**18.02.2022 Учебная группа 1ТМ, 2-я пара**

**Преподаватель Иванова Наталия Викторовна**

**ОДП.03 Информатика и ИКТ**

**Тема:** *Многотабличные базы данных. Схема БД. Связанные таблицы. Целостность данных. Этапы создания многотабличной БД с помощью реляционной СУБД..*

**Цель занятия:** *Обучающая* – сформировать знания о базах данных как основе информационной системы.

*Развивающая* – умения применять полученные знания при решении задач различной направленности.

*Воспитывающая* – создавать условия для воспитания потребности в овладении специальными знаниями, умениями, навыками.

**Задачи занятия:** изучение нового материала, первичное закрепление.

**Мотивация:** знание базы данных поможет поднять свой профессиональный уровень.

**Задание студентам:**

1. Изучить видеоурок <https://youtu.be/RxQyZyM0vC4>
2. Изучить пар. 6-7, учебника Семакин И.Г., Хеннер Е.К., Шейна Т.Ю. Информатика, учеб, 11 кл. базовый уровень/ И.Г. Семакин, Е.К. Хеннер, Т.Ю. Шейна изд.-М.: Бином. Лаборатория знаний, 2015.-264 с.
3. В тетради ответить на вопросы из лекции.

Фотографию с выполненным заданием прислать на электронный адрес **atata17@yandex.ru** в срок **до 08.00 21.02.2022** **г.**

**Лекция № 8**

**План**

1. Табличная форма модели данных
2. Создание БД

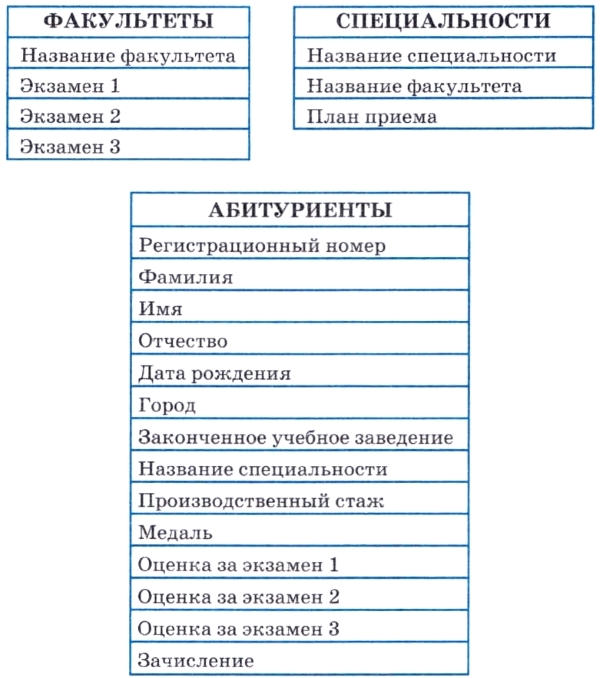
**Вопрос 1**. Табличная форма модели данных

Построим модель данных, состоящую из трех взаимосвязанных таблиц.

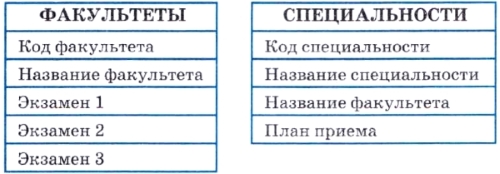
Эти три таблицы можно рассматривать как модель данных в реляционной СУБД. Но работать с БД в таком виде неудобно. Помимо того что реляционная БД должна состоять из таблиц, к ней предъявляется еще ряд требований.

Одним из главных требований является требование **отсутствия избыточности** (или минимизация избыточности) данных. Избыточность приводит к лишнему расходу памяти. Память нужно экономить. Это не только увеличивает информационную плотность базы данных, но и сокращает время поиска и обработки данных.

Очевидный недостаток описанных таблиц — многократное повторение длинных значений полей в разных записях. Например, название специальности «Радиофизика и электроника» будет повторяться в 100 записях для 100 абитуриентов, которые на нее поступают. Проще сделать так. В таблице **СПЕЦИАЛЬНОСТИ**для каждой специальности ввести свой короткий код. Тогда полное название запишется в БД только один раз, а в анкетах абитуриентов будет указываться только код. Точно так же можно закодировать названия факультетов.

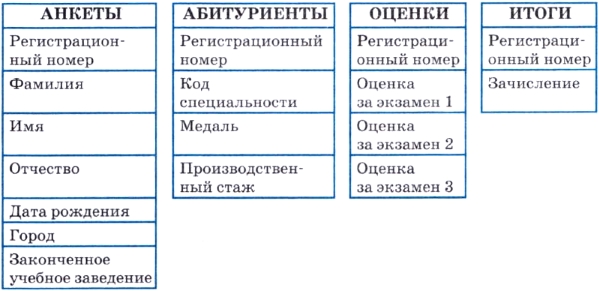


Внесем изменения в таблицы **ФАКУЛЬТЕТЫ и СПЕЦИАЛЬНОСТИ.**



Здесь предполагаются два упрощающих допущения: пусть на разных специальностях одного факультета сдаются одни и те же экзамены, а число экзаменов на всех факультетах равно трем (это вполне разумно).

Очень неудобной для работы является таблица АБИТУРИЕНТЫ. В ней слишком много полей. В частности, такую таблицу неудобно будет просматривать на экране, легко запутаться в полях. Поступим следующим образом. Разделим «большую» таблицу АБИТУРИЕНТЫ на четыре таблицы поменьше:

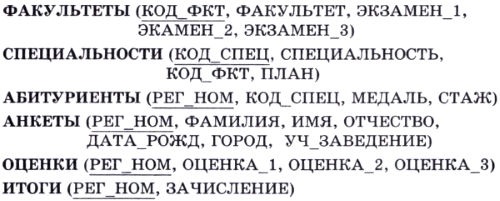


С такими таблицами работать гораздо проще. На разных этапах работы приемной комиссии каждая из этих таблиц будет иметь самостоятельное значение.

Таблица АНКЕТЫ содержит анкетные данные, не влияющие на зачисление абитуриента в вуз. В таблице АБИТУРИЕНТЫ содержатся сведения, определяющие, куда поступает абитуриент, а также данные, которые могут повлиять на его зачисление (предположим, что это может быть производственный стаж и наличие медали). Таблица ОЦЕНКИ — это ведомость, которая будет заполняться для всех абитуриентов в процессе приема экзаменов. Таблица ИТОГИ будет содержать результаты зачисления всех абитуриентов.

**Отношения и связи**

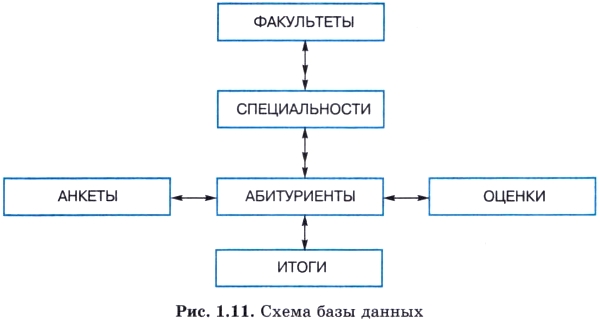
Каждая из спроектированных выше таблиц будет представлена в БД отдельным отношением. Опишем все их в строчной форме, дав в некоторых случаях полям сокращенные имена и подчеркнув главные ключи.



http://xn----7sbbfb7a7aej.xn--p1ai/img/galochka_znak1.png**Чтобы эти шесть таблиц представляли собой систему, между ними должны быть установлены связи**.

Фактически связи уже имеются через общие имена полей. Первые два отношения связаны между собой кодом факультета, второе и третье — кодом специальности, а четыре последних — регистрационным номером. Связи позволяют определить соответствия между любыми данными в этих таблицах, например между фамилией некоторого абитуриента и его оценкой по математике; между названием города и результатами экзамена по русскому языку выпускников школ этого города и пр. Благодаря этим связям возможно получение ответов на запросы, требующие поиска информации в нескольких таблицах одновременно.   
 **Схема базы данных**

 Для явного указания связей между таблицами должна быть построена схема базы данных. В схеме указывается наличие связей между таблицами и типы связей. Схема для нашей системы представлена на рис. 1.11.



В схеме использованы два типа связей: один к одному и один ко многим. Первый обозначен двунаправленной одинарной стрелкой, второй — одинарной стрелкой в одну сторону и двойной в другую. При связи «один к одному» с одной записью в таблице связана одна запись в другой таблице. Например, одна запись об абитуриенте связана с одним списком оценок. При наличии связи «один ко многим» одна запись в некоторой таблице связана с множеством записей в другой таблице. Например, с одним факультетом связано множество специальностей, а с одной специальностью — множество абитуриентов, поступающих на эту специальность.

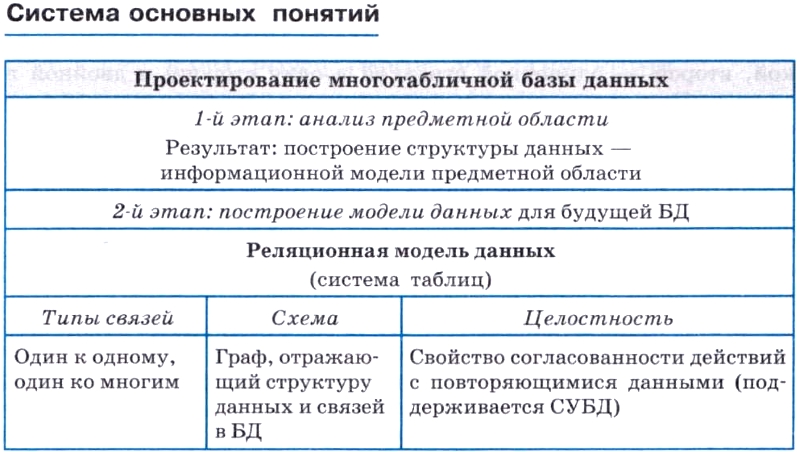
Связь «один ко многим» — это связь между двумя соседними уровнями иерархической структуры. А таблицы, связанные отношениями «один к одному», находятся на одном уровне иерархии. В принципе все они могут быть объединены в одну таблицу, поскольку главный ключ у них один — РЕГ\_НОМ. Но чем это неудобно, было объяснено выше.

**Что такое целостность данных**

СУБД поддерживает организацию связей между таблицами БД, обеспечивающую одно важное свойство базы данных, которое называется целостностью данных.

Система не допустит, чтобы одноименные поля в разных связанных между собой таблицах имели разные значения. Согласно этому принципу, будет автоматически контролироваться ввод данных. В связанных таблицах может быть установлен режим каскадной замены: если в одной из таблиц изменяется значение поля, по которому установлена связь, то в других таблицах одноименные поля автоматически изменят свои значения. Аналогично действует режим каскадного удаления: достаточно удалить запись из одной таблицы, чтобы связанные записи исчезли из всех остальных таблиц. Это естественно, поскольку, например, если закрывается ка- кой-то факультет, то исчезают и все его специальности. Или если у абитуриента изменяют регистрационный номер в таблице АБИТУРИЕНТЫ, то автоматически номер должен обновиться и в других таблицах.

На этом проектирование базы данных завершается. Это был теоретический этап. Практическая работа по созданию базы данных будет проходить в рамках компьютерного практикума.

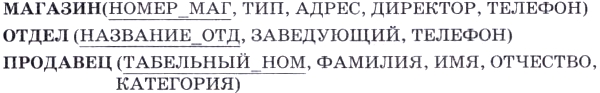


**Вопросы и задания**

1. а) Перечислите задачи, которые должна решать проектируемая информационная система «Приемная комиссия».  
  
б) Какие информационные процессы происходят на различных этапах приемной кампании в вузе?   
в) Какая информация добавляется к базе данных на каждом этапе?

2. а) В чем заключается построение модели данных?  
  
б) Что означает свойство целостности БД?   
в) Какие данные следует добавить в БД приемной комиссии, если требуется учитывать преподавателей, принимающих экзамены, и деление абитуриентов на экзаменационные группы?   
г) Какие данные следует добавить в БД приемной комиссии, если дополнительно к требованиям предыдущего задания нужно учитывать расписание экзаменов, т. е. сведения о том, где, когда и какому преподавателю сдает экзамен данная группа?   
д) Постройте схему БД с учетом выполнения заданий 2, в и 2, г.

3. а) При проектировании БД были определены следующие отношения:

   
Являются ли эти отношения связанными? Добавьте всё, что необходимо для их связи; изобразите схему БД в графическом виде,   
б) Спроектируйте базу данных для информационной системы «Наша школа», содержащей сведения об учителях, учениках, классах, изучаемых предметах.   
**Замечание:** данное задание носит творческий характер и может быть выполнено во многих вариантах. Устройте конкурс на лучшее решение этой задачи.

**Вопрос 2 Создание БД**

База данных создается средствами СУБД. Создание происходит в два этапа.  
1) Построение структуры таблиц и установка связей.   
2) Ввод данных в таблицы.

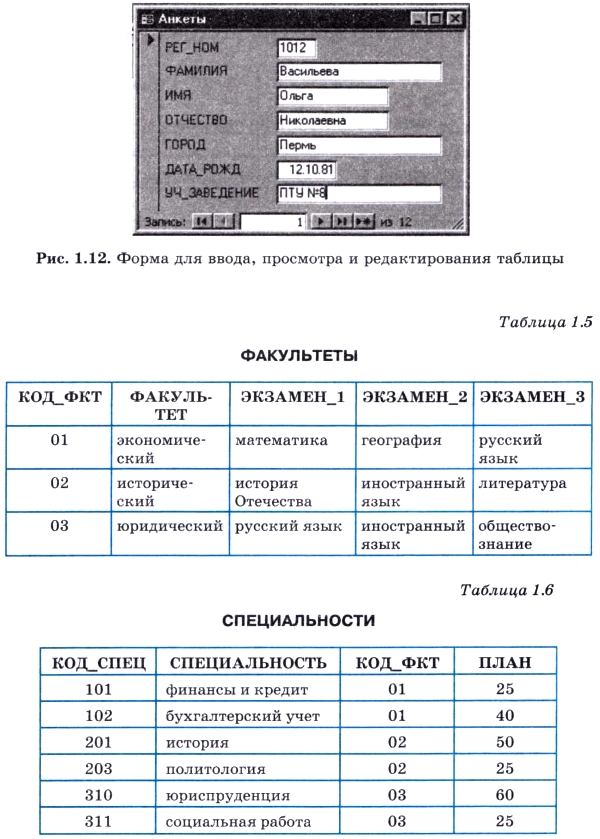
На первом этапе в каждой таблице определяются имена полей, их типы и форматы. Совсем не обязательно все таблицы БД должны быть построены одновременно. В нашем примере на начальном этапе работы приемной комиссии могут быть созданы таблицы ФАКУЛЬТЕТЫ и СПЕЦИАЛЬНОСТИ. Структуры этих таблиц представлены в табл. 1.3 и 1.4.



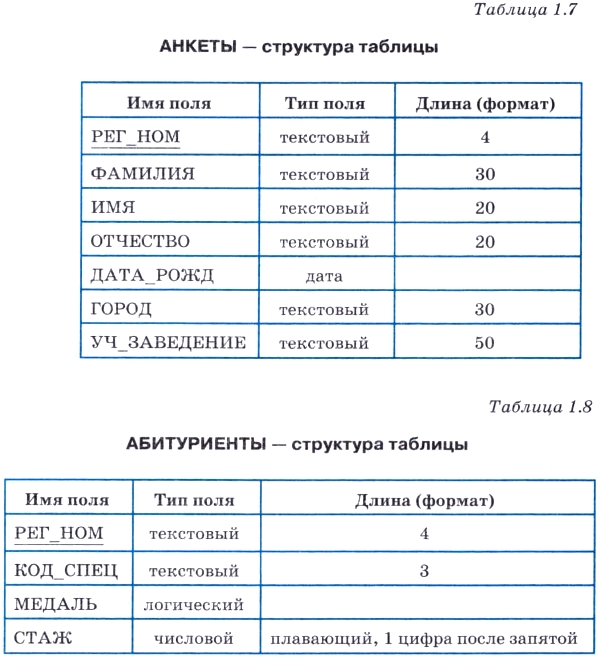
Затем средствами СУБД устанавливаются связи между таблицами через общее поле КОД ФКТ.

После этого таблицы можно заполнять данными. Современные СУБД предоставляют пользователю удобные средства ввода. Данные можно вводить непосредственно в строки таблиц, отражаемых на экране, или через диалоговые окна — формы (рис. 1.12). В процессе ввода данных СУБД осуществляет автоматический контроль соответствия вводимых данных объявленным типам и форматам полей.

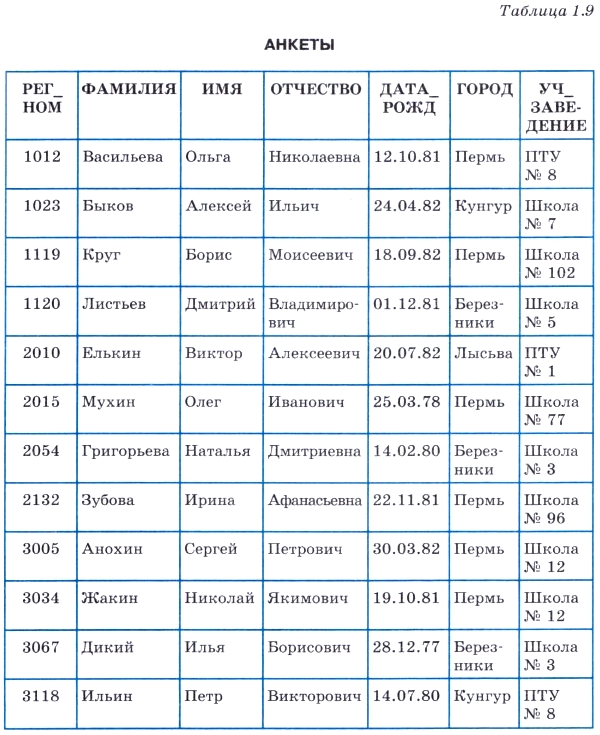
В таблице 1.5 приведены первые три записи таблицы ФАКУЛЬТЕТЫ, а в табл. 1.6 — шесть записей таблицы СПЕЦИАЛЬНОСТИ.

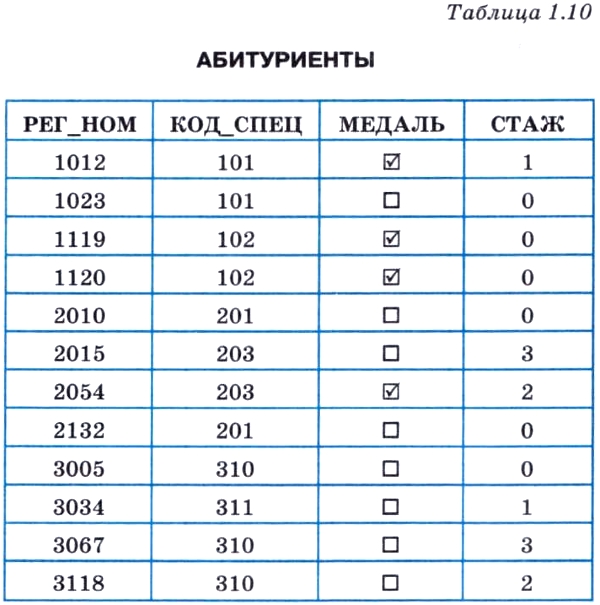


На этапе приема документов в базу данных будут добавлены таблицы АНКЕТЫ и АБИТУРИЕНТЫ. Их структуры представлены в табл. 1.7 и 1.8.

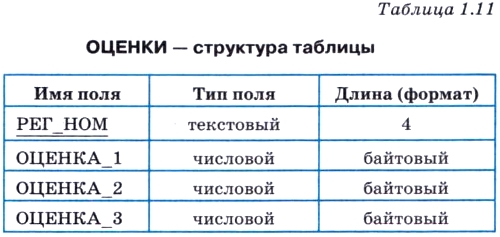


После установки связей таблицы будут заполняться данными. Первые двенадцать записей в этих таблицах приведены в табл. 1.9 и 1.10.



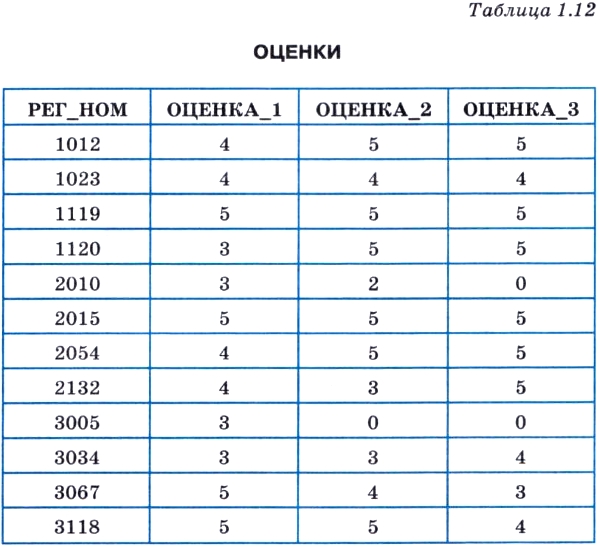


Когда начнутся приемные экзамены, понадобится таблица ОЦЕНКИ. Опишем ее структуру в табл. 1.11.

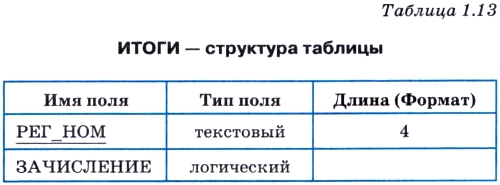


Тип «байтовый» является разновидностью типа «целый». Он применяется для целых положительных чисел в диапазоне от О до 255 и занимает в памяти 1 байт. Поскольку оценки принимают значения от 2 до 5, этот тип оказывается наиболее «экономным».

А вот какой вид (после установки связи с таблицей АБИТУРИЕНТЫ и ввода данных) примет таблица с результатами сдачи экзаменов перечисленными выше двенадцатью абитуриентами — табл. 1.12 (здесь 0 — неявка на экзамен).



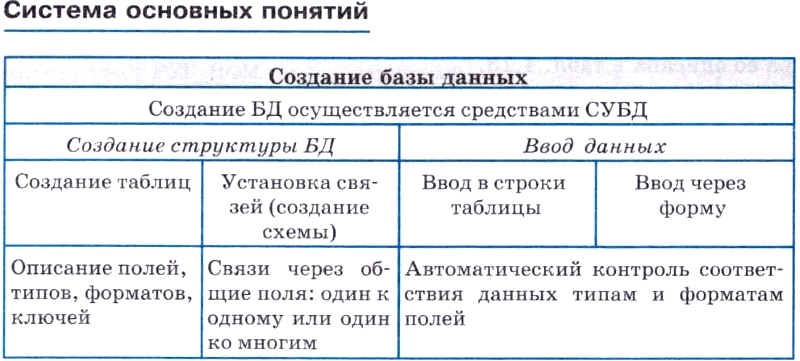
И наконец, осталось создать таблицу ИТОГИ для занесения в нее результатов зачисления абитуриентов в университет. Структура ее описана в табл. 1.13.



Содержание таблицы приведено в табл. 1.14.



Логические значения поля ЗАЧИСЛЕНИЕ первоначально отмечаются пустыми квадратиками, обозначающими ЛОЖЬ («нет») (значение по умолчанию логического поля — ЛОЖЬ). После объявления итогов для принятых абитуриентов это значение будет заменено на значение ИСТИНА («да») — будет выставлена галочка. Осталось подключить эту таблицу к схеме через поле РЕГ\_НОМ.

   
**Вопросы и задания**

1. Что нужно иметь для того, чтобы начать процесс создания базы данных?

2. Какую информацию нужно указать СУБД для создания таблиц БД?

3. Каким способом можно вводить данные в таблицы?

4. Как СУБД помогает пользователю производить безошибочный ввод данных?

Литература

1. Семакин И.Г., Хеннер Е.К., Шейна Т.Ю. Информатика, учеб, 11 кл. базовый уровень/ И.Г. Семакин, Е.К. Хеннер, Т.Ю. Шейна изд.-М.: Бином. Лаборатория знаний, 2015.-264 с.
2. **Электронные источники**

* Видеоурок <https://youtu.be/RxQyZyM0vC4>
* Презентация <http://иванов-ам.рф/informatika_11_sim/ur_06/ur_06_06.swf>