

Таблица 5.1. Классы автоматизированных ИС

1. Отрасли применения					
Деловая информация	Профессиональная информация	Потребительская информация	Электронная коммерция		
2. Вид и структура информации					
Библиографическая	Полнотекстовая	Справочная	Табличная	Графическая	
3. Взаимодействие с пользователями					
Системы разделения времени		Поддержка групповых решений			
4. Типы принимаемых решений					
Степень неопределенности		Уровень руководства			
5. Масштаб АИС					
АИС организации	Локальные АИС		Глобальные АИС		
6. Тип организации					
Производство	Учреждение	Библиотека	Вуз	Госпитали	Прочие организации
7. Программное обеспечение					
СУБД	АИПС	Универсальные оболочки	Программируемые среды		
8. Тип интерфейса с пользователем					
Командный язык	Шаблоны	Меню	Оконный		
9. Наличие и тип моделей, встраиваемых в АИС					
СОПР, КИС (EPR/BRP)	САПР	ЭС	ГИС		

Определение АИС

- АИС комплекс автоматизированных информационных технологий, предназначенный для *информационного обслуживания* — организованного непрерывного технологического процесса подготовки и выдачи потребителям научной, управленческой и др. информации, используемой для принятия решений, в соответствии с нуждами для поддержания эффективной деятельности



Классификации БД по моделям данных



- иерархические,
- сетевые,
- реляционные
- объектно-ориентированные-данные моделируются в виде объектов, их атрибутов, методов и классов.

- **Иерархические базы данных.** Иерархические базы данных графически могут быть представлены как перевернутое дерево, состоящее из объектов различных уровней. Верхний уровень (корень дерева) занимает один объект, второй - объекты второго уровня и так далее.

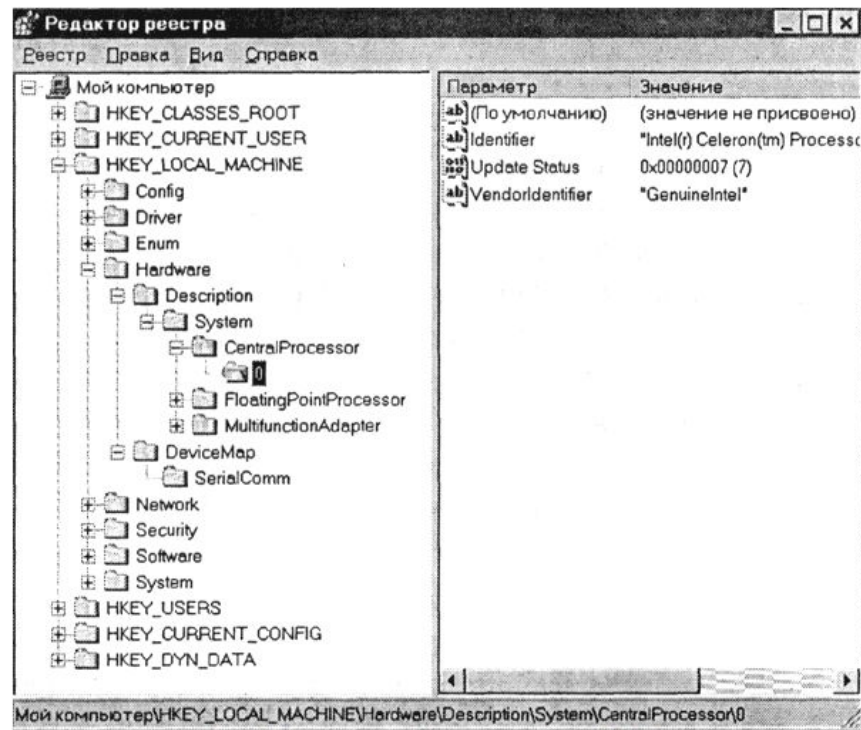
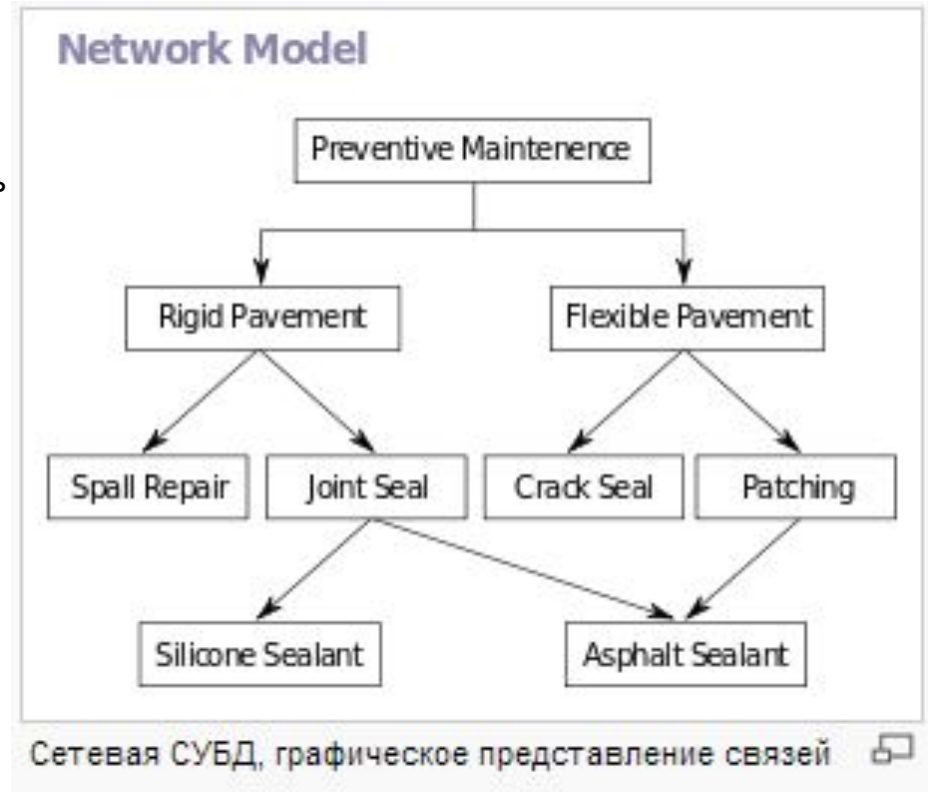


Рис. 3.2 Иерархическая база данных Реестр Windows

СЕТЕВЫЕ

- К основным понятиям **сетевой модели базы данных** относятся: уровень, элемент (узел), связь.
- Узел — это совокупность атрибутов данных, описывающих некоторый объект.
- В сетевой структуре каждый элемент может быть связан с любым другим элементом.
- Сетевые базы данных подобны **иерархическим**, за исключением того, что в них имеются указатели в обоих направлениях, которые соединяют родственную информацию.



объектно-ориентированные

- **Объектно-ориентированная (объектная) СУБД** — [система управления базами данных](#) — система управления базами данных, основанная на [объектной модели данных](#).
- Эта система управления обрабатывает данные как [абстрактные объекты](#) Эта система управления обрабатывает данные как абстрактные объекты, наделённые [свойствами](#) Эта система управления обрабатывает данные как абстрактные объекты, наделённые свойствами и использующие [методы](#) взаимодействия с другими объектами окружающего мира.
- **Абстракция** в [объектно-ориентированном программировании](#) в объектно-ориентированном программировании — это придание [объекту](#) в объектно-ориентированном программировании — это придание объекту характеристик, которые чётко определяют его концептуальные границы, отличая от всех других объектов. Основная идея состоит в том, чтобы отделить способ использования составных объектов данных от деталей их реализации в виде более простых объектов, подобно тому, как [функциональная](#) абстракция разделяет способ использования функции и деталей её реализации в терминах более примитивных функций, таким образом, данные обрабатываются функцией высокого уровня с помощью вызова функций низкого уровня.

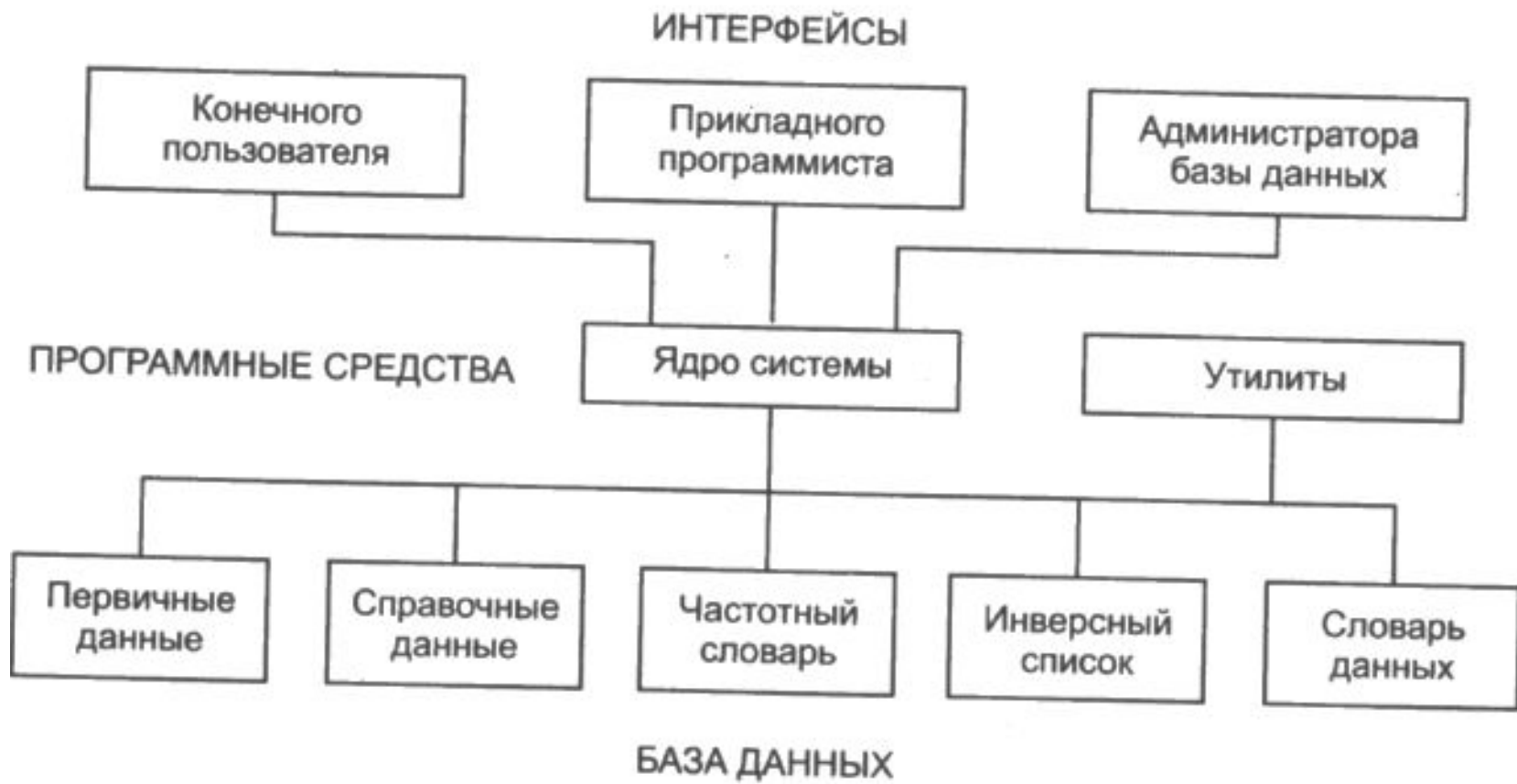
СУБД

СУБД - - комплекс языков и программ, позволяющий создавать БД и управлять ее работой. СУБД обрабатывает поступающие от пользователей и прикладных процессов обращения к БД, а затем выдает необходимые им сведения. СУБД характеризуется используемой моделью и средствами администрирования, разработки прикладных процессов, работы в информационной сети.

СУБД обеспечивает:

- описание и контроль данных;
- манипулирование данными (запись, поиск, выдачу, изменение содержания);
- физическое размещение (изменение размеров блоков данных, записей, использование занимаемого пространства, сортировку, сжатие, кодирование и пр.);
- защиту от сбоев, поддержку целостности и восстановление;
- работу с транзакциями и файлами;
- безопасность данных.

В структурном составе СУБД могут быть выделены ядро и среда



Ядро СУБД — программный комплекс (модуль или модули), обеспечивающий непосредственное выполнение физических операций над БД (в ранних системах функции Ядра выполняли программы методов доступа ОС ЭВМ).

Среда — совокупность интерфейсных модулей, обеспечивающих связь пользователей с Ядром и через него с БД. Среда включает в себя пользовательские интерфейсы и утилиты администратора БД (АБД).

Утилиты АБД образуют библиотеку программ обслуживания БД в привилегированном режиме (работа пользовательских средств параллельно утилитам не разрешена) и выполняют основные функции, к которым относятся:

- физическая подготовка дисковой памяти к размещению БД;
- подготовка справок о составе БД, структуре файлов, количестве данных и занимаемом объеме:
- загрузка файла БД из последовательного набора данных
- дозагрузка (расширение существующего файла);
- модификация БД: расширение или перемещение физических наборов данных, реорганизация;
- модификация файла (таблицы, группы таблиц): добавление новых полей в структуру записи; инвертирование полей или освобождение (превращение инвертированных полей в сканируемые);
- выгрузка образа БД (файла таблицы) для сохранения в архивном наборе данных;
- создание и ведение словаря данных и др.

Средства пользователя. Стандартными средствами этого типа, предоставляемыми фирмой-разработчиком, являются следующие:

- диалоговые интерфейсы;
- генераторы отчетов;
- система конструирования и поддержки интерактивных технологий в информационных системах (ЯП АИС).

Табличные базы данных

Подобные БД называются *табличными* или *реляционными* (от relation — отношение), и их теория рассматривается во многих источниках. Здесь мы ограничимся лишь базовыми понятиями:

- *файл* соответствует совокупности однородных объектов и содержит их более или менее подробные описания в зависимости от приложений. Файл имеет имя.
- элементом файла является *запись* (record) или агрегат разнотипных данных, описывающих объект (точнее, экземпляр объекта).
- Записи имен не имеют, но им соответствуют физические номера в файле;
- элементом записи (здесь -- неделимым) является *поле* — данное, описывающее какой-либо аспект (или атрибут) объекта. Поля имеют имена (ФИО, дата рождения и пр.). Разные файлы могут иметь поля с одинаковыми именами, но лучше этого избегать;

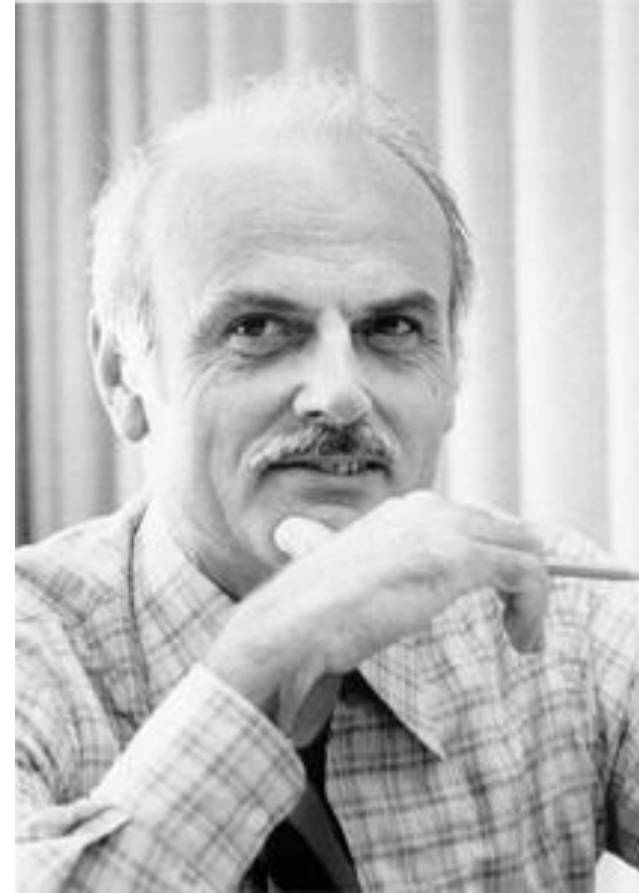
Табличные базы данных

- при представлении файла в виде таблицы столбцам соответствуют атрибуты или поля, строкам -объекты или записи.
- каждый столбец имеет имя, которое обычно записывается в верхней части таблицы. Оно должно быть уникальным в таблице, однако различные таблицы могут иметь столбцы с одинаковыми именами. Любая таблица должна иметь по крайней мере один столбец; столбцы расположены в таблице в соответствии с порядком следования их имен при ее создании. В отличие от столбцов, строки не имеют имен; порядок их следования в таблице неопределен, а количество логически не ограничено;
- *открытый файл* — файл, доступный в данный момент данному приложению. Открытие файла создает в памяти буфер, в который с внешнего накопителя считываются записи. В разные моменты времени могут быть открыты различные множества файлов, количество открытых файлов обычно стараются ограничить, чтобы не расходовать оперативную память;
- *активный или текущий* - тот из открытых файлов, который обрабатывается в данный момент времени. Все операции над файлами (добавление записи; удаление записи; редактирование записи) адресуются именно к активному файлу;
- *активная или текущая запись* - запись открытого файла, доступная для обработки в данный момент времени (редактирование, ввод полей, корректировка, удаление). Указатель текущей записи есть физический номер доступной записи. Текущая запись находится в оперативной памяти. При переходе к другой записи данного файла указатель записи изменяется, и содержание оперативной памяти замещается содержимым новой текущей записи

Эдгар Франк Кодд

(23 августа(23 августа 1923(23 августа 1923 — 18 апреля(23 августа 1923 — 18 апреля 2003)

- Британский математик и сотрудник фирмы IBM 1969 г предложил *реляционную модель* баз данных
- 12 правил Кодда (http://ru.wikipedia.org/wiki/12_правил_Кодда)
- 2002 г журнал Forbes поместил *реляционную модель данных* в список важнейших инноваций последних 85 лет





а

#	PRSN#	FIO	YEAR	SEX	ADRESS	PROFESSION
1	576	Распутин	1941	м	Москва	Программист
2	231	Петрова	1937	ж	Гомель	Водитель
3	256	Иванов	1945	м	Самара	Преподаватель
4	578	Сидорова	1987	ж	Тамбов	Продавец
5	132	Грачев	1978	м	Москва	Секретарь

б

DRV#	MAKE	BODY	YEAR	COLOR	HORSES
256	OPEL	СЕДАН	1991	Синий	78
578	ЖИГУЛИ	УНИВЕР	1994	Черный	56
576	BMW	СЕДАН	1987	Белый	125

в

OVN#	BANK#	ACCOUNT#	CURRENT
576	Автобанк	2345./34	23,345.00
132	Сбербанк	25058-6	1,000.00
578	Интербанк	5476-34	765,243.00

г

Проектирование базы данных

Основные задачи:

- Обеспечение хранения в БД всей необходимой информации.
- Обеспечение возможности получения данных по всем необходимым запросам.
- Сокращение избыточности и дублирования данных.
- Обеспечение целостности данных (правильности их содержания): исключение противоречий в содержании данных, исключение их потери и т.д.

Этапы проектирования

- 1. Концептуальное (инфологическое) проектирование** — построение информационной модели наиболее высокого уровня абстракции.
 - На данном этапе выделяются *сущности (объекты)*, описываются их свойства (*только существенные*), устанавливаются связи между сущностями

Этапы проектирования

2. Логическое (дatalogическое)

проектирование — создание схемы
базы данных — создание схемы базы
данных на основе конкретной модели
данных.

Пример, сущность «Человек»

ИНН	Фамилия	Имя	Отчество	Дата рождения

- *Каждая строка таблицы (одна запись) соответствует одному экземпляру сущности, а каждое поле описывает некоторое свойство (атрибут).*
- *Реляционная модель требует, чтобы любые две строки различались значением хотя бы одного атрибута.*

Реляционная БД

- **Ключом таблицы** называется поле или группа полей, содержащие уникальные в рамках данной таблицы значения. **Ключи: первичные и составные, суррогатные**

Дата	Время суток	Температура

Этапы проектирования


3. Физическое проектирование

- **Физическое проектирование** — создание схемы базы данных — создание схемы базы данных для конкретной СУБД.
- Необходимо учитывать специфику конкретной СУБД, ктр включает в себя ограничения на именование объектов базы данных, ограничения на поддерживаемые типы данных и т.п.
- Выбор методов управления дисковой памятью, разделение БД по файлам и устройствам, методов доступа к данным), создание индексов и т.д.

Нормализация данных в базе

- Не всякая таблица может удовлетворять требованиям реляционной модели!
- Из 12 правил Кодда следует:

1. Все данные в пределах одного столбца должны иметь один и тот же тип.



Имя	Возраст
Петя	9
Ваня	семь

2. В таблице должен быть назначен первичный ключ.

Нормальные формы (НФ)

1 НФ:

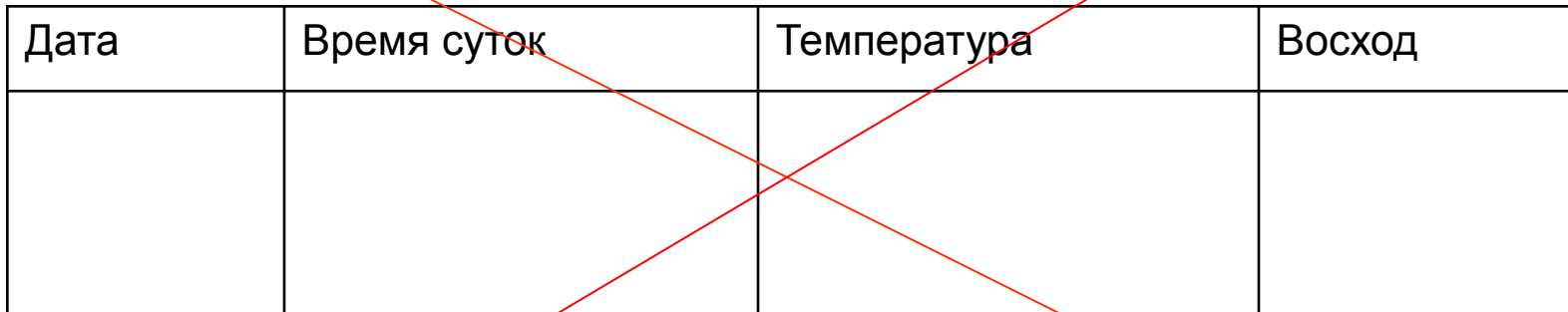
- Все данные, содержащиеся в таблице должны быть атомарными (неделимыми)!



Номер в журнале	Фамилия	Оценки
1	Иванов	2, 3, 2, 2, 3, 2, 3
2	Петрова	5, 5, 5, 5, 5

Нормальные формы (НФ)

2. Если соблюдаются требования 1 НФ и каждый не ключевой столбец полностью зависит от первичного ключа



Дата	Время суток	Температура	Восход

Нормальные формы (НФ)

3 НФ:

- Если соблюдаются требования 2 НФ и все не ключевые столбцы взаимно независимы.
- *Столбцы взаимозависимые: нельзя изменить один из них не изменяя другой*

Номер	Фамилия	Имя	Город	Код города	Телефон
1	Иванов	Иван	Ивановск	1001	54321
2	Петров	Петя	Петровск	20023	7788

Многотабличные БД

- Пример: база данных школа



Даталогическая модель БД «Школа»

Таблица
"Ученик"

Номер ученик	фамилия	Имя

Таблица
"Класс"

Номер класса	Номер параллели	Буква

Таблица
"Учитель"

Номер учителя	Фамилия	Имя	Отчество

Таблица
"Предмет"

Номер предмета	Название предмета

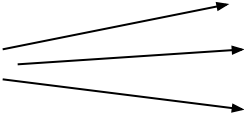
СВЯЗЬ «ОДИН КО МНОГИМ»

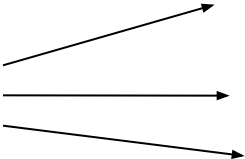


СВЯЗЬ «МНОГИЕ КО МНОГИМ»

Таблица "Учитель –
Предмет"

Номер учителя	Номер предмета

Один учитель  Много предметов

Один предмет  Много учителей

Правила Целостности и Правило ссылочной целостности

- **Правило целостности:** *первичные ключи таблиц не содержат неопределенных (пустых) значений.*
- **Правило ссылочной целостности:** внешние ключи не содержат несогласованных, с родительскими ключами значений.

Ученики

А
Б
В
Г
~~Д~~

Классы

А
Б
В
Г

Описание данных

- Типы данных в реляционных СУБД:
- числовые;
- строковые;
- логические;
- даты.



The image shows a screenshot of a database management system window titled "Friends : таблица". The window displays a table definition with two columns: "Имя поля" (Field Name) and "Тип данных" (Data Type). The table has three rows: "id" (Number), "name" (Text), and "birthday" (Text Memo). A dropdown menu is open for the "id" field, showing various data types: "Числовой" (Number), "Текстовый" (Text), "Поле МЕМО" (Text Memo), "Числовой" (Number), "Дата/время" (Date/Time), "Денежный" (Currency), "Счетчик" (Counter), "Логический" (Boolean), "Поле объекта OLE" (OLE Object Field), "Гиперссылка" (Hyperlink), and "Мастер подстано" (Master Substitution).

Имя поля	Тип данных
id	Числовой
name	Текстовый
birthday	Поле МЕМО

Физический этап проектирования БД

- *Определение имени таблицы.*
- *Определение имен и типов полей.*
- *Определение первичного ключа.*
- *Определение (при необходимости) индексированных полей.*

Создание таблиц в СУБД Access.

- Создание Кнопка Таблица
- Кнопка Конструктор таблиц позволяет создавать таблицу начиная с ее структуры
- Режим таблицы – позволяет просмотреть поля и записи

Открыть базу данных

- Типы файлов: *.accdb – Access 2007, *.mdb – более ранние версии
- Параметры кнопки открыть:
- Открыть
- Открыть для чтения
- Монопольно – другие пользователи не смогут просмотреть данные
- Монопольно для чтения

Ввод и редактирование данных в базу