**(2 слайд) Я поставил перед собой цель: научиться проектировать и работать с базами данных.**

**Для достижения данной цели были решены следующие задачи: изучение истории развития баз данных, изучение системы управления базами данных, продемонстрировать результаты изучения с помощью создания приложения “Электронный дневник”.**

**Решению цели способствовали следующие методы: применялось программное обеспечение** SQLiteStudio, QtDesigner, среда разработки PyCharm, язык программирования Python, язык программирования SQL.

Система “Электронный дневник”, разработанная мной, позволит каждому ученику знать свои оценки, а учителю выставлять оценки.

История развития баз данных

Толчком создания баз данных послужили недостатки файловых систем. Такие как, сложность создания программ для работы с файлами разной структуры, отсутствие централизованных методов управления доступом к файлам, замедленная организация многопользовательской работы с файлом. В результате появился новый подход к управлению информацией. Этот подход был реализован в рамках новых программных систем, названных впоследствии Системами Управления Базами Данных (СУБД), а сами хранилища информации, которые работали под управлением данных систем, назывались базами или банками данных (БД и БнД).

История развития СУБД насчитывает более 50 лет. В 1968 году была введена в эксплуатацию первая промышленная СУБД система IMS фирмы IBM. В 1975 году появился первый стандарт ассоциации по языкам систем обработки данных — Conference of Data System Languages (CODASYL), который определил ряд фундаментальных понятий в теории систем баз данных, которые и до сих пор являются основополагающими для сетевой модели данных.

В дальнейшее развитие теории баз данных большой вклад был сделан американским математиком Э. Ф. Коддом, который является создателем реляционной модели данных, которая стала основой реляционной базы данных.

Когда персональные компьютеры стали появляться у многих людей, начали создаваться простые программы для неподготовленных пользователей. Такие изменения коснулись и СУБД. Раньше изменения приходилось делать вручную, но с появлением программ с удобным интерфейсом данные заполнять легче.

Множится количество локальных сетей, все больше информации передается между компьютерами, остро встает задача согласованности данных, хранящихся и обрабатывающихся в разных местах, но логически друг с другом связанных. Успешное решение этих задач приводит к появлению распределенных многопользовательских баз данных, сохраняющих все преимущества настольных СУБД и в то же время позволяющих организовать параллельную обработку информации и поддержку целостности БД.

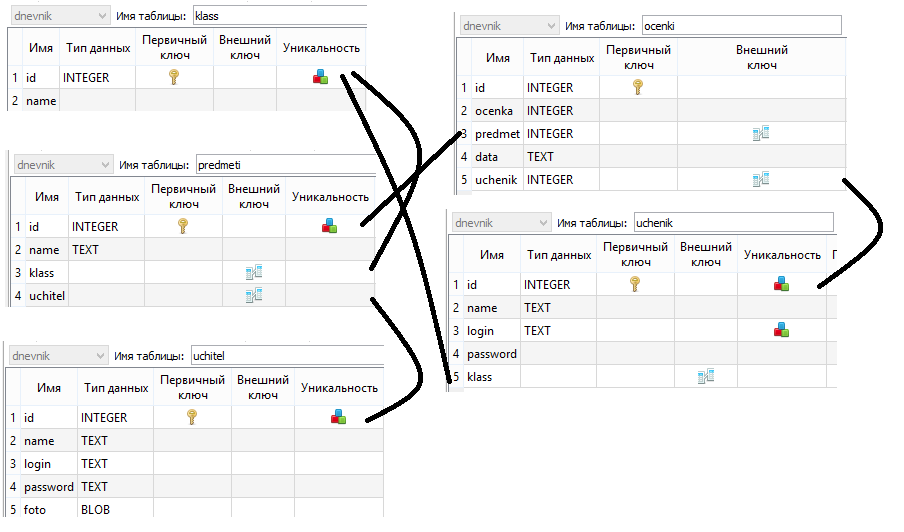
**(3 слайд)** Современные СУБД поддерживают реляционную модель данных, которая лежит в основе реляционных БД. Реляционная база данных – это набор данных с предопределенными связями между ними. Эти данные организованны в виде набора таблиц, состоящих из столбцов и строк. В таблицах хранится информация об объектах, представленных в базе данных. В каждом столбце таблицы хранится определенный тип данных, в каждой ячейке – значение атрибута. Каждая стока таблицы представляет собой набор связанных значений, относящихся к одному объекту или сущности. Каждая строка в таблице может быть помечена уникальным идентификатором, называемым первичным ключом, а строки из нескольких таблиц могут быть связаны с помощью внешних ключей. К этим данным можно получить доступ многими способами, и при этом реорганизовывать таблицы БД не требуется.

**(4 слайд)** Основным языком программирования, используемым в СУБД, является SQL. Он основан на отправке запросов базе данных. Есть четыре основных типа запросов данных в SQL, которые относятся к языку манипулирования данными (DML): SELECT – поможет выбрать необходимые строки из таблицы, INSERT – добавит строки в таблицу, UPDATE – обновит строки в таблице, DELETE – удалит строки из таблицы. Каждый из этих запросов имеет различные операторы и функции, которые помогают произвести различные действия с данными. Например, основной командой для получения какой-либо информации из БД является команда SELECT. Ее базовый синтаксис выглядит так:

SELECT перечень\_полей FROM имя\_таблицы WHERE условие

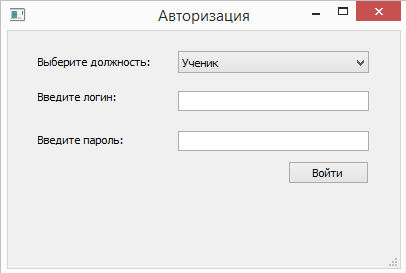
Для наглядного отображения указанной выше информации я решил создать приложение “Электронный дневник”, который повторяет основной функционал настоящего электронного дневника.

**(5 слайд)** Изначально я с помощью менеджера SQLiteStudio баз данных создал базу данных, состоящую из таблиц (Рис. 1). Линиями показаны связи между таблицами, т.е. значение ключа из одной таблицы (внешний ключ) соответствует значению ключа из другой таблицы. Иногда внешний ключ также называют ссылочным ключом.



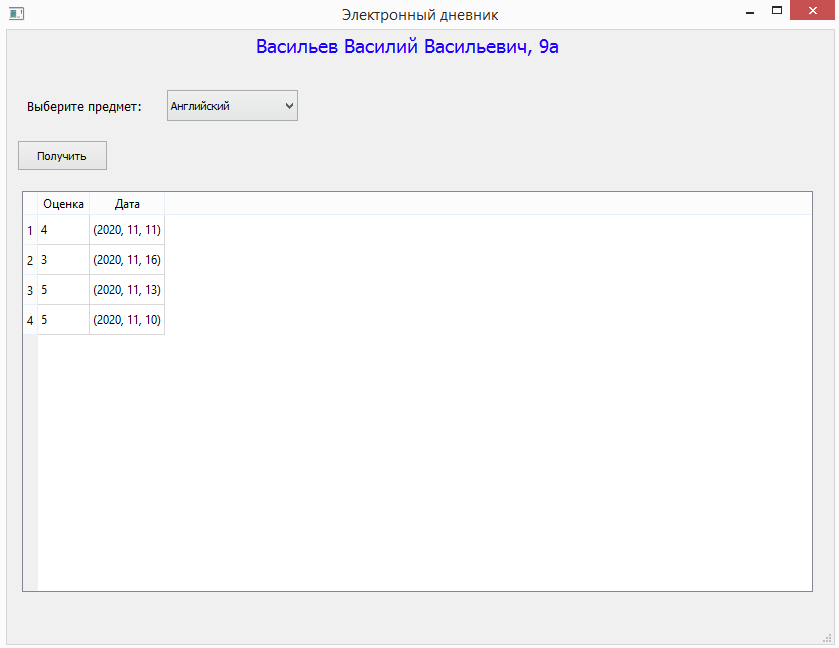
Дальше я создал формы для авторизации, для ученика и для учителя в программе QtDesigner, а затем написал программу на языке программирования Python, с помощью которой удалось связать формы с созданной базой данных и добиться работоспособности моего приложения.

**(6 слайд)** При запуске программы открывается форма для авторизации (Рис. 2). В ней необходимо будет выбрать должность (учитель, ученик), ввести логин и пароль.



После успешного входа в зависимости от выбранной должности открывается нужная форма.

**(7 слайд)** Рассмотрим форму ученика (Рис. 3).



Сверху формы отображается ФИО ученика и класс, в котором он обучается. Далее необходимо выбрать предмет. При нажатии на кнопку “Получить” мы видим таблицу, в которой каждой записи соответствует оценка и дата ее получения.

**(8 слайд)** Перейдем к форме учителя (Рис. 4).

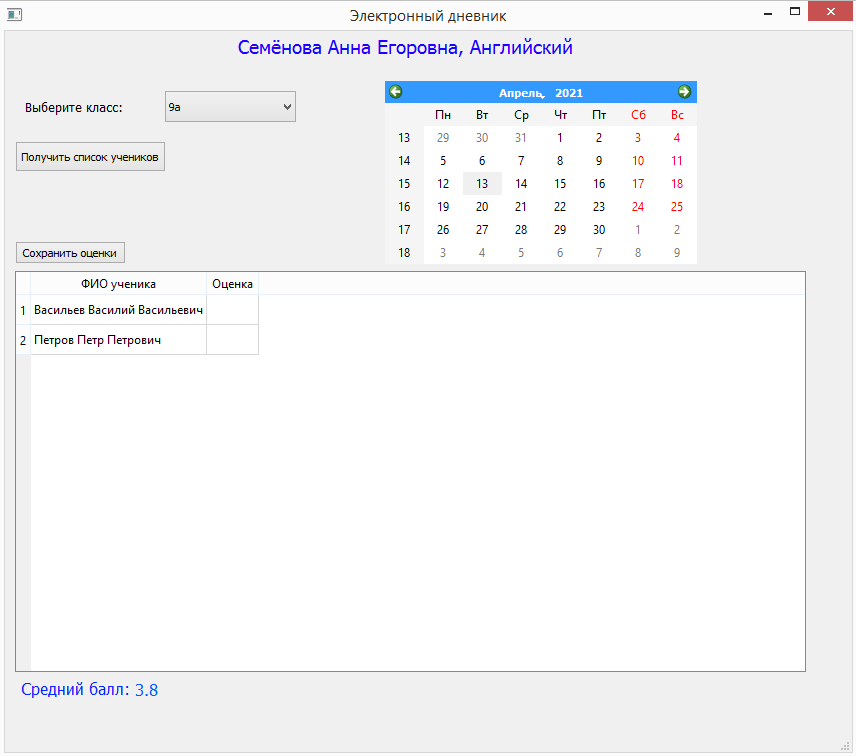


Рис. 4. Форма учителя

Сверху отображается ФИО учителя и предмет, который он преподает. Далее предлагается выбрать класс и дату на календаре. При нажатии на кнопку “Получить список учеников” отображается таблица, каждая запись которой соответствует ФИО ученика и оценке за выбранный на календаре день. Так же можно увидеть средний балл класса. После можно выставить оценку (или убрать ее), а затем нажатием на кнопку “Сохранить оценки” применить изменения.