

# Базы данных





# План

- ❖ Понятие БД
- ❖ Архитектура БД
- ❖ Модели данных
- ❖ Нормальные формы
- ❖ Операции реляционной алгебры
- ❖ Операции языка SQL
- ❖ СУБД

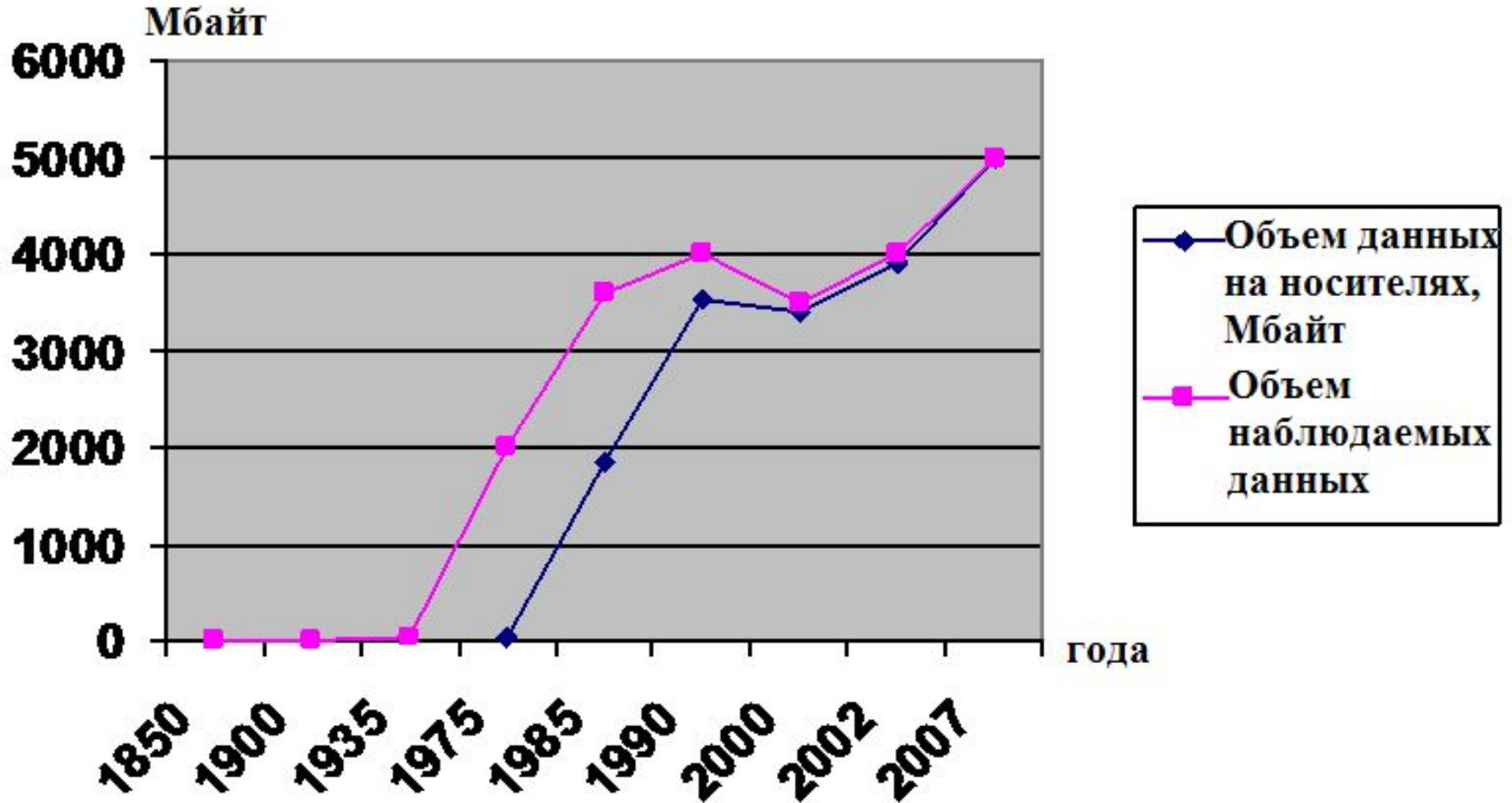


**То, как человек собирает,  
управляет и использует  
информацию, будет  
определять, окажется ли  
он в выигрыше или в  
проигрыше**

**Билл Гейтс,  
глава компании Microsoft**



# Увеличение объемов данных на примере состояния морской среды



Рост потоков информации так велик, что его часто называют **информационным взрывом.**





# Оценки объемов данных по некоторым разделам наук о земле в России

Дисциплина	Количество источников	Объем данных, Мбайт
Дистанционное зондирование Земли	440377 записей	143 217
Гидрогеология, геозкология	46 738 записей	776
Минеральные ресурсы	1 822 359 записей	13 152
Геофизика	2 992 026 записей	535 976
Недропользование	23 496 записей	3 786
Океанография	5 000 000 профилей, 200 действующих прибрежных станций	11 000 000
Морская геология	103 680 363 записей	29 615 312
Метеорология	1900 действующих метеорологических станций и постов	30 000 000
Гидрология суши	2 300 действующих гидрологических станций и постов	2 000 000
Аэрология	60 действующих аэрологических станций	1 000 000



# Общее определение базы данных

В широком смысле слова **база данных** – это совокупность сведений о конкретных объектах реального мира в какой-либо предметной области.

Предметная область – часть реального мира, подлежащего изучению для организации управления и автоматизации, например, предприятие, вуз и т.д.



# Пример неструктурированных данных

Личное дело № 16493, Сергеев Петр Михайлович, дата рождения 1 января 1976; личное дело № 16593, Петрова Анна Владимировна, дата рождения 15 марта 1975; личное дело № 16693, Анохин Андрей Борисович, дата рождения 14 апреля 1976.

Сложно организовать поиск необходимых данных, хранящихся в неструктурированном виде, а упорядочить подобную информацию практически не представляется возможным.



# Пример структурированных данных

№ личного дела	Фамилия	Имя	Отчество	Дата рождения
16493	Сергеев	Петр	Михайлович	01.01.76
16593	Петрова	Анна	Владимировна	15.03.75
16693	Анохин	Андрей	Борисович	14.04.76

**Структурирование** — это введение соглашений о способах представления данных.





# Определение базы данных

**База данных (БД)** – это поименованная совокупность структурированных данных, относящихся к определенной предметной области.

**Система управления базами данных (СУБД)** – это комплекс программных и языковых средств, необходимых для создания баз данных, поддержания их в актуальном состоянии и организации поиска в них необходимой информации.



# Классификация баз данных

## Технология обработки данных

### Централизованные

хранится в памяти одной вычислительной системы, которая может быть мэйнфреймом (доступ с помощью терминалов) или файловым сервером локальной сети.

### Распределенные

состоит из нескольких частей, которые хранятся в различных ЭВМ вычислительной сети.



# Классификация баз данных





# Классификация баз данных

Централизованные базы данных с сетевым доступом могут иметь следующую архитектуру:

- ❖ файл-сервер
- ❖ клиент-сервер
- ❖ двухуровневая модель
- ❖ трехуровневая модель



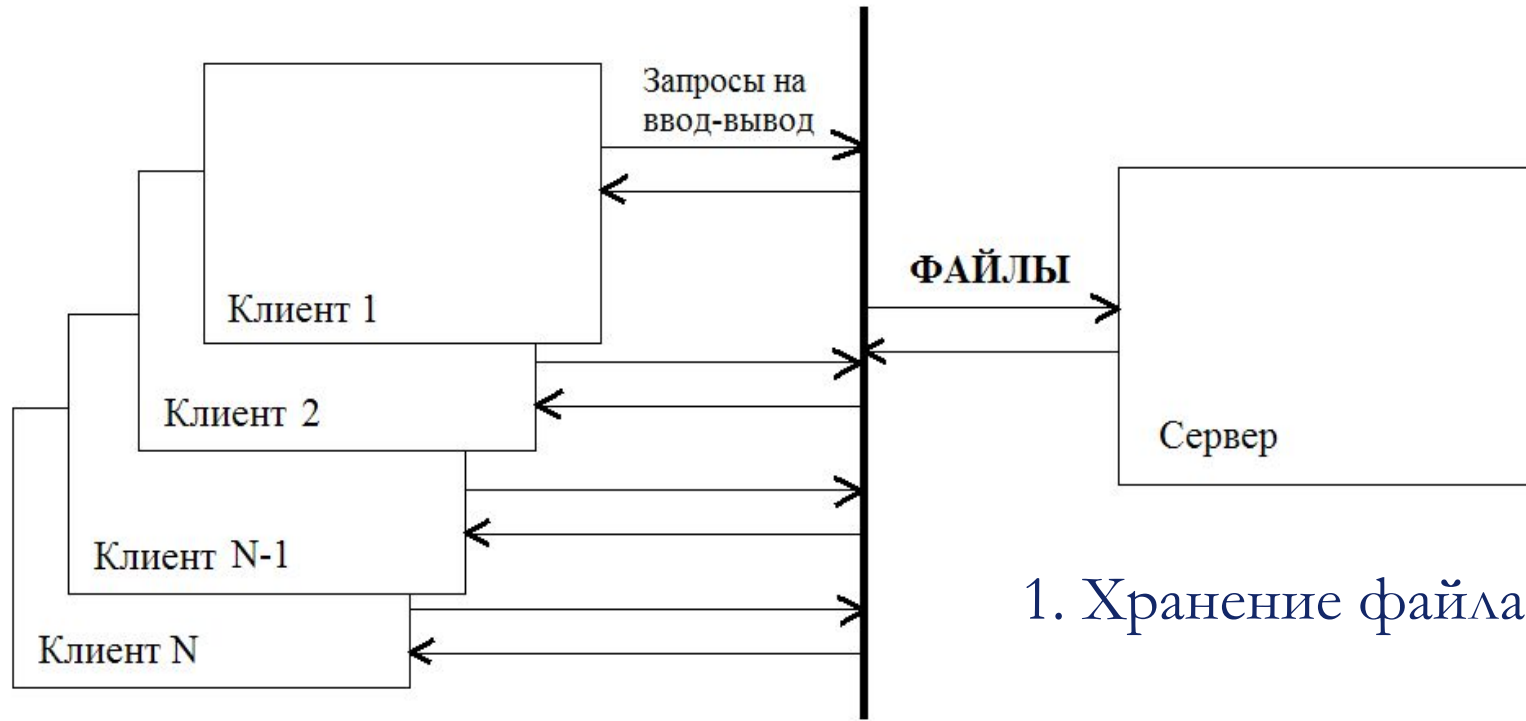


# Архитектура файл-сервер





# Архитектура файл-сервер



1. Хранение файла БД

1. Ввод и отображение данных
2. Доступ к данным и поиск по критериям
3. Реализация вычислительных функций над данными



# Достоинства и недостатки архитектуры файл-сервер

## Достоинства:

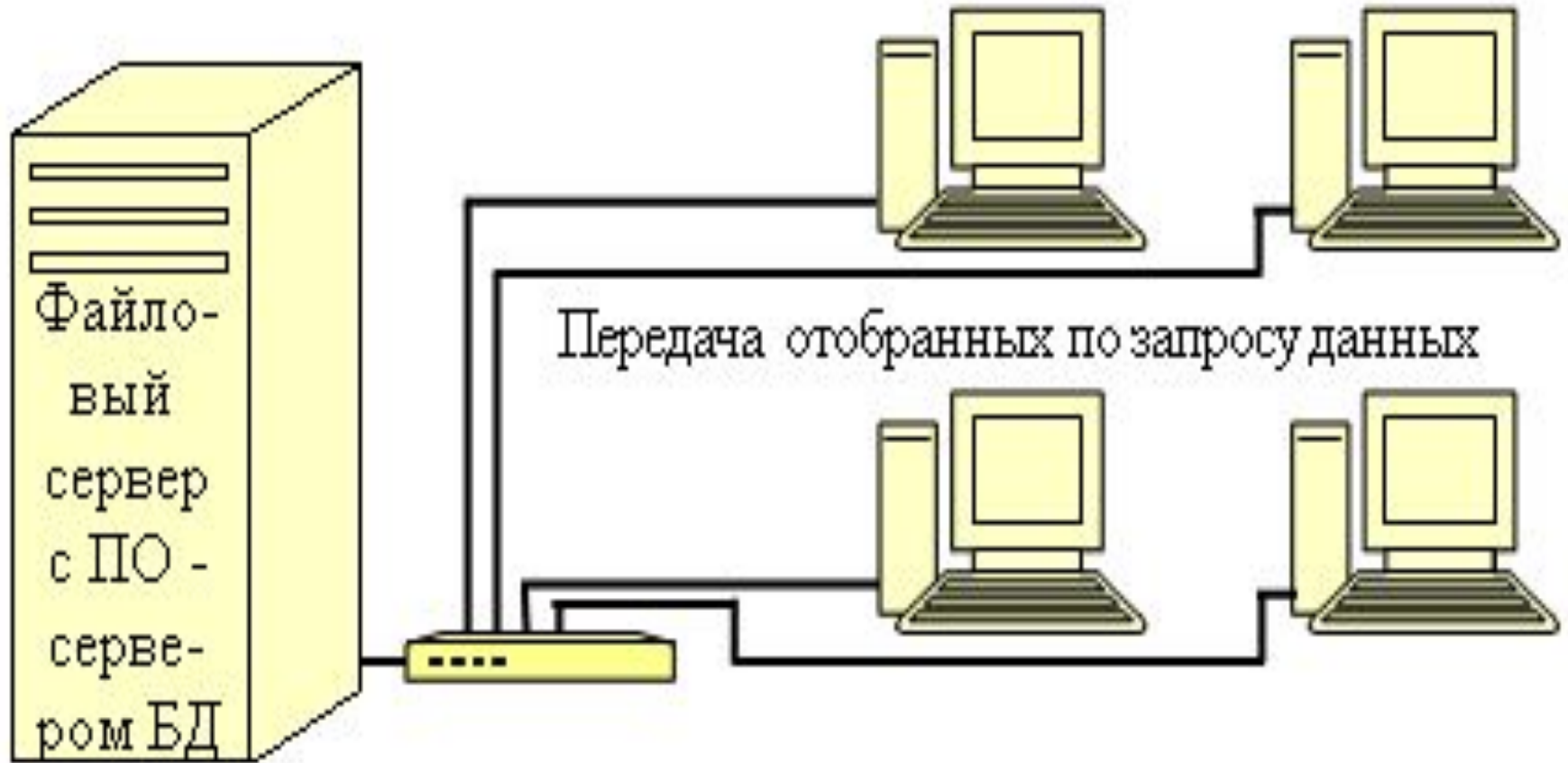
1. Отсутствие высоких требований к производительности сервера (главное – требуемый объем дискового пространства)
2. На сервере СУБД не размещается и не устанавлируется

## Недостатки:

1. Высокий сетевой трафик
2. Отсутствие специальных механизмов безопасности файла (файлов) БД со стороны СУБД



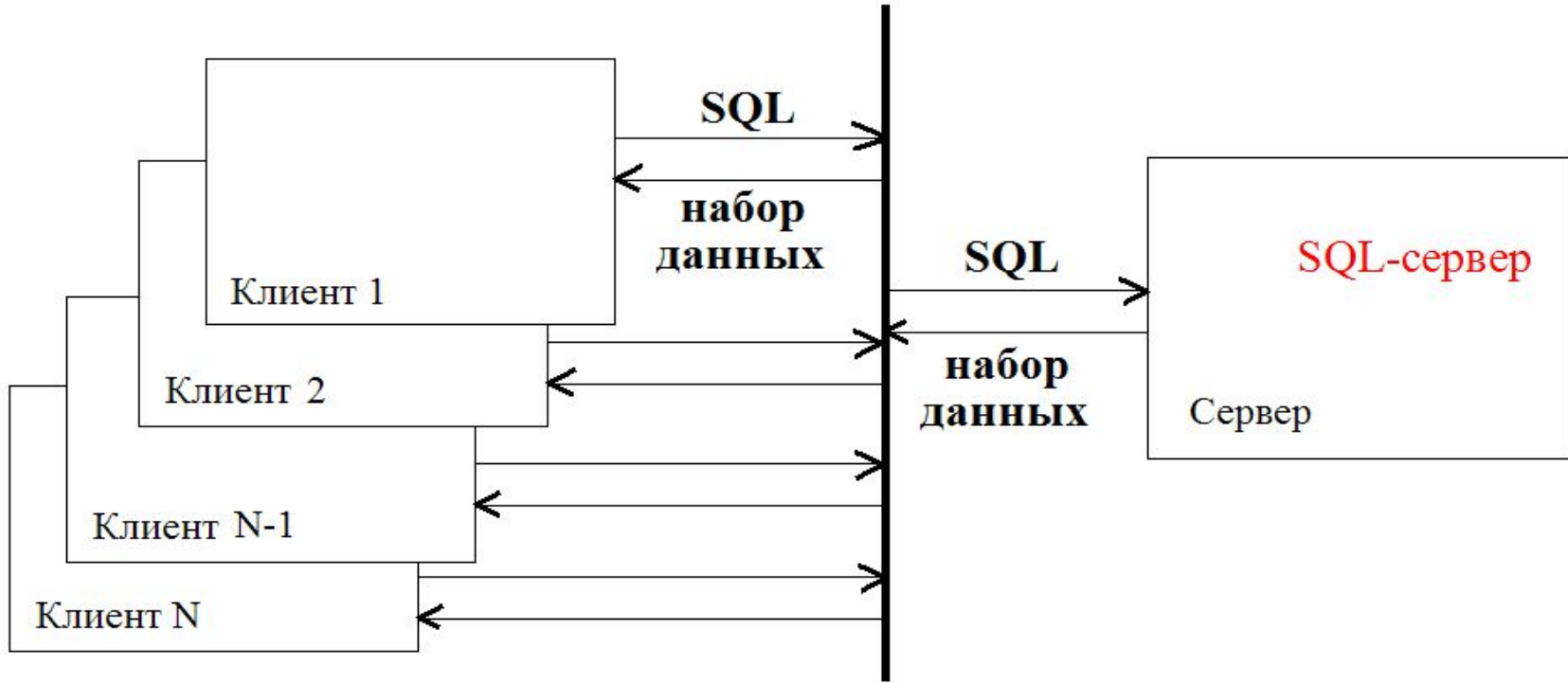
# Архитектура клиент-сервер







# Последовательность действий при использовании архитектуры клиент-сервер

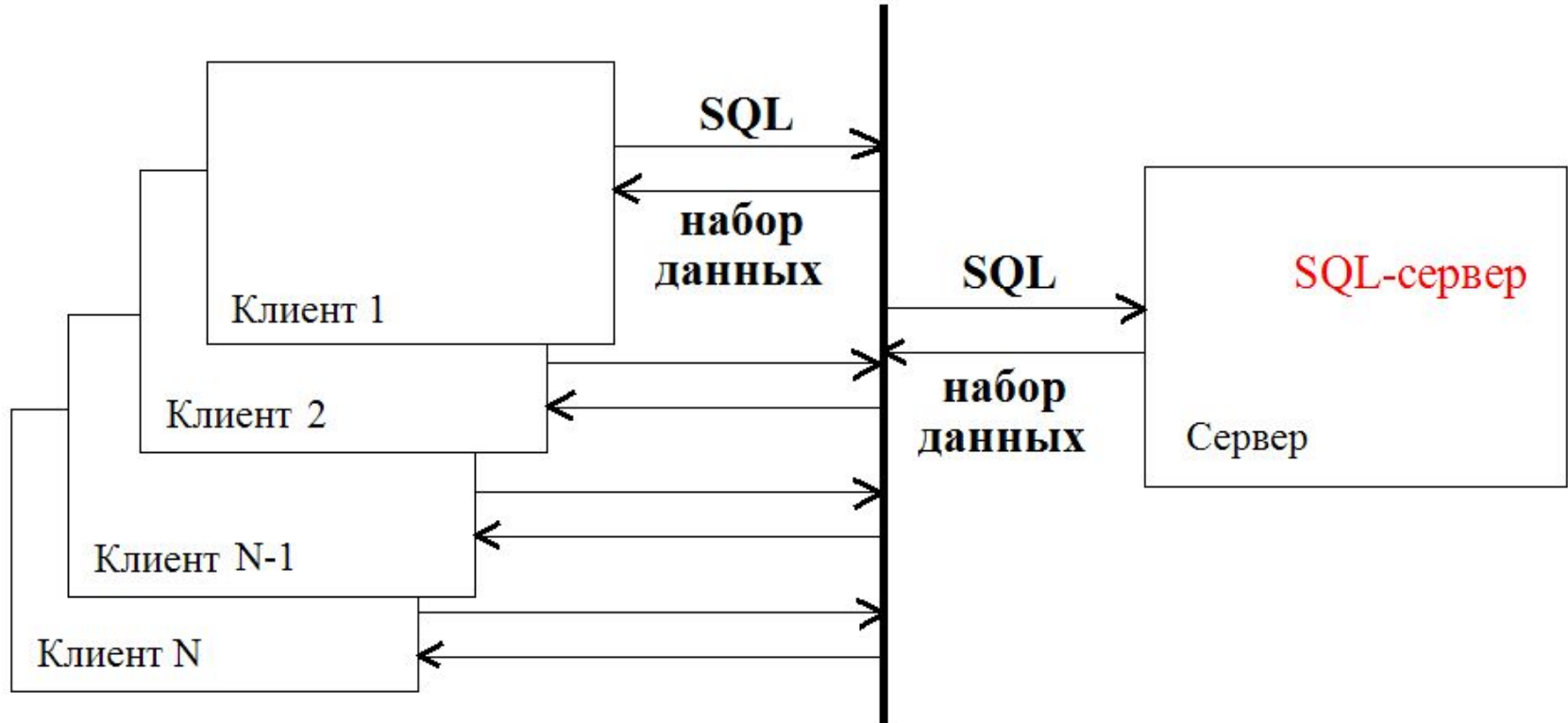


1. Регистрация на сервере при входе в клиентскую часть
2. Формирование SQL-запроса и его отправка SQL-серверу
4. Реализация вычислительных функций над полученным набором данных

3. Выполнение полученного SQL-запроса и отправка наборов данных серверу



# Архитектура клиент-сервер



1. Ввод и отображение данных
2. Реализация вычислительных функций над наборами данных

1. Хранение файла БД
2. Доступ к данным и поиск по определенным критериям



# Достоинства и недостатки архитектуры клиент-сервер

## Достоинства:

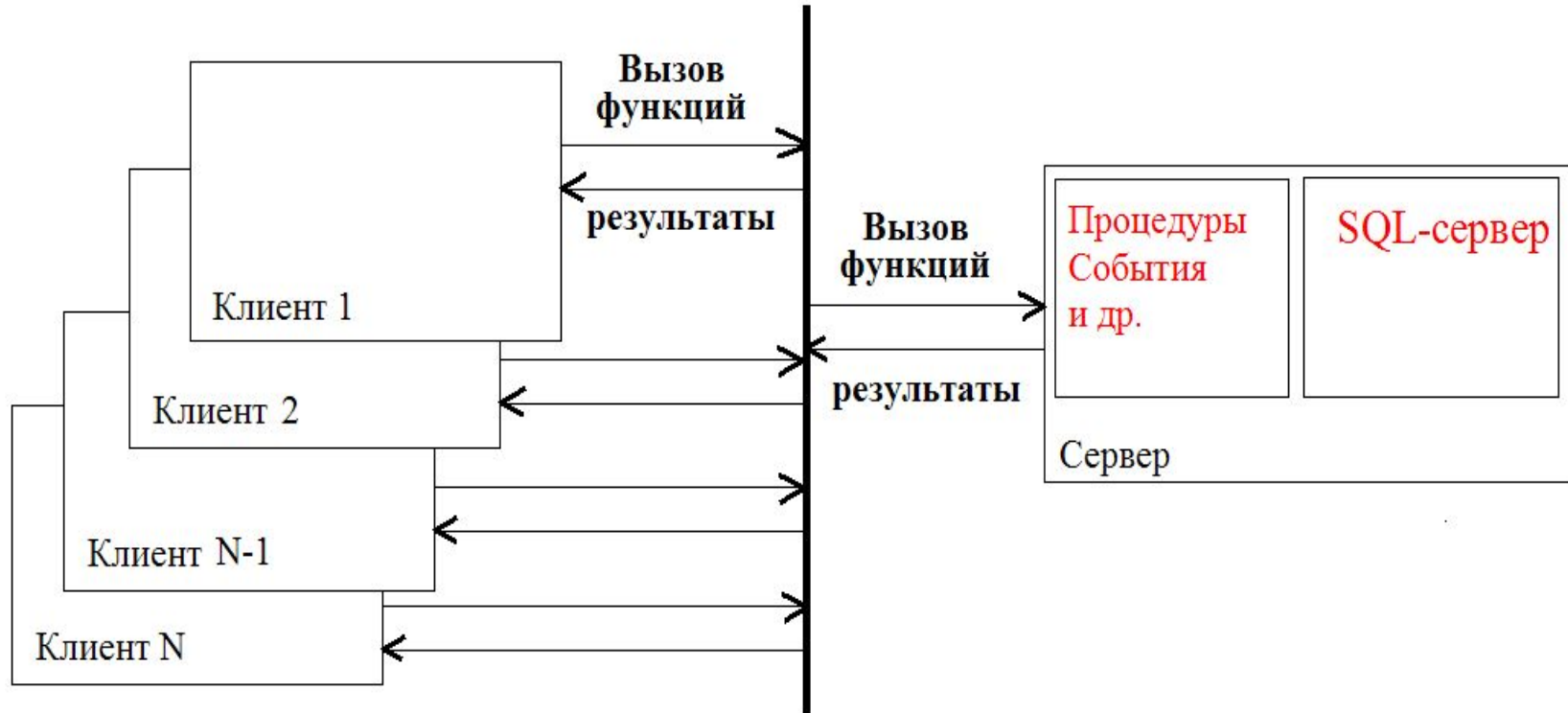
1. Более низкий трафик сети, чем в модели файл-сервер
2. SQL-сервер обеспечивает функции по обеспечению целостности и безопасности данных

## Недостатки:

1. В определенных случаях некоторые наборы данных могут занимать достаточно существенный объем



# Двухуровневая архитектура



1. Ввод и отображение данных

1. Хранение файла БД

2. Доступ к данным и поиск по критериям

3. Реализация вычислительных функций над наборами данных





# Достоинства и недостатки двухуровневой архитектуры

## Достоинства:

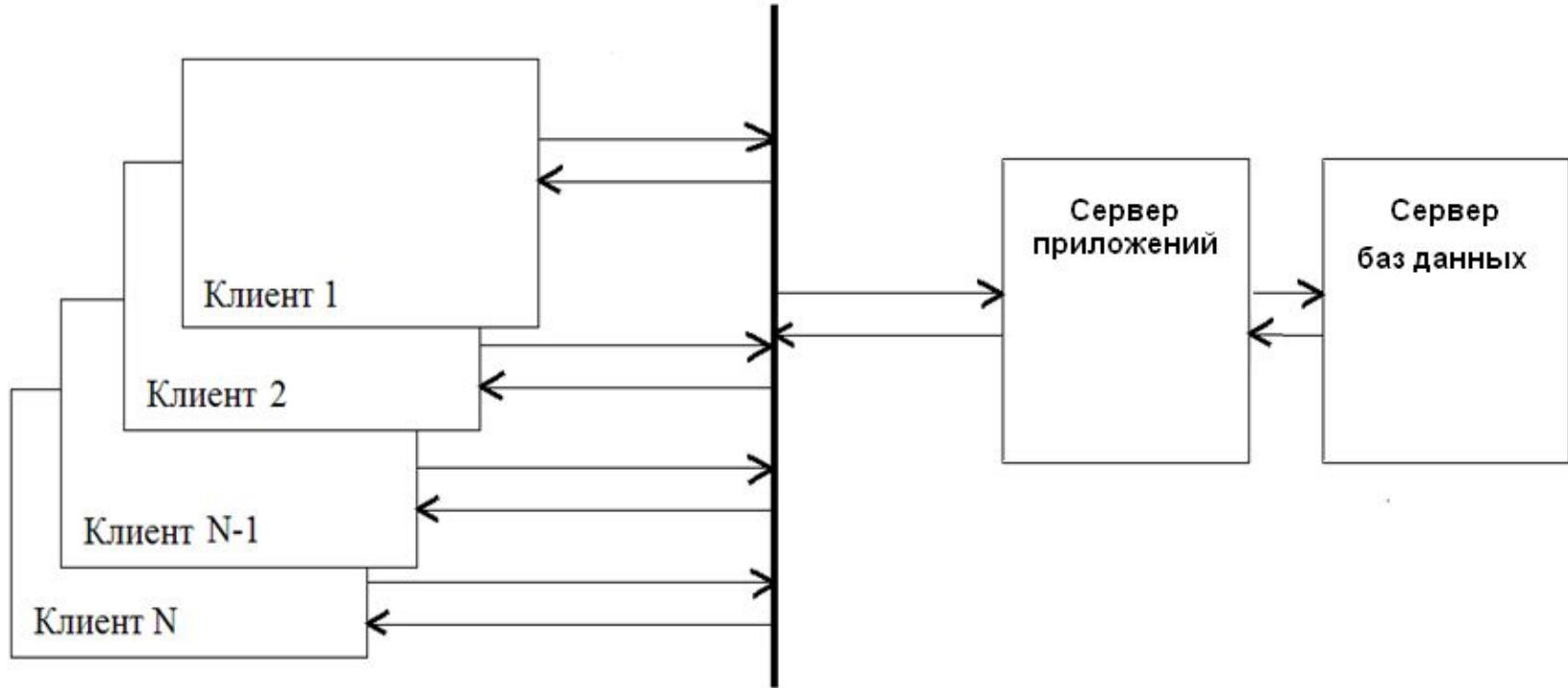
1. Существенное снижение трафика сети по сравнению с моделью клиент-сервер
2. Высокая надежность хранения и обработки данных

## Недостатки:

1. Высокие требования к вычислительной установке сервера по объему дискового пространства и быстродействию



# Трёхуровневая архитектура



**Трёхуровневая архитектура** (трёхзвенная архитектура) предполагает наличие следующих компонентов приложения: **клиентское приложение** ("тонкий клиент" или терминал), подключенное к **серверу приложений**, который в свою очередь подключен к серверу базы данных.



# Достоинства трехуровневой архитектуры

1. Изолированность уровней друг от друга позволяет (при правильном развертывании архитектуры) быстро и простыми средствами переконфигурировать систему при возникновении сбоев или при плановом обслуживании на одном из уровней
2. Высокая безопасность
3. Низкие требования к скорости сети между терминалами и сервером приложений
4. Низкие требования к производительности и техническим характеристикам терминалов, как следствие снижение их стоимости



# Недостатки трехуровневой архитектуры

1. Высокие требования к производительности серверов приложений и сервера базы данных, а, значит, и высокая стоимость серверного оборудования
2. Высокие требования к скорости сети между сервером базы данных и серверами приложений
3. Более сложные операции разворачивания и администрирования



# Модели данных

Хранимые в базе данные имеют определенную логическую структуру – описываются некоторой моделью представления данных (моделью данных), поддерживаемой СУБД.

**Модель данных** определяет способ организации данных, ограничения целостности и множество операций, допустимых над объектом.



К числу классических относятся следующие модели данных:

- ❖ иерархическая
- ❖ сетевая
- ❖ реляционная





# Иерархическая модель данных

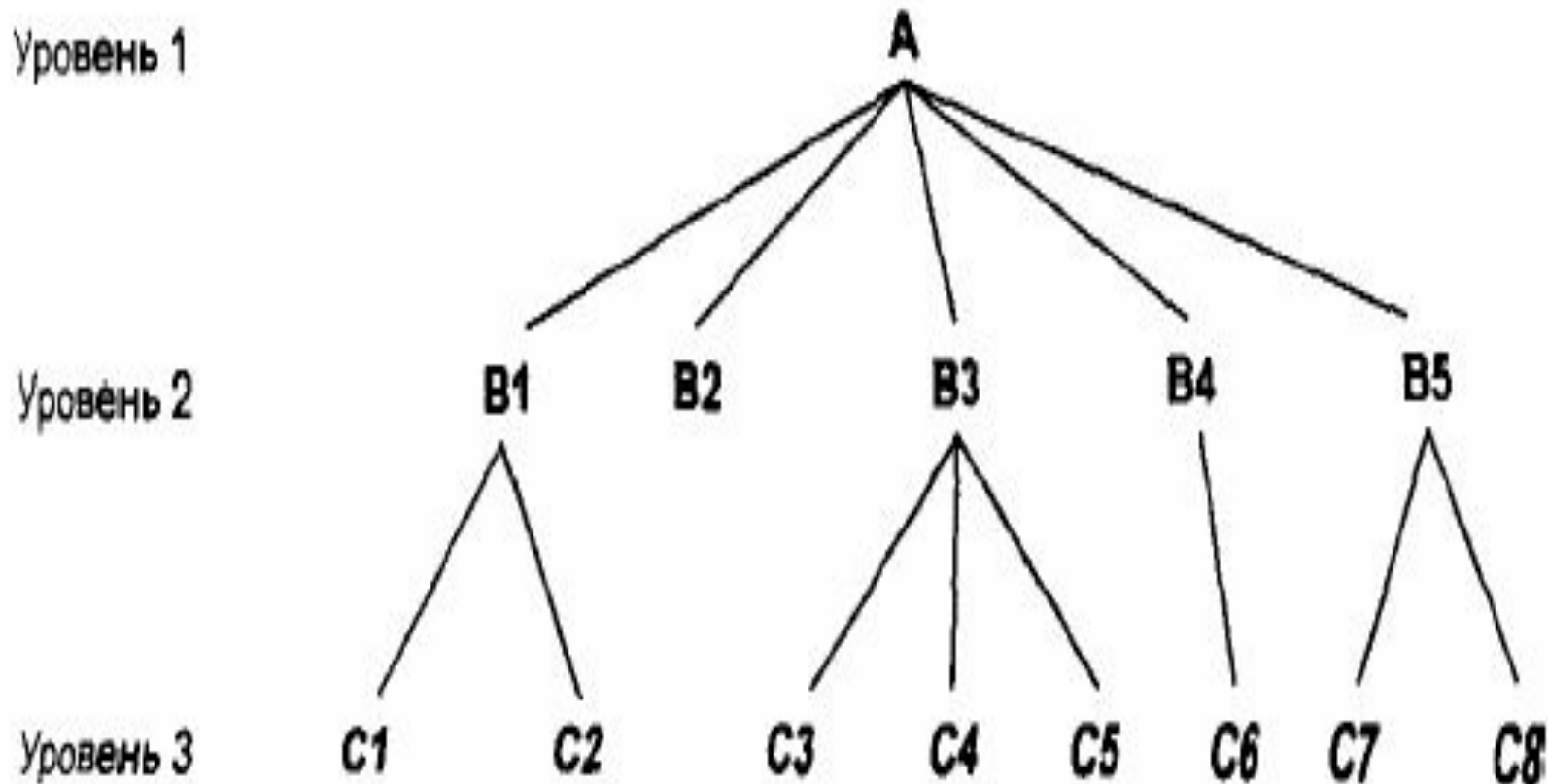
Иерархическая модель была разработана исторически первой.

На основе данной модели в конце 60 – начале 70 гг. была создана первая профессиональная СУБД **IMS** фирмы IBM.



# Иерархическая модель данных

Связи между данными описываются с помощью упорядоченного графа или дерева





# Иерархическая модель данных





# Достоинство и недостатки иерархической модели

## Достоинство:

1. Достаточно высокие показатели времени выполнения операций над данными

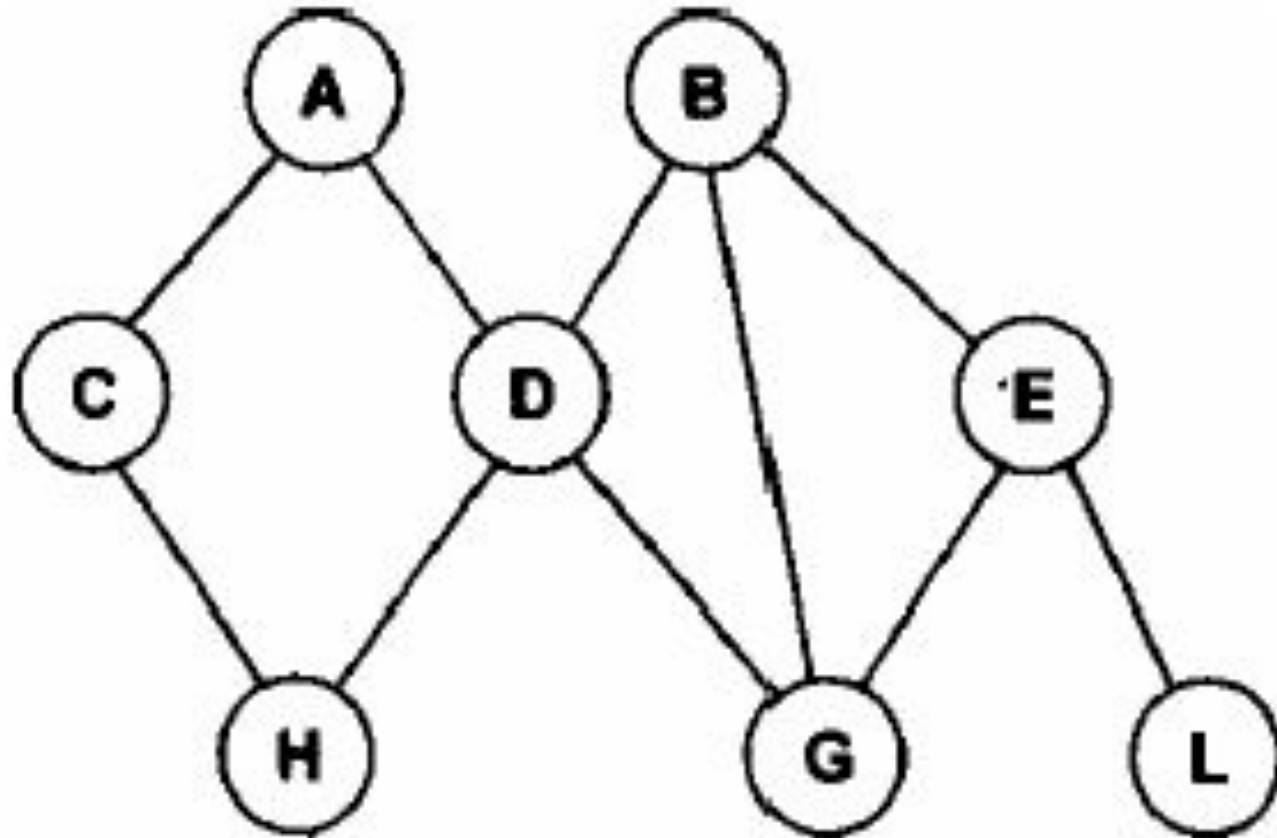
## Недостатки:

1. Сложность понимания для обычного пользователя
2. Присутствие избыточности



# Сетевая модель данных

Связи между данными описываются с помощью произвольного графа





# Сетевая модель данных

Студент (номер зачетной книжки, фамилия, группа)

87695  
Иванов  
111

85495  
Петров  
112

87495  
Сидоров  
113

Работа (шифр, руководитель, область)

1006  
Сергеев П.И.  
Информатика

1009  
Некрасова Г.П.  
Экономика

1008  
Кириллов В.П.  
Экология

1005  
Павлова И.М.  
История







# Достоинства и недостатки сетевой модели

## Достоинства:

1. Минимальная избыточность
2. В сравнении с иерархической моделью сетевая модель предоставляет большие возможности в смысле допустимости образования новых связей
3. Эффективная реализация по показателям затрат памяти.

## Недостатки:

1. Сложность понимания для обычного пользователя
2. Ослаблен контроль правильности образования связей



# Реляционная модель данных

Реляционная модель впервые предложена  
Эдгаром Коддом в 1970 г.

Основывается на понятии **отношение** (relation).  
Графически отношение представляется в виде  
**двумерной таблицы**.

В реляционной СУБД предполагается, что  
пользователь воспринимает БД как набор  
таблиц.



# Реляционная модель данных

Примеры реляционных СУБД:

- ❖ Microsoft Access
- ❖ Paradox
- ❖ dBASE
- ❖ FoxPro
- ❖ Clarion
- ❖ DB2
- ❖ Oracle

Последние версии реляционных СУБД имеют некоторые свойства объектно-ориентированных систем. Такие СУБД часто называют объектно-реляционными. Примером такой системы можно считать продукт **Oracle 10g**.



# Реляционная модель данных

№ личного дела	Фамилия	Имя	Отчество	Дата рождения	Группа
16493	Сергеев	Петр	Михайлович	01.01.76	111
16593	Петрова	Анна	Владимировна	15.03.75	112
16693	Петров	Андрей	Борисович	14.04.76	113



# Достоинства и недостатки реляционной модели

## **Достоинство:**

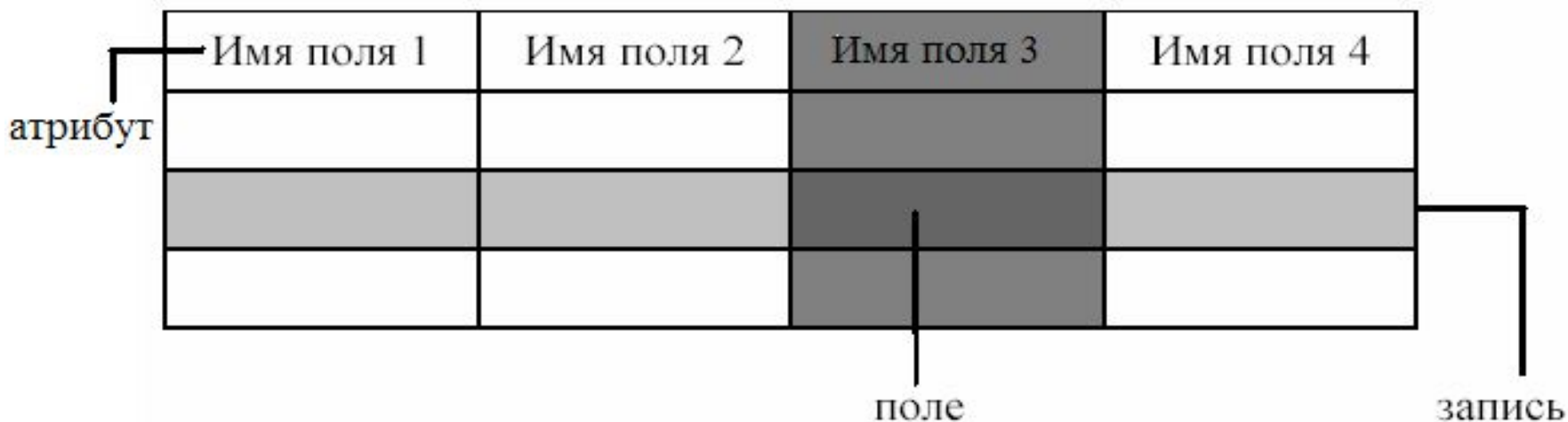
1. Простота и понятность для широкого пользователя, что явилось причиной ее широкого распространения.

## **Недостаток:**

1. Необходимая избыточность из-за связей между таблицами.



# Реляционная модель данных



В реляционной модели существуют альтернативные варианты терминов:

Альтернативный вариант	Альтернативный вариант	Альтернативный вариант
Таблица	Файл	Отношение
Строка	Запись	Кортеж
Столбец	Поле	Домен





# Первичный ключ

№ личного дела	Фамилия	Имя	Отчество	Дата рождения	Группа
16493	Сергеев	Петр	Михайлович	01.01.76	111
16593	Петрова	Анна	Владимировна	15.03.75	112
16693	Петров	Андрей	Борисович	14.04.76	113

**Ключом** отношения, или **первичным ключом**, называется атрибут отношения (набор атрибутов), однозначно идентифицирующий каждый из его кортежей.

Студент(№ личного дела, Фамилия, Имя, Отчество, Дата рождения, Группа)



# Внешний ключ

Логические связи между отношениями устанавливаются с помощью внешних ключей.

**Внешний ключ** – это атрибут (набор атрибутов) одного отношения, являющийся ключом другого отношения.

**Студент**

внешний  
ключ

<u>№ личного дела</u>	Фамилия	Имя	Отчество	Дата рождения	Группа
-----------------------	---------	-----	----------	---------------	--------

первичный  
ключ

**Группа**

<u>Группа</u>	Специальность	Староста
---------------	---------------	----------

первичный  
ключ



## Проблема: осуществить в таблице поиск по Фамилии

Располо- жение	...	Фамилия	...	Группа	...
1		Алексеев		151	
2		Яковлев		152	
3		Михайлов		151	
4		Афанасьев		511	
...		...			
1000		Кузнецов		221	



# Индексы

**Индекс** – средство ускорения операции поиска записей в таблице, а также выполнения других операций, использующих поиск (извлечение, модификация, сортировка и т.д.)

Фамилия	Расположение		Расположение	...	Фамилия	...	Группа	...
Алексеев	1	→	1		Алексеев		151	
Афанасьев	4	→	2		Яковлев		152	
Кузнецов	1000	→	3		Михайлов		151	
Михайлов	3	→	4		Афанасьев		511	
...			...		...			
Яковлев	2	→	1000		Кузнецов		221	



# Виды индексов

## Индексы

### Первичный

Ключевое поле таблицы всегда индексируется, поэтому для него не требуется дополнительно определять индекс.

### Вторичный

- Используются для ускорения выполнения запросов.
- Может быть несколько.
- Может входить любое количество полей.
- Одно и то же поле может входить в разные индексы.



# Проблема формирования отношений

**Пример:** для учебной части факультета создать БД о студентах

Имена выделенных атрибутов и их краткие характеристики:

**№** - номер личного дела студента

**Фамилия** – фамилия студента

**Имя** – имя студента

**Дата рождения** – дата рождения студента

**Группа** – номер группы, в которой учится студент

**Специальность** – специальность, на которой обучается студент





# Проблема формирования отношений

<b>№</b>	<b>Фамилия</b>	<b>Имя</b>	<b>Дата рождения</b>	<b>Группа</b>	<b>Специальность</b>
15345	Иванов	Иван	15.04.1989	392	Информатика-менеджмент
15349	Медведева	Анна	13.02.1989	392	Информатика-менеджмент
15310	Петров	Михаил	12.11.1989	392	Информатика-менеджмент
15259	Сидоров	Николай	26.01.1987	591	Информатика-английский язык
15263	Санин	Александр	20.10.1987	591	Информатика-английский язык



# Проблема формирования отношений

№	Фамилия	Имя	Дата рождения	Группа	Специальность
15345	Иванов	Иван	15.04.1989	392	Информатика-менеджмент
15349	Медведева	Анна	13.02.1989	392	Информатика-менеджмент
15310	Петров	Михаил	12.11.1989	392	Информатика-менеджмент
15259	Сидоров	Николай	26.01.1987	591	Информатика-английский язык
15263	Санин	Александр	20.10.1987	591	Информатика-английский язык



# Нормализация

**Нормализация отношений** – правила формирования отношений (таблиц), которые позволяют устранить дублирование, противоречивость хранимых в базе данных.



# Нормализация

Э. Коддом разработаны **три нормальные формы отношений** и предложен механизм, позволяющий **любое отношение преобразовать к третьей нормальной форме.**



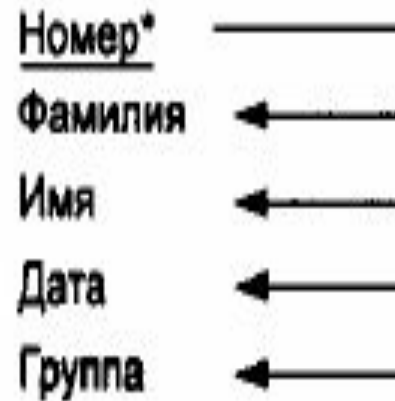
# Разбиение отношения

Студент в группе



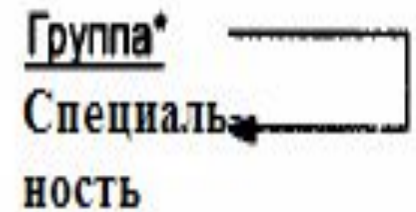
=

Студент



+

Группа







# Разбиение отношения

## Студент

<b>№</b>	<b>Фамилия</b>	<b>Имя</b>	<b>Дата рождения</b>	<b>Группа</b>
15345	Иванов	Иван	15.04.1989	392
15349	Медведева	Анна	13.02.1989	392
15310	Петров	Михаил	12.11.1989	392
15259	Сидоров	Николай	26.01.1987	591
15263	Санин	Александр	20.10.1987	591

## Группа

<b>Группа</b>	<b>Специальность</b>
392	Информатика-менеджмент
591	Информатика-английский язык





# Проблема формирования отношений

**Пример:** для учебной части факультета создать БД о студентах

Имена выделенных атрибутов и их краткие характеристики:

**№** - номер личного дела студента

**Фамилия** – фамилия студента

**Имя** – имя студента

**Дата рождения** – дата рождения студента

**Группа** – номер группы, в которой учится студент

**Специальность** – специальность, которой обучается студент

**Семестр** – номера семестров обучения

**Предмет** – предмет, изучаемый студентом

**Оценка** – экзаменационная оценка за предмет



# Проблема формирования отношений

№	Фамилия	Имя	Дата рождения	Группа	Специальность	Семестр	Предмет	Оценка
15345	Иванов	Иван	15.04.1989	392	ИМ	6	Англ.язык	Хорошо
15345	Иванов	Иван	15.04.1989	392	ИМ	6	Теор.алгоритмов	Отлично
15345	Иванов	Иван	15.04.1989	392	ИМ	6	Химия	Хорошо
15310	Петров	Михаил	12.11.1989	392	ИМ	6	Англ.язык	Удовлет.
15310	Петров	Михаил	12.11.1989	392	ИМ	6	Теор.алгоритмов	Удовлет.
15310	Петров	Михаил	12.11.1989	392	ИМ	6	Химия	Хорошо
15259	Сидоров	Николай	26.01.1987	591	ИА	10	Архитектура ЭВМ	Отлично
15259	Сидоров	Николай	26.01.1987	591	ИА	10	Комп.моделирование	Отлично



# Проблема формирования отношений

## Проблема обновления данных

№	Фамилия	Имя	Дата рождения	Группа	Специальность	Семестр	Предмет	Оценка
15345	Иванов	Иван	15.04.1989	492	ИМ	6	Англ. язык	Хорошо
15345	Иванов	Иван	15.04.1989	492	ИМ	6	Теор. алгоритмов	Отлично
15345	Иванов	Иван	15.04.1989	392 ?	ИМ	6	Химия	Хорошо
15310	Петров	Михаил	12.11.1989	392	ИМ	6	Англ. язык	Удовлет.
15310	Петров	Михаил	12.11.1989	392	ИМ	6	Теор. алгоритмов	Удовлет.
15310	Петров	Михаил	12.11.1989	392	ИМ	6	Химия	Хорошо
15259	Сидоров	Николай	26.01.1987	591	ИА	10	Архитектура ЭВМ	Отлично
15259	Сидоров	Николай	26.01.1987	591	ИА	10	Комп. моделирование	Отлично





# Проблема формирования отношений

## Проблема вставки новых данных

№	Фамилия	Имя	Дата рождения	Группа	Специальность	Семестр	Предмет	Оценка
15345	Иванов	Иван	15.04.1989	392	ИМ	6	Англ.язык	Хорошо
15345	Иванов	Иван	15.04.1989	392	ИМ	6	Теор.алгоритмов	Отлично
15345	Иванов	Иван	15.04.1989	392	ИМ	6	Химия	Хорошо
15310	Петров	Михаил	12.11.1989	392	ИМ	6	Англ.язык	Удовлет.
15310	Петров	Михаил	12.11.1989	392	ИМ	6	Теор.алгоритмов	Удовлет.
15310	Петров	Михаил	12.11.1989	392	ИМ	6	Химия	Хорошо
15259	Сидоров	Николай	26.01.1987	591	ИА	10	Архитектура ЭВМ	Отлично
15259	Сидоров	Николай	26.01.1987	591	ИА	10	Комп. моделирование	Отлично
15402	Степанов	Андрей	29.03.1991	191	ИТвО	1	-----	-----



# Первая нормальная форма (1НФ)

Отношение находится в **1НФ**, если в каждой ячейке всегда находится единственное атомарное значение, и никогда не может быть множества таких значений.



# Первая нормальная форма (1НФ)

№	Фамилия	Имя	Дата рождения	Группа	Специальность	Семестр	Предмет	Оценка
15345	Иванов	Иван	15.04.1989	392	ИМ	6	Англ. язык Теор. алгоритмов Химия	Хорошо Отлично Хорошо
15289	Петров	Михаил	12.11.1988	392	ИМ	6	Англ. язык Теор. алгоритмов Химия	Удовлет. Удовлет. Хорошо
15259	Сидоров	Николай	26.01.1987	591	ИА	10	Архитектура ЭВМ Комп. моделирование	Отлично Отлично

Отношение **не** находится в 1НФ





# Первая нормальная форма (1НФ)

№	Фамилия	Имя	Дата рождения	Группа	Специальность	Семестр	Предмет	Оценка
15345	Иванов	Иван	15.04.1989	392	ИМ	6	Англ.язык	Хорошо
15345	Иванов	Иван	15.04.1989	392	ИМ	6	Теор.алгоритмов	Отлично
15345	Иванов	Иван	15.04.1989	392	ИМ	6	Химия	Хорошо
15310	Петров	Михаил	12.11.1989	392	ИМ	6	Англ.язык	Удовлет.
15310	Петров	Михаил	12.11.1989	392	ИМ	6	Теор.алгоритмов	Удовлет.
15310	Петров	Михаил	12.11.1989	392	ИМ	6	Химия	Хорошо
15259	Сидоров	Николай	26.01.1987	591	ИА	10	Архитектура ЭВМ	Отлично
15259	Сидоров	Николай	26.01.1987	591	ИА	10	Комп.моделирование	Отлично

Отношение находится в 1НФ



# Функциональная зависимость

Нормализация основывается на наличии функциональной зависимости между атрибутами отношения.



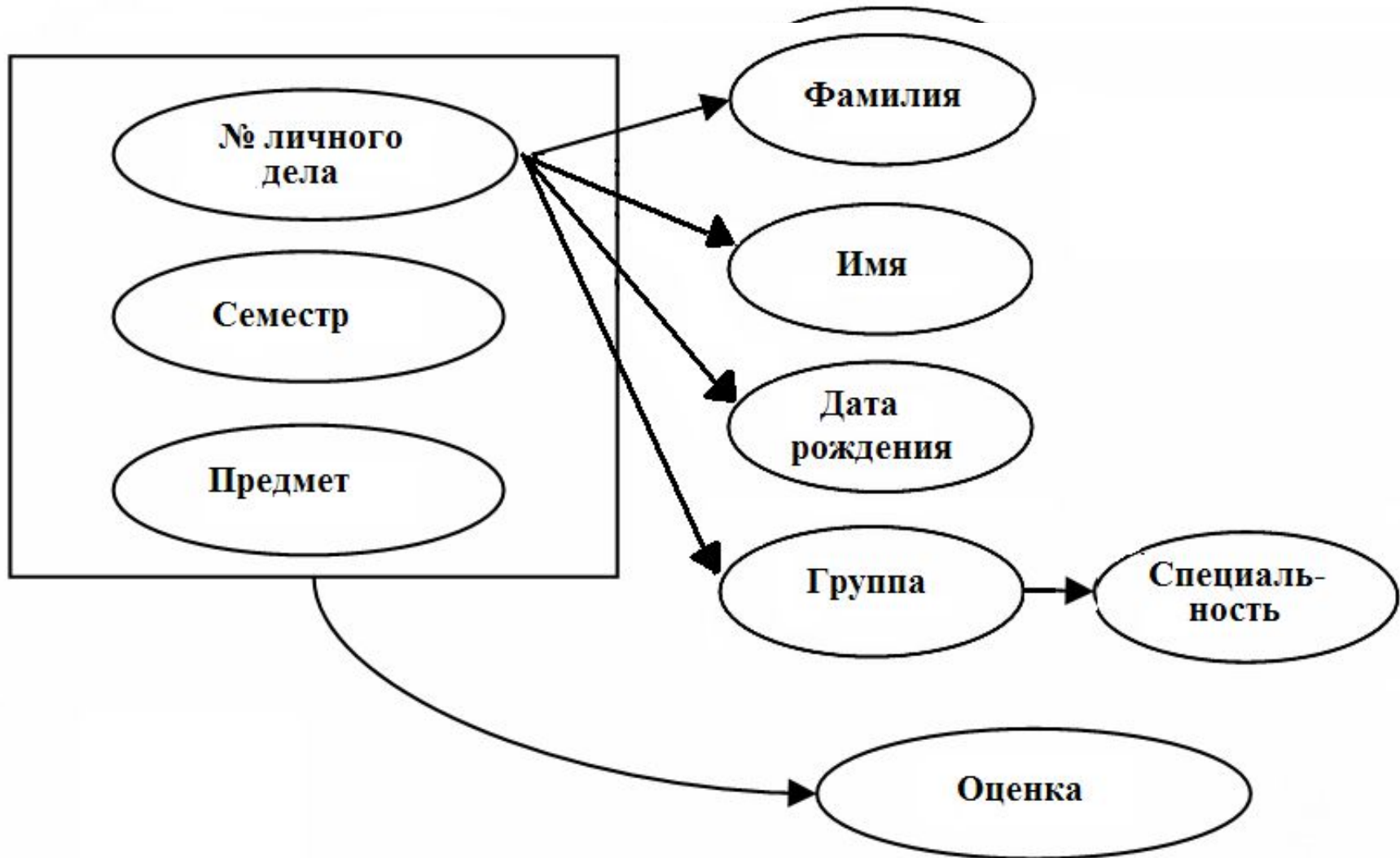
# Функциональная зависимость

Атрибут В отношения функционально зависит от атрибута А того же отношения в том и только том случае, когда в любой заданный момент времени для каждого из различных значений атрибута А обязательно существует только одно значение поля В. (допускается, что атрибуты А и В могут быть составными)





# Диаграмма функциональных зависимостей для примера БД «Студент»

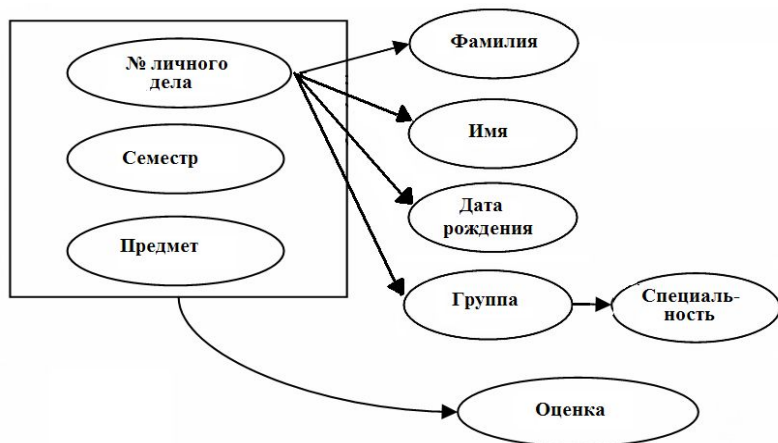






# Причина избыточности

№	Фамилия	Имя	Дата рождения	Группа	Специальность	Семестр	Предмет	Оценка
15345	Иванов	Иван	15.04.1989	392	ИМ	6	Англ.язык	Хорошо
15345	Иванов	Иван	15.04.1989	392	ИМ	6	Теор.алгоритмов	Отлично
15345	Иванов	Иван	15.04.1989	392	ИМ	6	Химия	Хорошо
15310	Петров	Михаил	12.11.1989	392	ИМ	6	Англ.язык	Удовлет.
15310	Петров	Михаил	12.11.1989	392	ИМ	6	Теор.алгоритмов	Удовлет.
15310	Петров	Михаил	12.11.1989	392	ИМ	6	Химия	Хорошо
15259	Сидоров	Николай	26.01.1987	591	ИА	10	Архитектура ЭВМ	Отлично
15259	Сидоров	Николай	26.01.1987	591	ИА	10	Комп.моделирование	Отлично



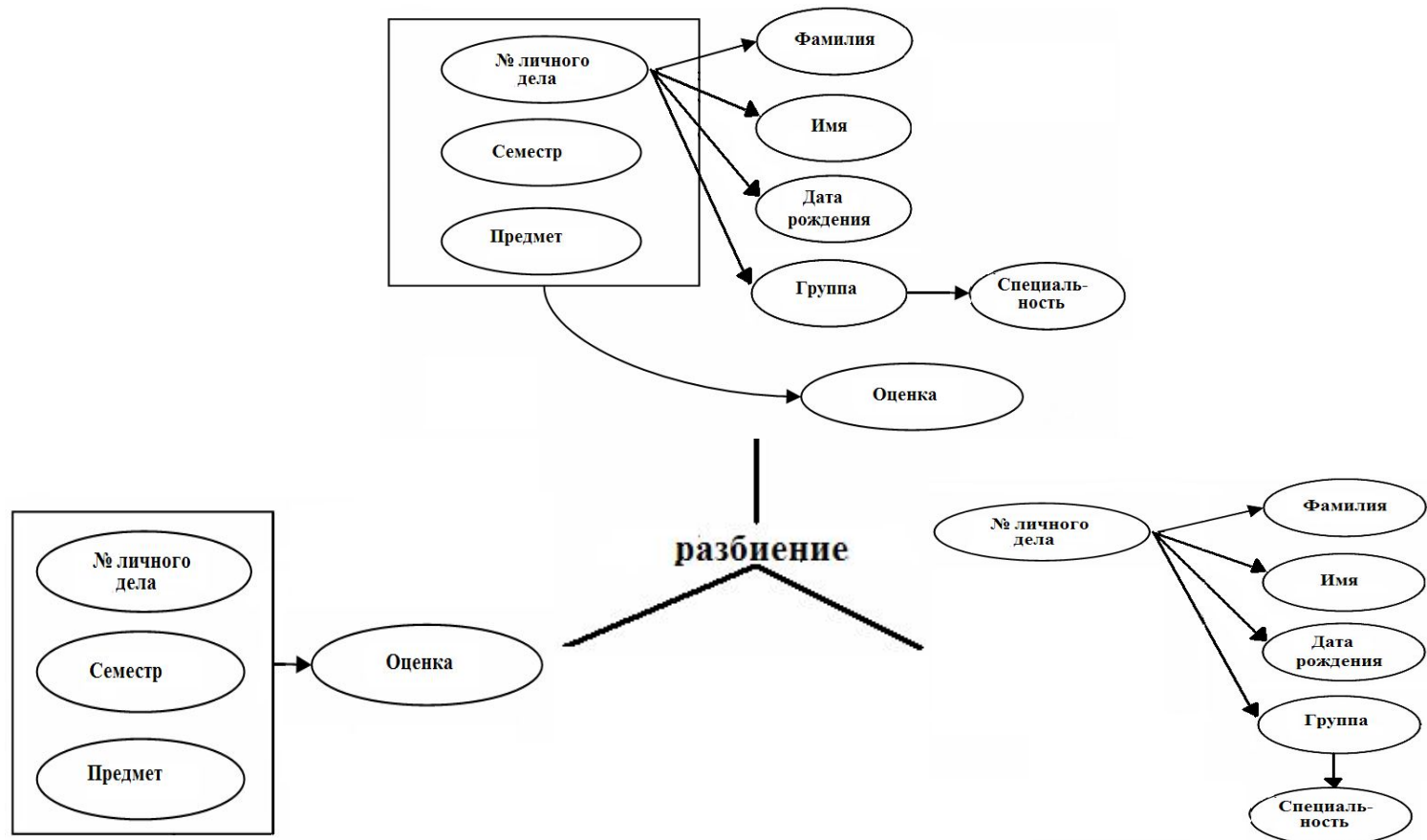
## Причина избыточности:

Фамилия, Имя, Дата рождения, Группа, Специальность зависят от атрибута № личного дела, являющегося **частью составного ключа**



# Вторая нормальная форма (2НФ)

Отношение находится в **2НФ**, если оно не содержит неключевых атрибутов, функционально зависящих от части ключа

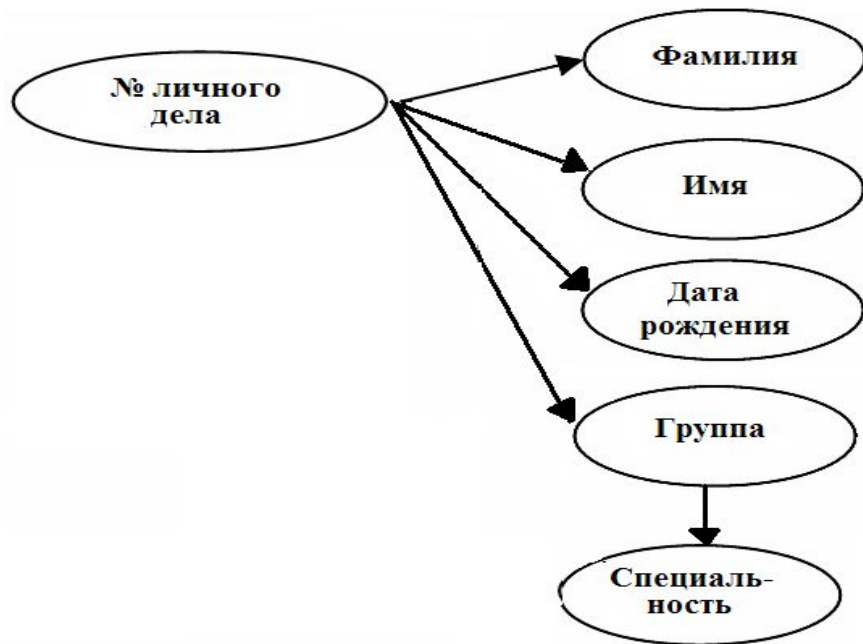






# Причина избыточности

№	Фамилия	Имя	Дата рождения	Группа	Специальность
15345	Иванов	Иван	15.04.1989	392	ИМ
15310	Петров	Михаил	12.11.1989	392	ИМ
15259	Сидоров	Николай	26.01.1987	591	ИА

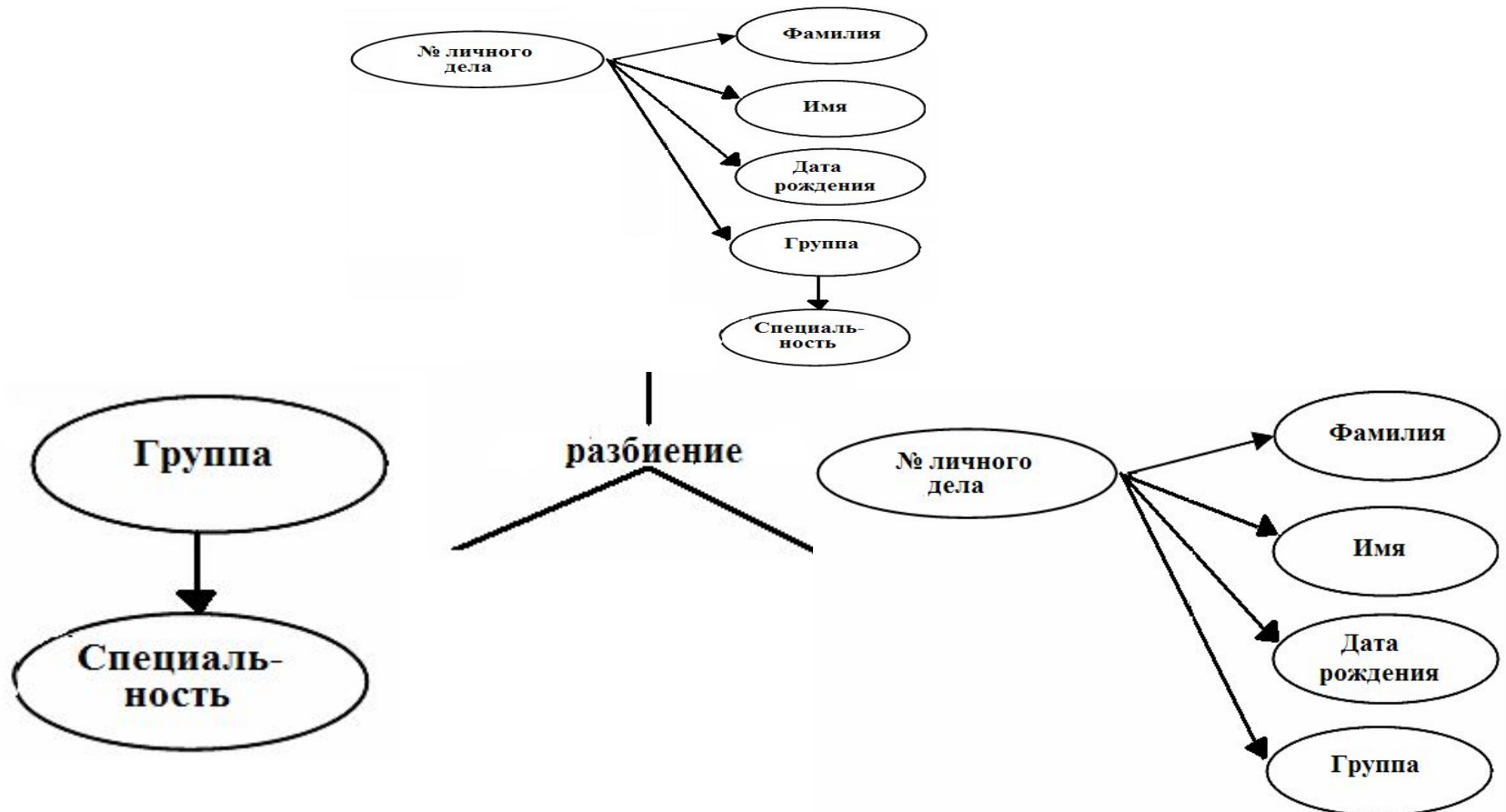


**Причина избыточности:**  
транзитивная зависимость  
между атрибутами отношения  
№личного дела->Группа ->  
->Специальность



# Третья нормальная форма (3НФ)

Отношение находится в **3НФ**, если оно не содержит неключевых атрибутов, транзитивно зависящих от части ключа





# Результат проектирования БД «Студент»

№	Фамилия	Имя	Дата рождения	Группа
15345	Иванов	Иван	15.04.1989	392
15310	Петров	Михаил	12.11.1989	392
15259	Сидоров	Николай	26.01.1987	591

Группа	Специальность
392	ИМ
591	ИА

№	Семестр	Предмет	Оценка
15345	6	Англ.язык	Хорошо
15345	6	Теор.алгоритмов	Отлично
15345	6	Химия	Хорошо
15310	6	Англ.язык	Удовлет.
15310	6	Теор.алгоритмов	Удовлет.
15310	6	Химия	Хорошо
15259	10	Архитектура ЭВМ	Отлично
15259	10	Комп.моделирование	Отлично



# SQL

Доступ к информации, содержащейся в реляционных базах данных, для пользователей, программ и вычислительных систем обеспечивает язык запросов **SQL** (Structured Query Language)





# Достоинства SQL

- Независимость от конкретных СУБД – все распространенные СУБД используют SQL.
- Приложения, созданные с помощью SQL, допускают использование как для локальных БД, так и для клиент-серверных систем.
- Операторы SQL употребляются как для интерактивного, так и программного доступа, поэтому части программ, содержащие обращение к БД, можно вначале проверить в интерактивном режиме, а затем встраивать в программу.



# Реляционная алгебра

SQL основан на операциях реляционной алгебры.

**Реляционная алгебра** – набор операций, выполняемых над отношениями.

Реляционная алгебра разработана Э.Коддом в рамках реляционной модели

Применяя операции реляционной алгебры к одним отношениям можно получить другие отношения





# Основные операции реляционной алгебры

## *Объединение*

Студенты 392

№	Фамилия	Имя	Дата рождения	Группа
15345	Иванов	Иван	15.04.1989	392
15349	Медведева	Анна	13.02.1989	392
15310	Петров	Михаил	12.11.1989	392

Студенты 591

№	Фамилия	Имя	Дата рождения	Группа
15259	Сидоров	Николай	26.01.1987	591
15263	Санин	Александр	20.10.1987	591

Объединение студентов 392 и 591 группы

№	Фамилия	Имя	Дата рождения	Группа
15345	Иванов	Иван	15.04.1989	392
15349	Медведева	Анна	13.02.1989	392
15310	Петров	Михаил	12.11.1989	392
15259	Сидоров	Николай	26.01.1987	591
15263	Санин	Александр	20.10.1987	591



# Основные операции реляционной алгебры

**Объединением** двух совместимых по типу отношений  $A$  и  $B$  называется отношение с тем же заголовком, что и у отношений  $A$  и  $B$ , и телом, состоящим из кортежей, принадлежащих или  $A$ , или  $B$ , или обоим отношениям.

Отношения называют **совместимыми по типу**, если они имеют идентичные заголовки, также атрибуты с одинаковыми именами определены на одних и тех же доменах.



# Основные операции реляционной алгебры

## *Пересечение*

Студенты 392 и 591 группы

№	Фамилия	Имя	Дата рождения	Группа
15345	Иванов	Иван	15.04.1989	392
15349	Медведева	Анна	13.02.1989	392
15310	Петров	Михаил	12.11.1989	392
15259	Сидоров	Николай	26.01.1987	591
15263	Санин	Александр	20.10.1987	591

Студенты 392

№	Фамилия	Имя	Дата рождения	Группа
15345	Иванов	Иван	15.04.1989	392
15350	Морозова	Ирина	21.12.1989	392
15359	Петрова	Мария	12.11.1989	392

Результат пересечения

№	Фамилия	Имя	Дата рождения	Группа
15345	Иванов	Иван	15.04.1989	392



# Основные операции реляционной алгебры

*Пересечением* двух совместимых по типу отношений  $A$  и  $B$  называется отношение с тем же заголовком, что и у отношений  $A$  и  $B$ , и телом, состоящим из кортежей, принадлежащих одновременно обоим отношениям  $A$  и  $B$ .





# Основные операции реляционной алгебры

## *Вычитание*

Студенты 392 и 591 группы

№	Фамилия	Имя	Дата рождения	Группа
15345	Иванов	Иван	15.04.1989	392
15349	Медведева	Анна	13.02.1989	392
15310	Петров	Михаил	12.11.1989	392
15259	Сидоров	Николай	26.01.1987	591
15263	Санин	Александр	20.10.1987	591

Студенты 392

№	Фамилия	Имя	Дата рождения	Группа
15345	Иванов	Иван	15.04.1989	392
15350	Морозова	Ирина	21.12.1989	392
15359	Петрова	Мария	12.11.1989	392

Результат вычитания

№	Фамилия	Имя	Дата рождения	Группа
15349	Медведева	Анна	13.02.1989	392
15310	Петров	Михаил	12.11.1989	392
15259	Сидоров	Николай	26.01.1987	591
15263	Санин	Александр	20.10.1987	591





# Основные операции реляционной алгебры

**Вычитанием** двух совместимых по типу отношений  $A$  и  $B$  называется отношение с тем же заголовком, что и у отношений  $A$  и  $B$ , и телом, состоящим из кортежей, принадлежащих отношению  $A$  и не принадлежащих отношению  $B$ .



# Основные операции реляционной алгебры

## *Выборка*

Студенты 392 и 591 группы

<b>№</b>	<b>Фамилия</b>	<b>Имя</b>	<b>Дата рождения</b>	<b>Группа</b>
15345	Иванов	Иван	15.04.1989	392
15349	Медведева	Анна	13.02.1989	392
15310	Петров	Михаил	12.11.1989	392
15259	Сидоров	Николай	26.01.1987	591
15263	Санин	Александр	20.10.1987	591

Результат выборки студентов 591 группы

<b>№</b>	<b>Фамилия</b>	<b>Имя</b>	<b>Дата рождения</b>	<b>Группа</b>
15259	Сидоров	Николай	26.01.1987	591
15263	Санин	Александр	20.10.1987	591



# Основные операции реляционной алгебры

*Выборкой* на отношении  $A$  с условием  $s$  называется отношение с тем же заголовком, что и у отношения  $A$ , и телом, состоящем из кортежей, значения атрибутов которых при подстановке в условие  $s$  дают значение ИСТИНА.



# Оператор выбора языка SQL

**SELECT [DISTINCT] элементы**  
**FROM** таблица(цы)  
**[WHERE** условие]  
**[GROUP BY** поле(я) **[HAVING** условие]  
**[ORDER BY** поле(я)]

Производит выборку указанных элементов из указанных таблиц в соответствии с указанными условиями. Результатом является новая таблица.



# Оператор выбора языка SQL

**SELECT [DISTINC] элементы**  
**FROM** таблица(цы)  
**[WHERE** условие]  
**[GROUP BY** поле(я) **[HAVING** условие]  
**[ORDER BY** поле(я)]

SELECT – выбрать

DISTINC – устранить в результирующей таблице  
одинаковые строки

FROM – из (таблиц)

WHERE – где





# Оператор выбора языка SQL

**SELECT [DISTINCT]** элементы

**FROM** таблица(цы)

**[WHERE]** условие

**[GROUP BY]** поле(я) **[HAVING]** условие

**[ORDER BY]** поле(я)

**GROUP BY** – выборка с точностью до группы строк

**HAVING** – условие выборки группы

**ORDER BY** – упорядочивание результата по указанным

ПОЛЯМ



# Оператор выбора языка SQL

Выбрать фамилии всех студентов

Студенты

№	Фамилия	Имя	Дата рождения	Группа
15345	Иванов	Иван	15.04.1989	392
15349	Медведева	Анна	13.02.1989	392
15310	Петров	Михаил	12.11.1989	392
15259	Сидоров	Николай	26.01.1987	591
15263	Санин	Александр	20.10.1987	591

Фамилия
Иванов
Медведева
Петров
Сидоров
Санин

`SELECT Фамилия FROM Студенты`



# Оператор выбора языка SQL

Вывести все сведения о студентах 591 группы,  
упорядочив их по фамилии

Студенты

№	Фамилия	Имя	Дата рождения	Группа
15345	Иванов	Иван	15.04.1989	392
15349	Медведева	Анна	13.02.1989	392
15310	Петров	Михаил	12.11.1989	392
15259	Сидоров	Николай	26.01.1987	591
15263	Санин	Александр	20.10.1987	591

```
SELECT * FROM Студенты WHERE Группа=591  
ORDER BY Фамилия
```



# Оператор выбора языка SQL

Вывести дату рождения студента Петрова

Студенты

№	Фамилия	Имя	Дата рождения	Группа
15345	Иванов	Иван	15.04.1989	392
15349	Медведева	Анна	13.02.1989	392
15310	Петров	Михаил	12.11.1989	392
15259	Сидоров	Николай	26.01.1987	591
15263	Санин	Александр	20.10.1987	591

```
SELECT Дата рождения FROM Студенты  
WHERE Фамилия="Петров"
```





# Возможности SQL

- Создание базы данных и таблицы с полным описанием их структуры
- Выполнение основных операций манипулирования данными, в частности, вставки, модификации и удаления данных из таблиц.
- Выполнение простых и сложных запросов, осуществляющих преобразование данных





# Тенденции развития СУБД

Направление развития реляционных СУБД в последние годы заметно меняется. Если предыдущее десятилетие они развивались, чтобы **обеспечить быстрый доступ к данным**, то теперь часто нужно хранить еще **графику и звук**. Существенно изменилась аппаратная среда - она стала сетевой. С развитием Web появилась необходимость поддерживать HTML - страницы.



# Основные функции СУБД

- ❖ управление данными во внешней памяти (на дисках)
- ❖ управление данными в оперативной памяти
- ❖ журнализация изменений, резервное копирование и восстановление базы данных после сбоев
- ❖ поддержка языка манипулирования данными



## В состав СУБД входят средства для:

- ❖ создания БД и модификации их структуры, создания индексных файлов
- ❖ работы с базами в табличном формате или в виде стандартной формы с расположением полей построчно
- ❖ разработки экранных форм
- ❖ генерации печатных форм
- ❖ генерации запросов очень сложной структуры
- ❖ в системах, ориентированных на разработчика, разработка меню, справочной системы и проекта, включающего все перечисленные выше компоненты и компилирующегося в исполняемую программу



# СУБД

**Система управления базами данных (СУБД) – это комплекс программных и языковых средств, необходимых для создания баз данных, поддержания их в актуальном состоянии и организации поиска в них необходимой информации.**

Термин "сервер баз данных" обычно используют для обозначения всей СУБД, основанной на архитектуре "клиент-сервер", включая и серверную, и клиентскую части.



# СУБД

**Сервер баз данных - СУБД, которая принимает запросы по сети и возвращает информацию, соответствующую запросу.**

**Наиболее распространенными серверами являются в настоящее время Interbase, Microsoft SQL Server, Oracle, IBM DB2, Informix**





# Классификация СУБД

**По типу управляемой базы данных СУБД  
разделяются на:**

- ❖ Сетевые (CronosPlus )
- ❖ Иерархические (IMS, System 2000)
- ❖ Реляционные (MS Access, Paradox, Interbase, FireBird, MySQL, DB2, Oracle, Ingres)
- ❖ Объектно-реляционные (Oracle Database, Microsoft SQL Server 2005)



# Сравнение СУБД

Характеристики	ORACLE	IBM DB2	MS SQL SERVER	SYBASE 12.0
Язык программирования	Java, Delphi PL/SQL	Java, SQL 2000	Transact-SQL	Java, Transact-SQL
Объектно-ориентированное проектирование БД	Да	Да	Нет	Да (через Java)
Мультимедийные типы данных	Да	Да	Ограниченно	Ограниченно
Макс. размер таблиц	Не огр.	64 Гбайт		Не огр.
Макс. число таблиц в БД	Не огр.	Практически не ограничено	Не огр.	Не огр.
Макс. число таблиц на каждое соединение	Не огр.	31	Не огр.	Не огр.
Макс. число пользователей БД	Не огр.	Практически не ограничено	Не огр.	Не огр.
Рекомендуемая емкость ОП на одного пользователя	Изменяемая величина	Локальный пользователь: 550 Кб Удаленный - 250 Кб		50 Кбайт



# Возможности СУБД

**Производительность СУБД** оценивается:

- ❖ скоростью поиска информации;
- ❖ скоростью выполнения операций обновления, вставки, удаления данных;
- ❖ временем выполнения операций импортирования базы данных из других форматов;
- ❖ максимальным числом параллельных обращений к данным в многопользовательском режиме;
- ❖ временем генерации отчета.



# Возможности СУБД

**Обеспечение целостности данных** на уровне базы данных.

Эта характеристика подразумевает наличие средств, позволяющих удостовериться, что информация в базе данных всегда остается корректной и полной:

- ❖ проверка уникальности первичных ключей,
- ❖ ограничение операций над данными,
- ❖ каскадное обновление и удаление данных.



# Возможности СУБД

**Обеспечение безопасности.** Некоторые СУБД предусматривают средства обеспечения безопасности данных. Такие средства обеспечивают выполнение следующих операций:

- ❖ шифрование прикладных программ;
- ❖ шифрование данных;
- ❖ защиту паролем;
- ❖ ограничение уровня доступа (к базе данных, к таблице).





# Возможности СУБД

Доступ к данным посредством языка **SQL**.  
Язык запросов **SQL** реализован в целом ряде популярных СУБД для различных типов ЭВМ либо как базовый, либо как альтернативный. В силу своего широкого использования является международным стандартом языка запросов.