

Информационные системы

Информационные системы – это совокупность тем или иным способом структурированных данных (базы данных) и комплекса аппаратно-программных средств для хранения, поиска, обработки и передачи значительных объемов информации, имеющей определенную практическую сферу применения.



Объект - это нечто существующее и различимое, для которого существуют название и способ отличить один подобный объект от другого.

Объектами могут быть не только материальные предметы, но и более абстрактные понятия, например, события, произведения искусства, философские теории и т.д.
Группа всех подобных объектов образует набор объектов. Конкретный объект в такой группе - экземпляр объекта.

Информационная система оперирует наборами объектов, используя при этом конкретные значения атрибутов (данных) о тех или иных объектах.

Атрибут (или данное, или свойство) – это показатель, который характеризует объект и принимает для конкретного экземпляра объекта числовое, текстовое или иное значение.

Атрибут набора объектов сам может быть набором объектов, имеющим собственные атрибуты.

Классификаторами (справочниками, словарями) – списки возможных значений атрибутов.

Повторяющаяся группа – совокупность (группа) однотипных данных, характеризующих экземпляр объекта, объем которых различен для каждого экземпляра.

Иногда для конкретного объекта одно и то же данное может принимать несколько значений. Такие данные образуют так называемые повторяющиеся группы. Например, премии ученого, увлечения человека, заболеваемость сотрудника.

Базы данных

Основа информационной системы, объект ее обработки - **база данных**.

База Данных (в общем смысле) - совокупность сведений о конкретных объектах реального мира в какой-либо предметной области или ее разделе.

(Шафрин Ю.А. Основы компьютерной технологии, 1998)

База Данных - это один или несколько файлов данных, предназначенных для хранения, изменения и обработки больших объемов взаимосвязанной информации.

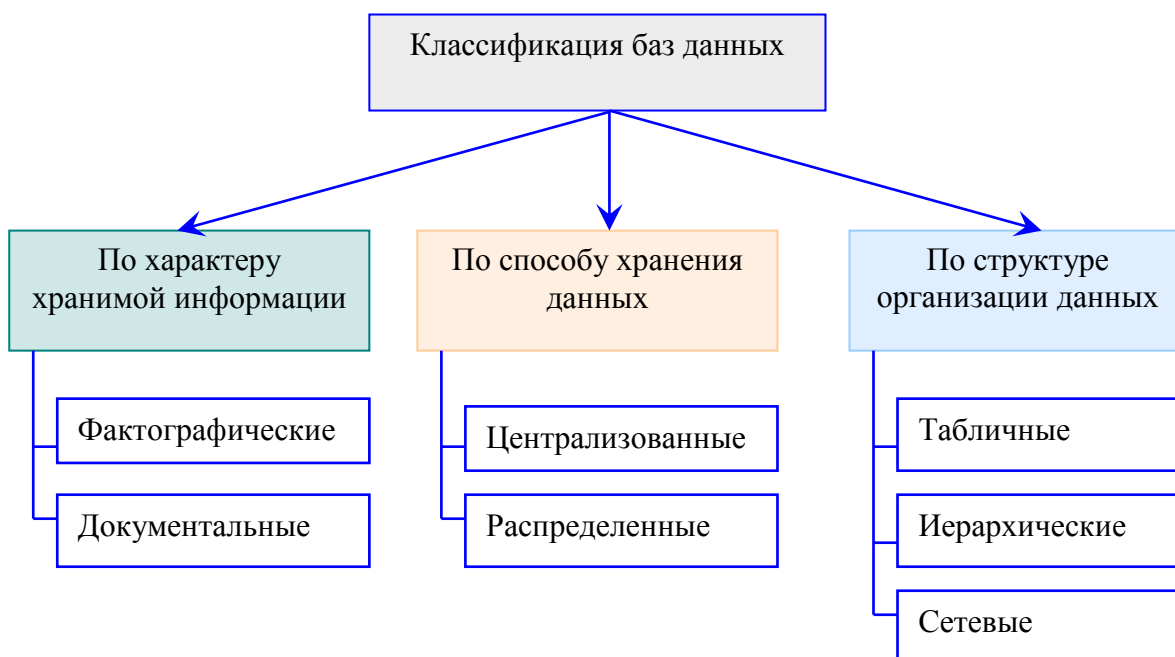
(Шауцукова Л.З. Информатика для 10-11кл, 2002)

База Данных – это информационная модель, позволяющая упорядоченно хранить данные о группе объектов, обладающих одинаковым набором свойств.

(Угринович Н.Д. Информатика и Информационные технологии для 10-11кл, 2001)

База Данных – организованная совокупность данных, предназначенная для длительного хранения во внешней памяти ЭВМ, постоянного обновления и использования.

(Ершов А.П. Школьный словарь по информатике)



Фактографические база данных – картотеки; краткая информация в строго определенном формате.

Документальные база данных – всевозможные документы (текстовые, графические, видео, звук).

Централизованные база данных – вся информация хранится на одном компьютере.

Распределенные база данных – информация может храниться на нескольких компьютерах (используются в локальных, глобальных сетях).

Геоинформационные системы – (Гео – земля; Земля + Информация) - системы обеспечивают наглядное представление различных параметров земной поверхности в форме соответствующих карт.

Синоним термина «база данных» - «банк данных».

Создавая базы данных, мы стремимся обеспечить возможность:

- упорядочивать информацию по различным признакам;
- быстро извлекать выборки с произвольным сочетанием признаков.

При использовании компьютера для этого информация должна быть структурирована.

Структурирование - это введение каких-то соглашений о способах представления данных.

Текстовые файлы (и документы WORD) содержат «неструктурированную» или в лучшем случае «плохо структурированную» информацию, не пригодную для эффективной обработки автоматом-компьютером.

Системы управления БД (СУБД)

СУБД - система программ, позволяющая создавать базы данных, обновлять хранимую в ней информацию, обеспечивать удобный к ней доступ с целью просмотра и поиска.

СУБД – это система программного обеспечения, позволяющая обрабатывать обращения к базам данных, поступающие от прикладных программ конечных пользователей. (Шауцукова Л.З. Информатика для 10-11кл, 2002)

СУБД – это программа, позволяющая создавать БД, а также обеспечивающая обработку (сортировку) и поиск данных.
(Угринович Н.Д. Информатика и Информационные технологии для 10-11кл, 2001)

Режимы работы с СУБД:

- создание;
- редактирование (замена, вставка, удаление записей);
- поиск;
- манипулирование (создание копий, сортировка по заданному признаку и др).

СУБД должна обеспечить:

- возможность внесения и чтения информации;
- работу с большим объемом данных;
- редактирование содержания базы данных;
- редактирование структуры базы данных;
- быстроту поиска информации;
- сортировку данных;
- целостность данных, их непротиворечивость;
- защиту от разрушения, уничтожения, несанкционированного доступа;
- систему «дружественных» подсказок.

Реляционные базы данных

Можно доказать, что любую структуру данных можно преобразовать в простую двумерную таблицу. Такое представление является наиболее удобным и для пользователя, и для машины.

Реляционная база данных - совокупность данных состоящих из *связанных* двумерных таблиц.

Название произошло от английского «relation» - отношение.

Номер	Имя абонента	Адрес
233-48-19	Петров Евгений	Садовая ул., 18
265-04-15	Дядя Коля	Зеленая ул., 45-2-56
570-14-50	Химчистка	Киевская ул., 123

Основная идея реляционного подхода –

представить произвольную структуру данных в виде простой двумерной таблицы. Такой процесс называется **нормализацией** структуры.

Основные понятия реляционных баз данных

1. Любые совокупности данных представляются в виде **двумерных таблиц**, каждая из которых содержит информацию об объектах определенного типа.
2. Каждая таблица состоит из **фиксированного числа столбцов** и **переменного числа строк**.
3. **Запись** – строка таблицы.
Каждая запись содержит информацию об отдельном экземпляре объекта.
4. **Поле** – столбец таблицы.
Каждый столбец представляет конкретное данное – одну характеристику объекта (атрибут). Для каждого поля разработчик должен определить:
 - уникальное имя поля;
 - тип поля;
 - дополнительные характеристики (длину, формат) поля.
5. Описание полей, определяемое разработчиком, называется структурой таблицы.
6. Каждое поле может входить в несколько таблиц.
7. Изменение количества полей и (или) их типов является особой операцией.

Кодирование информации

В информационных системах вместе с наименованиями различных объектов часто используют их цифровые или буквенные обозначения, называемые **кодами** (или **шифрами**).

! Коды должны быть уникальны.

Использование кодов решает две проблемы:

- уменьшает объем вводимой и хранимой информации;
(если поле входит не в одну, а в несколько таблиц)
- устраняет путаницу в написании одних и тех же значений разными способами.

При использовании кодов в базу данных обычно включают таблицы, условно именуемые классификаторами (словарями, справочниками), в которых эти коды расшифровываются. Словари формально не отличаются от любой другой таблицы в базе данных. В словаре можно указать не только наименование, но и другие более или менее постоянные данные объекта.

Таблица **УЧЕНИКИ**

Код ученика	Фамилия	Имя	Отчество	Класс
94-13	Орлов	Сергей	Петрович	11Б
95-07	Петров	Алексей	Иванович	10А
94-11	Викторов	Алексей	Юрьевич	11А
96-21	Воронов	Виктор	Викторович	9Б

Таблица **ОЦЕНКИ**

Код ученика	Код предмета	Оценка	Дата
94-13	МАТ	4	03.11.04
95-07	МАТ	5	04.11.04
94-11	ФИЗ	3	04.11.04
94-13	МАТ	5	04.11.04
94-13	ИНФ	3	05.11.04

Таблица **ПРЕДМЕТЫ**

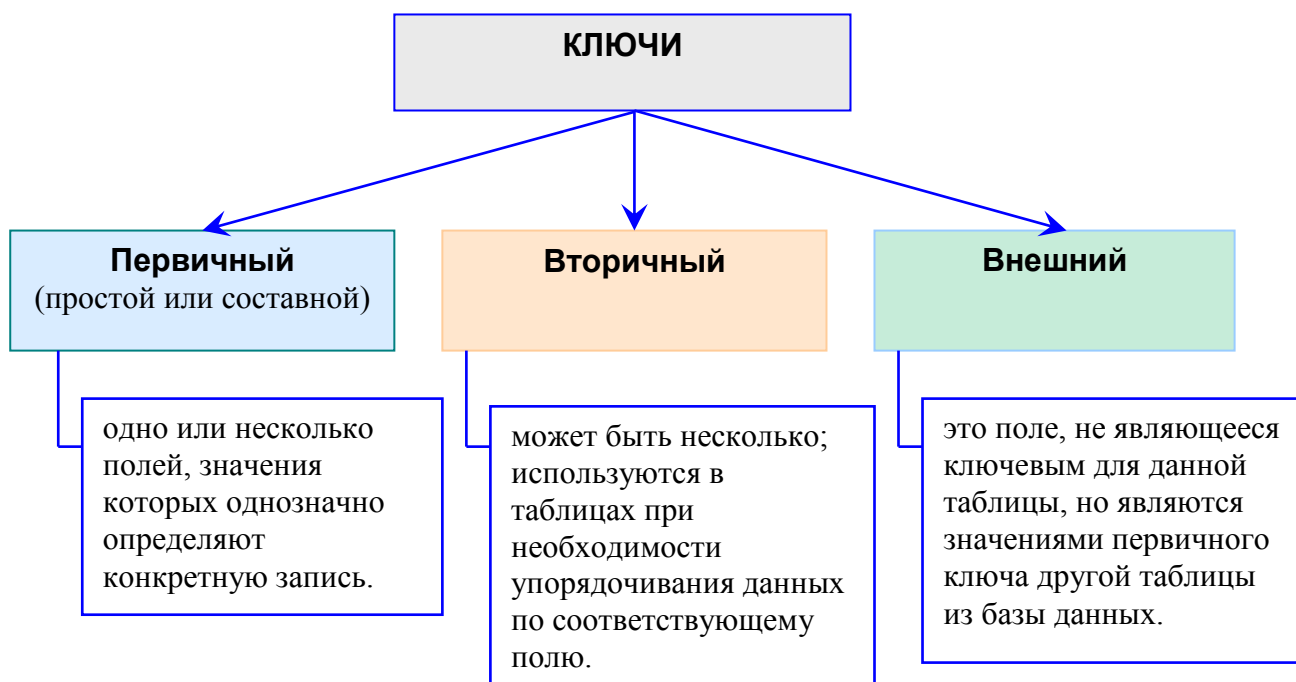
Код предмета	Название предмета
МАТ	Математика
ФИЗ	Физика
ИНФ	Информатика
РУС	Русский язык
ЛИТ	Литература

Значность кода, т.е. количество позиций в коде, зависит от количества возможных значений данного объекта.

Как и какие данные следует кодировать, зависит и от характера данных, и от целей и возможностей разработчика. В современных системах часто предусматривается автоматическое кодирование данных.

Ключи таблицы

Ключ – одно или несколько полей для идентификации записей таблицы.



Первичный ключ

Простой ключ



У разных книг могут совпадать значения полей, но инвентарный номер у каждой книги свой.

Таблица КНИГИ

Номер	Автор	Название	Год	Полка
001	Семакин И.Г.	Структурированный конспект базового курса	2001	2
002	Угринович Н.В.	Информатика и информационные технологии	2001	3
003	Окулов С.В.	Основы программирования	2002	4

Составной ключ



У разных записей одновременно не могут совпадать только сочетание двух полей: город и номер школы.

Таблица ШКОЛЫ

Город	Номер школы	Директор	Адрес школы	Телефон
Москва	444	Крючкова И.И.	Нижняя Первомайская, 14	465-23-52
Тверь	135	Григорьев С.В.	Пушкинская, 34	3-75-45
Пермь	143	Сысоева Т.И.	Мира, 78	47-30-12

Каждая запись в таблице должна иметь **первичный ключ**.

Свойства первичного ключа

1. **Уникальность записи**, т.е. запись должна однозначно определяться значением ключа.
2. **Отсутствие избыточности**, т.е. никакое поле нельзя удалить из ключа, так, чтобы он не потерял уникальности.

Вторичные ключи

- Значение простого ключа может быть *неуникальным*.
- Если первичный ключ в таблице всегда только один, вторичных (простых) ключей может быть много.
- Понятие вторичного (простого) ключа используются в таблицах обычно при необходимости упорядочивания данных по соответствующему полю.

Внешний ключ –

это поле, не являющееся ключевым для данной таблицы значениями первичного ключа другой таблицы из базы данных.

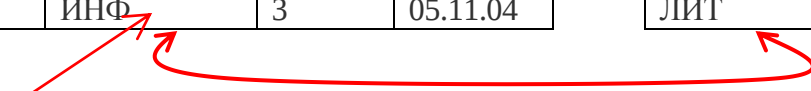
Таблица **ОЦЕНКИ**

Код ученика	Код предмета	Оценка	Дата
94-13	МАТ	4	03.11.04
95-07	МАТ	5	04.11.04
94-11	ФИЗ	3	04.11.04
94-13	МАТ	5	04.11.04
94-13	ИНФ	3	05.11.04

Таблица **ПРЕДМЕТЫ**

Код предмета	Название предмета
МАТ	Математика
ФИЗ	Физика
ИНФ	Информатика
РУС	Русский язык
ЛИТ	Литература

Внешний ключ



Достоверность информации

Поскольку первичное заполнение таблиц и ввод их в машину ведет человек, ошибки в данных являются не исключением, а правилом, и любая информационная система должна иметь средства для диагностики и исправления ошибок.

Данные как в одной таблице, так и в разных таблицах логически связаны между собой.

Логические (семантические) ошибки – нарушение логической взаимосвязи; смысловые ошибки, которые могут быть обнаружены аппаратом формального логического контроля, построенным для информационных систем.

Конкретная информационная система может иметь собственные средства дополнительного («нестандартного») контроля, так как стандартные средства не могут охватить все возможные случаи.

Современные СУБД имеют средства поддержания **целостности (согласованности)** данных, которые не позволят вам ввести строку с неправильными кодами.

В современных информационных системах можно указать условия, которым должны удовлетворять значения некоторых полей (условия верификации данных).

Например, один из критериев верификации для шахматной базы данных:
код белых не может быть равен коду черных (партнер не может играть сам с собой).

Гораздо сложнее обстоит дело с ошибками в **допустимых** значениях данных.

Например, если в той же таблице партий вместо кода Ботвинника вы указали код Каспарова, компьютер обнаружить эту ошибку не в состоянии.

Связь таблиц: главная и подчиненная таблицы

После создания различных таблиц, содержащих данные, относящиеся к базе данных, необходимо обеспечить целостность базы данных. Для этого необходимо **связать** таблицы между собой.

Таблица **УЧЕНИКИ**

Код ученика	Фамилия	Имя	Отчество	Класс
94-13	Орлов	Сергей	Петрович	11Б
95-07	Петров	Алексей	Иванович	10А
94-11	Викторов	Алексей	Юрьевич	11А
96-21	Воронов	Виктор	Викторович	9Б

Таблица **ОЦЕНКИ**

Код ученика	Код предмета	Оценка	Дата
94-13	МАТ	4	03.11.04
95-07	МАТ	5	04.11.04
94-11	ФИЗ	3	04.11.04
94-13	МАТ	5	04.11.04
94-13	ИНФ	3	05.11.04

Таблица **ПРЕДМЕТЫ**

Код предмета	Название предмета
МАТ	Математика
ФИЗ	Физика
ИНФ	Информатика
РУС	Русский язык
ЛИТ	Литература

Таблица **ОБЩИЕ СВЕДЕНИЯ ОБ УЧЕНИКЕ**

Код ученика	Отец	Мать	Дом. телефон	Адрес
94-13	Орлов П.Ф.	Орлова Т.Е.	234-67-89	Садовая ул., 23-67
95-07	Петров И.К.	Алексеева Т.Т.	123-78-34	Зеленая ул., 123/34, 24
94-11	Викторов Ю.С.	Викторова О.С.	765-23-01	Рабочая ул., 67-123
96-21	Воронов В.Г.	Воронова Д.С.	223-74-05	Строительная ул., 34-11

Схема связи таблиц

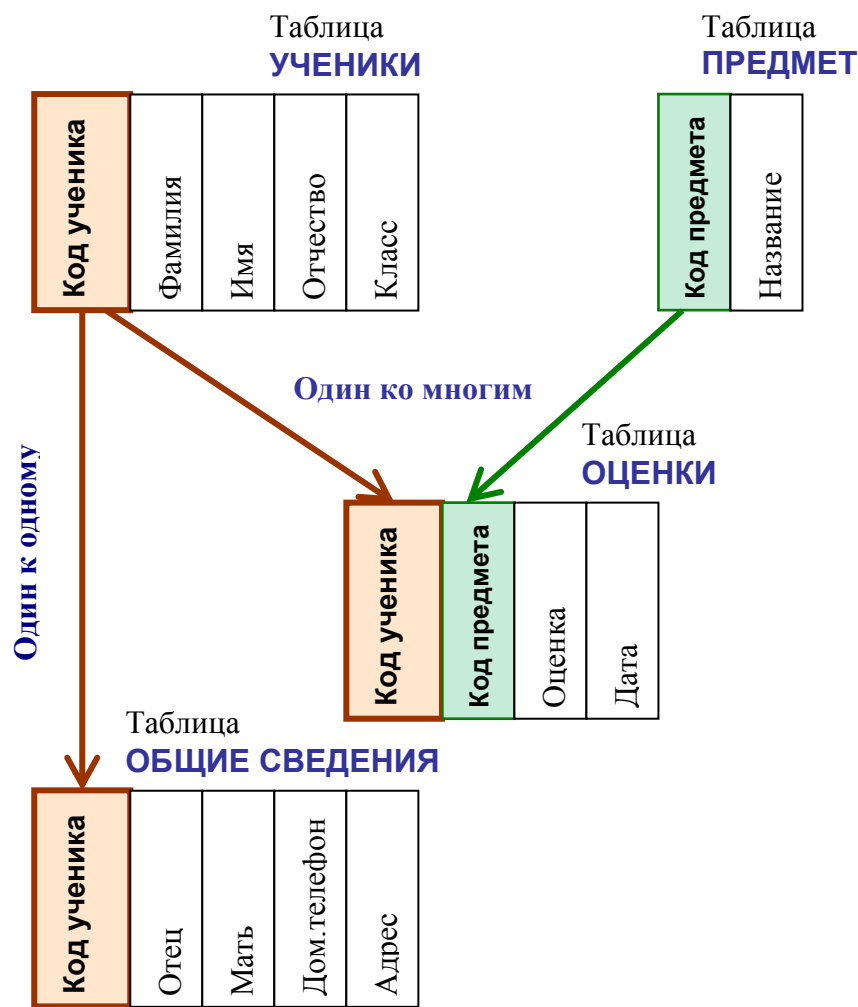


Таблица ОЦЕНКИ связана с таблицей УЧЕНИКИ по полю *Код ученика*, а с таблицей ПРЕДМЕТЫ по полю *Код предмета*.

Составляя таблицу УЧЕНИКИ, мы можем указать в ней и тех учеников, для которых в ОЦЕНКИ пока нет записей (например, если ученик болел и его еще не спрашивали на уроках).

Обратное неверно: не имеет смысла включать в таблицу ОЦЕНКИ оценки для ученика, которого мы еще не внесли в таблицу учеников.

В паре УЧЕНИКИ – ОЦЕНКИ первая таблица считается главной, а вторая — подчиненной. Поле *Код ученика* для первой таблицы является **первичным** ключом, а для второй таблице — **внешним** ключом.

Таблица ОЦЕНКИ имеет собственный первичный ключ, состоящий из трех полей.

Назовите эти поля (код ученика, код предмета, дата).

Для пары связанных таблиц (одна из которых главная, а другая — подчиненная) всегда определен один из двух взаимоисключающих типов связи:

«Один-ко-Многим» или «Один-к-Одному».

Отношение «Один – ко – Многим»

Каждому значению первичного ключа в *главной таблице* соответствует **одна, несколько или ни одной** записи в подчиненной таблице.

В нашем случае в паре УЧЕНИК-ОЦЕНКИ, в зависимости от того, сколько оценок получил данный ученик.

В паре ПРЕДМЕТЫ-ОЦЕНКИ первая таблица является главной, а вторая — подчиненной. И здесь мы видим отношение (связь) «Один-ко-Многим».

Отношение «Один-ко-Многим» - наиболее распространенный тип связи между двумя таблицами в реляционных БД.

Отношение «Один – к – Одному»

Каждому значению первичного ключа в *главной таблице* соответствует **одна или ни одной** записи в подчиненной таблице.

Например, в кадровых системах и службах безопасности часто используют *несколько* таблиц для хранения разных данных о работниках. В главной таблице хранится общая (формальная и несекретная) информация о всех лицах: табельный номер, фамилия, дата рождения и т. п. В подчиненные таблицы включают особую (или секретную) информацию (например, номер страхового полиса, имя пользователя компьютерной сети и т. п.), и число записей в таких таблицах может быть меньше (но не больше!) числа записей главной таблицы. Например, не каждый человек из главной таблицы — пользователь сети, но каждый пользователь сети обязательно должен входить в главную таблицу.

Существует также отношение «Много-ко-Многим»

Поддержка целостности данных

Целостность данных – согласованность данных между связанными таблицами в реляционных базах данных.

Механизм поддержки целостности данных обеспечивает выполнение работы с данными с учетом следующих правил:

1. Невозможно ввести в связанное поле подчиненной таблицы значение, которое отсутствует в связанном поле главной таблицы.
2. Не допускается удаление записи из главной таблицы, если существуют связанные с ней записи в подчиненной таблице.
3. Невозможно изменить значение ключевого поля в главной таблице, если существуют записи, связанные с данной таблицей.

Чтобы преодолеть ограничения на удаление или изменение связанных записей, сохраняя при этом целостность данных, следует поручить СУБД выполнение операций *Каскадное обновление* и/или *Каскадное удаление связанных полей*.

Операция *Каскадное удаление* – при удалении записи в главной таблице автоматически удаляются и все связанные записи в подчиненной таблице.

Операция *Каскадное обновление* – при изменении ключевого поля главной таблицы автоматически будут изменены и соответствующие значения поля связанных записей.

Проблемы реляционного подхода

Основные последствия неправильного проектирования:

- избыточность информации;
- противоречивость информации;
- потеря целостности (взаимосвязь между данными).

Можно доказать, что любую структуру данных можно преобразовать в простую двумерную таблицу.

Процесс превращения любой структуры данных в реляционную структуру называется **нормализацией**.

Технология проектирования баз данных, разработана в 70-х годах Е.Коддом.

Основные правила

Правило 1: Каждое поле таблицы должно представлять уникальный тип информации.
Т.е. необходимо избавиться от повторяющихся полей (групп) и разделить составные поля на отдельные элементы данных.

Правило 2: Каждая таблица должна иметь уникальный идентификатор (первичный ключ), который может состоять из одного или нескольких полей.

Правило 3: В таблице не должно быть данных, не относящихся к объекту, определяемому первичным ключом.

Правило 4: Независимость полей.
Т.е. возможность изменять значения любого поля (не входящего в первичный ключ) без воздействия на данные других полей.

Задание

Пусть некая фирма занимается оптовой торговлей, например кондитерскими изделиями. Покупателями (клиентами) фирмы являются различные кафе, рестораны, клубы и т.п. Для учета и анализа заказов в фирме создается БД, содержащая следующие поля:

Имя поля	Тип поля	Примечание
Номер заказа	Число	
Дата поставки	Дата	ДД.ММ.ГГ
Код клиента	Число	Можно текст
Наименование клиента	Текст	
Адрес клиента (город, улица, дом)	Текст	
Код продукта	Число	Можно текст
Наименование продукта	Текст	
Количество	Число	кг
Цена	Число	Руб./кг
Стоимость	Число	Руб.

Каждая строка этой таблицы содержит информацию о конкретном заказе.

Правило №1

выполняется частично – повторяющихся групп нет, но есть составное поле Адрес клиента (город, улица, дом).

Имя поля	Тип поля	Примечание
Номер заказа	Число	
Дата поставки	Дата	ДД.ММ.ГГ
Код клиента	Число	Можно текст
Наименование клиента	Текст	
Адрес клиента (город)	Текст	
Адрес клиента (улица, дом)	Текст	
Код продукта	Число	Можно текст
Наименование продукта	Текст	
Количество	Число	кг
Цена	Число	Руб./кг
Стоимость	Число	Руб.

Правило №2

выполняется – первичным ключом таблицы является *Номер заказа* (если нумерация заказов сквозная, но можно использовать составной ключ *Номер заказа – Дата поставки*).

Правило №3

не выполняется – в таблице присутствуют поля, которые непосредственно не связаны с первичным ключом таблицы.

Имя поля	Тип поля	Примечание
Номер заказа	Число	
Дата поставки	Дата	ДД.ММ.ГГ
Код клиента	Число	Можно текст
Наименование клиента	Текст	
Адрес клиента (город)	Текст	
Адрес клиента (улица, дом)	Текст	
Код продукта	Число	Можно текст
Наименование продукта	Текст	
Количество	Число	кг
Цена	Число	Руб./кг
Стоимость	Число	Руб.

Значение Адреса клиента не зависит от ключа нашей таблицы – номера заказа, а зависит от значения кода клиента. То же самое относится к полю Наименование клиента. Поэтому эти поля нужно удалить из таблицы и включить их в таблицу-классификатор, содержащей сведения о клиентах.

Аналогично, Наименование продукта и его Цена зависят от кода продукта, но не зависят от номера заказа. Поэтому их следует поместить в другую таблицу – классификатор (справочник продуктов).

Таблица ЗАКАЗЫ

Имя поля	Тип поля	Примечание
Номер заказа	Число	
Дата поставки	Дата	ДД.ММ.ГГ
Код клиента	Число	Можно текст
Код продукта	Число	Можно текст
Количество	Число	кг
Стоимость	Число	Руб.

Таблица КЛИЕНТ

Имя поля	Тип поля	Примечание
Код клиента	Число	Можно текст
Наименование клиента	Текст	
Адрес клиента (город)	Текст	
Адрес клиента (улица, дом)	Текст	

Таблица ПРОДУКТЫ

Имя поля	Тип поля	Примечание
Код продукта	Число	Можно текст
Наименование продукта	Текст	
Цена	Число	Руб./кг

Правило №4 – поля должны быть независимы. Поле *Стоимость* – это произведение цены на количество, а поэтому это поле вообще не следует включать в таблицы и хранить его значение: система просто вычислит стоимость заказа при необходимости.

Таблица ЗАКАЗЫ

Имя поля	Тип поля	Примечание
Номер заказа	Число	
Дата поставки	Дата	ДД.ММ.ГГ
Код клиента	Число	Можно текст
Код продукта	Число	Можно текст
Количество	Число	кг

Таблица КЛИЕНТ

Имя поля	Тип поля	Примечание
Код клиента	Число	Можно текст
Наименование клиента	Текст	
Адрес клиента (город)	Текст	
Адрес клиента (улица, дом)	Текст	

Таблица ПРОДУКТЫ

Имя поля	Тип поля	Примечание
Код продукта	Число	Можно текст
Наименование продукта	Текст	
Цена	Число	Руб./кг

Теперь обратим внимание на то, что фактически в нашей задаче учета заказов введено ограничение – одновременно фирма-покупатель в одном заказе может оформить покупку только одного наименования продукта: если нужно заказать продуктов больше, необходимо оформить несколько отдельных заказов, а это, конечно, не удобно.

Если же снять это ограничение, то получается, что количество различных наименований продуктов в одном заказе в общем случае может быть различным, т.е. поля *Код продукта* и *Количество* являются **повторяющейся группой**. Фактически для каждого заказа количество этих полей должно быть различным. Но в реляционной БД количество полей в таблице **фиксированно**, и по Правилу №1 в таблице не должно быть повторяющихся полей.

Задача решается просто – для повторяющихся групп заводится отдельная таблица, причем для каждого элемента группы отводится одна запись со своим ключом. В нашем случае эта таблица будет содержать сведения о содержимом каждого заказа, т.е. какие продукты нужно поставить заказчику одновременно в определенную дату. Например, если в заказе с номером 127 заказано три продукта, то в этой таблице **СОДЕРЖИМОЕ** появится три записи с одним номером заказа 127, при этом в каждой из этих записей будет различный код продукта и соответствующее заказанное количество этого продукта.

Таким образом, из таблицы **ЗАКАЗЫ** выделяется еще одна таблица **СОДЕРЖИМОЕ**, в которой будут следующие поля: *Номер заказа*, *Код продукта*, *Количество*.

В итоге после нормализации исходных данных мы получили четыре таблицы:

Таблица ЗАКАЗЫ

Имя поля	Тип поля	Примечание
Номер заказа	Число	
Дата поставки	Дата	ДД.ММ.ГГ
Код клиента	Число	Можно текст

Таблица СОДЕРЖИМОЕ ЗАКАЗА

Имя поля	Тип поля	Примечание
Номер заказа	Число	
Дата поставки	Дата	ДД.ММ.ГГ
Код продукта	Число	Можно текст
Количество	Число	кг

Таблица КЛИЕНТ

Имя поля	Тип поля	Примечание
Код клиента	Число	Можно текст
Наименование клиента	Текст	
Адрес клиента (город)	Текст	
Адрес клиента (улица, дом)	Текст	

Таблица ПРОДУКТЫ

Имя поля	Тип поля	Примечание
Код продукта	Число	Можно текст
Наименование продукта	Текст	
Цена	Число	Руб./кг

Задача

Создать базу данных Поликлиника, которая должна содержать сведения о посещении пациентами поликлиники своего участкового врача. Вся необходимая информация приведена в таблице.

ФИО пациента	Дата рождения	Адрес	Номер участка	ФИО врача	Дата посещения	Диагноз
Лосев О.И.	20.04.65	Лесная ул., 34-17	1	Петрова К.В.	11.10.2004	Грипп
Орлова Т.Н.	15.09.70	Парковая ул., 12-4-56	2	Антонова Т.Р.	11.10.2004	ОРВИ
Быкова Н.Ф.	13.12.77	Мира пр., 23-78	3	Титов Н.А.	12.10.2004	Стенокардия
Дуров С.С.	19.05.60	Лесная ул. 36-12	1	Петрова К.В.	13.10.2004	Гастрит
Орлова Т.Н.	15.09.70	Парковая ул., 12-4-56	2	Антонова Т.Р.	14.10.2004	Гастрит

Таблица ПАЦИЕНТ

Имя поля
Код пациента
Фамилия
Имя
Отчество
Дата рождения
Код улицы
Дом
Квартира
Код участка

Таблица ВРАЧ

Имя поля
Код врача
Фамилия
Имя
Отчество

Таблица УЧАСТОК

Имя поля
Код участка
Код врача

Таблица УЛИЦЫ

Имя поля
Код улицы
Название улицы

Таблица ДИАГНОЗЫ

Имя поля
Код диагноза
Наименование диагноза

Таблица СВЕДЕНИЯ О ПОСЕЩЕНИИ ВРАЧА

Имя поля
Код пациента
Код участка
Дата посещения
Код диагноза

Самостоятельная работа Тема «Базы данных» Вариант № 1 Вопросы 1. Что такое База данных? 2. Основная идея реляционного подхода. Задача Создать базу данных Видеотека, которая должна содержать сведения о конкретной кассете. Примерный набор атрибутов для каждой кассеты: 1) Номер кассеты, 2) Название видеоматериала, 3) Год создания, 4) Страна, 5) Рубрика (исторический фильм, фантастика и т.д.), 6) Фамилия режиссера, 7) Фамилия исполнителя главной роли, 8) Уровень (отлично, хорошо, средне и т.д.), 9) Краткое содержание, 10) Длительность, 11) Где находится и т.д.	Самостоятельная работа Тема «Базы данных» Вариант № 2 Вопросы 1. Что такое реляционная База данных? 2. Основные понятия реляционной базы данных: запись, поле, ключ. Задача Создать базу данных Аудиотека, которая должна содержать сведения о конкретном диске. Примерный набор атрибутов для каждого диска: 1) Номер диска, 2) Название диска, 3) Год выпуска, 4) Название произведений, 5) Страна, 6) Автор музыки и слов, 7) Фамилия исполнителя или название группы, 8) Уровень (отлично, хорошо, средне и т.д.), 9) Длительность, 10) Где находится и т.д.
Самостоятельная работа Тема «Базы данных» Вариант № 1 Вопросы 1. Что такое База данных? 2. Основная идея реляционного подхода. Задача Создать базу данных Видеотека, которая должна содержать сведения о конкретной кассете. Примерный набор атрибутов для каждой кассеты: 1) Номер кассеты, 2) Название видеоматериала, 3) Год создания, 4) Страна, 5) Рубрика (исторический фильм, фантастика и т.д.), 6) Фамилия режиссера, 7) Фамилия исполнителя главной роли, 8) Уровень (отлично, хорошо, средне и т.д.), 9) Краткое содержание, 10) Длительность, 11) Где находится и т.д.	Самостоятельная работа Тема «Базы данных» Вариант № 2 Вопросы 1. Что такое реляционная База данных? 2. Основные понятия реляционной базы данных: запись, поле, ключ. Задача Создать базу данных Аудиотека, которая должна содержать сведения о конкретном диске. Примерный набор атрибутов для каждого диска: 1) Номер диска, 2) Название диска, 3) Год выпуска, 4) Название произведений, 5) Страна, 6) Автор музыки и слов, 7) Фамилия исполнителя или название группы, 8) Уровень (отлично, хорошо, средне и т.д.), 9) Длительность, 10) Где находится и т.д.