



**КОМПЬЮТЕРНЫЕ
ТЕХНОЛОГИИ В
ЭКСПЕРТНОЙ
ДЕЯТЕЛЬНОСТИ**

Тема: «Введение в технологию баз данных».

Учебные вопросы:

- 1. Подходы к организации данных**
- 2. Уровни представления данных**
- 3. Модели данных**

Литература

основная:

1. **Мишин А.В.** Информационные технологии в профессиональной деятельности: учебное пособие / А.В. Мишин, Л.Е. Мистров, Д. В. Картавцев. – М.: РАП, 2011. – С. 241-259.

дополнительная:

2. **ГОСТ 15971–90.** Системы обработки информации. Термины и определения. – М.: Изд-во стандартов, 1991. – 12 с.
3. **ГОСТ 34.320-96.** Информационные технологии. Система стандартов по базам данных. Концепции и терминология для концептуальной схемы и информационной базы. – М.:ИПК Изд-во стандартов, 2001. – 43 с.
4. **ГОСТ 20886–85.** Организация данных в системах обработки данных. Термины и определения. – М.: Изд-во стандартов, 1986. – 12 с.

1. Подходы к организации данных

Данные – информация, представленная в формализованном виде, пригодном для передачи, интерпретации или обработки с участием человека или автоматическими средствами.

Обработка информации – систематическое выполнение операций над данными, представляющими предназначенную для обработки информацию.

Система обработки информации (СОИ) – совокупность технических средств и программного обеспечения, а также методов обработки информации и действий персонала, обеспечивающая выполнение автоматизированной обработки информации.

Исходные термины

Носитель данных – материальный объект, предназначенный для записи и хранения данных.

Информационная система – концептуальная схема, информационная база и информационный процессор, составляющие вместе формальную систему для хранения и манипулирования информацией.

Управление данными – совокупность функций обеспечения требуемого представления данных, их накопления и хранения, обновления, удаления, поиска по заданному критерию и выдачи данных.

Представление данных – характеристика, выражающая правила кодирования элементов и образования конструкций данных на конкретном уровне рассмотрения в вычислительной системе.

Подходы к организации данных

Организация данных – представление данных и управление данными в соответствии с определенными соглашениями.

Два **подхода** к организации массивов данных:

- файловая организация и
- организация в виде базы данных.

Файловая организация предполагает специализацию и хранение данных, ориентированных, как правило, на одну прикладную задачу, и обеспечивается самим прикладным программистом.

Недостатки файловой организации

- 1. Значительная избыточность данных.** Узкая специализация обрабатывающих программ и файлов обуславливает необходимость хранения одних и тех же элементов данных в разных системах.
- 2. Невозможность выявления противоречивых данных.** Причина очевидна – для выполнения одних и тех же операций на различных файлах требуются разные программы, а управление осуществляется разными лицами (группами лиц).
- 3. Низкий уровень удовлетворения информационных запросов пользователей.** Разработанные для специализированных прикладных программ файлы нельзя использовать для удовлетворения запросов пользователей, перекрывающих две и более предметных области. Более того, файловая организация данных вследствие отличий структуры записей и форматов представления данных не обеспечивает выполнения многих информационных запросов даже в тех случаях, когда все, необходимые элементы содержатся в имеющихся файлах.

Организация в виде **базы данных**

Необходимость отделения данных от их описания, определения такой организации хранения данных с учетом существующих связей между ними, которая позволила бы использовать эти данные для многих приложений привела к созданию **баз данных (БД)** и **систем управления базами данных (СУБД)**.

База данных – совокупность данных, организованных по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования данными, независимо от прикладных программ.

Основные требования к БД

- 1. Интегрированность данных**, т.е. хранение данных в едином хранилище – БД.
- 2. Независимость данных**. Различают *логическую* (изменение схемы БД не требует корректировки прикладных программ) и *физическую* (изменение метода организации данных не влияет ни на прикладные программы, ни на схему БД) независимость данных.
- 3. Адекватность предметной области**, т.е. возможность отображения любых фактов, характеризующих предметную область.
- 4. Целостность данных** (удовлетворение предъявляемым логическим требованиям).

Основные требования к БД

- 5. Минимальная избыточность** (дублирование) хранимых данных, обеспечивающая требуемую производительность БД.
- 6. Способность БД к расширению.**
- 7. Возможность поиска по нескольким ключам.**
- 8. Обеспечение защиты данных от несанкционированного доступа или случайного уничтожения хранимых данных.** Предполагает введение мер по идентификации пользователей и контролю их действий с точки зрения предоставленных полномочий, а также восстановление БД при аппаратных сбоях ЭВМ.

Банк данных и СУБД

Банк данных – основанная на технологии БД система программных, языковых, организационных и технических средств, предназначенных для централизованного накопления и коллективного использования данных.

В операционных системах, в среде которых функционирует банк данных, необходим комплекс программ – система управления базами данных.

Система управления базами данных (СУБД) – это совокупность программ и языковых средств, предназначенных для управления данными в базе данных, ведения базы данных и обеспечения взаимодействия её с прикладными программами.

Виды управления данными

- **Ведение базы данных** – деятельность по обновлению, восстановлению и перестройке структуры базы данных с целью обеспечения ее целостности.
- **Защита данных** – организационные, программные и технические методы и средства, направленные на удовлетворение ограничений, установленных для типов данных или экземпляров типов данных в системе обработки данных.
- **Доступ к порции данных** – предоставление процессу обработки данных порции данных или принятие от него порции данных посредством последовательности операций поиска, чтения и (или) записи данных.

Характеристика СУБД

- Главное **назначение** СУБД – предоставление пользователям БД средств манипулирования данными в абстрактных терминах, не связанных со способом их хранения в ЭВМ.
- СУБД **гарантирует** непротиворечивость, целостность, секретность и минимальную избыточность данных, хранимых в БД.
Эффективность СУБД определяется скоростью доступа к данным, рациональным использованием памяти ЭВМ, простотой разработки прикладных программ, оперирующими данными из базы.

Характеристика СУБД

СУБД обеспечивает **логическую** и **физическую** независимость данных.

Логическая независимость данных в БД означает, что:

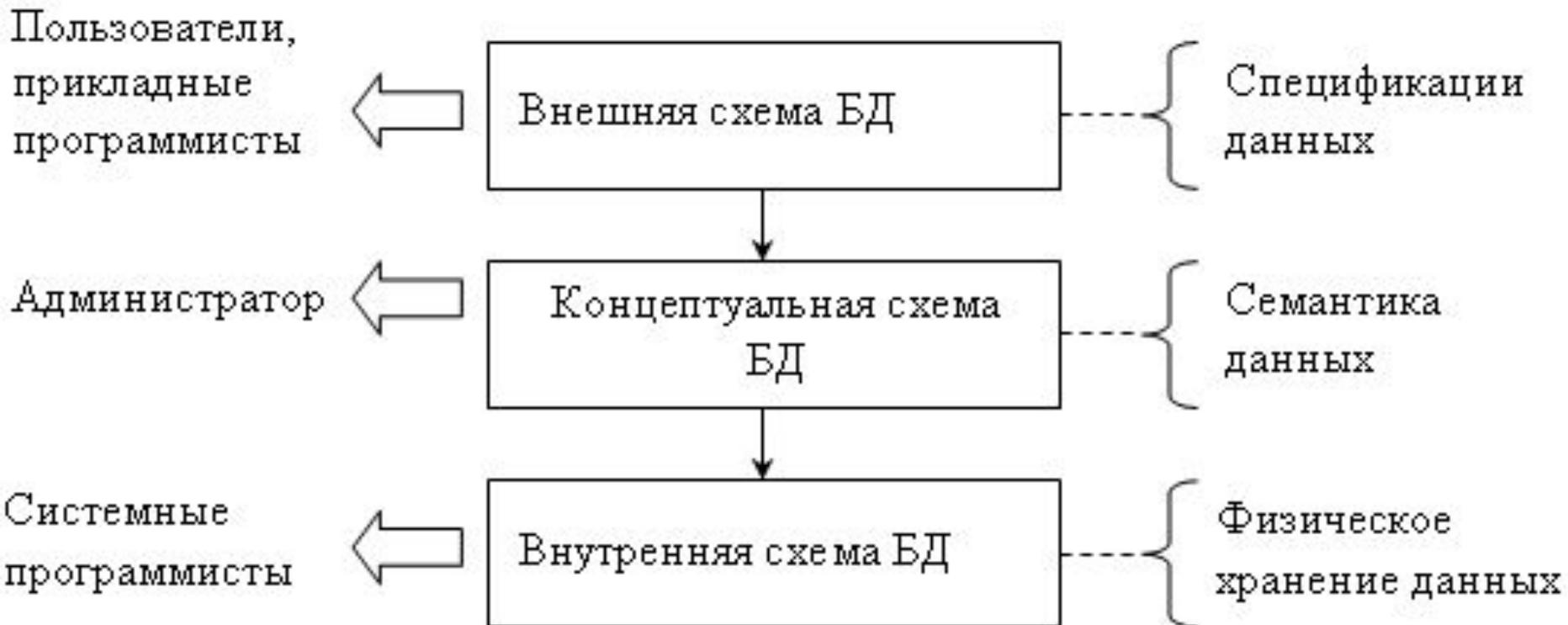
- логическая структура данных не зависит от особенностей прикладных программ, которые их используют (т.е. изменение прикладных программ не влечёт изменения логической структуры данных);
- прикладные программы не зависят от логической структуры данных (т. е. изменение логической структуры не влечет изменения прикладных программ).

Физическая независимость данных в БД означает, что:

- особенности размещения данных на физических носителях не зависят от логической структуры данных и особенностей прикладных программ;
- логическая структура данных и прикладные программы не зависят от способа размещения данных на физических носителях.

2. Уровни представления данных

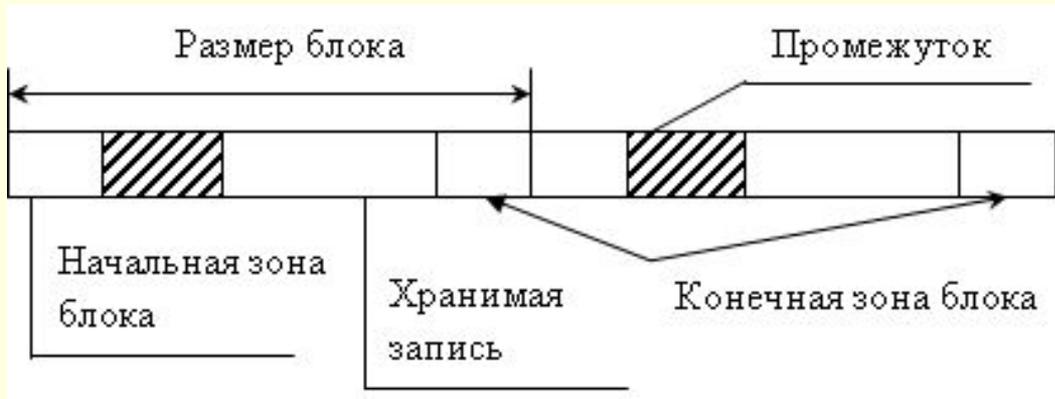
СУБД поддерживают три уровня описания данных, соответствующих **физическому представлению** данных в среде хранения; общему **логическому представлению** всей совокупности данных, хранящихся в БД; **частным представлениям** данных с точки зрения конкретных пользователей и прикладных программистов.



Характеристика внутренней схемы БД

Внутренняя схема БД – схема БД, определяющая представление данных в среде хранения и пути доступа к ним.

Элементарными единицами внутренней схемы БД являются физические блоки (записи), хранимые записи, указатели, данные переполнения и межблочные промежутки.



Структура физического блока



Структура хранимой записи

Характеристика концептуальной схемы БД

Концептуальная схема БД – схема БД, определяющая представление данных, единое для всех ее приложений и независящее от используемого в системе управления этой базой данных представления данных в среде хранения и пути доступа к ним.

Концептуальная схема БД базируется на следующих трех **понятиях**: объекты, атрибуты и связи между этими элементами и их свойствами.

С **объектом** связаны два понятия: тип и экземпляр объекта.

Понятие **тип объекта** относится к набору однородных предметов, выступающему как единое целое.

Экземпляр объекта относится к конкретному предмету.

Атрибуты предназначены для определения свойств и идентификации объектов, а также представления связей между объектами.

Под **связями** понимаются ассоциации (соответствия) между одинаковыми или различными типами объектов.

Характеристика внешней схемы БД

Внешняя схема БД – схема БД, поддерживаемая системой управления БД для приложений.

С точки зрения пользователя, внешняя схема БД представляет совокупность требований к данным, определяемым функциональными спецификациями (реальными форматами).

Внешние и концептуальные схемы БД строятся на основе одной из трех **моделей данных**: иерархической, сетевой и реляционной.

3. Модели данных

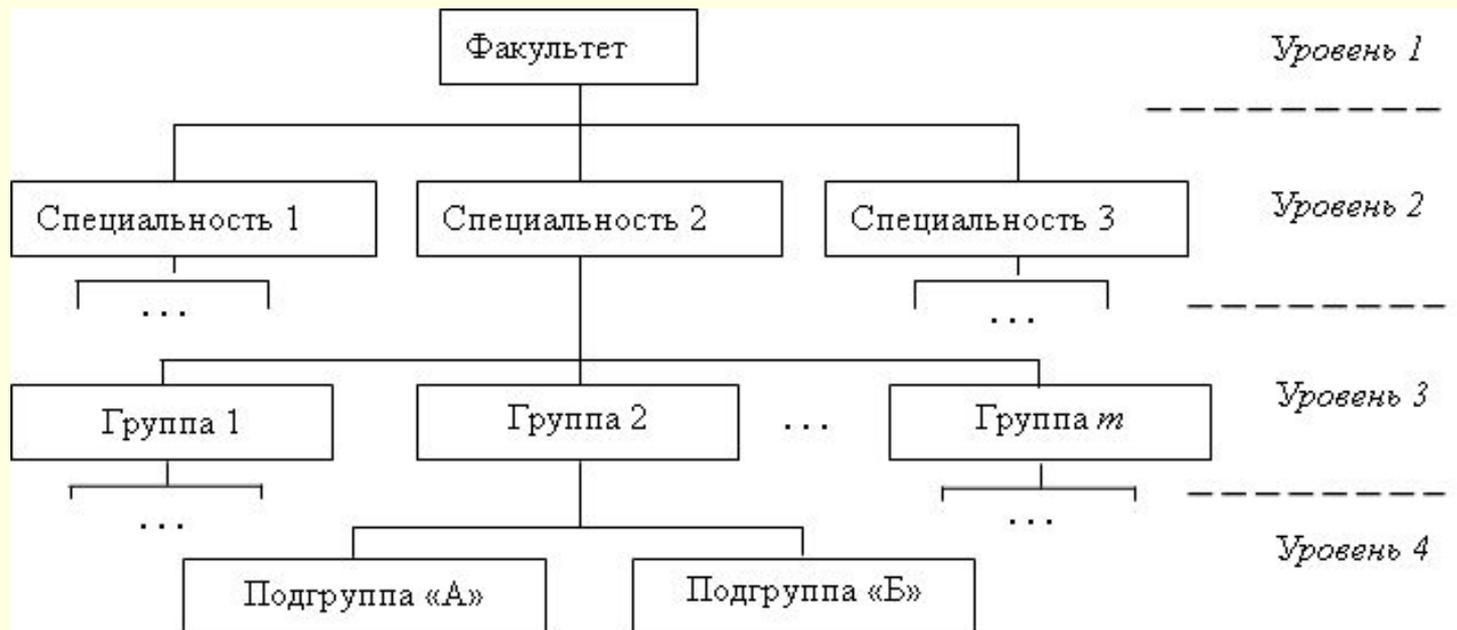
Под **моделью данных** понимается совокупность правил порождения структур данных в БД, операций над ними, а также ограничения целостности, определяющей допустимые связи и значения данных, последовательности их изменения.

Для задания модели данных используется язык описания данных и язык манипулирования данными.

Иерархическая модель данных

Иерархическая модель данных – модель данных, предназначенная для представления данных иерархической структуры и манипулирования ими.

Иерархическая модель организует данные в виде древовидной структуры и является реализацией логических связей: родо-видовых отношений и отношений «часть-целое».

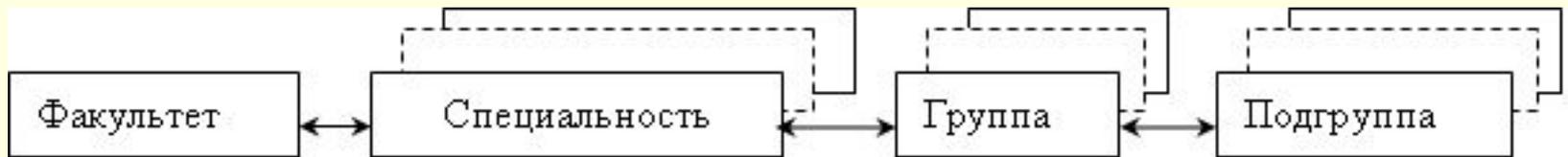


Иерархическая модель данных

Дерево представляет собой иерархию элементов, называемых **узлами** (вершинами). Под элементами понимается список (совокупность, набор) атрибутов, описывающих объекты. Выделяют *корневой* узел (*корень* дерева) и *порожденные* узлы.

Корень находится на самом верхнем уровне и не имеет предшествующих узлов. Остальные узлы, называемые **порожденными**, связаны между собой следующим образом: каждый узел имеет исходный (предшествующий) узел, находящийся на более высоком уровне, и произвольное количество порожденных им узлов на последующем уровне.

Узлы, не имеющие порожденных ими узлов, называют **листьями**. Между исходным узлом и порожденными узлами существует отношение «один ко многим (многие к одному)». Использование данного отношения позволяет представлять иерархическую структуру так:



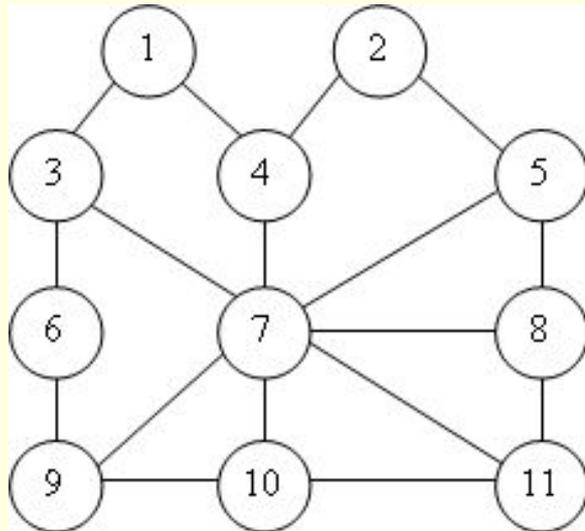
Требования к иерархии

1. Дерево может иметь только **один корень**.
2. **Узел** содержит **не менее одного атрибута**, описывающего объект в данном узле.
3. Порожденные узлы могут добавляться как в горизонтальном, так и в вертикальном направлении.
4. **Доступ** к порожденным узлам возможен только **через исходный узел**. Поэтому существует только один путь доступа к каждому узлу.
5. Количество экземпляров узла каждого уровня **ограничивается только физическим объемом БД**.

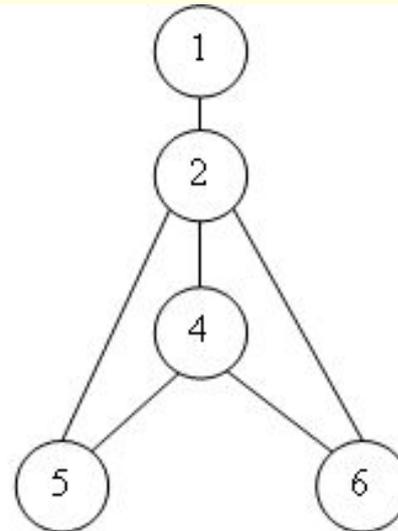
Сетевая модель данных

Сетевая модель данных – модель данных, предназначенная для представления данных сетевой структуры и манипулирования ими.

Сетевая модель организует данные в виде сетевой структуры, отличающейся от древовидной тем, что любой узел в сети может быть связан с любым другим узлом.



a)



б)

Реляционная модель данных

Реляционная модель данных – модель данных, основанная на представлении данных в виде набора отношений, каждое из которых представляет собой подмножество декартова произведения определенных множеств, и манипулировании ими с помощью множества операций реляционной алгебры или реляционного исчисления.

Домен – это множество значений. Примерами доменов могут служить множество целых чисел, множество вузов, множество фамилий студентов и т.д.

Отношением R называется подмножество декартова произведения одного или более доменов. Элементами отношения являются кортежи.

Примеры отношений

Пусть $D1 = \{1, 2\}$, $D2 = \{a, b, c\}$.

Тогда

$$D1 \times D2 = \{(1, a), (1, b), (1, c), (2, a), (2, b), (2, c)\}$$

Укажем несколько отношений:

$$R1 = \{(1, a), (1, b)\} \subseteq D1 \times D2,$$

$$R2 = \{(2, a)\} \subseteq D1 \times D2,$$

$$R3 = \{(1, a), (1, b), (1, c), (2, a), (2, b), (2, c)\} = D1 \times D2,$$

$$R4 = \emptyset.$$

Пример представления данных

В практике реляционных БД отношения представляются в виде двумерной таблицы, в которой строка есть кортеж, каждый столбец соответствует только одной компоненте этого отношения.

Такие таблицы обладают следующими **свойствами**.

1. Каждый элемент таблицы представляет собой один элемент данных, повторяющиеся группы отсутствуют.
2. Все столбцы в таблице однородные, т.е. элементы столбца имеют одинаковую природу.
3. В таблице нет двух одинаковых строк.
4. Строки и столбцы таблицы могут просматриваться в любом порядке и в любой последовательности.

Списочный состав подгруппы

Наименование подгруппы	Ф.И.О.	Дата рождения	Адрес местожительства	Аттестат о среднем образовании

Понятие ключа

Минимальное подмножество атрибутов X , совокупность значений которых однозначно идентифицируют кортежи $\langle A_1, A_2, \dots, A_k \rangle$ отношения R (каждую строку таблицы), называют **ключом**.

Ключ должен удовлетворять следующим двум **условиям**:

1. Функциональная зависимость $X \rightarrow A_1, A_2, \dots, A_k$ принадлежит полному множеству функциональных зависимостей F^+ , которое логически следует из заданного множества этих зависимостей F .
2. Ни для какого собственного подмножества X зависимостей $Y \rightarrow A_1, A_2, \dots, A_k$, где $Y \subset X$, не принадлежит F^+ .

При удовлетворении первого условия считается, что ключ обладает свойством однозначной идентификации кортежа, а при удовлетворении второго – свойством отсутствия избыточности.

Отношение может иметь несколько ключей, называемых **возможными** ключами. Один из возможных ключей, выбираемый для идентификации записей, называется **первичным** ключом.

Достоинства и недостатки моделей данных

Модель данных	Основные достоинства	Основные недостатки
Иерархическая	Простота понимания используемого принципа иерархии. Несложные процедуры поиска данных	Невозможность отображения связи «Многие ко многим». Сложность включения информации о новых объектах и удаления устаревшей. Избыточность данных
Сетевая	Отсутствие избыточности данных. Возможность описания сложных структур данных	Сложность понимания структуры. Возможная потеря независимости данных при реорганизации БД
Реляционная	Доступность для понимания. Обеспечение независимости данных. Строгая математическая основа	Экспоненциальное возрастание времени поиска данных при увеличении (более > 50000) количества записей. Избыточность данных

Подготоввил

А.В. Мишин

31.01.2016

Спасибо за



внимание!

**Кафедра права и информатики,
информационного права и
естественно-научных дисциплин
Центрального филиала
Российского государственного
университета правосудия**