**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

**(СПбГУТ)**

**ОТЧЕТ**

**по учебной практике студента**

**Абдуллаев Турал Фирдовси оглы**

**(Фамилия, имя, отчество)**

ИСиТ\_\_\_\_\_\_\_\_\_

Кафедра \_\_\_ИУС\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Специальность /Направление \_\_\_\_09.03.02\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Группа \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель практики

от базы практики \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ФИО, должность) (подпись)

Руководитель практики

от каф..ИУС \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ФИО, должность) (подпись)

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ФИО ) (подпись)

Санкт-Петербург

2017 г.

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА**

**Учебную практику студента группы ИСТ-511**

**Абдуллаев Турал Фирдовси оглы**

**(фамилия, имя, отчество)**

**1.  Тема задания на практику: Работа с базой данных**

**2. Срок сдачи студентом отчета: 17.09.18**

**3. Календарный план**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Этапы работы | Срок | Примечание |
| 1 | Приемная комиссия |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 | 17.09.18 |  |

**АННОТАЦИЯ**

Данная аналитическая работа посвящена теме базы данных приемной комиссии в университете СПБГУТ))). Она содержит в себе введение, три основные главы и заключение.

Объём работы составляет 21 страницу, на которых размещены 8 изображений.

В процессе учебной практики в качестве предмета исследования будет рассмотрена база данных, ее описание.

Ключевые слова: студент, поступление, база данных, образование, направление.

**СОДЕРЖАНИЕ**

ВВЕДЕНИЕ 5

Глава 1. Теоретические аспекты СУБД

1. Основные понятия……………………………………………………………6

2. Функциональные возможности СУБД………………………………………12

Глава 2. Разработка базы данных…………………………………………………15

ЗАКЛЮЧЕНИЕ 18

СПИСОК ЛИТЕРАТУРЫ 19

ПРИЛОЖЕНИЕ…………………………………………………………...……......20

**ВВЕДЕНИЕ**

Под базой данных (БД) понимают хранилище структурированных данных, при этом данные должны быть непротиворечивы, минимально избыточны и целостны.

Жизненный цикл любого программного продукта, в том числе и системы управления базой данных, состоит из стадий проектирования, реализации и эксплуатации.

Естественно, наиболее значительным фактором в жизненном цикле приложения, работающего с базой данных, является стадия проектирования. От того, насколько тщательно продумана структура базы, насколько четко определены связи между ее элементами, зависит производительность системы и ее информационная насыщенность, а значит - и время ее жизни.

Приёмная коми́ссия — уполномоченное собрание представителей среднего или высшего образовательного учреждения во главе с председателем приёмной комиссии, занимающееся регулировкой процесса перехода учащихся от начальной и средней общеобразовательных ступеней к высшей. Таким образом, желающие получить высшее образование превращаются из школьников в абитуриентов, подавших заявление о приёме и прочий набор требуемых документов (аттестат или диплом и т. д.), и, наконец, в студентов в случае успешной сдачи экзаменов и прохождения по конкурсу.

В современных условиях российского образования приемная комиссия становится одним из ключевых звеньев учебного заведения. От качества работы этого подразделения зависит успех приемной кампании, а значит и успех самого учебного заведения. Более того, во многих случаях речь идёт и вовсе о его выживании. Поэтому грамотное управление процессом приемной кампании является архиважным.

Теоретические аспекты СУБД.

1. Основные понятия БД.

Всякая прикладная программа является отображением какой - то части реального мира и поэтому содержит его формализованное описание в виде **данных.** Крупные массивы данных размещают, как правило, отдельно от исполняемого программы, и организуют в виде **Базы данных.** Начиная с 60-х годов для работы с данными, стали использовать особые программные комплексы, называемые **системами управления базами данных (СУБД)**

Системы управления базами данных отвечают за:

·физическое размещение данных и их описаний;

·поиск данных;

·поддержание баз данных в актуальном состоянии;

·защиту данных от некорректных обновлений и несанкционированного доступа;

·обслуживание одновременных запросов к данным от нескольких пользователей (прикладных программ).

**Модели данных**.

Хранимые в базе данных имеют определенную логическую структуру, то есть, представлены некоторой моделью, поддерживаемой СУБД. К числу важнейших относятся следующие модели данных:

- иерархическая;

- сетевая;

- реляционная;

- объектно – ориентированная;

В иерархической модели данные представляются в виде древовидной (иерархической) структуры. Она удобна для работы с иерархически упорядоченной информацией и громоздка для информации со сложными логическими связями.

Сетевая модель означает представление данных в виде произвольного графа. Достоинством сетевой и иерархической моделей данных является возможность их эффективной реализации показателей затрат памяти и оперативности. Недостатком сетевой модели данных является высокая сложность и жесткость схемы БД, построенной на ее основе.

Реляционная модель данных (РМД) название получила от английского термина Relation – отношение. Модель данных описывает некоторый набор родовых понятий и признаков, которыми должны обладать все конкретные СУБД и управляемые ими БД, если они основываются на этой модели.

Объектно-ориентировочная модель данных – это когда в базе хранятся не только данные, но и методы их обработки в виде программного кода. Это перспективное направление, пока также не получившее активного распространения из-за сложности создания и применения подобных СУБД.

**База данных** - это совокупность записей различного типа, содержащая перекрестные ссылки.

Файл - это совокупность записей одного типа, в котором перекрестные ссылки отсутствуют.

Более того, в определении нет упоминания о компьютерной архитектуре. Дело в том, что, хотя в большинстве случаев БД действительно представляет собой один или (чаще) несколько файлов, физическая их организация существенно отличается от логической. Таблицы могут храниться как в отдельных файлах, так и все вместе. И, наоборот, для хранения одной таблицы иногда используются несколько файлов. Для поддержки перекрестных ссылок и быстрого поиска обычно выделяются дополнительные специальные файлы.

Поэтому при работе с базами данных обычно применяются понятия более высокого логического уровня: запись и таблица, без углубления в подробности их физической структуры.

Таким образом, сама по себе база данных - это только набор таблиц с перекрестными ссылками. Чтобы универсальным способом извлекать из нее группы записей, обрабатывать их, изменять и удалять, требуются специальные программы, называются **СУБД.**

По характеру использования СУБД делят на персональные (СУБДП) и многопользовательские (СУБДМ).

К персональным СУБД относятся VISUALFOXPRO, ACCESS и др. К многопользовательским СУБД относятся, например, СУБД ORACLE и INFORMIX. **Многопользовательские СУБД** включают в себя сервер БД и клиентскую часть, работают в неоднородной вычислительной среде допускаются разные типы ЭВМ и различные операционные системы. Поэтому на базе СУБДМ можно создать информационную систему, функционирующую по технологии клиент-сервер. Универсальность многопользовательских СУБД отражается соответственно на высокой цене и компьютерных ресурсах, требуемых для поддержки

СУБДП представляет собой совокупность языковых и программных средств, предназначенных для создания, ведения и использования БД.

**Персональные СУБД** обеспечивают возможность создания персональных БД и недорогих приложений, работающих с ними, и при необходимости создания приложений, работающих с сервером БД.

Для обработки команд пользователя или операторов программ в СУБДП используются **интерпретаторы команд**(операторов) и **компиляторы**. С помощью компиляторов в ряде СУБДП можно получать исполняемые автономно приложения-ехе-программы.

Обеспечение целостности БД-необходимое условие успешного функционирования БД. Целостность БД-свойство БД, означающее, что база данных содержит полную и непротиворечивую информацию, Для обеспечения целостности БД накладывают ограничения целостности в виде некоторых условий, которым должны удовлетворять хранимые в базе данные. Примером таких условий может служить ограничение диапазонов возможных значений атрибутов объектов, сведения о которых хранятся в БД, или отсутствие повторяющихся записей в таблицах реляционных БД.

**Обеспечение безопасности** достигается СУБД шифрованием прикладных программ, данных, защиты паролем, поддержкой уровней доступа к базе данных, к отдельной таблице.

Расширение возможностей пользователя СУБДП достигается за счет подключения систем распространения Си или Ассемблера.

**Поддержка функционирования в сети обеспечивается:**

- средствами управления доступом пользователей к совместно используемым данным, т.е. средствами блокировки файлов (таблиц), записей, полей, которые в разной степени реализованы в разных СУБДП;

- средствами механизма транзакций, обеспечивающими целостность БД при функционировании в сети.

Теперь рассмотрим **функции СУБД** немного подробнее:

**Определение данных.**

СУБД должна допускать определения данных (внешние схемы, концептуальную схему, внутреннюю схему, а также все связанные отображения) в исходной форме и преобразовывать эти определения в форму соответствующих объектов. Иначе говоря, СУБД должна включать в себя компонент языкового процессора для различных языков определений данных. СУБД должно также «понимать» синтаксис языка определений данных.

**Обработка данных.**

СУБД должна уметь обрабатывать запросы пользователя на выборку, изменение или удаление существующих данных в базе данных или на добавление новых данных в базу данных. Другими словами, СУБД должна включать в себя компонент процессора языка обработки данных.

Запросы языка обработки данных бывают «планируемые» и «не планируемые».

1. **Планируемый запрос-это** запрос, необходимость которого предусмотрена заранее. Администратор базы данных, возможно, должен настроить физический проект БД таким образом, чтобы гарантировать достаточное быстродействие для таких запросов.

**2. Не планируемый запрос-это,**наоборот, специальный запрос, необходимость которого не была предусмотрена заранее. Физический проект БД может подходить, а может и не подходить для рассматриваемого специального запроса. В общем, получение возможной наибольшей производительности для не планируемых запросов представляет собой одну из проблем СУБД. (Подробнее эта проблема будет обсуждаться в следующих частях книги.)

**Безопасность и целостность данных.**

СУБД должна контролировать пользовательские запросы и пресекать попытки нарушения правил безопасности и целостности, определенные АБД.

**Восстановление данных и дублирование.**

СУБД или другой связанный с ней программный компонент, обычно называемый администратором транзакций, должны осуществлять необходимый контроль над восстановлением данных и дублированием. Подробности использования этих функций системы приводятся далее в этой книге.

**Словарь данных.**

СУБД должна обеспечить функцию **словаря данных**. Сам словарь данных можно по праву считать БД (но не пользовательской, а системой). Словарь «содержит данные о данных» (иногда называемые **метаданными**), т.е. **определения** других объектов системы, а не просто «сырые данные». В частности, исходная и объектная формы различных схем (внешних, концептуальных и т.д.) и отображений будут сохранены в словаре. Расширенный словарь будет включать также перекрестные ссылки, показывающие, например, какие из программ какую часть БД используют, какие отчеты требуются тем или иным пользователям, какие терминалы подключены к системе и т.д. Словарь может быть (а на самом деле даже должен быть) интегрирован в определяемую им БД, а значит, должен содержать описание самого себя. Конечно, должно быть возможность обращения к словарю, как и к другой БД, например, для того узнать, какие программы и/или пользователи будут затронуты при предполагаемом внесении изменения в систему. (Дальнейшее обсуждение этого вопроса приводится в следующих главах книги.)

**Производительность.**

Очевидно, что СУБД должна выполнять все указанные функции с максимально возможной эффективностью.

Подводя итог сказанному, можно сделать вывод, что в целом назначением СУБД является предоставление **пользовательского интерфейса** с БД. Пользовательский интерфейс может быть определен как граница в системе, ниже которой все невидимо для пользователя. Следовательно, по определению пользовательский интерфейс находится на внешнем уровне. Тем не менее, иногда встречаются случаи, когда внешнее представление вряд ли значительно отличается от относящейся, по мере в современных коммерческих продуктах.

В заключении вкратце сопоставим описанную СУБД с системой файлами (или с управлением файлами). В своей основе система управления файлами является компонентом общей системы, которая управляет хранимыми файлами; проще говоря, она «ближе к диску», чем СУБД. Таким образом, пользователь системы управления файлами может создавать и уничтожать хранимые файлы, а также выполнять простые операции выборки и обновления хранимых записей в таких файлах. Однако, в отличие от СУБД, системы управления файлами имеют некоторые недостатки.

**2. Функциональные возможности СУБД.**

Управляющим компонентом многих СУБД является ядро, выполняющее следующие функции:

- управление данными во внешней памяти;

- управление буферами оперативной памяти (рабочими областями, в которые осуществляется подкачка данных из базы для повышения скорости работы);

- управление транзакциями.

**1. Непосредственное управление данными во внешней памяти.**

Эта функция включает обеспечение необходимых структур внешней памяти, как для хранения данных, непосредственно входящие в базу данных так и для служебных целей. Например, для убыстрения доступа к данным в некоторых случаях (обычно для этого используется индекс).

В некоторых реализациях СУБД активно используется возможность существующих файловых систем. В других работа производится вплоть до уровня устройств внешней памяти. Но подчеркнем, что в развитых СУБД пользователь в любом случае не обязан знать использование СУБД файловую систему и если использует, то, как организованные файлы. В частности, СУБД поддерживает собственную систему и наименование объектов баз данных.

**2. Управление буторами оперативной памяти.**

СУБД обычно работает с БД, по крайней мере, этот размер обычно существует, больше доступен объему оперативной памяти. Что если при обращении к любому элементу данных будет производиться объем с внешней памятью, то вся система будет работать со скоростью устройства внешней памяти. Практическим единственным способом реально увеличение этой скорости является буферизация данных в оперативной памяти. При этом даже если операционная система производит общесистемную буферизацию. Этого недостаточно для цели СУБД, которая располагает гораздо больше информации о полезности буферизации, т.е. той или иной части БД. Поэтому в развитых СУБД поддерживается собственный набор буферов оперативной памяти, собственной дисциплины замены буферов. Заметим, что существуют отдельные направления СУБД, которые ориентированно, но постоянно присутствуют в оперативной памяти БД. Это направление основывается на предположение, что на столько велик, что позволит, не беспокоится о буферизации. (Пака эта работа находится в стадии развития).

**3. Управление транзакциями.**

Транзакция – это последовательность операций над БД, рассматриваемая СУБД как единое целое. При выполнении транзакция может быть либо успешно завершена, и СУБД зафиксирует произведенные изменения во внешней памяти, либо, например, при сбое в аппаратной части ПК, ни одного из изменений не отразится в БД. Понятие транзакция необходимо для поддержания логической целостности БД. Таким образом, поддержание механизма транзакции является обязательным условием даже однопользовательских СУБД. (Если такая система заслуживает СУБД). Но понятие транзакция гораздо более важно много пользователь СУБД, то свойство, то каждая транзакция начинается при целостном состоянии БД и оставляет это состояние целостное после своего завершения, делает очень удобным, использование понятие транзакция как единицы активности пользователя по отношению БД. При соответствующем управлении управляющимися транзакциями со стороны СУБД каждым использованием может в принципе ощущать себя единственным пользователем СУБД. Управление транзакции многопользовательской СУБД связаны важные понятия сериализация транзакции и сериального плана выполнения смеси транзакции. Под стерилизацией выполнении параллельно сериализация понимают такой порядок планирования их работ при которой суммарный эффект смеси транзакции эквивалентен эффекту их некоторого последовательного управления. Сериальный план выполнения смеси транзакции — это такой план, который приводит к сериализация транзакции. Что если удается добиться действительного сериального выполнения смеси транзакции, то для каждого пользователя по инициативе, которой образованна транзакция присутствие других транзакций будет незаметно (если не считать некоторого замедления работы по сравнению с одно пользованием режимом). Существует несколько базовых алгоритмов сериализация транзакции. Централизованных СУБД наиболее распространены алгоритмы, основанные на синхронизации захвата объектов БД. При использовании любого алгоритма возможная ситуация конфликта между двумя или более транзакциями по доступу объекта БД. В этом случае для поддержания сериализация необходимы, выполнять откат одной ли более транзакции. Это один из случаев, когда пользователь многопользовательской СУБД может реально (и достаточно неприятно) ощутить присутствие в системе транзакции других пользователей.

**Разработка базы данных.**

Процесс разработки (проектирования) базы данных включает два этапа: разработку логической организации базы данных и создание ее на носителе. Логическая организация базы данных – это предоставление пользователя о предметной области, информация о которой должна храниться в базе данных. Под физической организацией базы данных понимается совокупность средств и методов размещения данных во внешней памяти и на их основе внутренняя модель данных. Внутренняя модель является средством отображения логической модели данных, показывает, каким образом записи размещаются в базе данных, как они упорядочиваются, как организуются связи, каким путем можно осуществить выборку и так далее.

В настоящее время разработаны и используются на персональных компьютерах около двадцати систем управления базами данных. Они представляют пользователю удобные средства интерактивного взаимодействия с БД и имеют развитый язык программирования. Одной из самых популярных настольных программных СУБД является MicrosoftAccess.

Одной из основных причин такой популярности Access заключается в том, что, является по сути настольной СУБД, это приложение вобрало в себя многие возможности систем управления реляционными базами данных архитектуры клиент-сервер, называемой также SQL базой данных. Несмотря на то, что, Access включают в себя сложные функции и может послужить прекрасным инструментом для профессионального разработчика приложений БД, его использование не должно вызвать проблем и у непрофессиональной пользователей и даже тех, кто раньше не работал с СУБД. Кнопки на панелях инструментов дублируют основные команды меню, расширенный набор мастеров и настроек управляет практически всеми параметрами создания и изменения объектов БД (таблиц, форм, отчетов, запросов и т.д.). С помощью ACCESS можно создавать многопользовательских приложений, в которых файлы базы данных являются разделяемыми ресурсами в локальной сети. В ACCESS реализованного доступа к объектам базы данных. MicrosoftAccess для хранения объектов БД имеет собственную уникальную структуру для хранения всех связанных таблиц, форм, отчетов, запросов и макрософт в одном файле. Также имеет возможность импорта и экспорта данных во многие широкие распространенные форматы БД, электронных таблиц и текстовых файлов. ACCESS позволяет связывать БД с внешними таблицами в форматах dBase, FoxPro, Paradox и работать с ними в исходном формате. Также Access можно использовать в качестве клиентской части архитектуры клиент-сервер, что обеспечивает применение MicrosoftAccess не только в качестве профессиональной системы управления базы данных, но и как мощное инструментальное средство для создания приложений клиент-сервер.

База данных по бытовой технике города Улан-Удэ была разработана в программе MicrosoftAccess. Вся необходимая информация представлена в двух таблицах. Таблица базы данных – это совокупность сведений. Так, например, в таблице «торговые салоны» отображена информация о торговом салоне, адресе, телефоне , а в таблице «товары» - информация о предоставляемых торговым салоном товарах . Каждое поле предоставляет собой столбец таблицы и содержит определенную категорию информации. Каждая запись предоставляет собой строку таблицы и содержит информацию об определенном товаре. Можно создать взаимосвязи между таблицами, вместо того, чтобы хранить всю информацию в одной большой таблице, избегая тем самым ненужного дублирования данных, экономии памяти компьютера, а также увеличение скорости и точности обработки информации. Так, например, каждая запись в таблице «все товары» содержит информацию о фирме и предоставляемых ею товарах . Запросы используются примерно также, как и таблицы. Вы можете открыть запрос и просмотреть набор данных в табличном представлении. При создании запроса указываются таблицы, из которых будет производится выборка данных, указываются поля таблицы, которые должны быть внесены в результат запроса, указано условие отбора данных. В данном случае при выборе запроса «поиск по салону» , указав название салона, вы получите интересующую вас информацию о товарах в этом салоне.

При выборе запроса «поиск по товарам» вы получите информацию о представляемых фирмой товарах.

С помощью форм можно отображать данные, содержащиеся в таблицах или запросах, в еще более удобное для восприятия виде. При помощи форм можно добавлять в таблицы новые данные или изменять и удалять существующие. Форма может содержать рисунки, графики и другие объекты. Формы создаются из набора отдельных элементов, называемых элементами управления (графический объект для предоставления данных). Форму для предоставления данных Microsoft Access предоставляет автоматически. После ее заполнения можно просмотреть информацию полностью или частично при помощи процедур поиска, запроса, сортировки. Для просмотра необходимой информации с помощью форм в моей базе данных представлены 6 различных форм: кнопочная, все товары, все торговые салоны, поиск, поиск по салону, поиск по товару. При выборе той или иной формы, указав, если есть необходимость, название товара, вы получите интересующую вас информацию.

**ЗАКЛЮЧЕНИЕ**

Таким образом, БД является важнейшей составной частью информационных систем, которые предназначены для хранения и обработки информации.

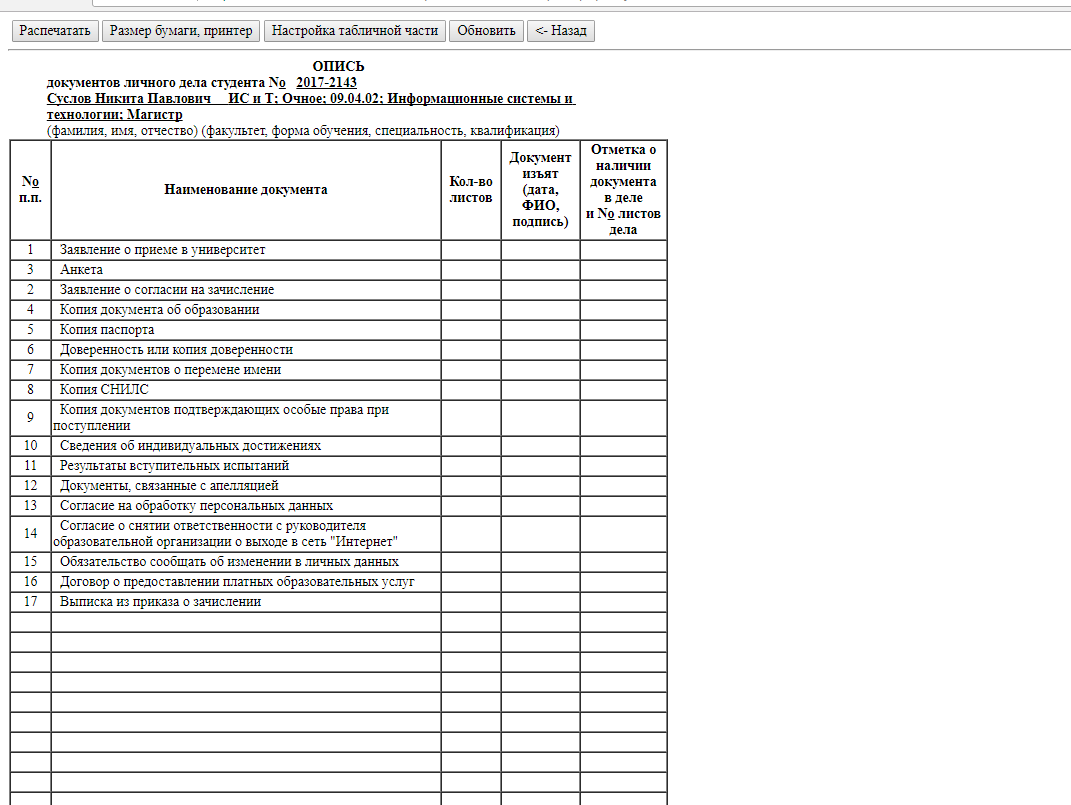
Изначально такие системы существовали в письменном виде. Для этого использовались различные картотеки, папки, журналы, библиотечные каталоги. Развитие средств вычислительной техники обеспечило возможность для создания и широкого использования автоматизированных информационных систем. Разрабатываются информационные системы для обслуживания различных систем деятельности, системы управления хозяйственными и техническими объектами, модельные комплексы для научных исследований, системы автоматизации проектирования и производства, всевозможные тренажеры и обучающие системы. Современные информационные системы основаны на концепции интеграции данных, характеризующих большими объектами хранимых данных, сложной организацией, необходимостью удовлетворять разнообразные требования многочисленных пользователей. Для управления этими данными и обеспечения эффективности доступа к ним были созданы системы управления данными.

Таким образом, СУБД называют программную систему, предназначенную для создания ЭВМ общей базы данных для множества приложений, поддержания ее в актуальном состоянии и обеспечения эффективности доступа пользователей к содержащимся в ней данным в рамках предоставленных им полномочий.

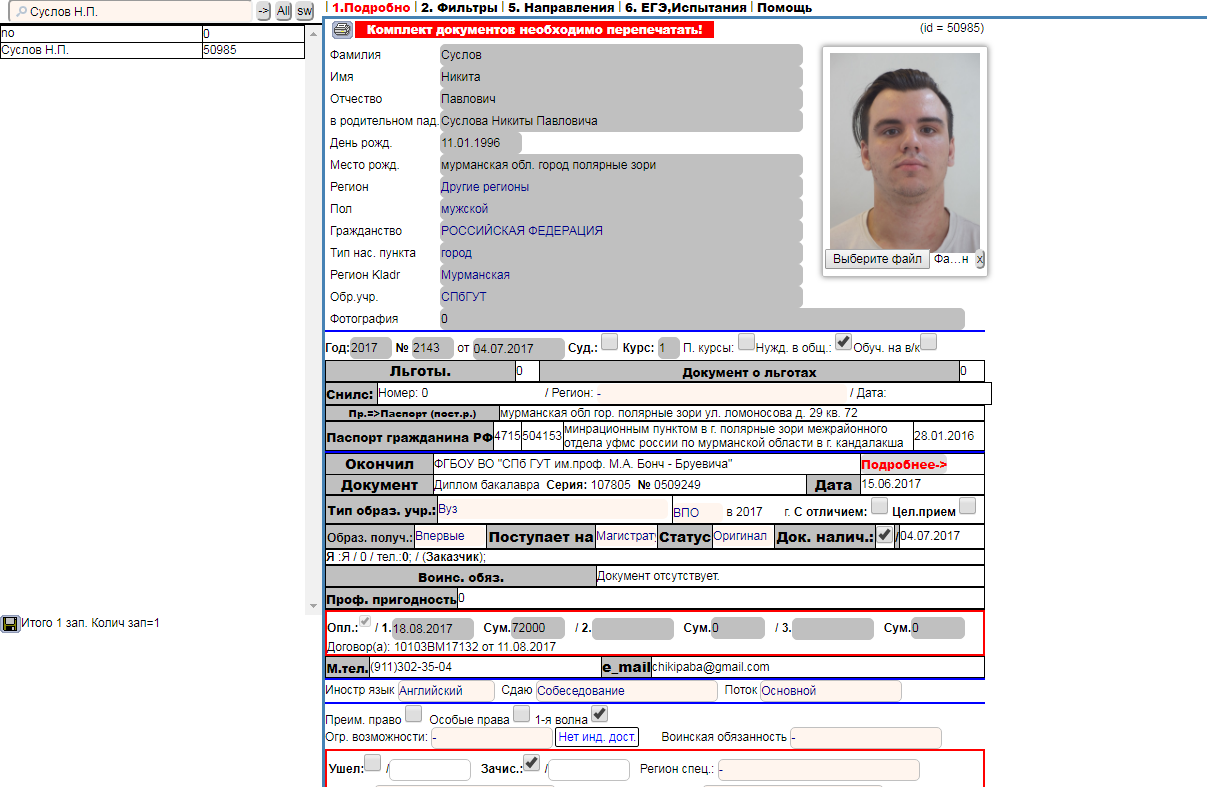
**СПИСОК ЛИТЕРАТУРЫ**

1. Бойко И., Объектно–ориентированные СУБД.- Киев: Высшая школа, 1999
2. Майкл. Хэлволсон, Майкл Янг, Эффективная работа с Microsoft Office. C.Петербург: Питер, 2001
3. Рыбакова О. О., Проектирование автоматизированных информационных систем. Методический материал для проведения аудиторных занятий и самостоятельной работы. Издание первое. – Запорожье: ЗЕТК, 2001
4. Карпова И.П. Базы данных: Учебное пособие. − СПб.: Питер, 2013.

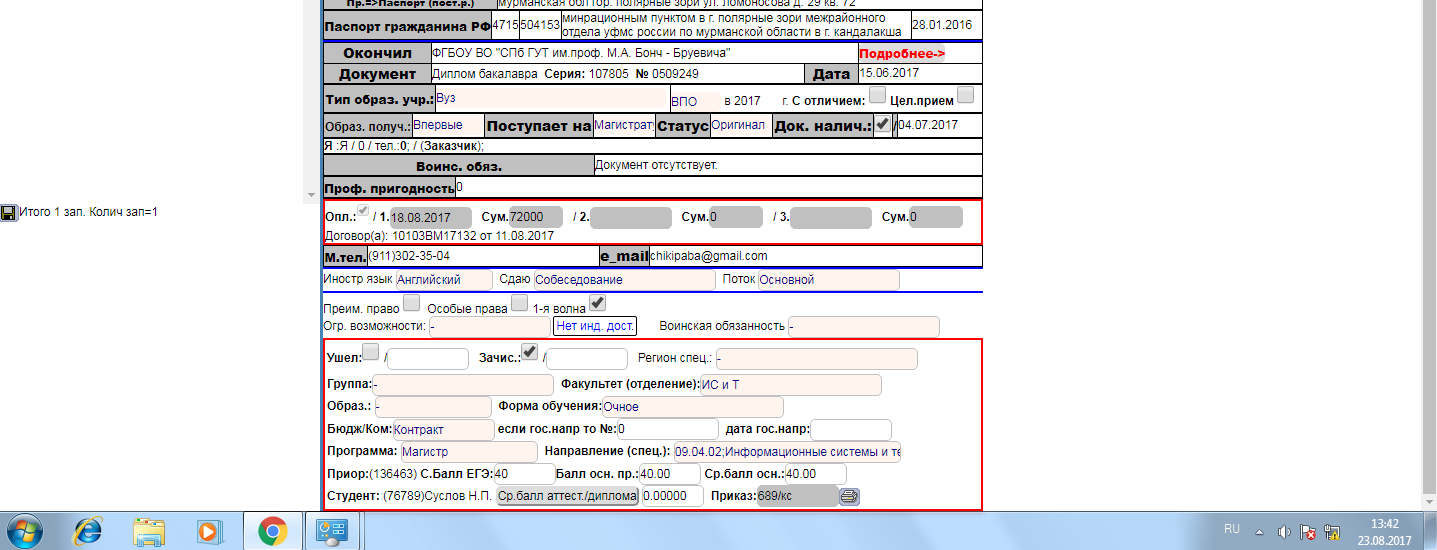
**Приложение**

****

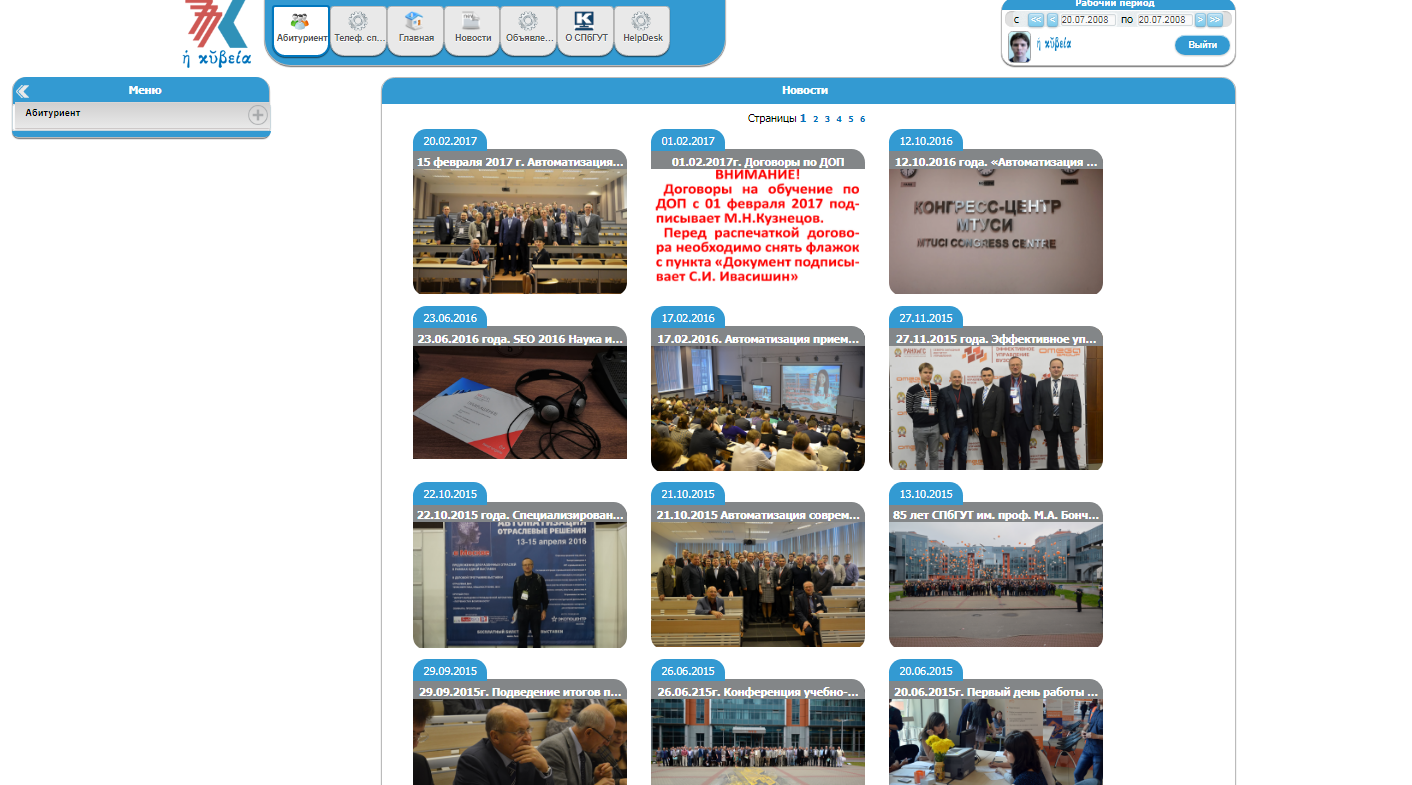
**Рис. 1. Опись документов в деле.**

****

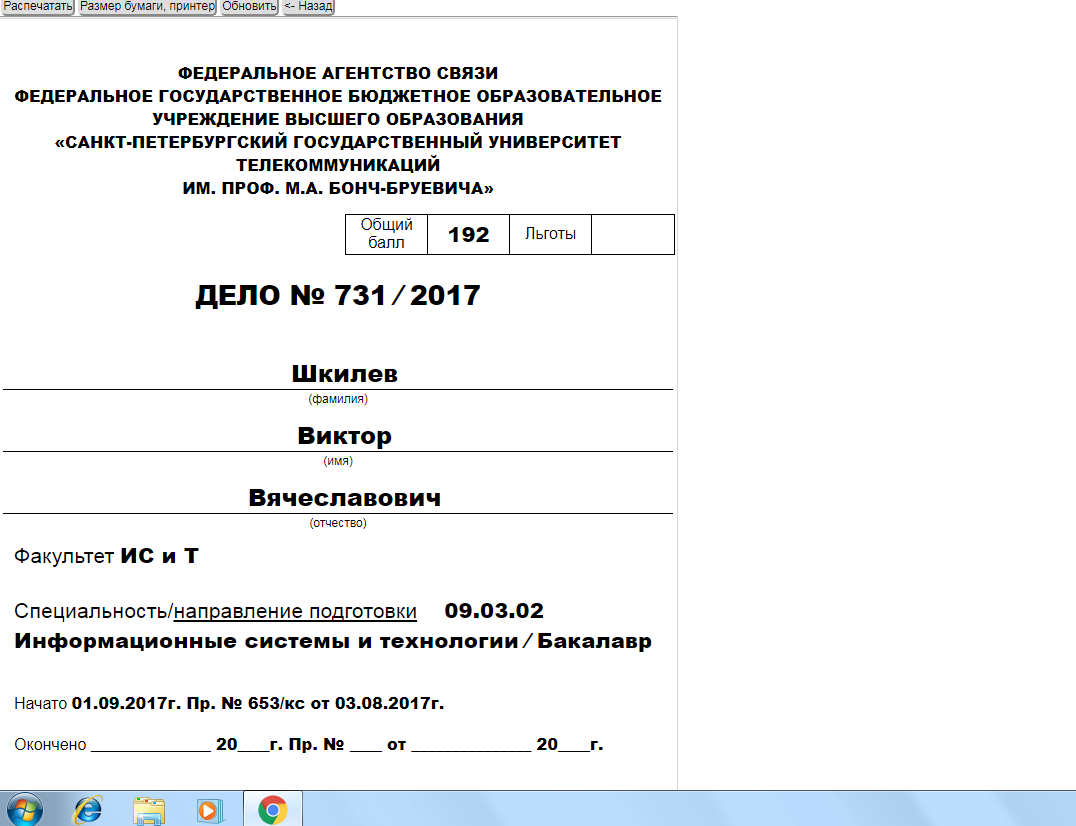
**Рис. 2. Поиск студента по базе**

****

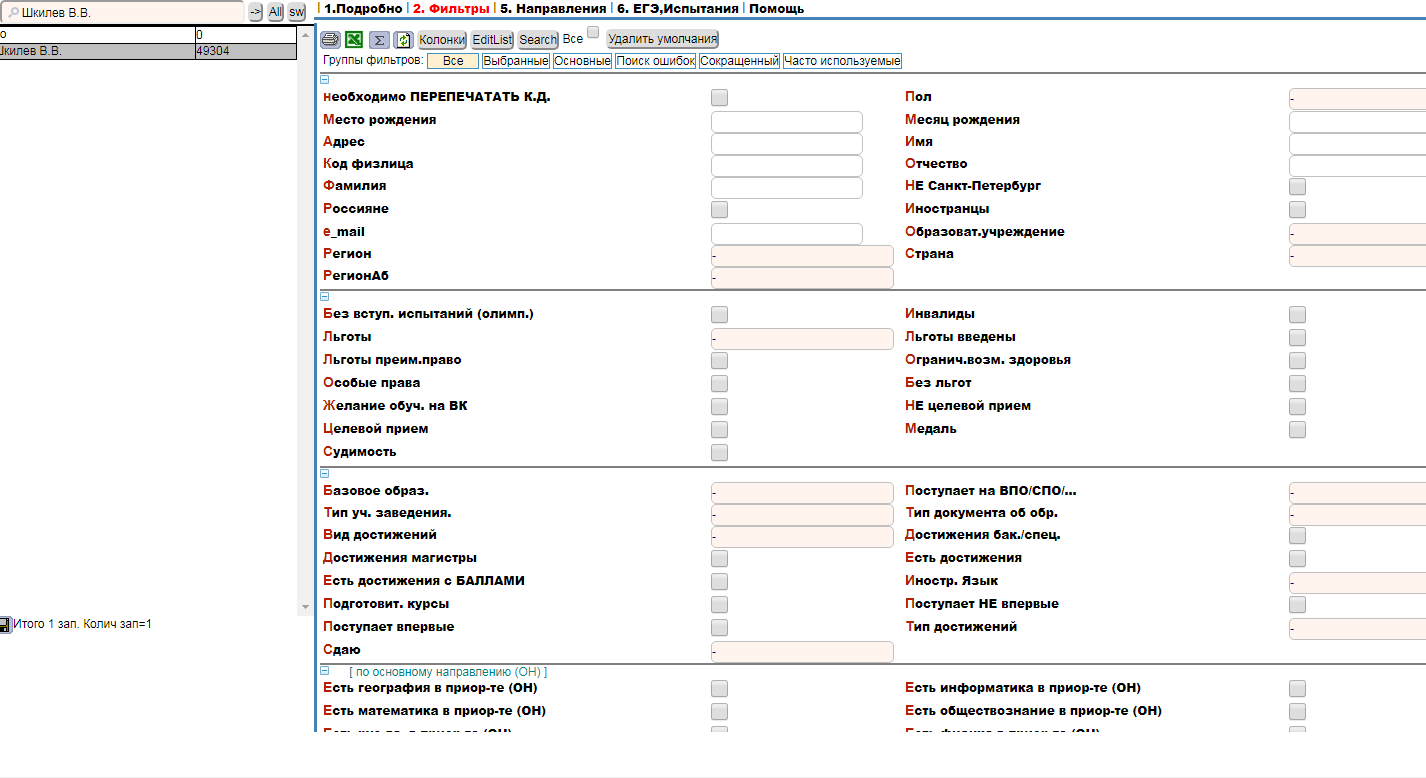
**Рис. 3. Приказ**

****

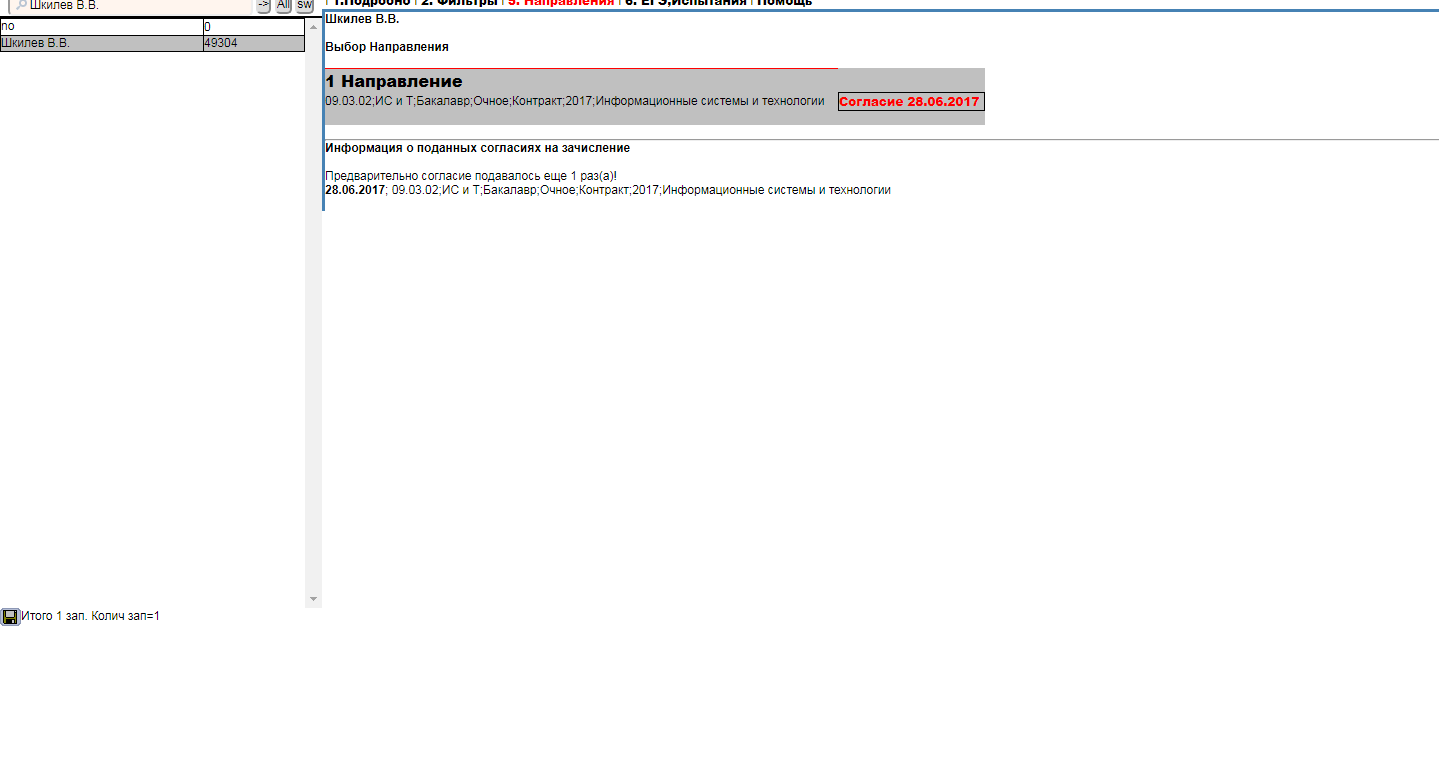
**Рис. 4. Вход в базу данных через лк.**

****

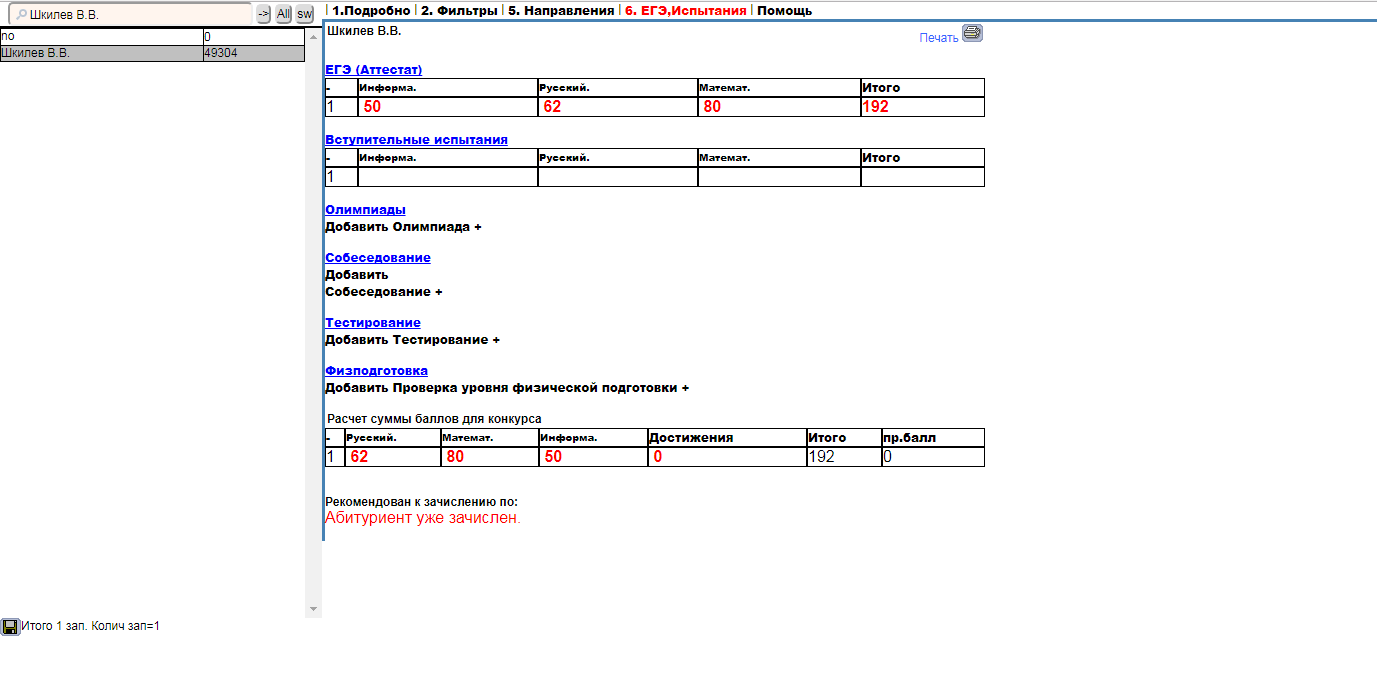
**Рис. 5. Дело**

****

**Рис. 6. Фильтры**

****

**Рис. 7. Направление**

****

**Рис. 8. Испытания**