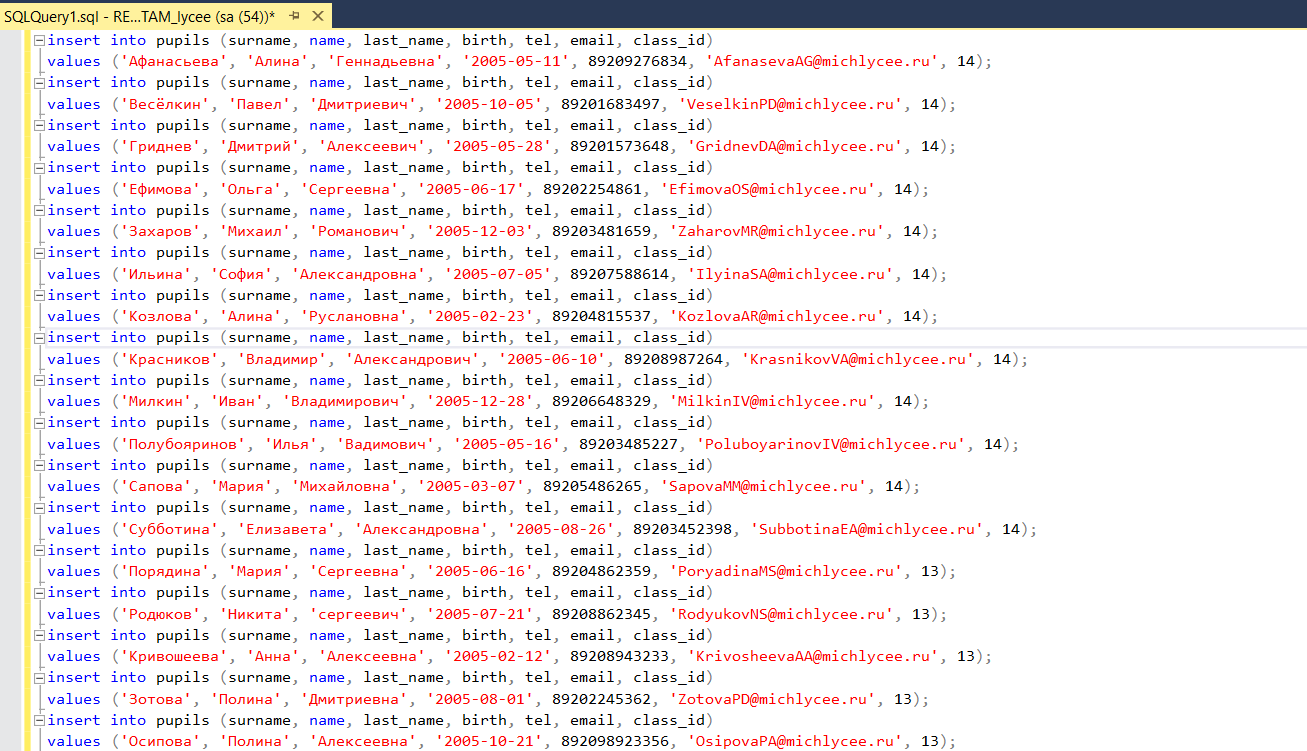
**Заполнение базы данных**

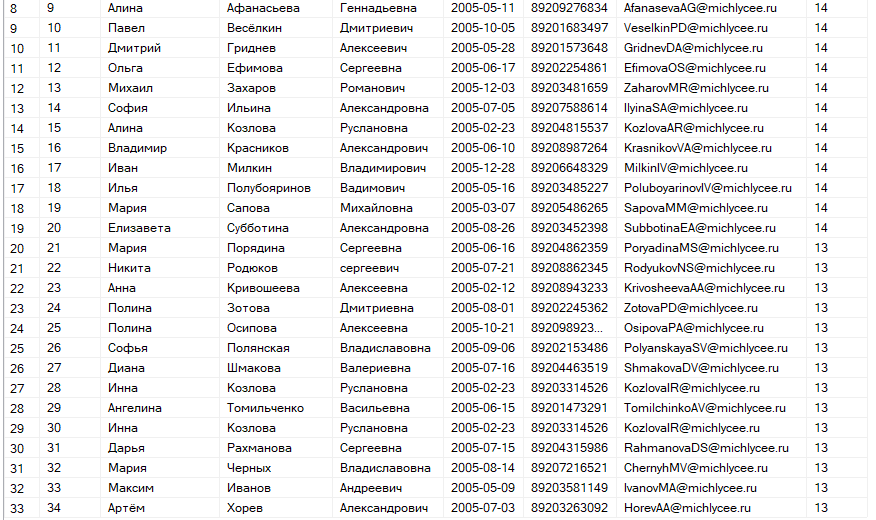
Для базы данных необходимо заполнить 9 созданных мной сущностей (“pupils”, “teachers”, “classes”, “marks”, “timetable”, “cabinets”, “subjects”, “direction”, “salary”.)

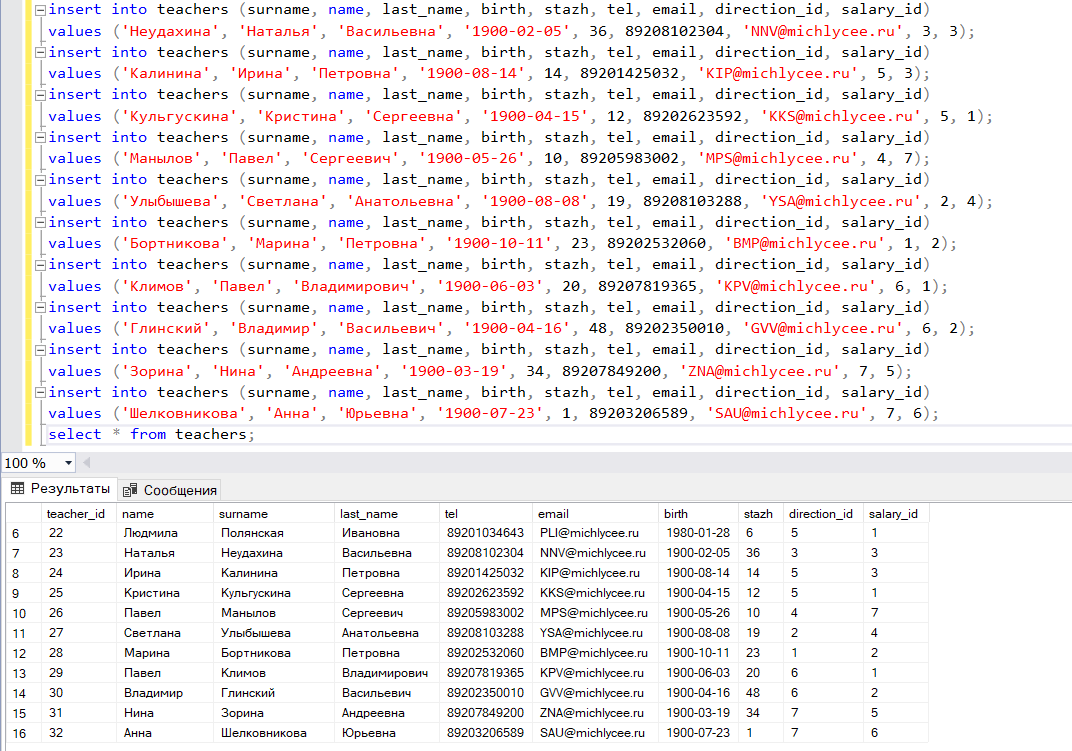
Каждая сущность заполняется по определённому строгому образцу, порядок в котором нарушать нельзя.

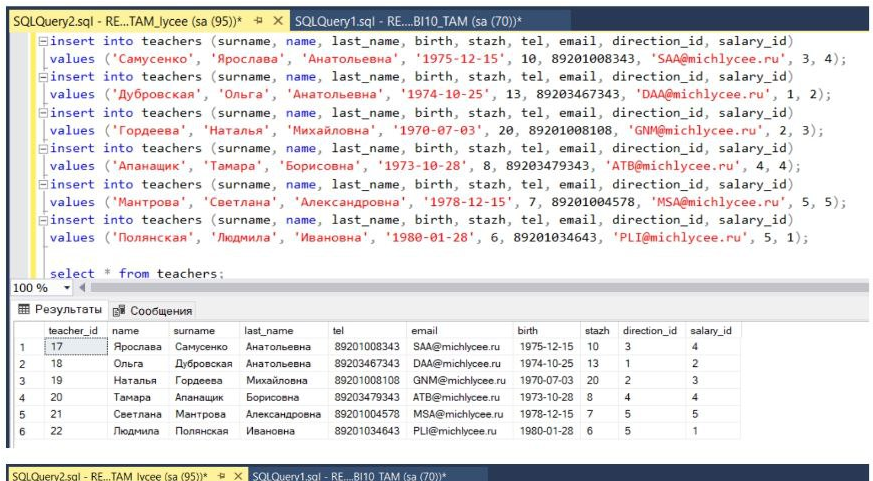
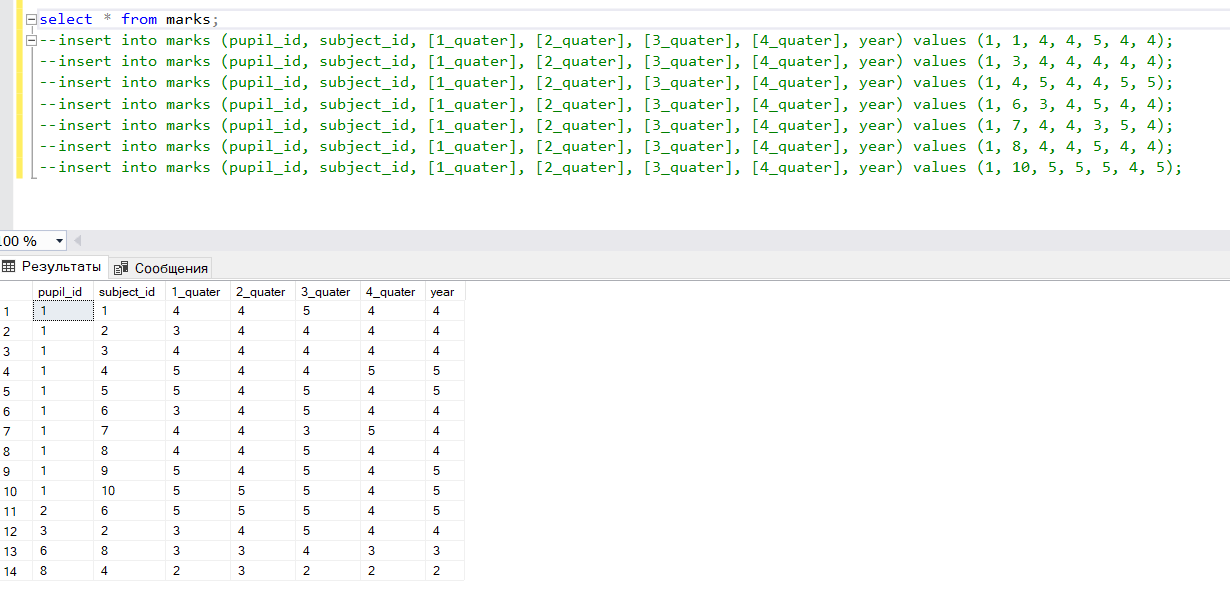
insert – оператор позволяющий добавить строки в таблицу, т.е. insert into pupils означает, то что я добавляю строки с информацией в сущность ученики.

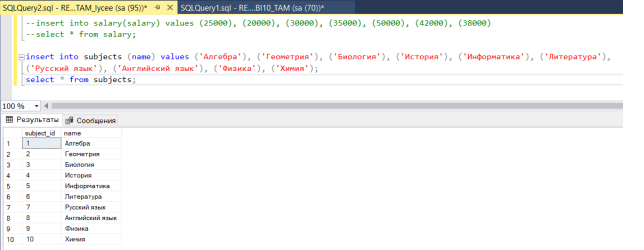
Дальше идёт обычное заполнение values (значений).

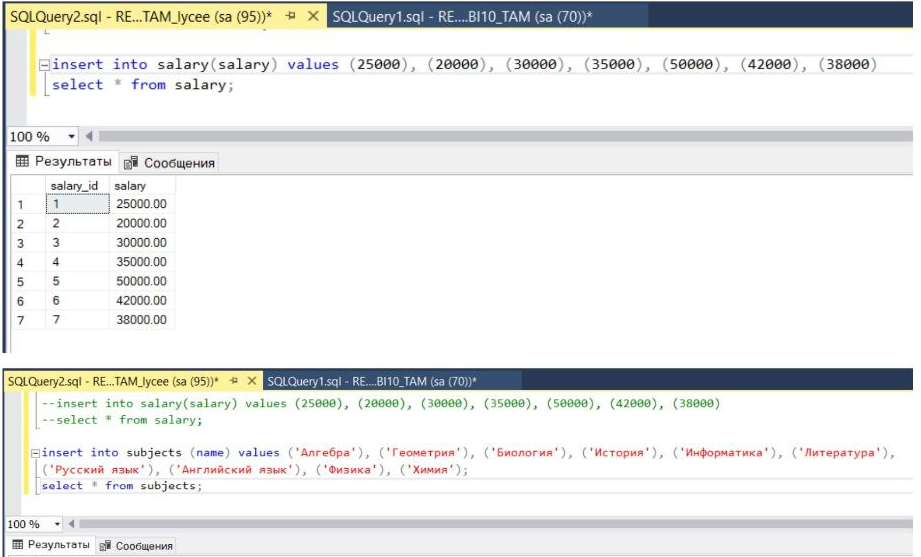
Pupils

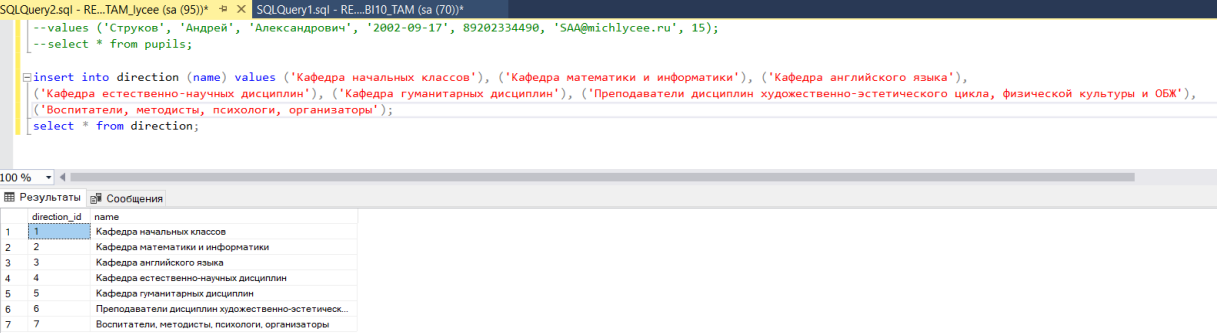


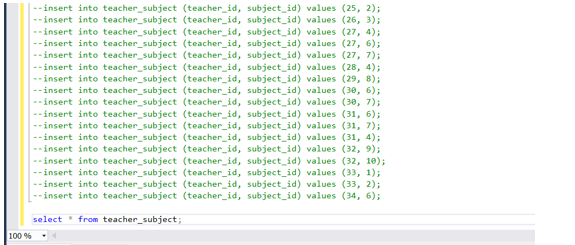
Teachers

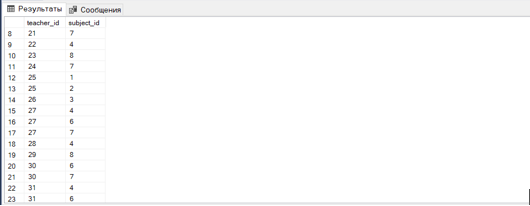
Marks

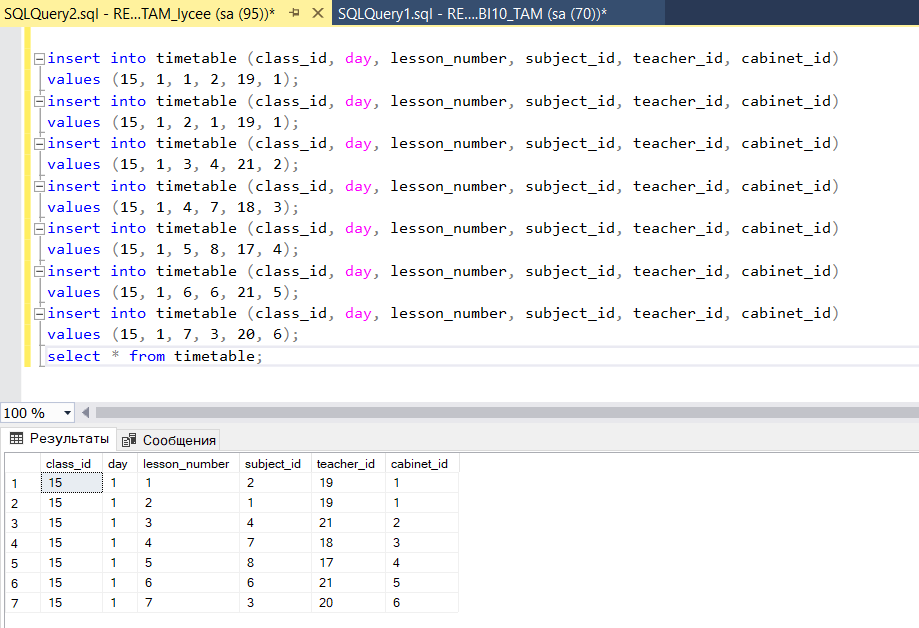
Subjects

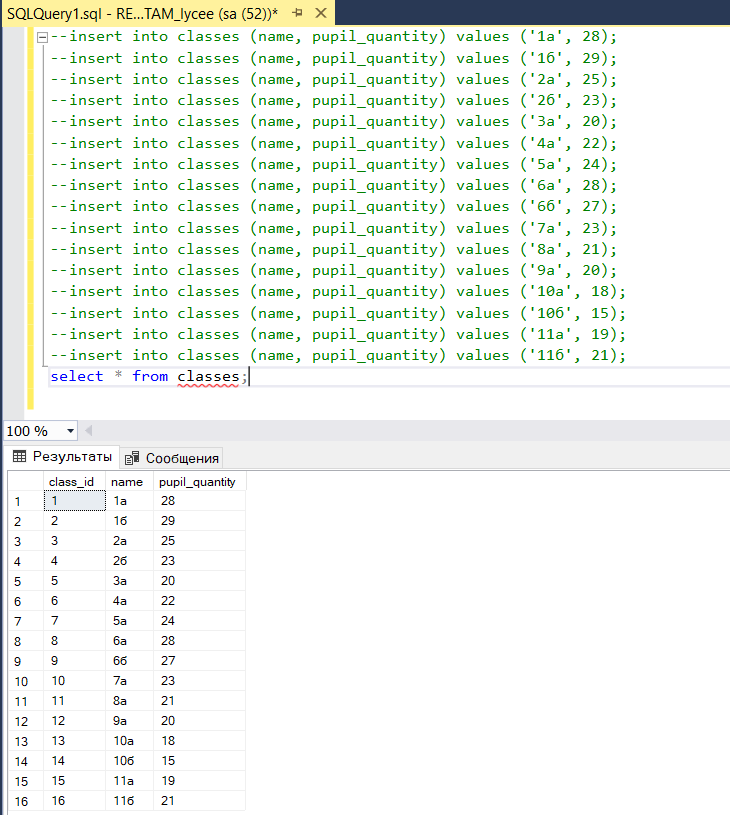
Salary

Direction

Cabinets

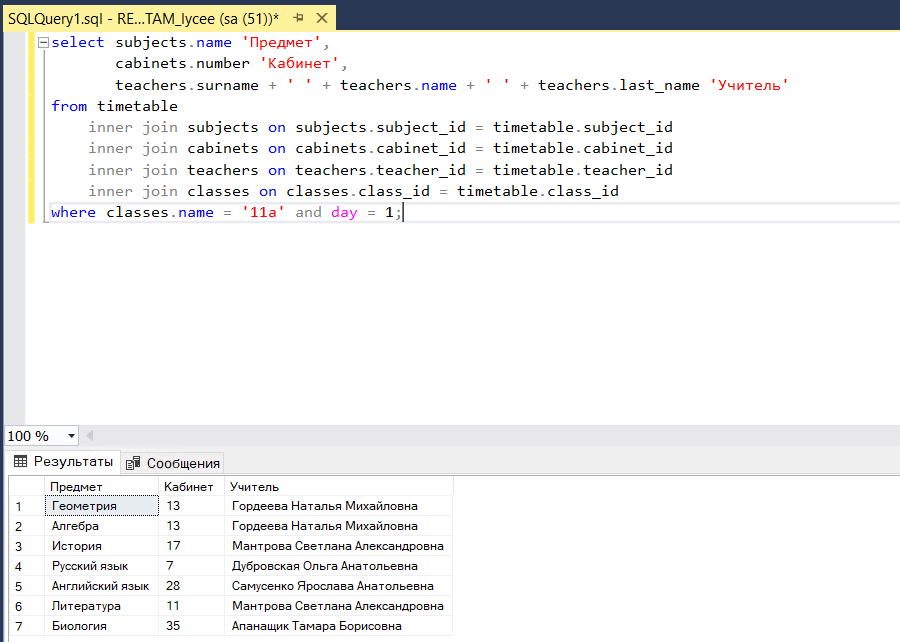
****

Timetable

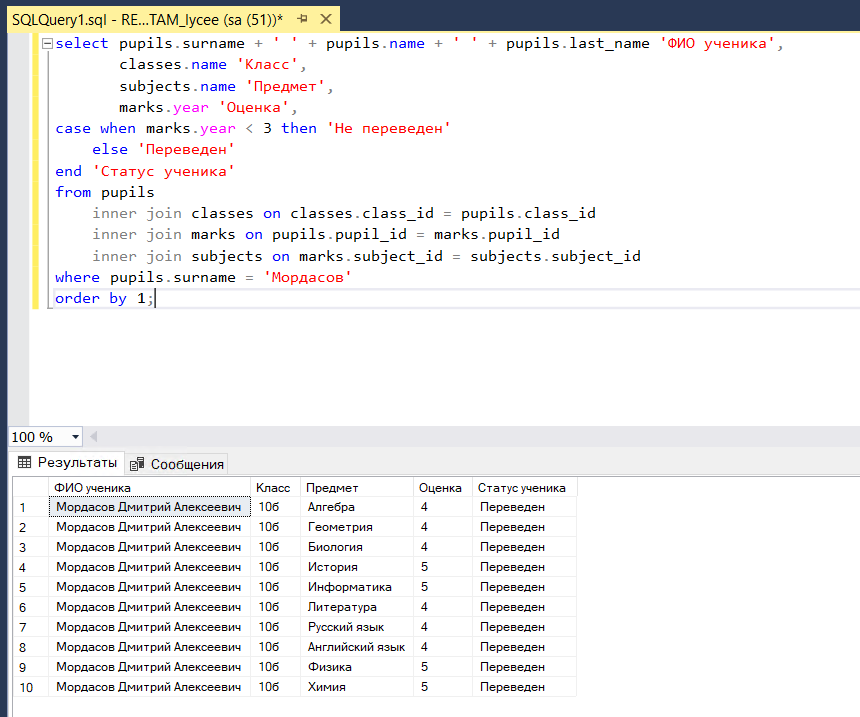
Classes

**Проверка запросами**

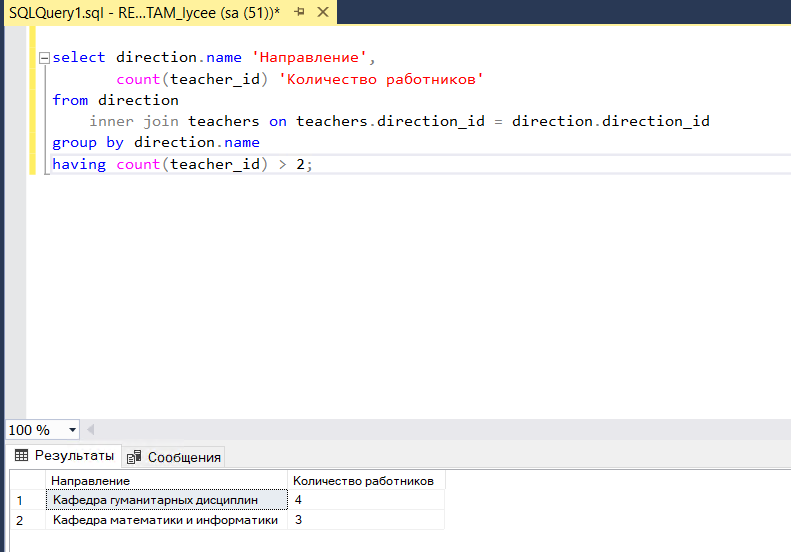
Вывести расписание для конкретного класса в конкретный день.



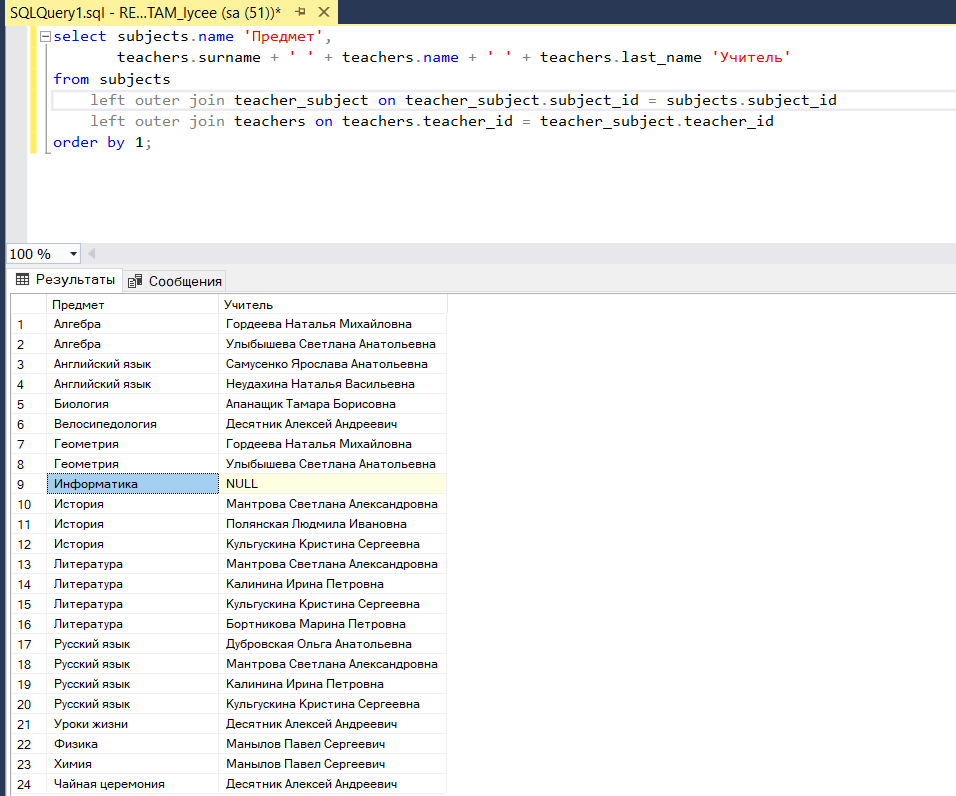
Вывести ФИО учеников, их класс обучения, годовые оценки по предметам, статус перевода на следующий год обучения.

****

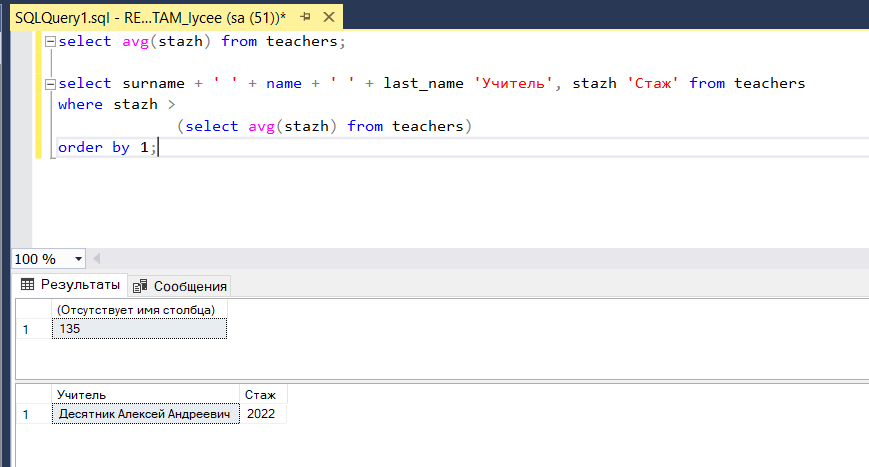
Вывести направления, где больше или равно 2 работников.

****

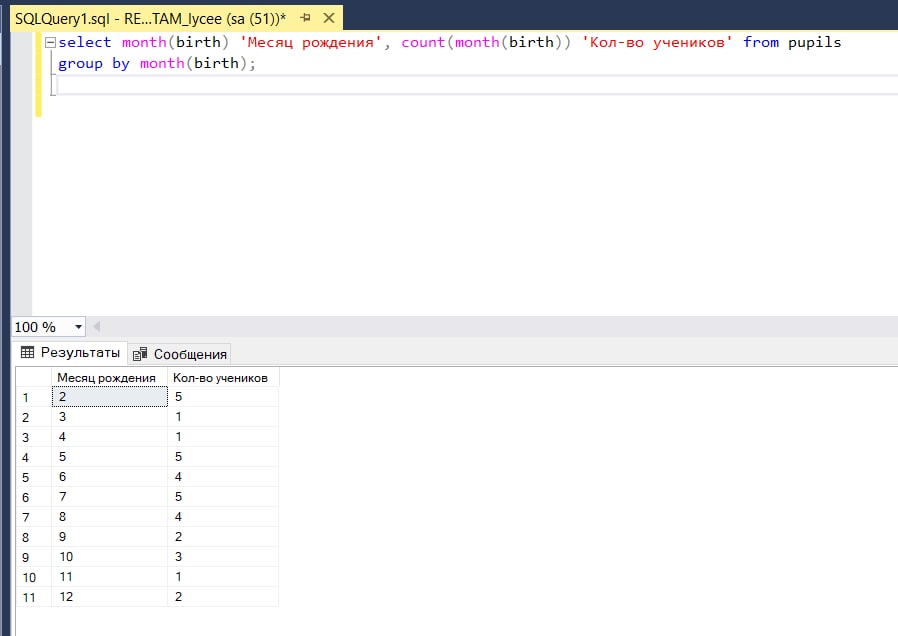
Проверить у всех ли предметов есть преподаватель.

****

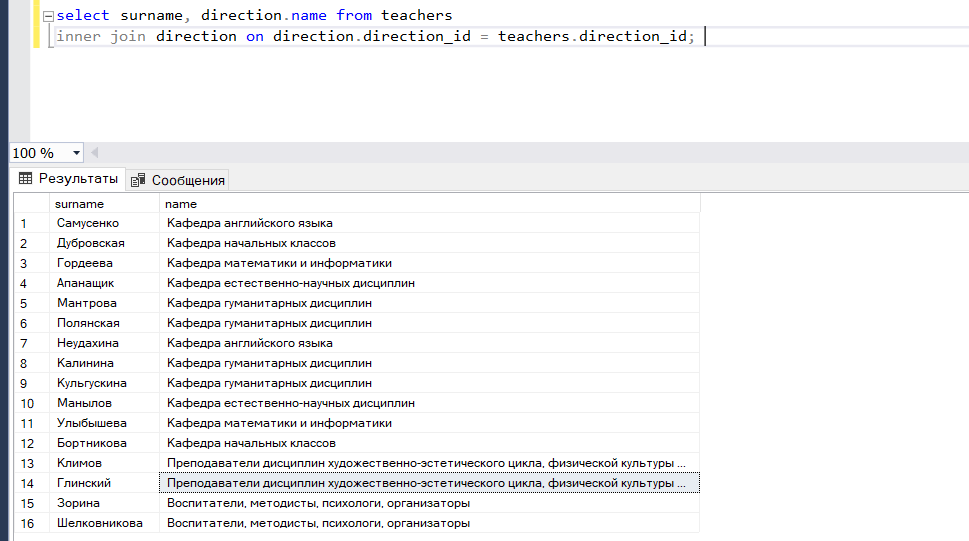
Вывести учителей, чей стаж больше среднего стажа всех учителей.

****

Вывести месяц и количество учеников рождённых в этот месяц.



Вывести к каким кафедрам принадлежат учителя.



**Итог**

Таким образом, была спроектирована база данных для общеобразовательного учреждения. С помощью созданной БД можно выполнять весь описанный в первой главе функционал.

Кроме того, информацию из БД можно использовать для автоматизации процесса составления расписания.

В дальнейшем можно расширить БД следующим образом: 1. Добавить таблицу не только рубежных оценок, но и ежедневных; 2. Добавить таблицу участия учеников в различных этапах олимпиад и их результаты; 3. Данная часть БД была сфокусирована именно на образовательном процессе, однако дальше можно масштабировать ее и добавить другие разделы школы и их отношения, например: библиотека, столовая, медицинское крыло, общежитие, администрация и прочее.