



Базы данных

- *База данных (БД)* – совокупность данных, организованных по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования данными, которые относятся к определенной предметной области.
- Под *данными* понимается информация, представленная в виде, пригодном для обработки автоматическими средствами при возможном участии человека.
- Под *предметной областью* понимается однородная часть реального мира, представляющая интерес для конкретного исследования.
- *СУБД* – пакет программ *обеспечивающих* создание БД и организацию данных.

Приложения пользователей – программы для работы с информацией
базы данных

Приложение 1

Приложение 2

Приложение N

СУБД – комплекс программных и языковых средств, необходимых для работы с базой данных

База данных – структурированная информация на диске, соответствующая определенной схеме

Схема базы данных – описание модели для конкретной базы

Модель данных – концептуальное описание предметной области

Предметная область – часть реально существующего мира

По технологии обработки данных базы данных
подразделяются на:

- **Централизованные** (хранится в памяти одной вычислительной системы. Эта вычислительная система может быть мэйнфреймом - тогда доступ к ней организуется с использованием терминалов - или файловым сервером локальной сети ПК)
- **Распределенные** (Состоит из нескольких, возможно, пересекающихся или даже дублирующих друг друга частей, которые хранятся в различных ЭВМ вычислительной сети. Работа с такой базой осуществляется с помощью системы управления распределенной базой данных (СУРБД).)



Централизованные базы данных с сетевым доступом могут иметь следующую архитектуру:

- файл-сервер;
- клиент-сервер базы данных;
- "тонкий клиент" - сервер приложений - сервер базы данных (трехуровневая архитектура).

- **Файл-сервер.** Архитектура систем БД с сетевым доступом предполагает выделение одной из машин сети в качестве центральной (файловый сервер). На этот компьютер устанавливается операционная система (ОС) для выделенного сервера. На нем же хранится совместно используемая централизованная БД в виде одного или группы файлов. Все другие компьютеры сети выполняют функции рабочих станций. Файлы базы данных в соответствии с пользовательскими запросами передаются на рабочие станции, где и производится обработка информации. При большой интенсивности доступа к одним и тем же данным производительность информационной системы падает.
- **Клиент-сервер.** В этой архитектуре на выделенном сервере, работающем под управлением серверной операционной системы, устанавливается специальное программное обеспечение (ПО) - сервер БД, например. СУБД подразделяется на две части: клиентскую и серверную. Основа работы сервера БД - использование языка запросов (SQL). Запрос на языке SQL, передаваемый клиентом (рабочей станцией) серверу БД, порождает поиск и извлечение данных на сервере. Тем самым, количество передаваемой по сети информации уменьшается во много раз.

Трехуровневая архитектура функционирует в Интранет- и Интернет-сетях. Клиентская часть ("тонкий клиент"), взаимодействующая с пользователем, представляет собой HTML-страницу в Web-браузере либо Windows-приложение, взаимодействующее с Web-сервисами. Вся программная логика вынесена на сервер приложений, который обеспечивает формирование запросов к базе данных, передаваемых на выполнение серверу баз данных. Сервер приложений может быть Web-сервером или специализированной программой



По способу доступа к данным базы данных
разделяются на :

- **базы данных с локальным доступом**
- **базы данных с сетевым доступом.**



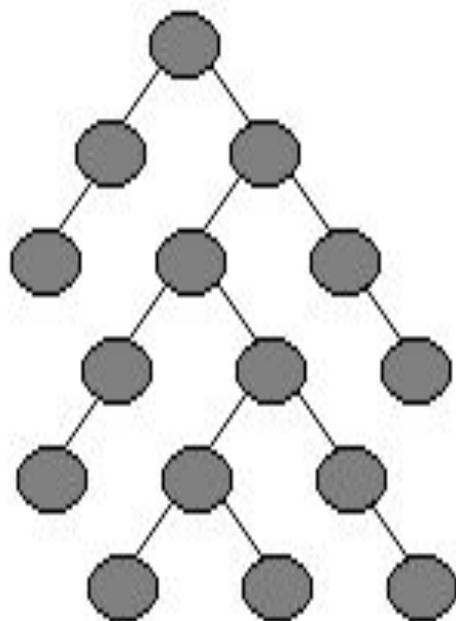
Классификация (по модели)

- Иерархическая
- Сетевая
- Реляционная

Классификация БД

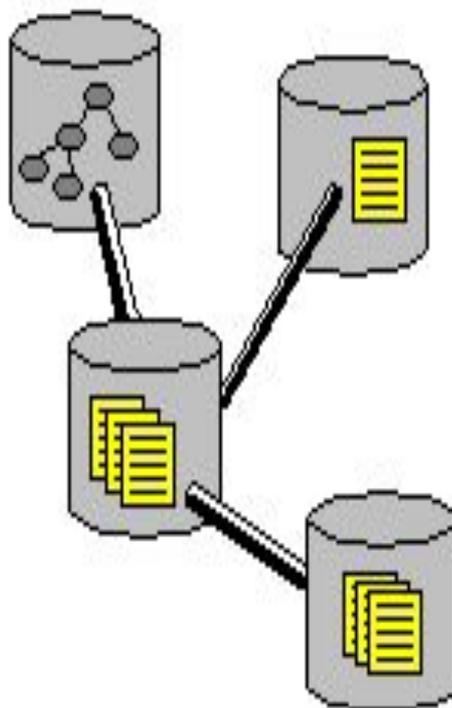
5%

Иерархические
(Древовидные)



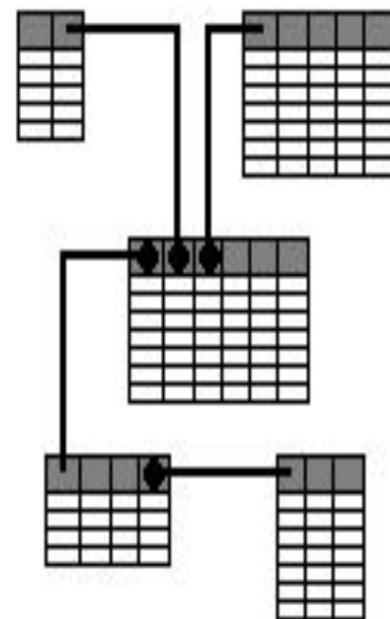
1%

Сетевые
(Распределенные)



94%

Реляционные
(Табличные)



В иерархической модели связи между данными можно описать с помощью упорядоченного графа (или дерева).

Основные операции манипулирования данными:

- поиск указанного экземпляра БД;
- переход от одного дерева к другому;
- переход от одной записи к другой внутри дерева;
- вставка новой записи в указанную позицию;
- удаление текущей записи и т. д.

К достоинствам иерархической модели данных относятся эффективное использование памяти ЭВМ и неплохие показатели времени выполнения основных операций над данными. Иерархическая модель данных удобна для работы с иерархически упорядоченной информацией.

Недостатком иерархической модели является ее громоздкость для обработки информации с достаточно сложными логическими связями, а также сложность понимания для обычного пользователя.

На иерархической модели данных основано сравнительно ограниченное количество СУБД, в числе которых можно назвать зарубежные системы IMS, PC/Focus, Team-Up и Data Edge, а также отечественные системы Ока, ИНЭС и МИРИС.

Сетевая модель данных позволяет отображать разнообразные взаимосвязи элементов данных в виде произвольного графа, обобщая тем самым иерархическую модель данных. Наиболее полно концепция сетевых БД впервые была изложена в Предложениях группы КОДАСИЛ (KODASYL).

Основные операции манипулирования данными:

- поиск записи в БД;
- переход от предка к первому потомку;
- переход от потомка к предку;
- создание новой записи;
- удаление текущей записи;
- обновление текущей записи;
- включение записи в связь;
- исключение записи из связи;
- изменение связей и т. д.

Достоинством сетевой модели данных является возможность эффективной реализации по показателям затрат памяти и оперативности. В сравнении с иерархической моделью сетевая модель предоставляет большие возможности в смысле допустимости образования произвольных связей.

Недостатком сетевой модели данных является высокая сложность и жесткость схемы БД, построенной на ее основе, а также сложность для понимания и выполнения обработки информации в БД обычным пользователем. Кроме того, в сетевой модели данных ослаблен контроль целостности связей вследствие допустимости установления произвольных связей между записями.

Системы на основе сетевой модели не получили широкого распространения на практике. Наиболее известными сетевыми СУБД являются следующие: IDMS, db VistaIII, СЕТЬ, СЕТОР и КОМПАС.

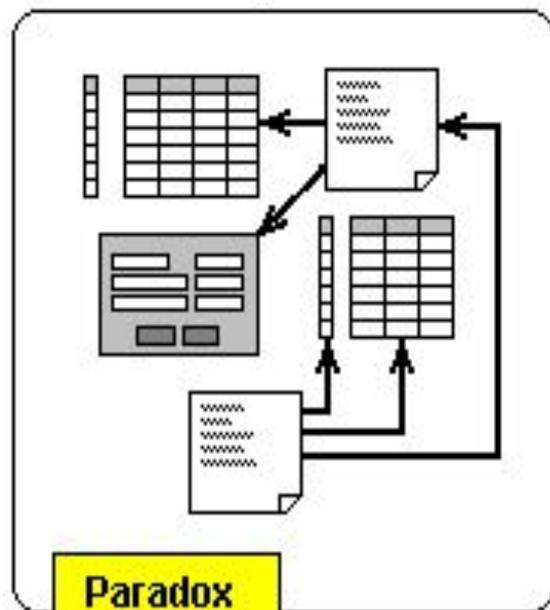
Реляционная модель данных предложена сотрудником фирмы IBM Эдгаром Коддом и основывается на понятии отношение (relation).

Достоинство реляционной модели данных заключается в простоте, понятности и удобстве физической реализации на ЭВМ. Именно простота и понятность для пользователя явились основной причиной их широкого использования. Проблемы же эффективности обработки данных этого типа оказались технически вполне разрешимыми.

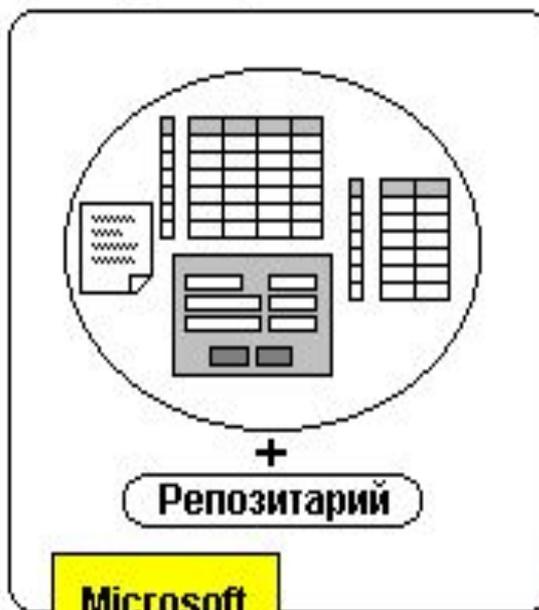
Основными недостатками реляционной модели являются следующие: отсутствие стандартных средств идентификации отдельных записей и сложность описания иерархических и сетевых связей.

Примерами зарубежных реляционных СУБД для ПЭВМ являются следующие:
все БД семейства dBase, DB2 (IBM), R:BASE (Microrim), FoxPro, FoxBase, Visual FoxPro, Microsoft Access, Clarion, Ingres, Oracle и т.д.

Многофайловая



Однофайловая



Репозиторий БД



1

Локальные БД

1

12

Удаленные БД

200

- **Реляционная БД** - основной тип современных баз данных. Состоит из таблиц, между которыми могут существовать связи по ключевым значениям.
- **Таблица** базы данных (table) - регулярная структура, которая состоит из однотипных строк (записей, records), разбитых на столбцы (поля, fields).
- В теории реляционных баз данных синоним таблицы - **отношение** (relation), в котором строка называется **кортежем**, а столбец называется **атрибутом**.
- В концептуальной модели реляционной БД аналогом таблицы является сущность (entity), с определенным набором свойств - **атрибутов**, способных принимать определенные значения (набор допустимых значений - **домен**).

- **Первичный ключ (primary key)** - главный ключевой элемент, однозначно идентифицирующий строку в таблице. Могут также существовать альтернативный (candidate key) и уникальный (unique key) ключи, служащие также для идентификации строк в таблице.
- **Внешний ключ (foreign key)** - ключевой элемент подчиненной (внешней, дочерней) таблицы, значение которого совпадает со значением первичного ключа главной (родительской) таблицы.

Элемент реляционной модели	Форма представления (характеристика)
Отношение	<i>Таблица</i> , имеющая уникальные строки, отличные друг от друга имена столбцов и допускающая произвольный порядок размещения строк
Схема отношения	<i>Строка заголовков таблицы</i>
Запись	Строка (запись) таблицы
Сущность	<i>Описание свойств объекта любой природы</i> , данные о котором хранятся в базе данных
Атрибут	<i>Заголовок столбца таблицы</i>
Домен	Множество допустимых значений атрибута
Значение атрибута	Значения поля в записи таблицы
Первичный ключ	<i>Один или несколько атрибутов отношения</i> однозначно определяющих каждую из его записей
Внешний ключ	Не ключевой атрибут таблицы (родителя), который является первичным ключом дочерней таблицы
Тип данных	Тип значений атрибута таблицы

Типы полей

1. **Символьный**
(текстовый, длина ≤ 256 символов)
2. **Поле МЕМО** (для хранения больших текстов, ≤ 65535 символов)
3. **Числовой**
(целые, вещественные)
4. **Дата/время**
5. **Логический** (для хранения логических данных, принимающих лишь 2 значения: true или false, Истина или Ложь, 1 или 0, Да или Нет)
6. **Поле объекта OLE** (для хранения графической, видео, аудио информации и т.п.)

Нормальная форма (НФ) – требование, предъявляемое к структуре таблиц в теории реляционных баз данных для устранения из базы избыточных функциональных зависимостей между атрибутами (полями таблиц).

Всего существует восемь НФ:

- 1–6 НФ;
- доменно-ключевая НФ;
- НФ Бойса-Кодда.

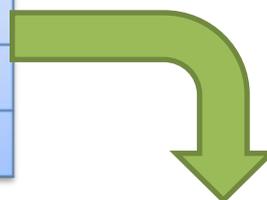
1 нормальная форма

Условия *первой нормальной формы*:

- 1) каждой сущности соответствует отдельная таблица;
- 2) каждый набор связанных данных идентифицирован с помощью первичного ключа;
- 3) поля не имеют дубликатов в каждой записи;
- 4) каждое поле содержит только одно значение (атомарно).

Атомарность (неделимость) поля означает, что содержащиеся в нем значения не должны делиться на более мелкие.

Преподаватели		
№	ФИО	Подразделение
1	Каратанов А. В.	1, 105
2	Крицкий Д. Н.	1, 105
3	Брега Д. А.	1, 104



Преподаватели			
№	ФИО	Факультет	Кафедра
1	Каратанов А. В.	1	105
2	Крицкий Д. Н.	1	105
3	Брега Д. А.	1	104

2 нормальная форма

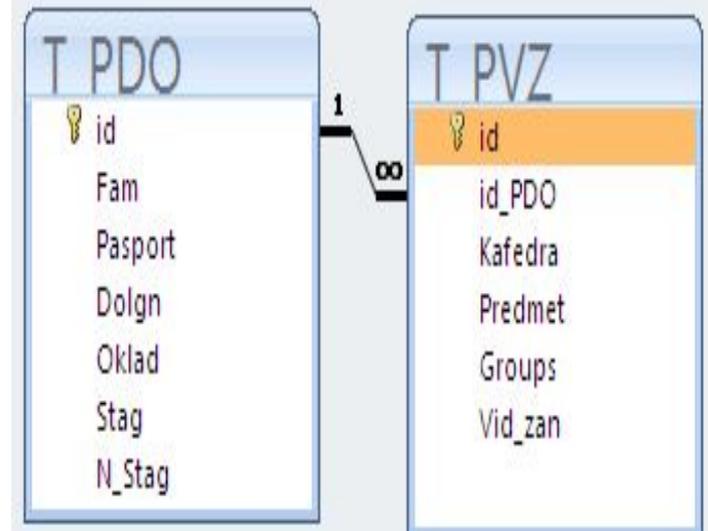
Условия **второй нормальной формы**:

- 1) модель удовлетворяет условиям 1 НФ;
 - 2) все поля каждой таблицы, не входящие в первичный ключ, связаны полной функциональной зависимостью с первичным ключом.
- Замечание. Если ключ отношения является простым, то отношение автоматически находится в 2НФ.

База данных приведенная к 2 НФ

T_PDO						
id	Fam	Pasport	Dolgn	Oklad	Stag	N_Stag
1	Иванов	5702111111	Преподаватель	3 000,00р.	10	2 000,00р.
2	Иванов	5702555555	Преподаватель	3 000,00р.	10	2 000,00р.
3	Петров	5403222222	Доцент	6 000,00р.	5	1 000,00р.
4	Федоров	5501333333	Профессор	8 000,00р.	10	2 000,00р.
5	Яковлев	5112444444	Преподаватель	3 000,00р.	5	1 000,00р.

T_PVZ					
id	id_PDO	Kafedra	Predmet	Groups	Vid_zan
1	1	Инф. технологий	АП	111	Практика
2	1	Инф. технологий	БД	222	Практика
3	2	Инф. технологий	ВМП	333	Лаб. работа
4	3	Инф. технологий	ВМП	333	Лекция
5	3	Инф. технологий	ПОКС	333	Лекция
6	3	Инф. технологий	БД	222	Лаб. работа
7	4	Инф. технологий	АП	111	Лекция
8	4	Математики	Математика	111	Лекция
9	5	Инф. технологий	ВМП	222	Практика
10	5	Инф. технологий	ПОКС	333	Практика
11	5	Математики	Математика	111	Практика



3 нормальная форма

Условия **третьей нормальной формы**:

- 1) модель удовлетворяет условиям 2 НФ;
- 2) все не ключевые поля полностью зависят от первичного ключа таблицы и не зависят друг от друга.

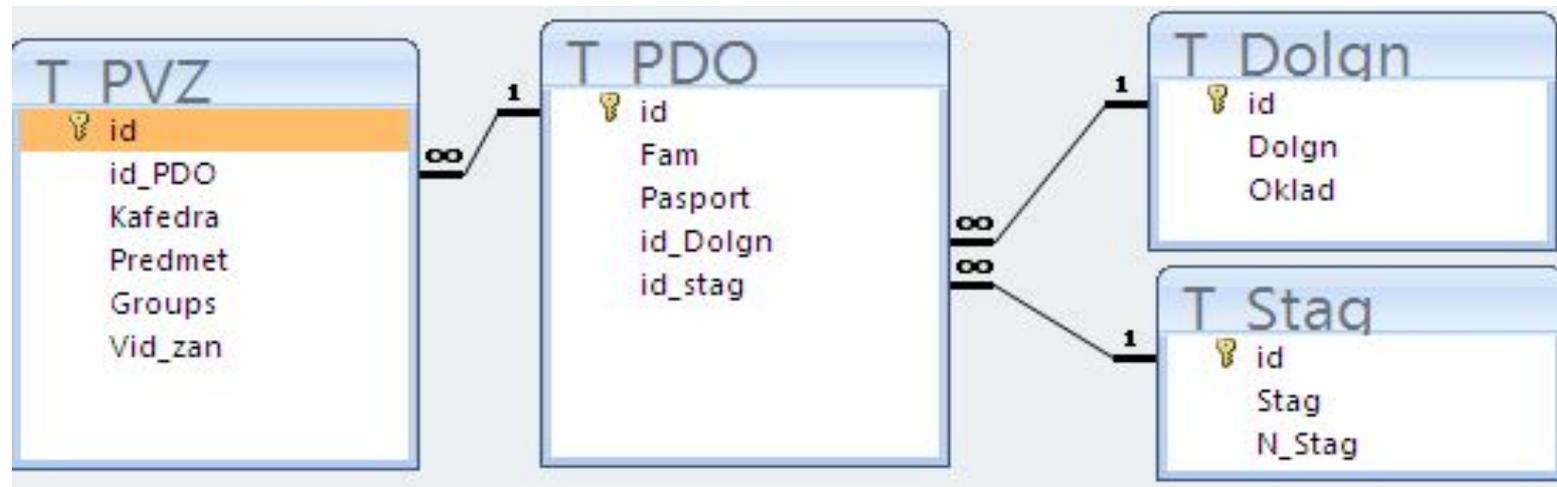
- **Неключевой атрибут** — это атрибут, который не входит в состав первичного ключа рассматриваемой переменной-отношения.
- Два или более атрибутов называются *взаимно независимыми*, если ни один из них функционально не зависит от какой-либо комбинации остальных атрибутов. Подобная независимость подразумевает, что каждый такой атрибут может обновляться независимо от значений остальных атрибутов.

База данных приведенная к 3 НФ

T_PDO				
id	Fam	Pasport	id_Dolgn	id_stag
1	Иванов	5702111111	1	2
2	Иванов	5702555555	1	2
3	Петров	5403222222	2	1
4	Федоров	5501333333	3	2
5	Яковлев	5112444444	1	1

T_Stag		
id	Stag	N_Stag
1	5	1 000,00p.
2	10	2 000,00p.

T_Dolgn		
id	Dolgn	Oklad
1	Преподаватель	3 000,00p.
2	Доцент	6 000,00p.
3	Профессор	8 000,00p.



Способ хранения не нормализованных данных в базе данных bdUchet

TUchet

Fam	Grup	Spec	Predmet	Chas	Oценка
Иванов	ООКС	ИС	Программирование	72	5
Иванов	ООКС	ИС	Базы данных	80	4
Петров	ООКС	ИС	Программирование	72	4
Петров	ООКС	ИС	Базы данных	80	5
Сидоров	Э-2007	ПИЭ	Программирование	72	3
Сергеев	КИТ-2007	ПК	Информатика	64	5

Способ хранения нормализованных данных в БД bdUchet_N

TStud

id	Fam	Grup	Spec
1	Иванов	ООКС	ИС
2	Петров	ООКС	ИС
3	Сидоров	Э-2007	ПИЭ
4	Сергеев	КИТ-2007	ПК

TPred

id	Predmet	chas
1	Программирование	72
2	Базы данных	80
3	Информатика	64

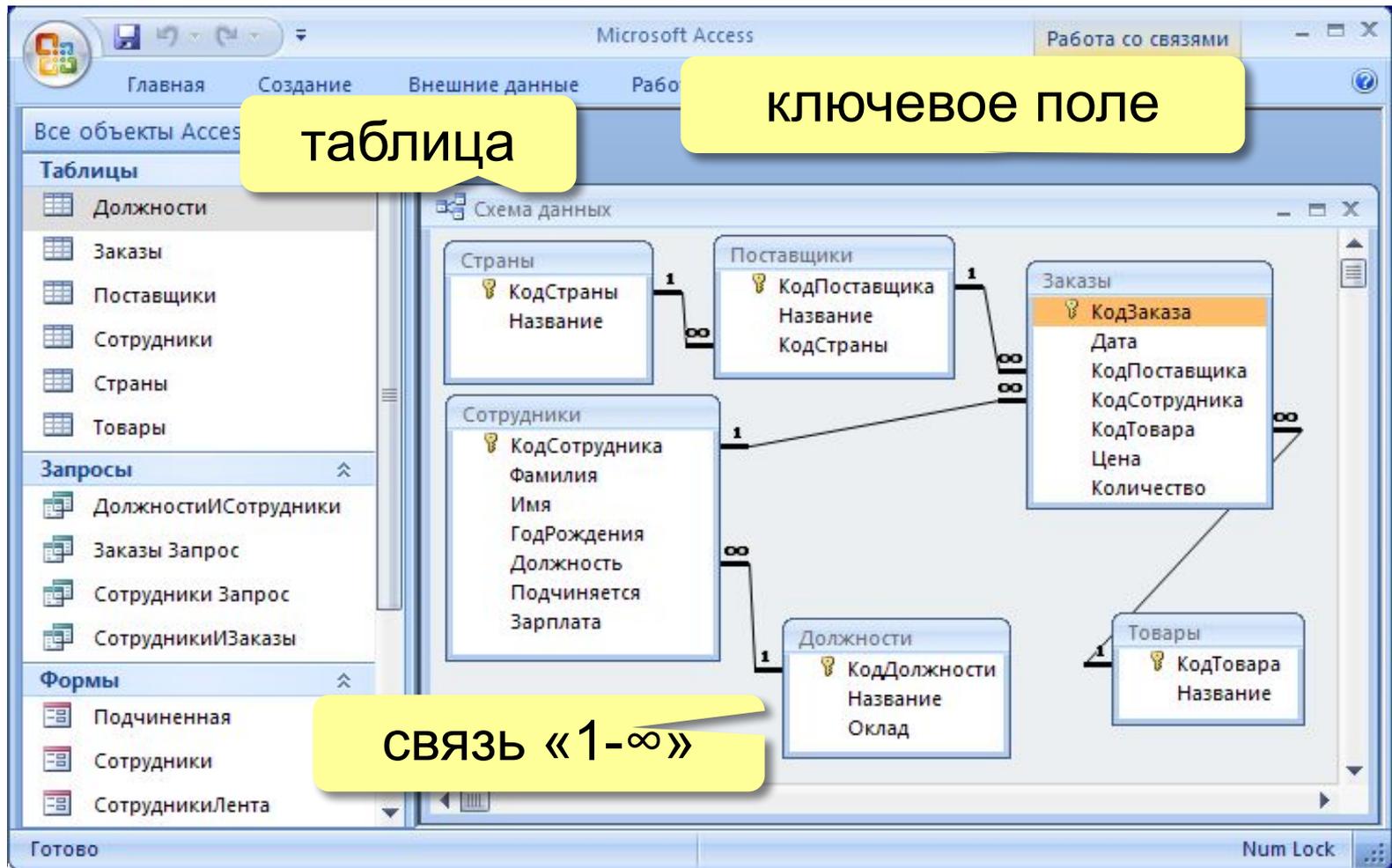
TStud_Pred

id	Id_stud	Id_Pred	Oценка
1	1	1	5
2	1	2	4
3	2	1	4
4	2	2	5
5	3	1	3
6	4	3	5

Объекты MS Access

- **Таблицы** – это основные объекты БД, предназначенные для хранения данных (реляционная БД может иметь несколько взаимосвязанных таблиц)
- **Запросы** – это спец.структуры, предназначенные для обработки данных базы
- **Формы** – это объекты, с помощью которых в базу вводят новые данные или просматривают имеющиеся
- **Отчеты** – это объекты, с помощью которых данные выводят на принтер в удобном наглядном виде
- **Макросы** – это макрокоманды. Если какие-то операции с БД производятся особенно часто, имеет смысл сгруппировать несколько команд в один макрос и назначить его выделенной комбинации клавиш
- **Модули** – это программные продукты, написанные на языке Visual Basic. Если стандартных средств MS Access пользователю не хватает, он может расширить возможности системы, написав необходимые модули

Схема данных



Удалить связь: ЛКМ + Delete.

Создать связь: перетащить нужное поле на соответствующее поле второй таблицы.