

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РЕСПУБЛИКИ КАЗАХСТАН

ИННОВАЦИОННЫЙ ЕВРАЗИЙСКИЙ УНИВЕРСИТЕТ

Кафедра «Машиностроение и строительство»

Магистерская диссертация

**ПРИМЕНЕНИЕ КЛИЕНТ-СЕРВЕРНОЙ ТЕХНОЛОГИИ ДЛЯ
АВТОМАТИЗАЦИИ УЧЕТА УСПЕВАЕМОСТИ СТУДЕНТОВ,
ОБУЧАЮЩИХСЯ ПО КРЕДИТНОЙ ТЕХНОЛОГИИ ОБУЧЕНИЯ**

6N0602 «Информатика»

Исполнитель _____ Р.В.Кравченко
(подпись, дата)

Научный руководитель

к.т.н., профессор _____ Н.М. Зайцева
(подпись, дата)

Допущен к защите:
Зав. Кафедрой «Машиностроение и строительство»

к.т.н., профессор _____ С.А.Стаценко
(подпись, дата)

Павлодар, 2011

СОДЕРЖАНИЕ

	Введение	3
1	Управление учебным процессом в условиях современных <u>информационных технологий</u>	6
1.1	Обзор современных информационных систем в образовательной среде	6
1.2	Переход вуза к электронной системе управления учебным процессом в условиях кредитной технологии	7
1.3	Технологические принципы компьютерной автоматизации учебного процесса	9
2	Применение инструментальных средств для построения информационной системы	18
2.1	Функции и задачи разрабатываемого автоматизированного рабочего места «Успеваемость» информационной системы «е-ВУЗ» ИнЕУ	18
2.2	Выбор операционной системы и протокола сети	18
2.3	Выбор СУБД	20
2.3.1	Обзор наиболее известных СУБД	20
2.3.2	Критерии выбора СУБД	27
2.4	Реализация технологии клиент-сервер на базе программных средств Microsoft	31
2.4.1	Выбор сервера баз данных	32
2.4.2	Microsoft Access как инструментальная среда разработки клиентского приложения.	36
2.5	Реализация базы данных для проектируемого автоматизированного рабочего места	37
3	Проектирование автоматизированного рабочего места информационной системы «е-ВУЗ ИнЕУ»	43
3.1	Анализ проблемы	43
3.2	Методы проверки корректности вводимой первичной информации	43
3.3	Разработка автоматизированного рабочего места для учета успеваемости студентов обучающихся по кредитной системе обучения	46
3.3.1	Структура программы	46
3.3.2	Описание работы АРМ «Успеваемость»	47
3.4	Апробация АРМ «Успеваемость»	59
	Заключение	60
	Список использованных источников	62
	Приложение А Рейтинговая ведомость	64
	Приложение Б Экзаменационная ведомость	65
	Приложение В Сводно-рейтинговая ведомость	66
	Приложение Г Листинг программы	67

ВВЕДЕНИЕ

Актуальность исследования. При переходе системы образования на кредитную технологию обучению для многих вузов Казахстана стал актуальным вопрос повышения эффективности управления учебным процессом. В современных условиях одним из основных направлений повышения эффективности управления является использование автоматизированных информационно-управляющих систем. Теперь споры и дискуссии ведутся не о необходимости автоматизации, а на тему способов ее осуществления. Каждый проект в области автоматизации должен рассматриваться вузом как стратегическая инвестиция средств, которая должна окупиться за счет улучшения управленческих процессов, повышения эффективности образования, сокращения издержек. В выборе правильного решения должно быть, в первую очередь, заинтересовано руководство предприятия.

В настоящее время информационная система Инновационного Евразийского университета (далее ИнЕУ), как правило, должна обеспечивать работу всех структурных подразделений. Процесс автоматизации начинается с анализа деятельности вуза и выработки основных рекомендаций к будущей информационной системе. Только после этого решается вопрос выбора той или иной готовой системы или разработки собственной. В Казахстане гораздо чаще, чем за рубежом, принимается решение о проведении разработки собственной информационной системы. В этом случае, приходится решать целый ряд проблем таких как, выбор базового программного и аппаратного обеспечения, проектирование функциональной структуры информационной системы, проектирование базы данных и расчет параметров ее функционирования.

Цель и задачи исследования. Цель диссертационной работы заключается в исследовании, разработке и обосновании подходов использования клиент-серверной технологии при автоматизации и систематизации учета успеваемости студентов, обучающихся по кредитной технологии обучения путем построения и внедрения одного из автоматизированного рабочего места (далее АРМ) информационной системы ИнЕУ, АРМ «Успеваемость».

Для достижения поставленной цели были определены и успешно решены следующие задачи:

- анализ состояния казахстанского рынка информационных систем в образовательной среде;
- выявление критерии выбора базовых программных продуктов;
- разработка принципов, моделей и методов построения базы данных для вуза;
- проектирование и внедрение АРМ «Успеваемость».

Предмет исследования - применение инструментальных средств Microsoft в задачах компьютерной обработки и ввода информации.

Объект исследования — инструментальные средства программирования баз данных, поставляемые фирмой Microsoft.

Методы исследования. Результаты диссертационной работы получены

на основе комплексного использования теорий баз данных, клиент-серверных технологий, вычислительных сетей и кредитной технологии, а также теории алгоритмов.

Научная новизна состоит в том, что была найдена логически завершенная технология применения инструментальных средств фирмы Microsoft не как традиционных средств программирования баз данных, а именно как инструментария построения информационных систем, для автоматизации управления учебным процессом. Это позволило создать специально предназначенную для этих целей программную платформу и сборочную технологию построения автоматизированного рабочего места информационной системы, за счет чего эффективность разработки была повышена в несколько раз.

Также в результате проведенных исследований, анализа и обобщения опыта проектирования и эксплуатации автоматизированных систем управления:

- разработан общий подход, теоретические положения и формализованная модель для проектирования АРМ в кредитной технологии обучения;
- поставлена и решена задача определения оптимальных параметров функционирования клиент серверной технологии для решения поставленной задачи;
- определены основные особенности автоматизированных рабочих мест как объектов управления и предложена новая информационная технология управления.

Практическая значимость исследования результатов работы заключается в открывающейся возможности широкого использования разработанного программного - аппаратно комплекса в сфере образования.

Получение общей технологической платформы, специально предназначенной для компьютерной реализации различных схем ведения управления учебным процессом в масштабах вуза на основе клиент-серверной технологий Microsoft;

Возможности свободного применения полученных технологических методов в связи с общедоступностью и повсеместным распространением программных средств Microsoft в нашей стране; реализации предложенных технологических методов построения в форме сетевых программных решений масштаба вуза, которые могут быть внедрены в промышленную эксплуатацию на большом количестве вузов Казахстана.

Кроме этого полученные методы могут быть использованы в иных сферах, как при проектировании новых корпоративных информационных систем управления вузом или предприятием, так и для повышения эффективности существующих систем.

Структура исследования. Диссертационная работа состоит из введения, трех глав, заключения, списка использованной литературы и приложений.

Во **введении** к диссертации обосновывается актуальность темы, раскрывается общее состояние проблемы и степень ее изученности, определяются цель и задачи исследования, представлена научная новизна, излагаются методические подходы к рассмотрению проблемы, практическая значимость и апробация полученных результатов.

В первой главе диссертации **«Управление учебным процессом в условиях современных информационных технологий»** проводится анализ возможностей применения электронной информации в образовании с целью поддержки принятия управленческих решений и рационализации использования ресурсов вуза. Рассматриваются современные технологические методы построения автоматизированных информационных систем по кредитной технологии обучения, проводится обзор современных информационных систем в образовательной среде Казахстана.

Во второй главе **«Применение инструментальных средств построения информационной системы»** проводится анализ применения функциональных возможностей инструментальных средств, для программной реализации функциональности системы управления учебным процессом в масштабах вуза. Технологический процесс построения стандартизируется с учетом кредитной специфики управления учебным процессом, вырабатывается общая технология построения компьютерных систем, основанная на принципах сборочного программирования на основе ядра из стандартных программных модулей с требуемым набором настроек.

В третьей главе **«Проектирование автоматизированного рабочего места информационной системы «е-ВУЗ ИнЕУ»»** проводится экспериментальная проверка разработанных технологических методов на реальных областях управления учебным процессом. Показывается применение технологии и высокая эффективность разработанных методов построения информационных систем.

В заключении диссертации сформулированы основные выводы и рекомендации.

Реализация результатов работы. Спроектированный программный комплекс АРМ «Успеваемость» был успешно внедрен в АСУ «е-ВУЗ» ИнЕУ.

Следует отметить, что применение результатов работы позволило существенно сократить время проектирования.

Апробация результатов и публикации. Основные положения и результаты диссертации докладывались и обсуждались руководством ИнЕУ. По теме диссертации опубликовано 3 статьи.

Разработанное АРМ введено в эксплуатацию, что подтверждается актом внедрения.

1 УПРАВЛЕНИЕ УЧЕБНЫМ ПРОЦЕССОМ В УСЛОВИЯХ СОВРЕМЕННЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

1.1 Обзор современных информационных систем в образовательной среде

В настоящее время на казахстанском рынке предлагается около 10 программных продуктов управления учебным процессом вуза. Наиболее известными ИС отечественной разработки являются: «PLATONUS»; «АЛМА DATA»; «Tamos University Suite» и др.

Исследование и анализ рынка программного обеспечения для ВУЗов Казахстана позволил рассмотреть программные продукты и выявить их преимущества и недостатки.

Система «PLATONUS» разработана вендором программного обеспечения компанией «Арта». Это автоматизированная информационная система для ВУЗов и колледжей, включающая в себя подсистемы администрирования студентов, поддержки учебного процесса и дистанционного обучения, объединенных системой электронного документооборота.

Преимущества:

- web-интерфейс (нет нужды в клиентской части);
- кредитная система обучения;
- большие возможности.

Недостатки:

- закрытый код (нет возможности подстраивать под себя);
- отсутствие некоторых функций.

АЛМА DATA. Конфигурация Автоматизация учебного процесса для Казахстана» разработана компанией «1С-Сапа».

Преимущества:

- централизация;
- большие возможности.

Недостатки:

- привязка к системе 1С;
- отсутствие некоторых функций.

Автоматизированная система управления обучением «Tamos University Suite» разработана компанией «Tamos Development Ltd» представляет собой комплексную программу, включающую в себя автоматизированную систему управления процессом обучения и технологию дистанционного обучения.

Преимущества:

- web-интерфейс (нет нужды в клиентской части);
- централизация.

Недостатки:

- закрытый код (нет возможности подстраивать под себя).

Общим недостатком для всех рассмотренных систем является высокая

стоимость приобретения

1.2 Переход вуза к электронной системе управления учебным процессом в условиях кредитной технологии

В современных условиях применение электронной учетной информации с целью повышения экономической эффективности работы вуза стало новым этапом развития управления учебным процессом при кредитной технологии обучения, а владение технологиями обработки информации — одним из ведущих факторов повышения рентабельности и конкурентоспособности. Увеличение роли информационного фактора в управлении подтверждается и повсеместным распространением информационных технологий (ИТ) в образовательной среде, а также стремлением многих вузов к внедрению автоматизированных систем управления. Происходящие изменения технологий управления в экономике, безусловно, не могут остаться без внимания исследователя. Здесь наблюдается именно та ситуация, когда технический прогресс, в данном случае, бум в секторе информационных технологий, открыл новые технологические возможности повышения экономических показателей и создал богатую нишу для направления экономико-информационных исследований. Тем не менее, постановка основной проблемы проектирования - проблемы наилучшего использования ограниченных ресурсов, в данном случае ресурсов вуза, осталась неизменной, поэтому в работе основное внимание будут уделяться новым технологическим подходам к решению этой проблемы — методам построения компьютерных систем и обработки внутренней электронной информации вуза. Формирования электронных баз данных, представляющих информационную модель деятельности вуза, которые можно использовать для экономической рационализации работы вуза с помощью программных средств обработки информации заслуживают детального рассмотрения, начиная с первооснов явления, что поможет в дальнейшем найти наиболее обоснованные решения для поставленных задач.

Таким образом, решаемая технологическая задача создания эффективной системы управления упирается в создание эффективной системы информационного обеспечения управления. Под инструментальной средой будем понимать совокупность информации и информационных технологий. Практически опыт показывает, что во многих случаях при решении локальных оперативных проблем бывает достаточным получить требуемые ключевые данные, после чего решение становится очевидным [1].

Построение электронных баз данных, моделирующих деятельность вуза, основывается на информационно-логическом моделировании системы регистрации информации, которая называется **учетной системой**. Система формируется с целью обеспечения информационных потребностей управления. По своей информационной функциональности управленческий учет ориентирован на получение необходимой информации для реализации управленческих целей вуза - рационализации использования процессов управления и улучшения результатов его работы.

Среди наиболее существенных технологических возможностей [2, 3], влияющих на систему управления, можно выделить следующие:

доступность: любая информация о работе объекта управления может быть получена в любое время с рабочего места руководителя;

оперативность: обработка и передача необходимой информации производится почти мгновенно;

информационная прозрачность: возможность получения любой отчетной информации для анализа, контроля и принятия решений.

Электронная информация системы с инструментальными средствами ее обработки формируют **информационную среду**, которая позволяет оперативно получать необходимые для управления вузом учебные процессы. Электронная система управления предоставляет требуемые технологические возможности для того, чтобы сделать среду управления **информационно совершенной**, то есть не имеющей ограничений, связанных с недоступностью информации по техническим и другим причинам. Те вузы, которые в большей степени владеют информацией, получают конкурентные преимущества по сравнению с другими, таким образом, возможность применения ИТ в образовании порождает новый вид конкурентных преимуществ - **информационную конкуренцию** - конкуренцию по владению технологиями и методами использования внутренней учетной информации для рационализации и повышения эффективности работы вуза. Информационная конкуренция существует наряду с такими видами конкуренции, как конкуренция по качеству и издержкам. Стремление получить конкурентные преимущества по уровню информационного обеспечения управления и готовность вуза во внедрение компьютерных систем управления учебным процессом выступает основным стимулом развития систем и программных средств автоматизации, поэтому данная сфера исследований имеет выраженную практическую направленность и ориентацию на условия кредитной технологии обучения.

На уровне вуза достижение указанных целей невозможно без применения клиент-серверной информационной системы, которая поддерживает совместную работу большого количества сотрудников в едином информационном пространстве при решении общих задач вуза.

В условиях современных информационных технологий указанные требования реализуются:

- **общим сервером баз данных**, на котором хранится база данных с информацией системы и который выполняет функции обработки данных по запросам пользователей и передачи итоговых наборов данных клиентским приложениям;

- **локальной вычислительной сетью** вуза (LAN), по которой осуществляется обмен информацией между пользователями системы;

- **клиентскими приложениями** пользователей, установленными локально на рабочих станциях, которые выполняют функцию графического интерфейса (оболочки) для работы с данными.

Многопользовательская среда работы с информацией системы является на сегодняшний день наиболее передовой и эффективной технологией построения систем автоматизации управленческого характера в масштабах вуза. **Сетевые технологии** баз данных позволяют создавать информационные системы, которые автоматизируют совместную работу большого количества сотрудников посредством обмена учетной информацией локальной компьютерной сети вуза LAN (local area network) через общую **базу данных**. Так создается единое **информационное пространство**, которое позволяет руководству автоматизировать оперативный анализ работы и вносить корректирующие управленческие решения в работу сотрудников и подразделений.

После того как определены основные технологические требования к компьютерным системам управления учебным процессом в вузе, необходимо установить критерии выбора инструментальных средств для работы с информацией систем, в том числе, с учетом специфики казахстанского программного рынка. В Казахстане с начала 2000-ых годов интерес к системам автоматизации управленческого характера устойчиво увеличивался, в значительной мере, в связи с возросшими возможностями и удешевлением программных средств автоматизации, при этом наблюдалось запаздывание относительно запада на несколько лет. Инструментарий для системы управления учебным процессом - одно из самых больных мест в силу большой трудо- и капиталоемкости. Нередко создание программного продукта обеспечивающего систему затмевает задачу создание системы в принципе. Кроме того, возможности использования внутренней учетной информации вуза во многом ограничены возможностями существующих программных продуктов, а сфера применения управления учебным процессом в вузе существенно шире, чем реализовано в существующих системах автоматизации. Из этого следует, что решение проблемы следует искать именно **в разработке универсальных, бюджетных и общедоступных методов построения компьютерных систем автоматизации в образовании**, которые позволяют с минимальными издержками осуществлять компьютерную поддержку любых схем ведения управления учебного процесса. В качестве базовой инструментальной платформы автором предлагаются программные средства фирмы Microsoft, выбор которых будет подробно обоснован ниже.

1.3 Технологические принципы компьютерной автоматизации учебного процесса

Перевод функций учебного процесса на программные средства автоматизации. Как показывает опыт, система управления учебным процессом, спроектированная на логическом уровне (включая модель учебных процессов, схему документооборота), не найдет своего практического применения до тех пор, пока ее учетные функции не будут автоматизированы с применением компьютерных технологий. Тот факт, что

система управления учебным процессом - это, прежде всего, **информационная система**, говорит о необходимости применения современных информационных технологий для перевода функций учета и анализа информации на программные средства автоматизации. Программный продукт, реализующий требуемые учетные функции системы, должен выступать как качественно новое практическое решение задач и управление учебным процессом вуза, которое принципиально изменяет процесс управления в образовании.

Электронная система управления должна содержать все необходимые данные о работе объекта, по которым можно полностью воспроизвести историю совершения операций, соблюдая все логические связи между ними. В соответствии с концепцией А. Саймона [4], электронная система учета выступает информационным отображением и аналогом реальной системы, поэтому реализация учетной системы в виде программного продукта представляет собой **метод компьютерного моделирования** работы реального объекта (в нашем случае - вуза). Удобство и практичность этого метода заключается в том, что модель системы в виде программного продукта достаточно просто внедрить в эксплуатацию, и, таким образом, осуществить верификацию сформированной модели ведения учета, которая заложена в программном продукте.

Процесс построения логически завершенной и полнофункциональной системы управленческого учета является итеративным, что, вообще говоря, является свойством систем управления. Это обосновывается как цикличностью их развития, подтвержденной практическим опытом большого количества разработчиков, так и логически тем, что на начальном этапе проектирования невозможно учесть все особенности работы сложной системы. При этом информация о работе реальной системы поступает к разработчику постепенно, по мере ее эксплуатации в вузе и обновления версий. В ходе итеративного разрешения логических противоречий между логикой работы реального объекта и ее программной реализацией, в конечном счете, получается завершенная модель системы управления, которая представляет собой информационный аналог работы объекта [5]. По опыту автора, для того чтобы получить завершенную модель системы управления, требуется выпустить не менее 5-10 версий программного продукта. Таким образом, наличие эффективной технологии и инструментов сопровождения работающего программного продукта имеет не меньшее значение, как и непосредственно технология разработки. Сам же метод внедрения компьютерных систем автоматизации и последовательного устранения несоответствий в логике работы программы и объекта учета является эффективным инструментом практического применения, развития и совершенствования моделей управления.

Экономическое значение факта получения АСУ, которая содержит и обрабатывает электронную информацию той или иной сферы бизнес-деятельности вуза, трудно переоценить, особенно в современных условиях.

Пользователи могут не только следить за совершением всех текущих операций, но и автоматически формировать отчетность по этой информации, которые предоставляют систему эффективности работы вуза, что особенно важно при поддержке принятия оперативных и стратегических решений.

Применение технологий баз данных в автоматизированных системах учета. Основным технологическим инструментом хранения и управления информацией в компьютерных системах управления учебным процессом является **база данных**, в которой накапливается учетная информация по совершаемым операциям. База данных как система хранения и управленческого анализа учетной информации работает под управлением программного средства - **системы управления базами данных (СУБД)**, которая выполняет все необходимые функции по:

- физическому размещению данных на носителе информации;
- поддержки целостности информации;
- выполнению обработки данных с целью получения требуемой отчетности;
- взаимодействию с пользователями.

Отсюда, вследствие своего функционального назначения, основным применением баз данных выступает автоматизация учетных и аналитических функций.

База данных как программное средство быстро нашло применение, как в бизнесе, так и в образовательной среде.

На ранних этапах развития технологии баз данных (начиная с конца 60-х) в качестве основных логических моделей организации данных применялись **сетевая и иерархическая** модели [4, 6, 7]. На сегодняшний день основным промышленным стандартом является **реляционная** модель данных, впервые предложенная в 1970 г. Э. Ф. Коддом [8] и нашедшая повсеместное распространение в различных сферах применения баз данных, в том числе в задачах построения учетных систем для приложений. В реляционной модели вся информация системы распределяется по связанным двумерным таблицам-отношениям [6], в соответствии с принципами нормализации [9], которые позволяют избежать ее дублирования.

По возможностям одновременной работы с учетной информацией базы данных условно разделяются на два вида:

- однопользовательские (настольные, desktop)
- многопользовательские (серверные, сетевые)

На ранних этапах развития систем управления базами данных (80-е, начало 90-х) из-за ограниченности возможностей аппаратных и программных средств, в основном, использовались настольные базы данных. С точки зрения решения информационных проблем управления на уровне вуза, однопользовательские базы данных имеют очень ограниченное применение, так как не позволяют обеспечить одновременную работу большого числа сотрудников, совместно выполняющих связанные друг с другом операции. Кроме того, однопользовательские базы данных

характеризуются относительно низкой производительностью и ограниченными функциональными возможностями.

Учитывая указанные недостатки, развитие баз данных пошло в сторону сетевых технологий и серверных систем управления базами данных, о чем свидетельствует большое количество публикаций в научной и профессиональной литературе, появившихся во второй половине 90-х годов [10]. С начала нового тысячелетия клиент-серверные системы уже стали промышленным стандартом автоматизации.

В сетевой архитектуре учетной системы выделяется общая информационная часть системы - база данных вуза, которая также называется **серверной частью** или **сервером**. Выделенный в сети компьютер, на котором размещена база данных, также носит название сервера. База данных реализует логику последовательной регистрации связанных операций, хранения информации документов, фиксирующих операции. Таким образом, реализуется система **электронного документооборота**, который является информационным аналогом системы бумажного документооборота [10]. Система электронного документооборота имеет ряд ключевых преимуществ, заключающихся в скорости доступа и обработки информации, в том числе формирования производных документов на основе уже созданных первичных, исключении ошибок и проблем анализа информации, связанных с многократным ручным вводом информации и ее дублированием.

Обмен информацией между пользователями производится через сервер баз данных по **локальной сети** вуза (LAN - Local Area Network). Работа в сетевом режиме избавляет от необходимости передавать документы в бумажном виде, создает оперативность работы и увеличивает скорость проведения операций, что немаловажно в современных условиях.

Функции графического пользовательского интерфейса (GUI) для обмена информацией с сервером баз данных и отображения полученных данных на компьютере пользователя выполняет **клиентское приложение**.

На сегодняшний день существующие программные средства создают практически неограниченные возможности по созданию сетевых систем, поддерживающих одновременную работу большого количества пользователей в сетевом режиме с общими базами данных вуза. Прежде чем перейти к применению конкретных СУБД и разработке архитектуры управления системы масштаба вуза, необходимо рассмотреть существующие сетевые технологии совместной работы с информацией систем управленческого учета и определить, какая из них наиболее подходит для разработки автоматизированных систем управленческого учета.

В качестве основных принципиально различных сетевых технологий выделяют технологии:

- файл-сервер;
- клиент-сервер.

Принципиальная разница между ними заключается в распределении вычислительной нагрузки по обработке данных между базой данных и пользовательскими приложениями [7], что проявляется в производительности системы, в том числе в:

- скорости обработки учетной информации;
- величине трафика информации по сети;
- ограничении на количество информации в базе данных;
- возможности одновременной работы с общей информацией.

Технология файл-сервер. Технология файл-сервер исторически была первой технологией работы с общими информационными ресурсами в сетевом режиме [11] и является базовой для локальных сетей. Модель файлового сервера использовалась во многих системах автоматизации управления учебного процесса в вузе, созданных на основе таких систем управления базами данных, как FoxPro, Clipper, Paradox и т.д. [12].

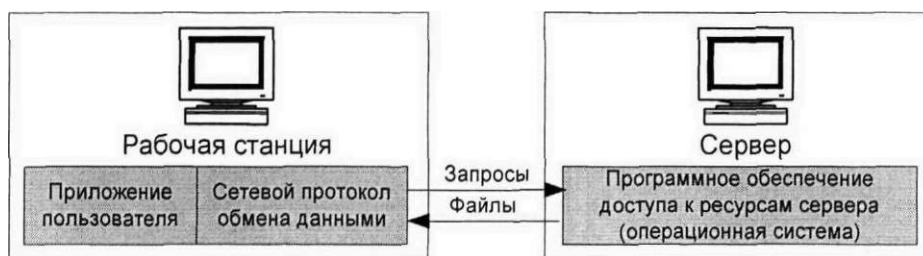


Рисунок 1.1. Модель доступа к данным по технологии файл-сервер

В модели файл-сервер один из компьютеров выполняет функции файлового сервера и предоставляет другим компьютерам сервис по передаче файлов. Файловый сервер работает под управлением сетевой операционной системы (например, Novell NetWare) и предоставляет сетевой доступ к общим информационным ресурсам (т.е. к файлам). На других компьютерах сети функционирует приложение, которое запрашивает нужные файлы постпредством коммуникационного протокола обмена файлами с сервером (Рисунок 1.1).

Технология характеризуется относительно низкой производительностью и представляется бесперспективной для разработки и развития системы управления учебного процесса. В процессе накопления учетной информации размер базы данных, как правило, достаточно быстро увеличивается, в результате чего возрастает сетевой трафик, снижается скорость работы приложения пользователя, таким образом, производительность системы снижается, иногда до критического уровня, когда оперативная работа с информацией становится невозможной. Кроме того, локальная обработка информации на рабочих станциях пользователей накладывает определенные ограничения на возможности совместной работы, а также не позволяет разграничить права доступа к информации на уровне

файла. Из этого следует, что архитектура файл-сервер обладает низким показателем масштабируемости, то есть является чувствительной к увеличению объемов обрабатываемой информации. Поэтому в современных условиях информационной системы стала модель клиент-сервер [10], которая технологически более совершенна и позволяет преодолеть указанные проблемы.

Технология клиент-сервер. Технология клиент-сервер возникла в начале 90-х годов и исходно была связана с парадигмой открытых систем [13, 14]. Главная оптимизационная идея технологии заключается в том, что пользователю для обеспечения текущих информационных потребностей при оперативной работе нужна не вся информация базы данных, а только небольшая ее часть, которую можно получить на сервере и переслать на рабочую станцию пользователя. Необходимая информация определяется условиями запроса, который формируется клиентом, затем принимается и обрабатывается сервером. Таким образом, оптимизируется трафик сети и перераспределяется нагрузка между сервером и клиентами так, чтобы обеспечить наилучшую производительность при работе системы. Система перестает быть чувствительной к объему информации в базе данных и обладает хорошей масштабируемостью, что имеет принципиальное значение для работы с базами данных.

На сегодняшний день технология клиент-сервер является наиболее перспективной технологией построения сетевой системы масштаба вуза, более подробное описание технологии представлено в [9, 10, 15, 16].

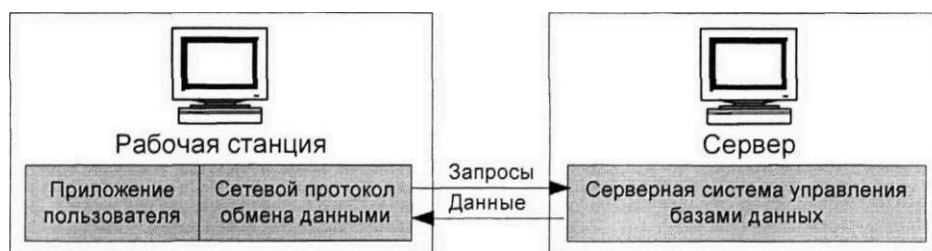


Рисунок 1.2. Модель доступа к данным по технологии клиент-сервер

В технологии клиент-сервер (Рисунок 1.2) база данных работает уже не под управлением сетевой операционной системы, а под управлением серверной системы управления базами данных (**сервера баз данных**), которая обрабатывает запросы пользователей к базе данных. Серверная СУБД может поддерживать одновременную работу большого количества пользователей с разными базами данных, размещенными на сервере.

Среди наиболее мощных на сегодняшний день серверных СУБД промышленного масштаба можно выделить Oracle, Microsoft SQL Server, Borland Interbase.

Концепция тонкого клиента. Из проведенного анализа можно сделать вывод, что секрет производительности информационной системы заключается в том, где производится обработка запросов [10].

В зависимости от типа сетевой архитектуры и от того, насколько оптимально с точки зрения обмена данными реализованы серверная и клиентская части учетной системы, зависит распределение вычислительной нагрузки и объемы передаваемой информации между сервером и клиентом или "толщина" клиента.

Тонкий клиент (thin client) несет на себе минимум нагрузки по обработке информации и выполняет функции графического представления информации для пользователя. Толстый клиент (fat client) берет на себя относительно большую часть обработки данных [10], что отрицательно сказывается на производительности ввиду того, что возрастает сетевой трафик и клиентское приложение выполняет функции обработки информации существенно хуже сервера.

Наиболее прогрессивная на сегодняшний день концепция разработки клиентского приложения - концепция **тонкого клиента**, основным положением которой является то, что главная функция клиентского приложения заключается в предоставлении пользователю графического интерфейса для работы с данными, то есть только функция графического представления данных. Вся вычислительная нагрузка по обработке данных (хранение данных, бизнес-логика) ложится на сервер, который производит обработку данных существенно быстрее, а клиент получает только итоговый обработанный набор записей.

Концепция тонкого клиента позволяет сделать систему максимально производительной, минимизировать трафик сети и сократить требования к аппаратному обеспечению рабочих станций пользователей.

Технологические принципы. В технологии клиент-сервер обмен информацией производится между сервером и клиентскими приложениями по запросам клиентов, что составляет основу работы системы информационного обеспечения управления. По мере того, как реляционная модель баз данных становилась основным мировым стандартом, различные поставщики СУБД объединили свои усилия в выработке общего стандарта языка для управления реляционными данными. Так, в 1974 году разработчик фирмы IBM Дональд Д. Чамберлин предложил первую спецификацию реляционного языка, который получил название SEQUEL (Structured English QUEry Language), который впоследствии был переработан, и в 1977 году он приобрел свое окончательное название SQL (Structured Query Language) [4].

На сегодняшний день основным стандартом реляционных СУБД является структурированный язык запросов SQL, последним вариантом которого является ANSI SQL - 92, при этом различными СУБД может поддерживаться свой вариант SQL.

SQL представляет собой декларативный (в отличие от процедурных) язык операций над множествами, который предоставляет возможность в максимально лаконичной форме сформировать запрос к СУБД [6] и избежать написания алгоритмических процедур на языке программирования для выполнения требуемого запроса.

Язык SQL предоставляет полный набор команд (запросов) для управления реляционными базами данных, начиная от создания базы данных (оператор CREATE DATABASE) и до извлечения требуемого табличного набора данных из таблиц базы данных (оператор SELECT). Например, для внесения изменений в реляционные данные используется три вида запросов (подробнее см. справочную информацию по языку SQL):

- запрос на добавление (оператор INSERT)
- запрос на изменение (оператор UPDATE)
- запрос на удаление (оператор DELETE)

Таким образом, в информационной системе язык SQL является основой автоматизированной обработки информации, так как позволяет выполнять все необходимые операции над данными и получать требуемые представления информации. В свою очередь, сервер баз данных выступает как основное инструментальное средство построения автоматизированной системы управления учебным процессом вуза. Клиентское приложение выполняет только функции графического представления полученных данных в удобной для пользователя экранной или печатной форме. В различных клиентских приложениях используются разные технологии создания графических объектов для представления данных, однако, в среде операционной системы Windows для всех них общим является стандартный графический интерфейс, порожденный классом Windows.Forms, и возможность применения функций для управления графическим интерфейсом Windows API.

Инструментальные средства построения информационных систем. Прежде чем перейти к применению технологий фирмы Microsoft, необходимо провести экспресс-анализ инструментальных средств, которые могут выступать программной платформой для построения системы и выделить преимущества и недостатки тех или иных платформ. Провести полный анализ не представляется возможным, так как программные продукты, которые много лет разрабатывались ведущими мировыми поставщиками, настолько сложны и функциональны, что их можно сравнивать по сотням критериев. Стоит отметить, что из существующих на рынке инструментальных средств разработки ни одна система не доминирует по всем критериям, в противном случае, все остальные потеряли бы свою нишу на рынке, поэтому ограничимся рассмотрением ключевых преимуществ инструментальных средств Microsoft для построения сетевого автоматизированного рабочего места в масштабе вуза.

Основой учетной системы масштаба вуза выступает серверная база данных, в которой хранится и обрабатывается учетная информация. На сегодняшний день наиболее распространены базы данных Oracle, Microsoft SQL Server, IBM DB/2, Sybase ASE, Borland Interbase, среди которых ведущие позиции занимают Oracle и MS SQL (первое и второе место по продажам в мире, соответственно [18]). Создавать АРМ, которая будет эксплуатироваться много лет, имеет смысл на платформе надежного поставщика, в котором

можно быть уверенным, что выбранная платформа достаточно функциональна и надежна, а разработанное АРМ просуществует на этой платформе не один год и с возможностью развития и доработки. Лидером, в этом смысле, безусловно, является Oracle, который выступает основной платформой для систем, в которых требуется обработка огромных массивов информации (например, банковские системы). Кроме того, Oracle не только является наиболее функциональной платформой, но и зарекомендовал как ведущий мировой поставщик, СУБД которого наиболее полно отвечают современным требованиям по функциональности, производительности и надежности. Недостатком Oracle является то, что это достаточно дорогая платформа, которую могут позволить себе крупные компании, внедряющие достаточно масштабные информационные системы. Говоря об автоматизации управления учебным процессом, здесь возникает та ситуация, когда цена базовой платформы, зачастую, перекрывает цену самой системы автоматизации. Для того чтобы решить проблему доступности базовых инструментальных средств построения информационных систем управленческого характера, была выбрана СУБД Microsoft SQL Server. Как платформа для компьютерных систем он удобен не только тем, что популярен в нашей стране и более доступен по цене, но и тем, что он может эффективно применяться в бесплатной настольной редакции, что является идеальным решением для автоматизации небольших рабочих групп (5-10 человек). Помимо этого, MS SQL достаточно функционален для задач управления учебного процесса на уровне вуза, для которого не требуется обработка терабайтов информации и удобен для программирования.

Выводы

В данной главе был проведен анализ возможностей применения технологий баз данных в автоматизированных системах учета, применение электронной информации с целью поддержки принятия управленческих решений и рационализации использования ресурсов.

Рассматривались современные технологические методы построения автоматизированных информационных систем.

Был проведен обзор современных информационных систем в образовательной среде Казахстана.

2 ПРИМЕНЕНИЕ ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ ПОСТРОЕНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ

2.1 Функции и задачи разрабатываемого автоматизированного рабочего места «Успеваемость» информационной системы «е-ВУЗ» ИнЕУ

При определении требований к структуре АРМ учитывался перечень функций и задач, подлежащих автоматизации, а также требования к составу информационного и организационного обеспечения информационной системы.

Создаваемое АРМ по характеру своего функционирования является информационно-вычислительным. Оно призвано накапливать, хранить, корректировать, обрабатывать, рассчитывать и выдавать информацию.

Информационным фондом данной системы является особым образом учет успеваемости студентов обучающихся по кредитной технологии.

Исходя из анализа недостатков обработки документации, можно сделать вывод, что основными недостатками являются неупорядоченность хранения информации, увеличение трудоемкости и времени обработки документации. Следовательно, основными функциями информационной системы будут такие функции, как эффективная обработка и организация централизованного хранения информации.

Исходя из определения информационной системы, представляется оправданным выделение функций проектируемой информационной системы:

- сбор и хранение информации;
- автоматизированный и ручной ввод информации;
- оперативный доступ к информации;
- обработка и формирование информации в адекватные временные рамки;
- автоматизированный расчет информации;
- вывод информации;
- оперативная защита информации.

2.2 Выбор операционной системы и протокола сети

Правильный выбор операционной системы требует тщательного взвешивания всех "за" и "против", так как ошибка в этом вопросе приводит к неоправданным затратам на разработку прикладных программ, эксплуатацию самой вычислительной системы и ее стыковки с вычислительными сетями.

В первую очередь следует отметить, что задача выбора операционной системы распадается на две подзадачи: выбор операционной системы сервера и выбор операционной системы для клиента.

Выбор операционной системы для клиентской части.

На выбор операционной системы самое прямое влияние оказывают аппаратные средства. Оставшихся после загрузки операционной системы ресурсов должно хватать для стабильной работы пользовательских приложений. Поэтому при выборе оцениваются потребности в оперативной памяти, дисковой памяти и вычислительных ресурсах процессора. Очень

важным является набор поддерживаемых аппаратных платформ. Поскольку корпоративная информационная система предполагает наличие компьютерной сети, операционная система должна поддерживать ряд сетевых протоколов.

Существенное влияние на выбор операционной системы оказывает специфика пользовательских приложений. Кроме этого операционная системы должна обеспечивать определенный уровень защиты данных от несанкционированного доступа. При выборе следует учитывать также стоимость операционной системы, поскольку даже небольшая разница в стоимости одной лицензии умноженная на число пользователей может вылиться в круглую сумму.

В качестве операционных систем клиентской части, как правило, выбирают Windows, OS/2, Linux и др. В настоящее время лидером является Windows. Несмотря на многочисленные упреки, именно она удовлетворяет основным потребностям пользователей. Поэтому практически все коммерческие продукты имеют версии для Windows.

Выбор операционной системы для сервера

К операционной системе для сервера предъявляются особые требования, поскольку от нее будет зависеть работа всей информационной системы. Поскольку сервер должен обладать высокой производительностью, применяются специализированные аппаратные средства. Обычно это многопроцессорные или кластерные архитектуры. Поэтому операционная система в первую очередь должна поддерживать используемую аппаратную платформу. Кроме этого, выбранные серверные приложения должны поддерживать данную версию операционной системы.

Для кластерных или многопроцессорных архитектур с числом процессоров более четырех обычно применяются Unix-подобные операционные системы. Серверы средних и малых предприятий, а также узлы ИС зачастую строятся на базе однопроцессорных архитектур или многопроцессорных с числом процессоров не более четырех. В этом случае спектр возможных операционных систем достаточно широк. На практике обычно выбирают Unix-подобную систему или Windows 2000 (Windows NT). Windows более проста в администрировании и не требует такой высокой квалификации как Unix. Операционную систему Unix применяют только тогда, когда необходимо обеспечить особую надежность и защищенность.

Выбор протокола сети.

Современная корпоративная информационная система должна обеспечивать как локальный, так и удаленный сетевой доступ пользователей через Internet. Поэтому наиболее простым решением является выбор широко известного сетевого протокола TCP/IP. Этот протокол поддерживается практически всеми современными СУБД и операционными системами.

2.3 Выбор СУБД

Основой информационной системы является система управление базой данных. Важно выбрать такую СУБД, которая не только в полной мере удовлетворяет текущим потребностям, но и имеет необходимый «запас прочности» для дальнейшего расширения и интеграции.

2.3.1 Обзор наиболее известных СУБД

На сегодняшний день известно более двух десятков серверных СУБД, однако наиболее популярными следует признать Oracle, Microsoft SQL Server, Informix, Sybase, DB2, Progress [19].

ORACLE. Была первой коммерческой реляционной СУБД, которая поддерживала язык SQL. Ее первая версия появилась в 1979 году. Фактически все это время Oracle является бессменным лидером на рынке производителей коммерческих СУБД и второй (после Microsoft) по величине компанией, производящей программное обеспечение[19].

Oracle была первой компанией, создавшей СУБД, которая использовала предоставляемые некоторыми серверными платформами средства параллельных вычислений. Oracle Parallel Server дает возможность нескольким процессорам обращаться к одной базе данных, что позволяет обеспечить высокую скорость обработки транзакций. Последние версии дают возможность осуществить декомпозицию операций с большими объемами данных с целью параллельного выполнения их на нескольких процессорах.

Существуют версии этой СУБД для различных серверных платформ: UNIX, VMS, Windows и др. СУБД Oracle поддерживает объектные расширения и соответствующие типы данных, такие как вложенные таблицы, массивы, объекты и др. Имеется возможность создания таблиц, содержащих агрегированные данные и возможность частичного их обновления при изменении данных, на основании которых они вычислены. Поддерживаются Java, XML, C/C++, PL/SQL.

СУБД дает возможность использовать дополнительные средства обеспечения безопасности, например шифрование данных, поддержку SSL, роли уровня базы данных и уровня предприятия.

Oracle 8i существует в трех редакциях: Oracle 8i, Oracle 8i Enterprise Edition, Oracle 8i Personal Edition. Для создания многомерных хранилищ данных существует и отдельный продукт – Oracle Express OLAP.

Помимо различных версий сервера баз данных среди продуктов Oracle имеется также Designer/2000 – ориентированное на эту СУБД CASE-средство для анализа бизнес-процессов и проектирования данных, а также средства разработки клиентских приложений. Одно из них – Developer/2000. Приложения, созданные с помощью Developer/2000, могут выполняться на различных платформах. Язык PL/SQL, используемый в этом средстве разработки, является интерпретируемым и представляет собой тот же самый язык, что используется в Oracle для написания серверного кода. Это позволяет отлаживать серверный код с помощью Developer/2000.

Производя собственные средства разработки, Oracle предоставляет своим пользователям возможность создавать клиентские приложения с помощью других средств. Помимо стандартного клиентского API (Oracle Call Interface), клиентская часть Oracle содержит объектную модель (Oracle Objects for OLE), позволяющую использовать клиентскую часть Oracle в качестве набора COM-объектов для доступа к данным. Кроме этого клиентская часть Oracle может содержать ODBC-драйвер для доступа к данным этой СУБД.

Отметим, что и многие другие компании производят ODBC-драйверы и OLE DB-провайдеры для доступа к Oracle (в частности, Microsoft). Компании, производящие средства разработки, использующие собственные библиотеки доступа к данным (такие как Inprise или Gupta/Centura), также включают библиотеки доступа к Oracle в состав наиболее дорогих версий своих продуктов.

Из готовых информационных систем на базе Oracle следует особо отметить одну из самых крупных систем управления предприятием SAP/R3. На Западе также нередко используются готовые решения от самой Oracle Corporation, объединенные под общим названием Oracle Applications, такие как Oracle Financials, Oracle Human Resources, Oracle Market Management, Oracle Project Systems и др.

MICROSOFT SQL SERVER. Первая версия Microsoft SQL Server [19], совместно разработанная в 1988 году компаниями Microsoft и Sybase, предназначалась для платформы OS/2. Последующие версии этого сервера баз данных предназначались для платформы Windows NT и со временем были тесно интегрированы с этой операционной системой. Позднее был осуществлен переход от Windows NT к Windows 2000. Для других платформ версии этого сервера не выпускались и не выпускаются.

СУБД отличается удобством пользовательского интерфейса утилит администрирования в сочетании с достаточно высокой производительностью и относительно невысокой стоимостью эксплуатации. Наибольший рост популярности этой СУБД пришелся на конец 90-х годов, когда были выпущены Microsoft SQL Server 6.0 (1995 год), обладавший централизованными функциями администрирования и встроенными возможностями репликации данных, Microsoft SQL Server 6.5 (1996 год) и Microsoft SQL Server 6.5 Enterprise Edition, поддерживающий параллельные вычисления в многопроцессорных системах.

СУБД обладает простыми утилитами администрирования. Сервисы преобразования данных (Data Transformation Services) облегчают перенос данных в SQL Server из других типов СУБД. Поддерживаются распределенные запросы и транзакции, а также расширенные функции для создания Web-приложений.

Вместо Microsoft SQL Server в качестве встроенной СУБД для настольных приложений и приложений для небольших рабочих групп можно использовать Microsoft Data Engine (MSDE). Это настольный сервер баз

данных совместим с Microsoft SQL Server и предназначен для использования в настольных системах или в сетевых приложениях с небольшим (до 2 Гбайт) объемом данных и малым количеством пользователей.

Клиентские приложения для Microsoft SQL Server и MSDE можно создавать как с помощью средств разработки Microsoft (Visual Basic, Visual C++, Access, Visual FoxPro), так и с помощью средств разработки других производителей. Для этой цели имеются ODBC-драйвер и OLE DB-провайдер, а также содержащий их набор библиотек Microsoft Data Access Components (MDAC), позволяющий использовать в средствах разработки объекты ActiveX Data Objects (ADO). MDAC является составной частью Windows 2000, а для пользователей других Windows-платформ доступен отдельно на Web-сайте Microsoft.

В отличие от Oracle, Microsoft не производит средств разработки, использующих тот же самый язык программирования, что и язык для создания кода триггеров и хранимых процедур, однако производит средства отладки серверного кода (например, SQL Server Debugger входит в состав Visual Basic и Visual C++).

SYBASE. СУБД компании Sybase, называемая Adaptive Server Enterprise существует для Windows NT и некоторых версий UNIX (включая Linux) [19]. СУБД предназначен для обслуживания крупных предприятий. Сервер поддерживает упреждающее асинхронное чтение, что повышает скорость выполнения сложных запросов. Возможна распределенная обработка запросов как для многопроцессорных, так и для кластерных систем. Расширенные хранимые процедуры позволяют осуществить легкий доступ к не-SQL функциям (Java, 3GL-системы, функции операционной системы и т.д.). СУБД обеспечивает интеграцию с популярными системами безопасности, такими как Kerberos.

Еще одна линия серверных продуктов Sybase ведет свое начало от сервера баз данных Watcom SQL Anywhere, отличавшегося компактностью и простотой администрирования. Этот сервер предназначен для обслуживания небольших рабочих групп, для применения в портативных компьютерах в качестве персонального сервера с периодической репликацией, а также в мобильных устройствах. Существуют версии этого сервера для Windows CE и версия UltraLite для разнообразных мобильных устройств.

Для управления распределенными транзакциями Sybase выпускает монитор транзакций Jaguar CTS. Для создания многомерных хранилищ данных у Sybase существует еще один серверный продукт – Adaptive Server IQ, позволяющий создавать хранилища на основе данных не только из СУБД производства Sybase, но и из СУБД других производителей. Отметим также, что существует ряд продуктов под общим названием Sybase Industry Warehouse Studio, ориентированных на обслуживание конкретных предметных областей: торговли (Retail Warehouse Studio), здравоохранения (Healthcare Warehouse Studio), страхования (Life Insurance Warehouse Studio) и др.

Помимо серверных продуктов Sybase производит средства проектирования данных и генерации кода приложений. Для создания клиентских приложений созданы средства разработки PowerBuilder и PowerJ, а также продукт PowerSite, предназначенный для создания Web-приложений. CASE-средство DataArchitect поддерживает широкий спектр СУБД различных производителей, а генератор приложений AppModeler способен генерировать код не только для PowerBuilder и Optima++, но и для Delphi, Visual Basic, Web-приложений с использованием ASP.

INFORMIX. Ведущим продуктом фирмы Informix является Informix Dynamic Server. Данный продукт поддерживает платформы UNIX и Microsoft Windows NT. Он обеспечивает эффективную работу как на одно-, так и на многопроцессорных системах, а также в кластерах. Сервер построен по архитектуре Dynamic Scalable Architecture (DSA), обеспечивающей мощные средства для параллельной обработки данных [19].

СУБД Informix управляет дисковым пространством как с помощью средств операционной системы (UNIX или Microsoft Windows NT), так и собственных функций, позволяющих обойти ограничения операционной системы и добиться более высокой производительности. Поддерживается фрагментация таблиц и индексов на нескольких дисках и управление разделением памяти для одновременного доступа к данным несколькими приложениями. Используются технологии распараллеливания запросов и зеркалирования данных.

Сервер поддерживает двухфазное завершение транзакций, гетерогенные транзакции (в этом случае в транзакциях может принимать участие и не-Informix сервер, доступный через Informix Enterprise Gateway).

Расширения функциональности сервера реализуются на базе DataBlade. Это коллекции объектов баз данных и подпрограмм на языке C, подключаемых к базе данных. Для разработки DataBlades необходимо использовать DataBlade Developer's Kit. Фирма Informix и целый ряд независимых производителей выпускают модули DataBlade, такие, например, как Excalibur Text DataBlade Module, Informix Geodetic DataBlade Module, Informix TimeSeries DataBlade Module, Excalibur Image DataBlade Module, Informix Web DataBlade Module и ряд других.

Входящие в состав Informix Dynamic Server клиентские утилиты предназначены для подключения к серверу и обработки информации (DB-Access) и для выполнения функций администрирования (DB/Cockpit).

Клиентские приложения могут создаваться с использованием языков Informix ESQL (средство для разработки на языке C, позволяющее включать в приложения запросы к данным на языке SQL), а также C, C++, Java, Visual Basic и Delphi. Помимо этого существуют собственные средства разработки – Informix-4GL и Informix Client Software Developer's Kit.

Фирма Informix выпускает Informix ODBC Driver, OLE DB Provider для Informix Dynamic Server и Informix JDBC Driver.

В состав продукта входят собственно сервер, а также Informix Connect, DataBlade Developer's Kit и Informix Server Administrator. Для генерации отчетов предлагается визуальное средство Informix-ViewPoint.

Говоря о сервере фирмы Informix, следует упомянуть и поддержку OLAP. Продукт под названием Informix MetaCube поставляется как часть Informix Decision Frontier – комплексного решения для создания хранилищ данных. Среди других продуктов фирмы Informix следует отметить i.Reach, i.Sell и Informix Internet Foundation.2000. Informix Internet Foundation.2000 представляет собой специально разработанный для Internet вариант Informix Dynamic Server. i.Reach это корпоративный репозиторий для хранения данных различного типа, интеллектуального управления информацией и извлечения данных. Основное назначение данного продукта – поддержка включения в содержимое корпоративных сайтов электронных документов и их последующее обслуживание. Продукт i.Sell это комплексное решение для электронной коммерции на базе Informix Dynamic Server.

DB2. Семейство серверных СУБД фирмы IBM, известное под названием DB2 Universal Database, представляет собой стратегию IBM по объединению продуктов DB2 для различных платформ в единую линию. Впервые появившееся в 1996 году семейство DB2 Universal Database объединяло в себе функциональные возможности таких продуктов фирмы, как DB2 Common Server, DB2 Parallel Edition (DB2 PE), Net.Data, Data Propagator и технологии DataHub. Семейство предназначалось для платформ UNIX, OS/2 и Microsoft Windows NT [19].

Отметим, что при переносе DB2 на не-IBM-платформы фирма старается максимально использовать уникальные функциональные возможности конкретной платформы. Например, в DB2 for Windows 2000 для обеспечения безопасности используется Windows NT LAN Manager, полностью поддерживаются Windows Performance Monitor, Systems Management Server, интеграция с Active Directory для каталогизации баз данных, а также такие интерфейсы доступа к данным, как ODBC, ADO и OLE DB. Помимо этого, DB2 for Windows 2000 поддерживает Microsoft Transaction Services (MTS) в качестве координатора при создании приложений, использующих распределенные транзакции. Для разработчиков, использующих Microsoft Visual Studio, становятся доступными дополнительные модули, например Stored ProcedureBuilder, включаемый непосредственно в среду Visual Studio. IBM также предлагает собственные средства разработки, например IBM VisualAge for Java. СУБД поддерживает создание хранимых процедур на языке Java (Java Stored Procedure Builder).

IBM предлагает бесплатное средство для миграции данных из Microsoft Access в DB2, а также средства для миграции данных из Oracle, Microsoft SQL, Sybase и Informix.

К основным характеристикам СУБД можно отнести поддержку реляционных и комплексных данных через объектные расширения, возможность работы на мультипроцессорных платформах, поддержку

кластеров, 64-битную архитектуру памяти, распараллеливание запросов, возможность создания Web-приложений (поддерживаются такие технологии, как Java, JDBC, SQLJ, XML) и наличие средств для гетерогенного администрирования и обработки данных.

Семейство DB2 функционирует на системах AS/400 и RISC System/6000, мэйнфреймах IBM, машинах от Hewlett-Packard и Sun Microsystems и под управлением таких операционных систем, как Windows NT, Windows 95/98, OS/2, AIX, HP-UX, SCO UnixWare, Linux, NUMA-Q и Sun Solaris. Поддерживаются портативные устройства под управлением Windows CE и Palm OS.

Кроме этого поставляется средство для онлайн-аналитической обработки данных и реализации хранилищ данных – DB2 OLAP Server, интегрирующее ядро Hyperion Essbase с семейством DB2 Universal Database. Продукт работает с Hyperion Integration Server (Hyperion), Hyperion Wired for OLAP (Hyperion), Brio.Insight (Brio Technology), BUSINESSOBJECTS (Business Objects), PowerPlay (Cognos), Lotus 1-2-3 (Lotus), Excel, Internet Explorer, Visual Basic (Microsoft) и Crystal Info (Seagate). Продукт DB2 Connect является средством для управления соединениями различных клиентов с DB2 на AS/400.

Для создания и тестирования приложений в архитектуре клиент-сервер имеется средство DB2 Universal Developer's Edition. Продукт DB2 DataJoiner позволяет получать доступ к данным, расположенным на различных платформах как к единому образу данных.

Кроме этого компанией IBM поставляется DB2 продукт Query Patroller. Это набор средств для создания запросов и управления ресурсами для систем принятия решений. DB2 Query Patroller получает ODBC-запросы от клиента, анализирует их и динамически распределяет запросы по различным узлам DB2 UDB Enterprise – Extended Edition. Приложение DB2 Net.Data позволяет Web-разработчикам создавать динамические Internet-приложения, используя Web Macros.

Предлагаются средства для внедрения масштабируемых мобильных решений и для управления удаленными пользователями. СУБД поддерживает функции репликации, централизованное администрирование и средства управления через Internet.

PROGRESS. Корпорация Progress Software Corp. основана в 1981 году. Progress Software Corp. является производителем средств разработки приложений (язык 4GL) и SQL сервера базы данных СУБД Progress. Компания производит средства для построения информационных систем масштаба промышленного предприятия, ориентированных на интенсивную обработку транзакций в распределенных вычислительных средах в реальном масштабе времени [19].

Progress Software Corp. занимает 21 процентов рынка встроенных в приложения СУБД и является лидером на этом рынке.

Предлагаемые Progress инструменты позволяют решать все задачи по разработке высокопроизводительных приложений от проектирования ядра базы данных до создания графических инструментов и бизнес-логики информационной системы, обеспечивая высокую межоперабельность и открытость.

Progress поддерживает как обычный процедурный, событийно-ориентированный, так и объектно-ориентированный стили программирования. Progress позволяет работать с данными, хранящимися в базах данных других форматов: Oracle, Sybase, Rdb, C-ISAM, OS/400 и др.

Приложения, написанные на Progress, полностью переносимы более чем на 160 аппаратных платформ, на которых функционирует Progress, включая MS-Windows, OS/2, Windows NT, Unix (UnixWare, SCO UNIX, Interactive, Solaris, AIX, HP-UX, IRIX), OpenVMS, IBM AS/400.

Гибкая архитектура сервера позволяет конфигурировать сервер базы данных для оптимальной поддержки требуемого размера базы данных, от маленькой, однопользовательской базы данных до системы, управляющей тысячами пользователей 4GL и SQL. Кроме этого, независимые продавцы ПО или конечные пользователи имеют возможность внедрять только те составляющие сервера базы данных, которые необходимы. Это минимизирует ресурсы и персонал, необходимый для работы с приложением, основанным на реляционной СУБД Progress. Динамическое ядро базы данных позволяет настраивать и оптимизировать выбранные параметры сервера базы данных во время работы сервера и под нагрузкой реальными пользователями. Это обеспечивает самую высокую степень точности, поскольку можно видеть изменения непосредственно под нагрузкой. В системах, работающих круглосуточно и без перерывов, параметры могут изменяться для того, чтобы использовать преимущества неполной загрузки – без выключения базы данных.

Для соответствия промышленным стандартам, реляционная СУБД Progress может использоваться с широким набором клиентов (Progress 4GL, Java, C, C++, Visual Basic, PowerBuilder и т.д.), включая хост/терминальные продукты третьих фирм, отвечающие требованиям ODBC и JDBC, такие как средства работы с отчетами, анализ данных, CASE, управление системой и множество других.

Для построения Internet-приложений разработаны продукты: Progress Apptivity 3, WebSpeed 3, ProtoSpeed. WebSpeed – это собственная разработка Progress Software предназначенная как для существующих web-приложений, так и для построения транзакционных Internet-приложений с использованием HTML. WebSpeed WorkShop – это среда, основанная на технологии мастеров, которая включает в себя язык скриптов серверной части, под названием SpeedScript, основанный на progress 4GL (также поддерживаются JavaScript, VBScript, CGI и апплеты Java). Apptivity, выпущенный Progress летом 1997 г, предназначен для построения серверной части приложений, которые могут поддерживать сложные бизнес-процессы посредством визуальной среды

разработки, генерирующей Java-код. Apptivity обеспечивает поддержку межбраузерных связей, а также включает в себя интеграцию с CORBA. В дополнение, данный инструментарий может быть использован для построения основанных на Java клиент-серверных приложений, и включает встроенный отладчик и Java-компилятор. ProtoSpeed – это отдельное средство отладки на Java, созданное для трассировки распределенных Internet-приложений в гетерогенных средах. Инструмент поддерживает Internet-протоколы: HTTP, FTP, SMTP, POP3 и IMAP4.

2.3.2 Критерии выбора СУБД

Наиболее простой подход при выборе СУБД основан на оценке того, в какой мере существующие системы удовлетворяют основным требованиям проекта информационной системы. Более сложным и дорогостоящим вариантом является создание испытательного проекта на основе нескольких СУБД и последующий выбор наиболее подходящей. Но и в этом случае следует ограничивать круг возможных систем, опираясь на некие критерии отбора. Перечень требований к СУБД может меняться в зависимости от поставленных целей. Тем не менее, можно выделить несколько групп критериев: стоимость, производительность, надежность, особенности разработки приложений, контроль работы системы, требования к рабочей среде, особенности архитектуры и функциональные возможности, моделирование данных, смешанные критерии.

Моделирование данных включает в себя следующие критерии:

Используемая модель данных. Существуют несколько моделей данных: иерархическая, сетевая, реляционная, объектно-реляционная, объектная. Выбор модели осуществляется исходя из основных потребностей информационной системы на начальном этапе проектирования.

Поддерживаемые типы данных. Некоторые современные системы помимо общепринятых базовых типов данных (integer, character) поддерживают дополнительные типы данных, а также возможности расширения типов данных.

Реализация языка запросов. Практически все реляционные СУБД совместимы с той или иной спецификацией стандартного языка доступа к данным SQL (SQL-89, SQL-92). Кроме того, во многих системах реализованы некоторые расширения данного стандарта (например, в свое время таким расширением были триггеры).

Средства поиска. Некоторые современные системы имеют встроенные дополнительные средства контекстного поиска.

Триггеры, хранимые процедуры, ограничения и утверждения как конструкции активных сред. Традиционные базы данных являются пассивными, т.е. изменения и выборка данных осуществляются только под влиянием внешнего источника. Однако в последнее время важную роль приобретает концепция активных баз данных, когда в базу данных закладываются возможности управления данными. В создании активных сред

помогают несколько основных конструкций базы данных: ограничения, утверждения, хранимые процедуры и триггеры. Триггер это программа базы данных автоматически выполняемая при некоторых событиях (создание записи, изменение записи, удаление записи и др.). Хранимая процедура это программа, которая относится (храниться) к базе данных, а не к внешней программной системе. Ограничения и утверждения это относительно простые конструкции, имеющие вид от спецификации связей первичного и внешнего ключей, используемых в ограничениях целостности по ссылке, до SQL-подобных ограничений, используемых для проверки принадлежности значений заданному диапазону или списку значений.

Особенности архитектуры и функциональные возможности.

Сетевые возможности. Практически все современные СУБД позволяют использовать широкий диапазон сетевых протоколов и служб для работы и администрирования.

Масштабируемость. Как правило, система должна предусматривать рост информационной системы. Различают увеличение объема хранимых данных, числа пользователей и объема обрабатываемой информации.

Распределенность. Обычно информационная система создается с целью объединения информации предприятия, но наиболее простая реализация в виде централизации данных на одном сервере не всегда применима из соображений повышенных требований к отказоустойчивости или сильного территориального распределения подразделений. Механизмы создания распределенных баз данных в различных системах реализованы по-разному.

Максимальный размер базы данных. При выборе системы следует учитывать, сможет ли база данных вместить планируемые объемы информации.

Контроль работы системы.

Средства мониторинга. Современная система должна обладать средствами мониторинга основных параметров базы данных для администрирования и принятия решений по оптимизации комплекса.

Автоматическая настройка. Некоторые СУБД включают возможность автоматически подстраиваться для достижения максимальной эффективности.

Требования к рабочей среде .

Минимальные требования к оборудованию. Поскольку большинство современных информационных систем построены на основе архитектуры клиент-сервер, различают требования к оборудованию клиентской и серверной части.

Поддерживаемые аппаратные платформы и операционные системы. Современные СУБД могут работать на различных аппаратных платформах под управлением нескольких операционных систем. Это дает возможность выбирать операционные системы и аппаратную конфигурацию для

достижения максимальной надежности и производительности в условиях ограниченного финансирования.

Производительность.

Рейтинг ТРС. Существуют несколько тестовых рейтингов. Одним из самых популярных и объективных является ТРС. Он представляет собой комплексный анализ СУБД и аппаратуры. Существует два вида рейтинга ТРС оценивающие производительность и отношение производительность к цене комплекса. Кроме этого рейтинг разбит на категории с точки зрения архитектуры серверной части (кластерные системы, не кластерные системы), производителя серверной части (Compaq, IBM, HP, Dell и т.п.), а также области применения (OLTP, электронная коммерция, поддержка принятия решений, бизнес отчетность и т.п.)

Возможности встроенной оптимизации запросов. Современные СУБД должна содержать оптимизатор запросов, который в результате семантический преобразований с учетом статистических данных вырабатывает оптимальный план выполнения запроса.

Возможности параллельной архитектуры. Для повышения производительности во многих современных СУБД применяется параллельная обработка. Например, возможна поддержка технологий параллельной обработки последовательности запросов на нескольких процессорах или использование нескольких компьютеров объединенных в параллельный сервер базы данных.

Возможности перераспределения обработки между сервером и клиентом. Некоторые СУБД позволяют частично или полностью переносить обработку данных с клиента на сервер. В результате становится возможным по выбору строить конфигурации с «толстым» или с «тонким» клиентом. В некоторых СУБД возможно построение приложений, которые в зависимости от ситуации сами решают, какая часть обработки должна быть перенесена на сервер или другого клиента.

Надежность.

Модель транзакций. Существуют многочисленные модели транзакций – от простейших, например плоских, до более изощренных, таких как вложенные или многозвенные. Большое значение имеет правильный выбор одной из сложных моделей транзакций с учетом специфики деятельности конкретной организации. Например, вложенные транзакции предоставляют исключительно гибкие возможности управления субтранзакциями, но они сложны в реализации и не во всех случаях эти возможности необходимы. Для построения распределенных архитектур СУБД должна поддерживать распределенные транзакции.

Защита информации от несанкционированного доступа. Любая современная информационная система должна иметь возможность разграничения доступа к информации. Существуют различные уровни защиты информации. В большинстве случаев не достаточно простой идентификации пользователей при входе в систему, поэтому СУБД должна

поддерживать средства, с помощью которых для каждого пользователя определяются полномочия доступа к данным и модификации данных.

Резервное копирование. В некоторых критических ситуациях восстановление потерянных данных, возможно только с использованием резервной копии. СУБД должна позволять варьировать планы резервного копирования в зависимости от ограничений на время восстановления, стоимость хранения и возможную потерю части информации (например, за последний день). Некоторые СУБД позволяют выполнять резервное копирование без остановки системы. Кроме этого существуют технологии снижающие стоимость хранения. В качестве примера можно привести инкрементальное резервное копирование, при котором сохраняются только те блоки базы данных, которые изменились с момента последнего резервного копирования.

Восстановление после сбоев. Программные или аппаратные сбои могут привести к нарушению целостности информации. В этом случае дальнейшее функционирование системы зависит от реализации механизмов восстановления. Например, ведение журнала незавершенных транзакций позволяет автоматически восстановить целостность базы данных после сбоя возникшего при выключении электричества.

Особенности разработки приложений.

Спектр поддерживаемых языков программирования. СУБД может поддерживать несколько языков программирования. Наличие широкого набора языков программирования повышает доступности системы для разработчиков и позволяет создавать оптимальные по производительности и функциональности приложения.

Поколение поддерживаемых языков программирования. Функциональность приложения, а также его производительность и время разработки, существенно зависят от поколения используемого языка программирования. Применение проблемно-ориентированного языка четвертого поколения (4GL), которые оперирует конкретными понятиями предметной области, существенно повышает функциональность приложения. Один оператор 4GL может соответствовать десяткам операторов 3GL. 4GL содержит средства визуальной разработки, что позволяет вести разработку в стиле RAD (Rapid Application Development – быстрая разработка приложений). Если же прикладная область ориентирована на Internet, то требуется применение языков 3GL, основанных на языке Java.

Средства разработки Web-приложений. При проектировании Internet-ориентированного приложения, желательно, чтобы СУБД содержала специальные средства разработки Web-приложений.

Собственные средства разработки приложений. Многие поставщики создали специальные средства разработки приложений для своих систем, которые позволяют наилучшим образом использовать возможности СУБД. Это могут быть средства создания отчетов, средства разработки Web-приложений, язык 4GL.

Многоязыковая поддержка. Поддержка большого количества национальных языков расширяет область применения СУБД и построенных на ее основе приложений.

Смешанные критерии

Поддержка национального языка. Поддержка национального языка может существенного сказываться на стоимости разработки и сопровождения приложений.

Качество и полнота документации. К сожалению, не все системы имеют полную и подробную документацию.

Стоимость системы. При оценке стоимости системы должны учитываться **и цена базовых продуктов, и стоимость их сопровождения.** Кроме этого, следует учитывать стоимость разработки приложения для выбираемой СУБД. На стоимость системы сильно влияет модель формирования стоимости базовых продуктов. Например, как правило, цена конкретного продукта зависит от числа пользователей. Поэтому, в некоторых случаях следует выбирать более дорогую СУБД, которая позволит снизить финансовые вложения при будущем расширении информационной системы.

2.4 Реализация технологии клиент-сервер на базе программных средств Microsoft

Что касается инструментальных средств разработки клиентского приложения, то тут тоже наблюдается большой выбор инструментальных средств, таких как Delphi, Microsoft Access, Visual Basic, C++, Power Builder. Microsoft Access выбран автором также из соображений доступности, так как в нашей стране Microsoft Office распространен повсеместно, а также, потому что Access предоставляет максимально удобную и трудосберегающую инструментальную среду программирования именно для целей построения учетных систем, в которых кроме экранных и печатных форм других возможностей программирования не требуется.

С точки зрения автора, инструментальные средства Microsoft являются наиболее доступной и дешевой платформой для построения информационных систем, кроме того, они обладают мощной и функциональной инструментальной средой разработки, поэтому были выбраны для детального изучения на предмет применения этой функциональности в экономике в задачах автоматизации управления учебного процесса.

Вопрос выбора инструментальных средств является очень дискуссионным, в котором долгое время не будет единого мнения. Как пишет участник форума [20], "Это психологический и социальный и финансовый вопрос, многие связывались с Oracle не только из-за его преимуществ, но и из-за зарплаты, престижа, и таких, надо сказать, немало". Другой участник пишет, что "некоторые пишут на C++ из соображений престижа, точно также пользуются Oracle, а не MS SQL". Таким образом, принимая во внимание цели работы, следует сконцентрироваться не на дискуссии по поводу выбора

инструментальных средств, а на поиске возможностей применения конкретного выбранного набора в реальной ситуации, что должно принести содержательные теоретические и практические результаты.

2.4.1 Выбор сервера баз данных.

В сетевой архитектуре клиент-сервер сервер баз данных представляет собой ядро информационной системы, так как выполняет основные функции по хранению и обработке информации системы, в том числе формированию требуемых наборов данных при оперативном анализе и поддержке принятия решений. Так как наиболее важная роль в архитектуре учетной системы уделяется серверу, начнем рассмотрение задачи реализации технологии клиент-сервер на основе программных средств Microsoft с выбора программного средства, выполняющего функции сервера баз данных.

Сетевые программные продукты, обычно, имеют достаточно высокую стоимость, поэтому важной экономической задачей работы выступает задача минимизации затрат на программное обеспечение для разработки информационной системы автоматизации управления учебным процессом. Цены на сервера баз данных колеблются в пределах нескольких тысяч долларов, в зависимости от фирмы-производителя, версии, конфигурации, поэтому цена сервера оказывает существенное влияние на себестоимость автоматизации.

Построение сетевых корпоративных систем возможно на основе двух серверов **реляционных** баз данных Microsoft, в зависимости от того, какая сетевая технология будет реализовываться:

Microsoft Jet - для реализации сетевой архитектуры учетной системы по технологии **файл-сервер** [21];

Microsoft SQL Server - для реализации технологии **клиент- сервер**.

Процессор баз данных Microsoft Jet является составным программным компонентом Microsoft Access, который наиболее популярен как настольная СУБД [21],

Microsoft SQL Server (MS SQL) выпускается в шести версиях, среди которых наиболее часто используемые:

- настольная версия (Desktop Edition);
- стандартная версия (Standart Edition);
- промышленная версия (Enterprise Edition).

Версия SQL Server выбирается в зависимости от объемов информации баз данных сервера, количества пользователей, требуемого быстродействия и других факторов масштаба автоматизации. С описанием возможностей и функциональных ограничений каждой из шести версий MS SQL можно ознакомиться в [21], а также на www.microsoft.com.

Приведем краткую сравнительную таблицу технических характеристик Jet и SQL Server (Таблица 2.1), в соответствии с [22]

Таблица 2.1 Основные отличия СУБД Microsoft

Характеристика	Jet	SQL Server Desktop Edition	SQL Server (Standart and Enterprise versions)
Поддерживаемые операционные системы	9x/NT/2000	9x/NT/2000	NT/2000
Максимальный размер базы данных	2GB	2GB	Терабайты
Максимальное количество процессоров	1	2	4 для Standart Edition, 32 для Enterprise Edition
Наличие лицензии	Нет	Нет	Да

В масштабе вуза объемы информации базы данных, по мере ввода операций системы, достаточно быстро увеличиваются, а производительность системы, соответственно, снижается, поэтому архитектуру ИС нужно создавать на основе технологии клиент-сервер. Клиент-серверные технологии на сегодняшний день являются стратегическим направлением развития корпоративных баз данных, поэтому информационные системы следует разрабатывать на основе продуктов, которые эффективно реализуют сетевые функции, то есть MS SQL Server, в данном случае.

Кроме того, важным недостатком Access как СУБД, является отсутствие инструментальных средств, таких как триггеры и хранимые процедуры, необходимые для поддержки сложных моделей данных, наличие которых имеет принципиальное значение для решения многих задач автоматизации.

Таким образом, Microsoft Access наилучшим образом может использоваться, как относительно простая настольная СУБД на основе процессора баз данных Jet, применение Access в корпоративном масштабе имеет ряд существенных ограничений. Microsoft, в свою очередь, рекомендует переводить базы данных предприятий на платформу SQL Server, о чем свидетельствует большое количество документации по этому вопросу в MSDN [22].

Из приведенной таблицы (Таблица 2.1) видно, что для создания ИС, которая предполагают одновременную работу небольшого числа пользователей, можно использовать настольную версию сервера - Microsoft SQL Server Desktop Edition, более известную под аббревиатурой MSDE. MSDE является бесплатным продуктом, который входит в состав Microsoft Office, что является довольно малоизвестным фактом, в том числе потому, что опция установки MSDE отсутствует в перечне компонент MS Office,

предлагаемых для установки инсталлятором Office. MSDE, в отличие от других программ Office, устанавливается не с помощью стандартной программы установки MS Office, а запуском утилиты `setupsql.exe`, находящейся в каталоге `\SQL\X86\SETUP\` на первом инсталляционном компакт-диске Office

MSDE официально поддерживает одновременную работу до пяти пользователей, однако, в принципе, ограничения на количество подключений у него нет, но при количестве подключений более пяти, работа сервера существенно замедляется.

Перечислим основные ограничения MSDE [15]:

- пять подключений;
- объем базы данных до 2 гигабайт;
- поддержка не более 2 процессоров;
- отсутствия поддержки функций серверной операционной системы (распределение транзакций между серверами, кластеризация и т. д.).

MSDE также относительно неудобен тем, что он не содержит графических инструментов управления, таких как Enterprise Manager или Query Analyzer, что требует знания специальных команд для управления сервером. Тем не менее, все функции по управлению MSDE, в том числе графический интерфейс, можно реализовать в Microsoft Access.

Важным преимуществом с точки зрения реализации программного функционала базы данных выступает то, что возможности MSDE аналогичны промышленной версии СУБД MS SQL Server Enterprise Edition, что делает MSDE особенно привлекательным продуктом для малых, средних, а в определенных случаях, и крупных предприятий.

Таким образом, программные возможности промышленного сервера баз данных содержатся в бесплатном приложении к Microsoft Office 2000 - MSDE, что делает платформу Microsoft SQL Server для построения корпоративных информационных систем особенно привлекательной. О перспективности MS SQL Server на текущий момент свидетельствует также статистика [18], по которой он является наиболее популярной СУБД промышленного масштаба, воплощая, таким образом, политику фирмы Microsoft "SQL Server everywhere". Также показательным примером, подтверждающим перспективность платформы SQL Server, выступает факт подписания в 2000 году между фирмами 1С и Microsoft соглашения, на основании которого 1С смогла предложить своим клиентам новые комплексные решения "1С MS SQL -Предприятие 2000" [23].

Клиентская часть системы и технология подключения к серверу. С выходом Microsoft Office 2000 у широкого круга пользователей появилась возможность создавать полнофункциональные клиент-серверные системы, соответствующие концепции тонкого клиента, на основе программного обеспечения, включенного в Office:

- СУБД Microsoft SQL Server (MSDE) для реализации серверной части КИС;

– Проект Microsoft Access для построения клиентского приложения SQL Server.

Клиентская часть учетной системы наиболее эффективно реализуется в формате проекта Microsoft Access (ADP - файл). Проект Microsoft Access представляет собой новый тип файла Microsoft Access, впервые появившийся в Office 2000, который устраняет ряд неудобств и функциональных ограничений при работе с SQL Server через интерфейс подключения ODBC. Проект Microsoft Access выступает как новый технологический шаг Microsoft в направлении повсеместного распространения сетевых технологий баз данных. Универсальным ядром учетной системы, таким образом, в данной архитектуре выступает SQL Server. В свою очередь, Access интересен не как СУБД, а как инструментальная среда построения пользовательского интерфейса (графической оболочки) для SQL Server.

На смену ODBC теперь пришла новая технология OLE DB, на основе которой создана новая объектная модель доступа к данным ADO (ActiveX Data Objects), во многом схожая с моделью DAO (Data Access Objects), применяемая в базах данных Microsoft Access. Встроенная в Microsoft Access поддержка OLE DB и ADO сформировала качественно новые возможности применения этого продукта для создания корпоративных учетных систем масштаба предприятия [18], которые по сегодняшний день не претерпели принципиальных изменений.

Проект Microsoft Access позволяет наилучшим образом реализовать концепцию тонкого клиента, то есть в данной сетевой архитектуре у Access выделяется только функция графического представления информации, тогда как все остальные функции по хранению и обработке данных выполняет MS SQL.

С точки зрения издержек на инструментальные средства программной реализации единственным платным продуктом, необходимым для построения системы, является Microsoft Access. Помимо этого, возможности Microsoft Access настолько упрощают процесс разработки и сопровождения полнофункциональной клиент-серверной ИС, что дешевле использовать Microsoft Access, чем другие среды программирования, например на C++, Delphi, Visual Basic.

Недостаток применения Access в качестве клиента заключается в том, что работоспособность клиента зависит от того, установлен ли Access на каждом компьютере, а также от версии Access, что, естественно, недопустимо для крупных ИС, с которыми работают сотни пользователей. Однако вполне приемлемо для менее масштабных систем автоматизации, кроме того, средствами VBA удалось достичь полной совместимости клиентского приложения под Microsoft Access 2000/XP/2003, если в распоряжении разработчика есть версия Microsoft Office Developer Edition (MOD). В комплект MOD, кроме самого Office, включены Office Developer Tools (ODE Tools - инструменты разработчика), которые позволяют создать инсталлятор для проекта Access, включив все необходимое программное

обеспечение, включая библиотеки Access, для установки продукта на любом компьютере с операционной системой Windows. Также учитывая высокую популярность и повсеместное распространение Microsoft Office в нашей стране, требование наличия Microsoft Access на рабочих станциях пользователей, как правило, выполняется автоматически и не требует дополнительных денежных расходов.

2.4.2 Microsoft Access, как инструментальная среда разработки клиентского приложения.

По своему названию Access переводится как "доступ", что неявно отражает наиболее сильную сторону этого продукта. Access представляет собой наилучшую инструментальную среду для построения графического пользовательского интерфейса **доступа к данным** с очень богатыми функциональными возможностями. Среда Access содержит необходимые для этого инструменты:

- экранные формы
- печатные формы (отчеты)
- модули и редактор VBA для программирования

пользовательского приложения

К проекту Access могут подключаться дополнительные библиотеки, использоваться ActiveX элементы управления и API функции, расширяющие возможности приложения за счет использования встроенных в операционную систему Windows функций. На текущий момент можно говорить о том, что в арсенале Access содержатся все функциональные возможности для создания мощного и удобного **графического интерфейса** доступа к данным, позволяющие решить практически любую задачу создания приложений пользователя для учетных систем.

С появлением технологии доступа к данным OLE DB, все неудобства при работе с ODBC ушли в прошлое, и в настоящий момент использование Access как СУБД в корпоративном масштабе нерационально и неперспективно, так как функции СУБД можно без труда перевести на существенно более мощную и функциональную СУБД Microsoft SQL Server. Таким образом, благодаря технологии OLE DB совместное применение Microsoft Access и Microsoft SQL Server стало максимально простым, удобным и эффективным, поэтому далее эта пара продуктов будет использоваться как универсальная среда разработки системы, которая, как будет показано ниже, позволяет эффективно **реализовать учетные и аналитические функции** программных систем.

Подводя итог вышесказанному, следует отметить, что возможности программного обеспечения Microsoft к 2000 году настолько возросли, что на основе продуктов популярного пакета Microsoft Office 2000 стало возможным создавать сетевые информационные системы, с применением наиболее совершенных на сегодняшний день технологий баз данных. Наличие практически неограниченных возможностей вызывает

естественное стремление использовать эти возможности для решения ряда задач автоматизации учета и управления, которые до настоящего момента не были решены ввиду несовершенства программных средств.

Отсюда следует, что ближайшая задача, которую необходимо решить - это найти **эффективное применение** функциональных возможностей программных средств Microsoft для решения класса задач компьютерной автоматизации на уровне вуза. Для этого потребуется провести анализ функциональных возможностей инструментальных средств и связать их с информационными функциями ведения управления и учета, для которых необходимо реализовать программную поддержку. Результатом должна стать **общая программная платформа для разработки приложений в образовательной среде.**

2.5 Реализация базы данных для проектируемого автоматизированного рабочего места

Таблицы базы данных разработаны с помощью среды разработки Microsoft Access на сервере баз данных MS SQL SERVER 2000. Применение реляционной СУБД помогает упростить структуру данных и таким образом облегчить выполнение работы.

База данных проектируемого автоматизированного рабочего места по учету успеваемости студентов содержит следующие таблицы:

- Рубежный контроль 1 (tb_RK1);
- Рубежный контроль 2 (tb_RK2);
- Рубежный контроль 3 (tb_RK3);
- Рубежный контроль 4 (tb_RK4);
- Экзамен 1(Ekzamen_1);
- Экзамен 2(Ekzamen_2);
- Таблица доступа пользователей(tb_usersINEU);
- Пользователи (BD);
- Студенты(tb_Abit);
- Справочник дисциплин (SprDis);
- Справочник по группам(SprGrp).

Таблица «Рубежный контроль 1» содержит поля представленные в таблице 2.2 и хранит в себе данные о рубежном рейтинговом контроле №1 по каждому студенту..

Таблица 2.2 – Таблица «Рубежный контроль 1»

Имя поля	Тип данных	Описание
KodSt	Int	Счетчик
ИКТ	Nvarchar	ИКТ студента(связь с таблицей tb_Abit)
Продолжение таблицы 2.2		

ID_PPS	Int	Код преподавателя (связь с таблицей BD)
ID_Dis	Int	Код дисциплины (связь с таблицей spr_Dis)
Date_BD	Datetime	Дата ввода
Date_Proved	Datetime	Дата проведения РК
Ball	Real	Количество баллов(%)
Kol_Kr	Int	количество кредитов

Таблицы «Рубежный контроль 2», «Рубежный контроль 3», «Рубежный контроль 4» содержит аналогичные поля и хранят в себе данные о рубежном рейтинговом контроле №2,3,4.

Таблицы «Экзамен 1» и «Экзамен 2» содержат поля представленные в таблице 2.3. Хранят данные по экзаменам студентов.

Таблица 2.3 – Таблица «Экзамен»

Имя поля	Тип данных	Описание
KodSt	Int	Счетчик
IKT	Nvarchar	ИКТ студента(связь с таблицей Abit)
ID_PPS	Int	Код преподавателя (связь с таблицей BD)
ID_Dis	Int	Код дисциплины (связь с таблицей spr_Dis)
Date_BD	Datetime	Дата ввода
Date_Proved	Datetime	Дата проведения экзамена
Ball	Real	Количество баллов(%)
Kol_Kr	Int	количество кредитов
Nom_semestr	Int	Номер семестра

Таблица 2.4 «Таблица доступа пользователей» служит для разграничения прав и аутентификации пользователей для работы по вводу рейтинга или экзаменационных баллов. По полю KodSotr осуществлена связь с таблицей BD, где хранится подробная информация по каждому пользователю.

Таблица 2.4 «Таблица доступа пользователей»

Имя поля	Тип данных	Описание
KodSotr	Int	Код сотрудника в таблице BD
KodPodrazd	Nvarchar	Код подразделения
Login	Nvarchar	Логин преподавателя
Passw	Nvarchar	Пароль

Таблица «Пользователи» хранит информацию по каждому преподавателю. Состоит из следующих полей, отображенных в таблице 2.5.

Таблица 2.5

Имя поля	Тип данных	Описание
Kod	Int	Счетчик
UniCode	Int	Код преподавателя
Fio	Nvarchar	Фамилия
Imy	Nvarchar	Имя
Otch	Nvarchar	Отчество
Pol	Int	Пол
Passport	Nvarchar	Паспортные данные
Date_vidachi	Datetime	Дата выдачи
Kem_vidan	Nvarchar	Кем выдан
PlaceBorn	Nvarchar	Место рождения
GodRog	Datetime	Год рождения
Kod_Org	Int	Код организации
Dolgnost	Nvarchar	Должность
Uvolen	Datetime	Дата увольнения

Таблица 2.6 «Студенты» хранит личную информацию по каждому студенту.

Таблица 2.6 «Студенты»

Имя поля	Тип данных	Описание
ИКТ	Nvarchar	Внутривузовский индивидуальный код студента
TEST_ИКТ	Nvarchar	Индивидуальный код абитуриента ЕНТ,КТ
Продолжение таблицы 2.6		

DateScan	Datetime	Дата сканирования бланка
KodTexSecret	Nvarchar	Код технического секретаря приемной комиссии
Fio	Nvarchar	ФИО абитуриента
KodGr	Nvarchar	Код гражданства
KodNaz	Nvarchar	Код национальности
KodDocLich	Int	Вид документа, удостоверяющий личность
NumDocLich	nvarchar	Номер документа личности
Pol	Bit	Пол
KodLangEKZ	Int	Язык обучения (1-казахский, 2-русский)
OrgDocLich	Int	Где выдан документ личности
DatDocLich	Datetime	Дата выдачи документа личности
BirthDate	Datetime	Дата рождения
Altin	Int	Медаль Алтын белги
SeriaAttDipl	Nvarchar	серия аттестата, диплома
KodOblOk	Nvarchar	Область, где учился
KodRaionOk	Nvarchar	Район, где учился
KodSchoolOK	Nvarchar	Код учебного заведения, которое закончил
KodOk	Int	Тип учебного заведения (1-школа, 2-колледж и т.д.)
IIN	Nvarchar	ИИН

Таблица 2.7 «Справочник дисциплин» хранит в себе информацию по дисциплинам: код дисциплины, год поступления, код приказа на дисциплину и т.д.

Таблица 2.7 «Справочник дисциплин»

Имя поля	Тип данных	Описание
KodDis	Int	Счетчик, код
GodPost	Nvarchar	Год поступления
Kod_Prizn_Dis	Int	Код дисциплины
Kod_Cikl_Dis	Int	Код дисциплины
KodStSpec	Nvarchar	Код специальности
Продолжение таблицы 2.7		

ShifrDis	Nvarchar	Шифр дисциплины
NameRus	Nvarchar	Наименование дисциплины
ShifrDisKz	Nvarchar	Шифр дисциплины на казахском
NameKz	Nvarchar	Наименование дисциплины на казахском
ShifrDisEn	Nvarchar	Шифр дисциплины на английском
NameEng	Nvarchar	Наименование дисциплины на английском

И последняя используемая таблица 2.8 это справочник групп, который содержит информацию по сформированным группам.

Таблица 2.8 «Справочник групп»

Имя поля	Тип данных	Описание
G_code	Int	Счетчик, код
KodKaf	Int	Код кафедры
G_name	nvarchar	Наименование группы
KodPPS	Int	Код ППС
Kurs	Int	Курс
KodLang	Int	Язык обучения
KodFormOb	Int	Форма обучения
Y_Create	smallInt	Год поступления
KodStSpec	Int	Код специальности
KodUchPlan	Int	Код учебного плана
Priznak_Vyp	Bit	Признак выпуска
Yearvypusk	Nvarchar	Год выпуска

Между созданными таблицами были созданы связи. Наличие связей между таблицами задает целостность базы данных. Задание в свойствах отношений необходимо для автоматического отслеживания изменений, что позволяет хранить целостность структуры. Все произведенные изменения, автоматически применяются на все таблицы, которые входят в отношение. Это позволяет исключить возможность появления записей в дочерних таблицах, поле внешнего ключа которых ссылается на несуществующую запись в родительской таблице. На рисунке 2.1 представлены созданные

СВЯЗИ.

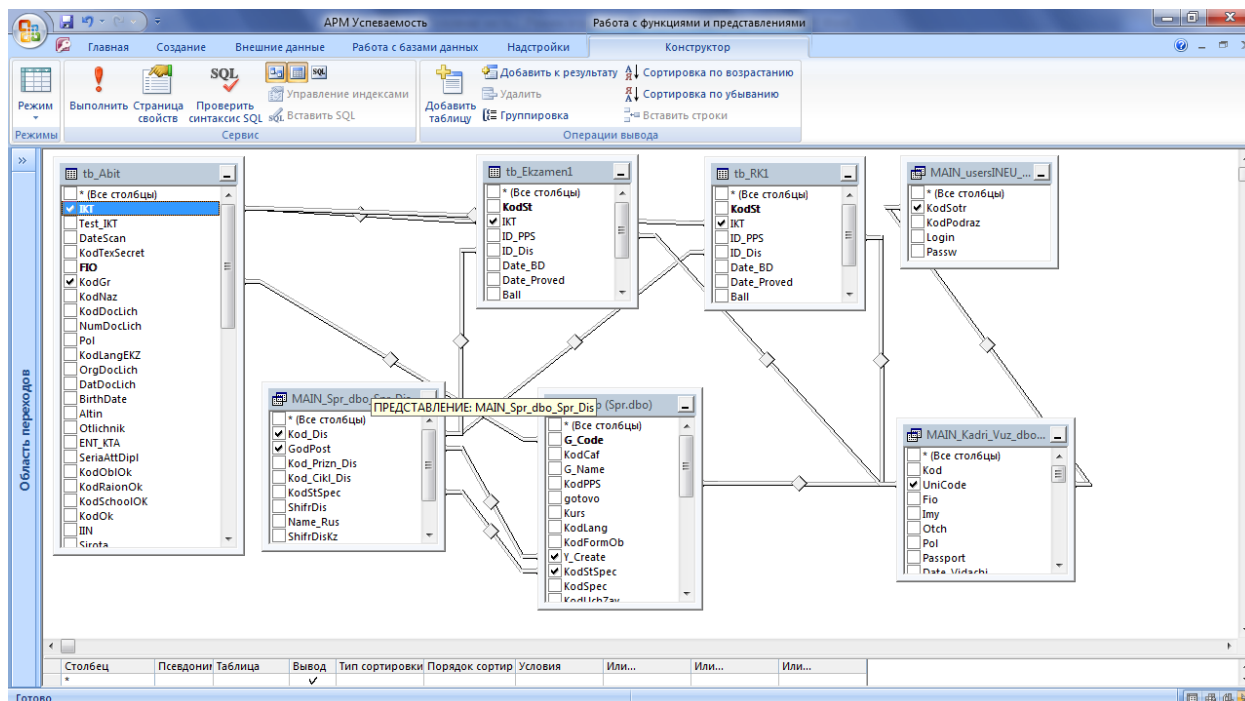


Рисунок 2.1 – Таблицы базы данных и созданные связи между ними

Выводы

Сформулированы и классифицированы основные критерии выбора СУБД для построения информационной системы.

На основании анализа в качестве основы для построения системы рекомендована СУБД MS SQL SERVER. Построена база данных для АРМ «Успеваемость» информационной системы «е-ВУЗ» ИнЕУ.

В качестве инструментальной среды разработки клиент-серверного приложения выбран Microsoft Access.

На основании анализа рекомендована операционная система Windows XP (Windows 2000) для клиентской части и Windows 2000 Server (Windows NT) для сервера.

В качестве сетевого протокола предложено использовать TCP/IP.

3 ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОГО РАБОЧЕГО МЕСТА ИНФОРМАЦИОННОЙ СИСТЕМЫ «Е-ВУЗ ИНЕУ»

3.1 Анализ проблемы

Концептуальные положения, ставшие основой для проектирования ИС определялись задачами, стоящими перед вузом, как перед структурной единицей сферы образования.

В современных условиях кредитной системы обучения экономически целесообразно выделить отделы вуза в качестве отдельных подразделений, где каждый отдел возьмет на себя часть функций.

3.2 Методы проверки корректности вводимой первичной информации

При проектировании клиентских приложений работающих с базами данных необходимо заложить основу, обеспечивающую безопасность информации в базах, корректность добавляемой информации, целостность базы при удалении и изменении данных. Реализация поставленной задачи возможна только при проверке вводимой информации на соблюдение требуемых положений по работе с базой данных. Следовательно, проверка корректности вводимой пользователем информации является актуальной задачей при проектировании клиентских приложений.

Проверка корректности выполняется путем решения ряда проблем увеличения точности ввода данных:

1. обязательное наличие значений в водимых полях;
2. проверка типа вводимых данных (цифровой, буквенный, дата);
3. ограничение диапазона значений данных;
4. корректность преобразовательных инструкций при работе с базой данных;
5. разработка обширной системы оповещения пользователя.

Перечисленные выше задачи были решены при разработке АРМ «Успеваемость». Для этого использовались следующие методики:

1. Проверка на наличие значения в поле выполняется с помощью ниже приведенного кода. Код содержит строку текста сообщения, которое появляется на экране, если не указаны вводимые:

```
If cmbGrp.Value = "" Then  
MsgBox "Необходимо выбрать группу!", vbInformation, "АРМ  
Успеваемость"  
Exit Sub  
End If
```

2. Проверка значения вводимого в поле на соответствие определенному типу данных можно осуществить следующим способом:

Например, вводимое значение должно быть в форме даты. Эта проблема решается использованием компонента календарь. Т.е. запрещается

ВВОД ЛЮБЫХ ИНЫХ ДАННЫХ В ПОЛЕ С ОПОВЕЩЕНИЕМ ПОЛЬЗОВАТЕЛЯ
соответствующим сообщением (рисунок 3.1)

Код, выполняющий эту задачу, приведен ниже:

```
If IsNull(plDateProved.Value) = True Then
```

```
MsgBox "Необходимо указать дату!", vbInformation, "APM  
Успеваемость"
```

```
End If
```

А также устанавливаем условие на корректность вводимой даты.

```
If plDateProved > Date Then
```

```
MsgBox "Указана НЕ КОРРЕКТНАЯ дата!", vbCritical, "APM  
Успеваемость"
```

```
End If
```

3. Условие ограничение на диапазон вводимых данных.

В АРМ Успеваемость для ввода баллов было использовано
ограничительное условие ввода данных в виде цифр. Для этого был
использован следующий код:

```
Private Sub Ball_KeyPress(KeyAscii As Integer)
```

```
If ((KeyAscii <> 8) And (KeyAscii < 48)) Or (KeyAscii > 57) Then KeyAscii  
= 0 'backspace = 8
```

```
End Sub
```

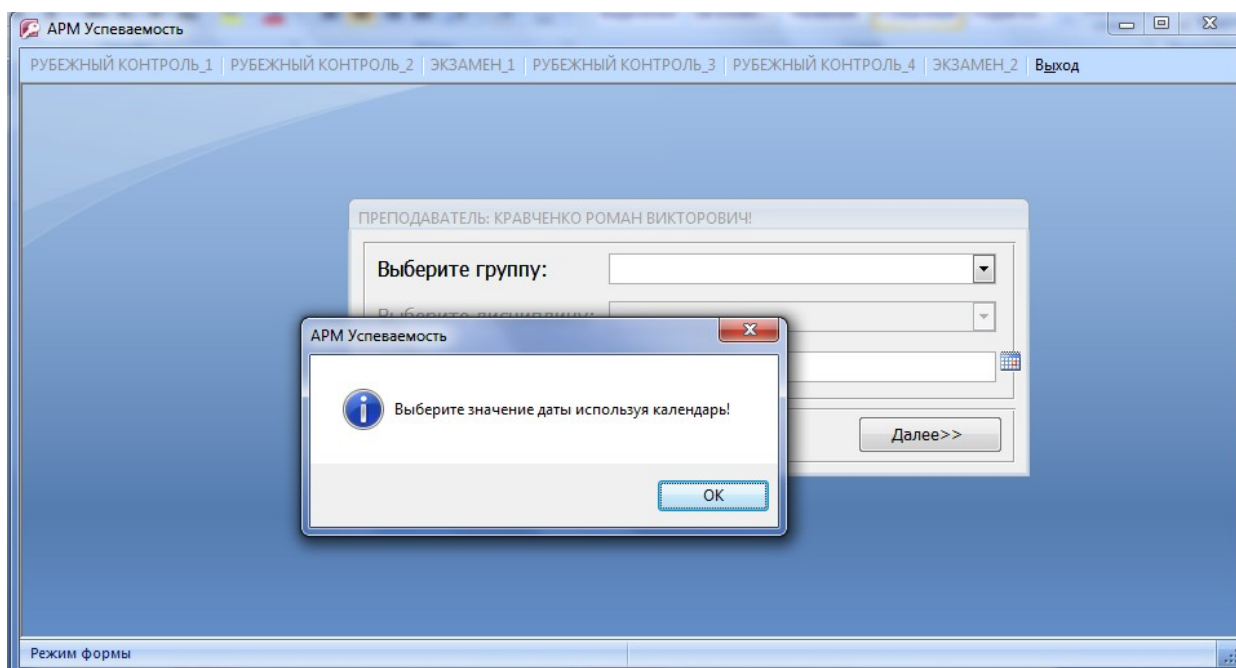


Рисунок 3.1 – Интерфейс ввода даты АРМ «Успеваемость»

Проверка на диапазон вводимых численных значений(от 0 до 30) может
быть выполнена кодом приведенным ниже:

```
Set rs = CurrentProject.Connection.Execute("SELECT IKT, G_CODE,  
ID_PPS, Ball, ID_DIS FROM dbo.tmp_pk WHERE (Ball > 30) AND (G_CODE =
```

```
" & Form_frmPriv.cmbGrp & ") and (ID_DIS = " & Form_frmPriv.cmbdis.Value  
& ") and (ID_PPS = " & Form_frmPriv.tmpUnicode.Value & ")")
```

```
If rs.RecordCount > 0 Then
```

```
MsgBox "Проставлены не корректные баллы!" & vbCrLf & "Значение  
должно быть в диапазоне от 0 до 30!", vbCritical, " APM Успеваемость "
```

```
End If
```

Интерфейс предупреждающего сообщения приведен на рисунке 3.2.

4. Проверку на корректность преобразовательных инструкций при работе с базой данных осуществляется с помощью двух технологий на основе SQL команд.

а) Режим проверки:

```
Set rs = CurrentProject.Connection.Execute("SELECT IKT, G_CODE,  
ID_PPS, Ball, ID_DIS FROM dbo.tmp_pk WHERE (Ball IS NULL) AND  
(G_CODE = " & Form_frmPriv.cmbGrp & ") and (ID_DIS = " &  
Form_frmPriv.cmbdis.Value & ") and (ID_PPS = " &  
Form_frmPriv.tmpUnicode.Value & ")")
```

```
If rs.RecordCount > 0 Then
```

```
MsgBox "Баллы проставлены не увсех!", vbCritical, " APM  
Успеваемость "
```

```
End If
```

б) Режим вставки:

```
Set rs = CurrentProject.Connection.Execute("INSERT INTO dbo.tb_RK1  
(IKT, Ball, ID_PPS, ID_Dis, Date_Proved, Kol_Kr) SELECT IKT, Ball, ID_PPS,  
ID_DIS, Date_Proved, Kol_kr FROM dbo.tmp_pk WHERE (G_CODE = " &  
Form_frmPriv.cmbGrp & ") and (ID_DIS = " & Form_frmPriv.cmbdis.Value &  
") and (ID_PPS = " & Form_frmPriv.tmpUnicode.Value & ")")
```

Тем самым увеличиваем стабильность работы приложения и сохранность данных.

5. Для пользователя одним из основных моментов увеличивающих точность ввода данных являются оповещающие сообщения. Они информируют, предупреждают пользователя о проделанных действиях, предотвращая возможность совершения ошибки.

Код:

```
If rs.RecordCount > 0 Then
```

```
MsgBox "Проставлены не корректные баллы!" & vbCrLf & "Значение  
должно быть в диапазоне от 0 до 30!", vbCritical, "APM Успеваемость "
```

```
Exit Sub
```

```
End If
```

Предлагаемые методики позволят существенно снизить количество ошибок при вводе, тем самым обеспечивая точность сохраняемых данных в базе, целостность данных и стабильность работы приложения.

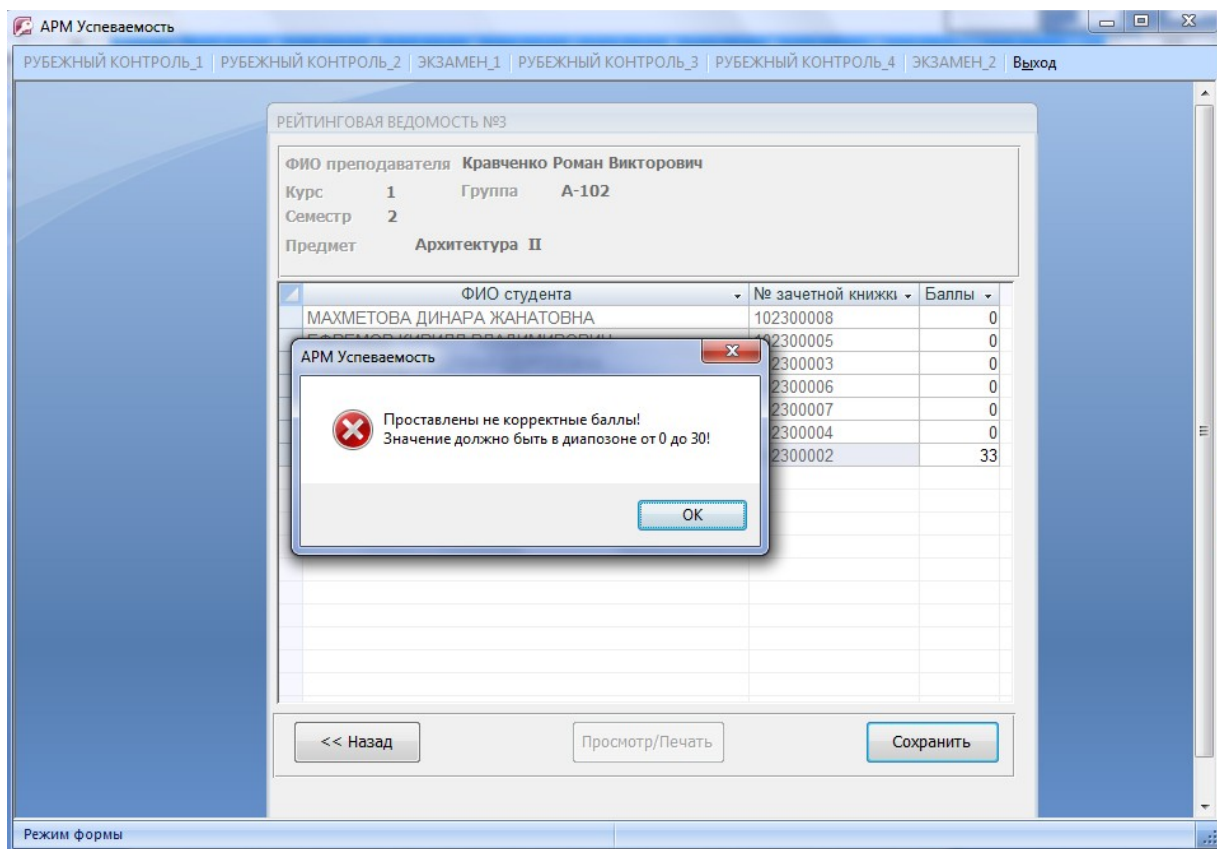


Рисунок 3.2 – Интерфейс проверки введенных данных на вхождение в установленный диапазон значений. АРМ «Успеваемость»

3.3 Разработка автоматизированного рабочего места для учета успеваемости студентов обучающихся по кредитной технологии.

Область применения. АРМ «Успеваемость» (объектно-ориентированная среда Visual Basic) может использоваться сотрудниками отдела регистрации для ввода преподавателями рейтинговых и экзаменационных баллов по каждому студенту.

Функциональные возможности. АРМ «Успеваемость» позволяет заносить данные по успеваемости каждого студента это – рейтинговые и экзаменационные баллы. А также на основе полученных данных формировать сводную ведомость по каждому студенту.

3.3.1 Структура программы

При запуске АРМ «Успеваемость» для удобства пользователю был создан интуитивно понятный интерфейс и структура программы.

Главное меню программы состоит из следующих пунктов меню:

- Рубежный контроль №1. При нажатии осуществляется функция ввода рейтинга за первый рейтинг первого семестра учебного года;
- Рубежный контроль №2. При нажатии осуществляется функция ввода рейтинга за второй рейтинг первого семестра учебного года;

- Рубежный контроль №3. При нажатии осуществляется функция ввода рейтинга за первый рейтинг второго семестра учебного года;
- Рубежный контроль №4. При нажатии осуществляется функция ввода рейтинга за второй рейтинг второго семестра учебного года;
- Экзамен №1. Осуществляется функция ввод экзаменационных баллов за первый семестр учебного года;
- Экзамен №2. Осуществляется функция ввод экзаменационных баллов за второй семестр учебного года;;

Структуру программы можно посмотреть на рисунке 3.3

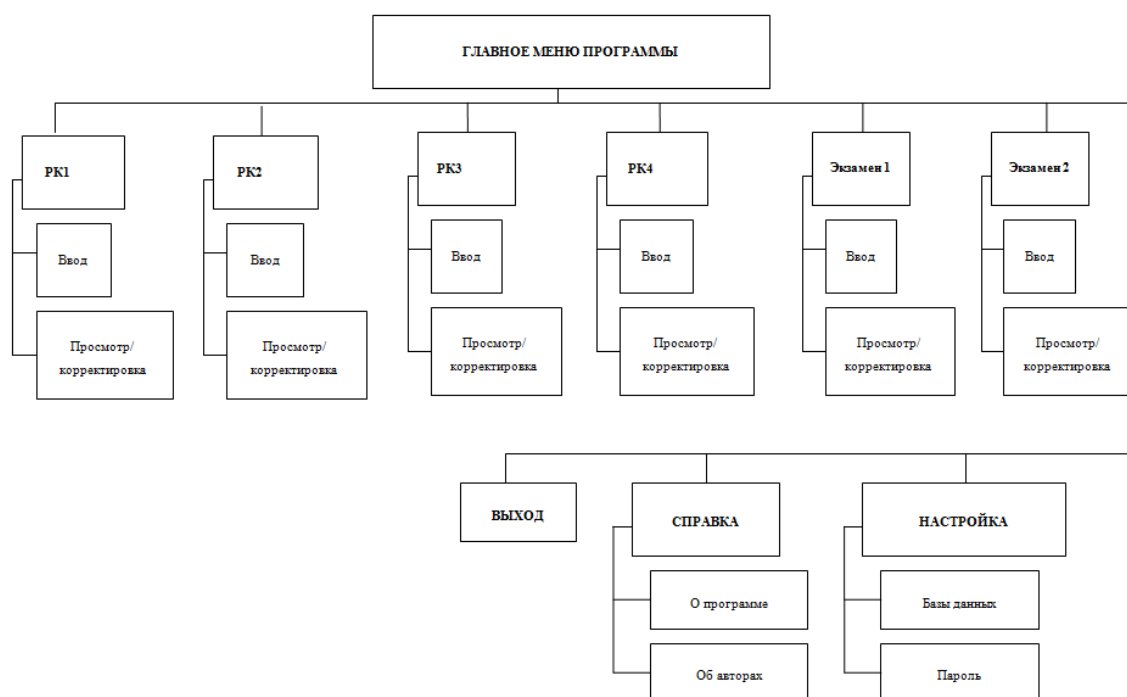


Рисунок 3.3 – Структура программы

3.3.2 Описание работы АРМ «Успеваемость»

При запуске программы появляется первое окно с доступным на данный момент времени функционалом. Окно изображено на рисунке 3.4.

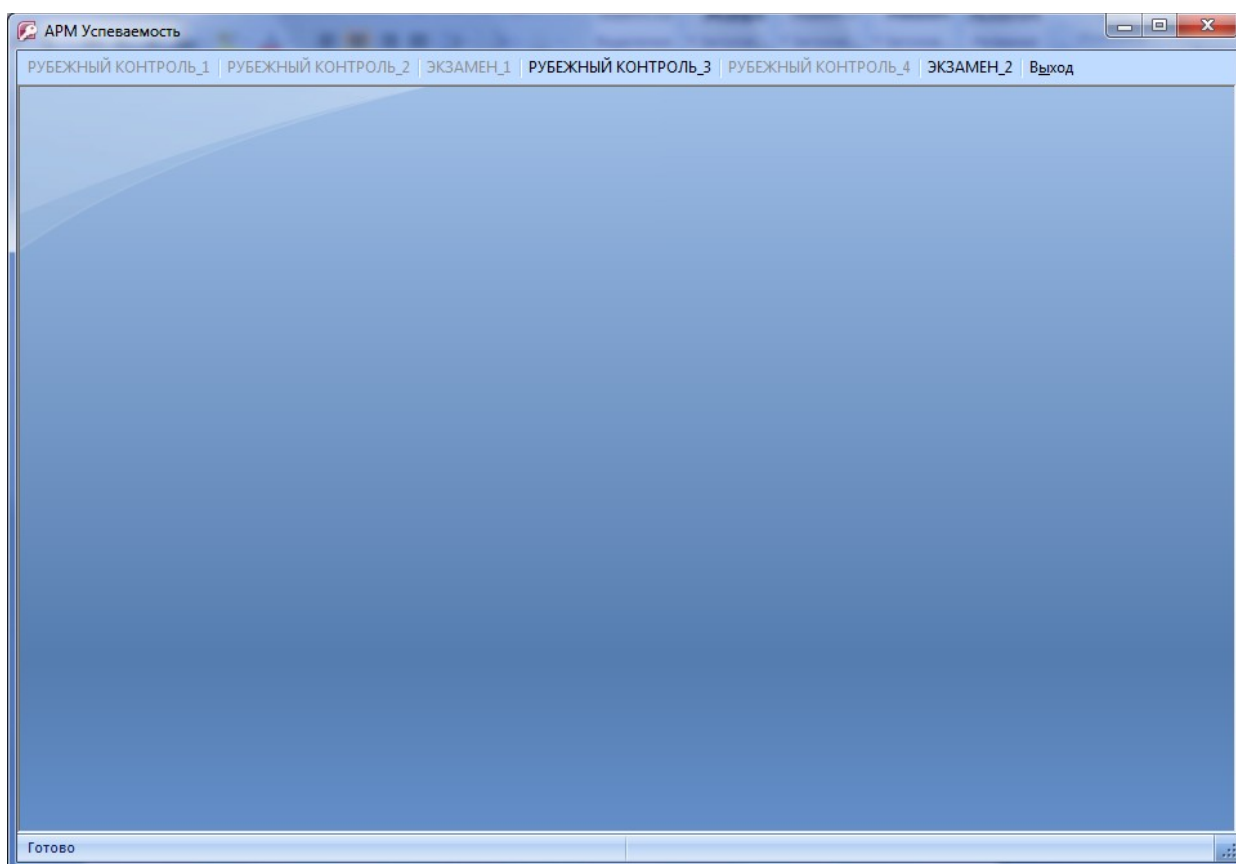


Рисунок 3.4 –главное окно АРМ «Успеваемость»

Для удобства навигации, в программе наглядно представлено главное меню. При входе в автоматизированную подсистему пользователь получает доступ к главному меню, в котором указан возможный функционал учета успеваемости на данный момент, т.е. если при открытии временной промежуток равен времени проставления третьего рубежного контроля текущего учебного года, то этот функционал будет доступен. При выборе доступного меню «Рубежный контроль 3» открывается окно входа в систему. Окно изображено на рисунке 3.5.

После успешной авторизации преподавателя система предлагает пользователю выбрать следующий шаг, войти для ввода рейтинга или для его повышения. Окно изображено на рисунке 3.

На следующем этапе пользователь должен выбрать группу, дисциплину и дату проведения рейтингового контроля. Окно изображено на рисунке 3.6.

Если преподаватель выберет группу, у которой рейтинг уже проставлен система выдаст сообщение, что рейтинг для данной группы уже введен, рисунок 3.7

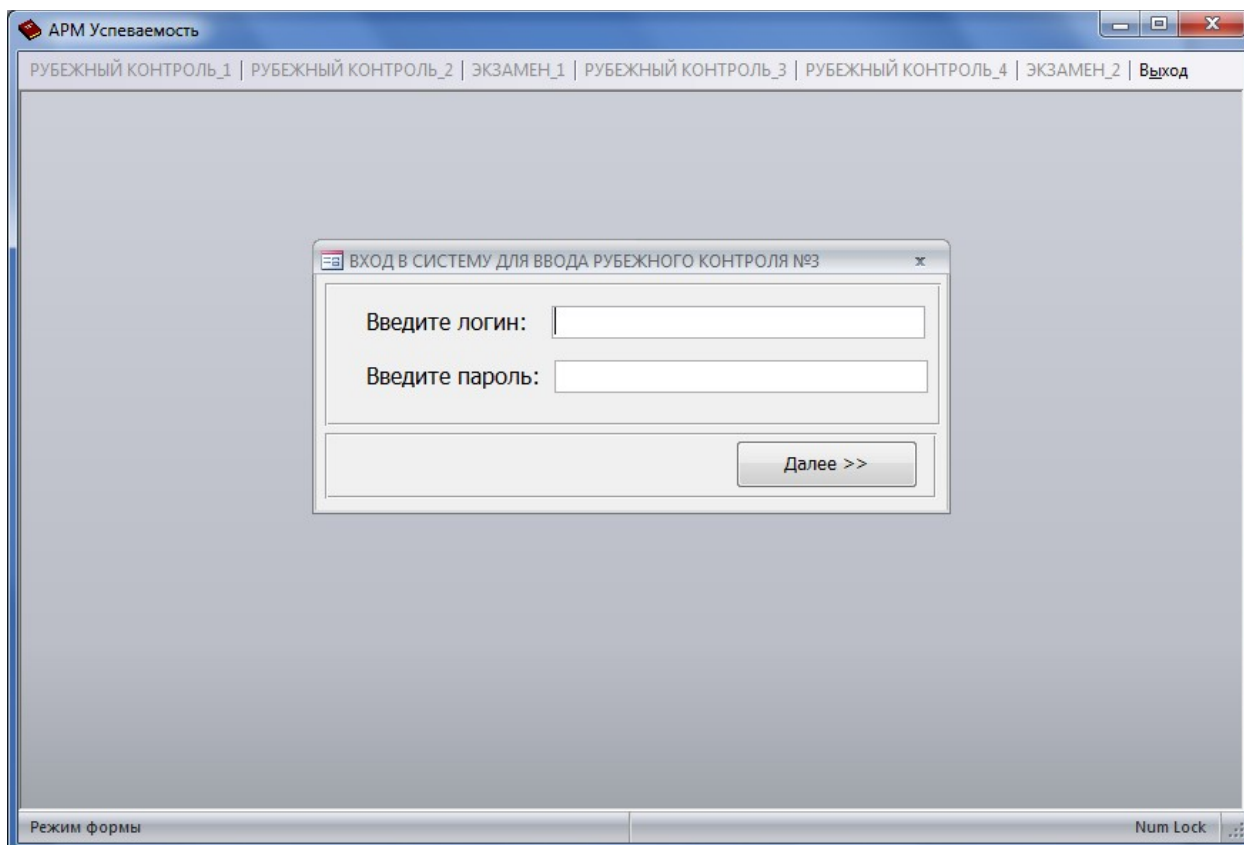


Рисунок 3.4 – Окно для входа в систему для ввода рубежного контроля №3

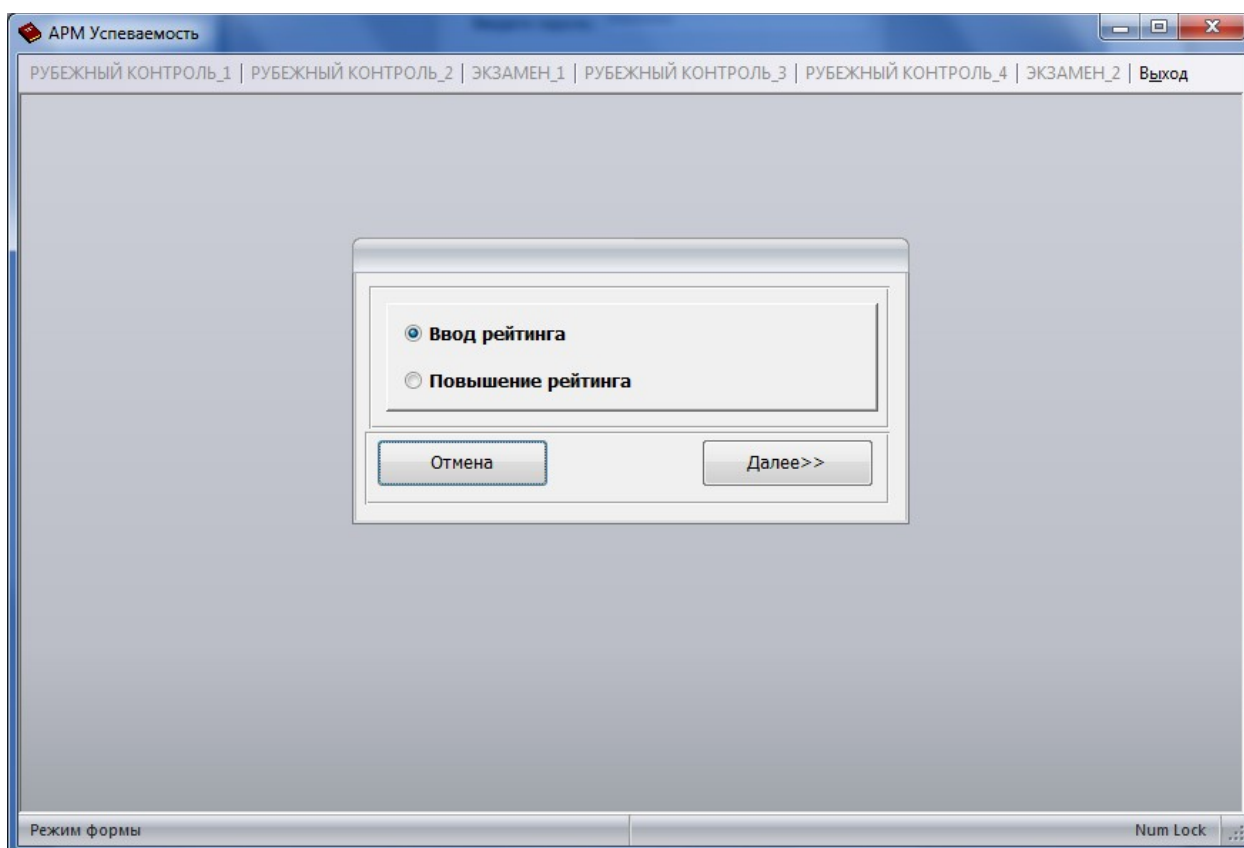


Рисунок 3.5 – Окно выбора следующего действия

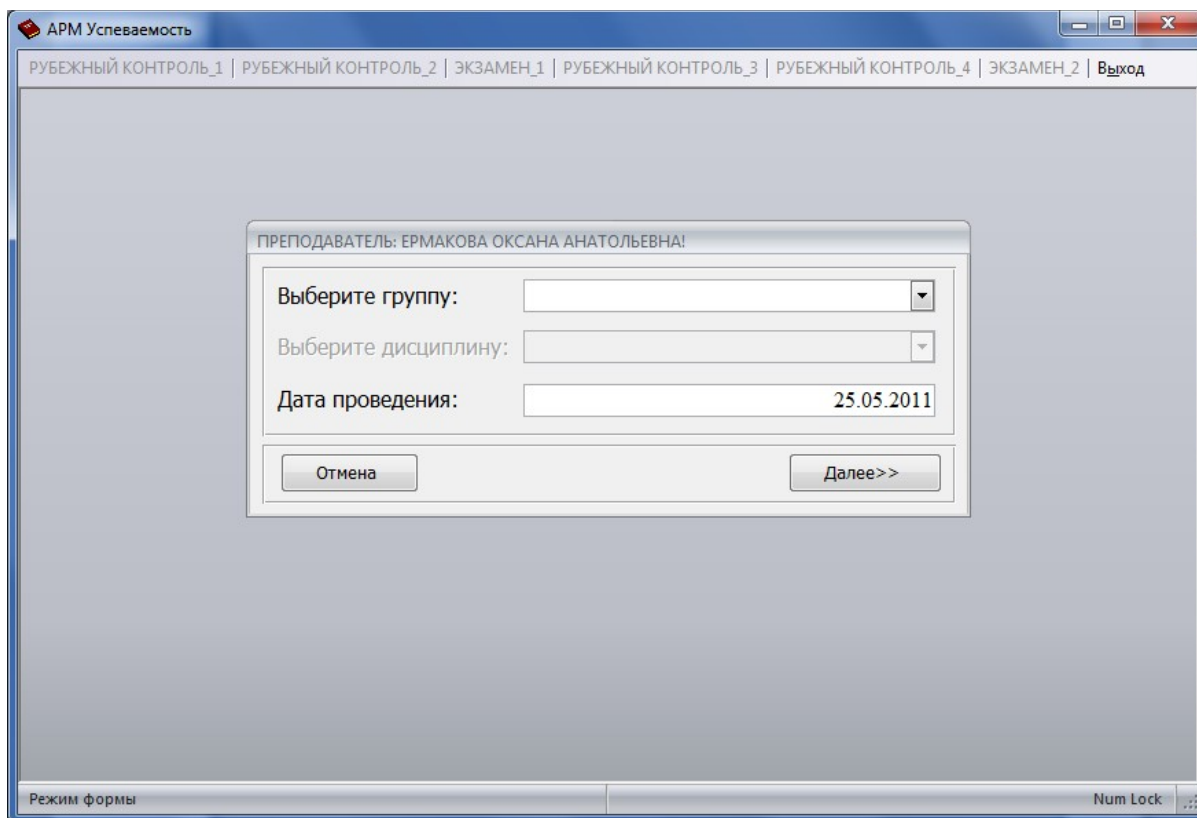


Рисунок 3.6 – Окно выбора группы для проставления рейтинга

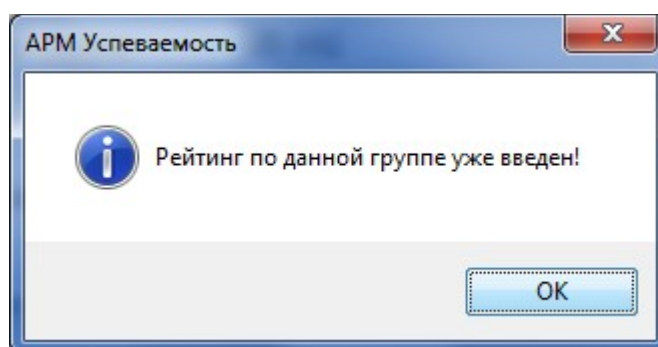


Рисунок 3.7 – Окно предупреждения «Рейтинг уже введен»

Далее появляется само окно ввода рейтинга (Рисунок 3.8), учитывая кредитную систему оценок преподаватель должен вводить баллы соответствующие для каждого студента не превышая максимальный предел 30 баллов (если преподаватель совершит ошибку система контроля корректности ввода выдаст соответствующее сообщение), см. рисунок 3.9.

АРМ Успеваемость

РУБЕЖНЫЙ КОНТРОЛЬ_1 | РУБЕЖНЫЙ КОНТРОЛЬ_2 | ЭКЗАМЕН_1 | РУБЕЖНЫЙ КОНТРОЛЬ_3 | РУБЕЖНЫЙ КОНТРОЛЬ_4 | ЭКЗАМЕН_2 | Выход

РЕЙТИНГОВАЯ ВЕДОМОСТЬ №3

ФИО преподавателя: Ермакова Оксана Анатольевна

Курс: 1 Группа: БП-102

Семестр: 2

Предмет: Биохимия

ФИО студента	№ зачетной книжки	Баллы
БАБАЙ КАРАШАШ	102600113	0
БАХИРЕВА ЕЛЕНА АЛЕКСАНДРОВНА	102600119	0
БИЛЯЛОВА ГУЛЬЖАН БЕКБУЛАТОВНА	102600114	0
ЖУСУПОВА АНАР МАРАТОВНА	102600120	0
КАРАГАЕВА ГУЛЬМИРА КАБДУЛЛАЕВНА	102600115	0
МАГЗУМОВА ГУЛЬСУМ ЖАНАТОВНА	102300017	0
МАХСУТ АЙГЕРИМ	102600116	0
МОРГУНОВА КСЕНИЯ СЕРГЕЕВНА	102600117	0
ШИХАЛЕВ АЛЕКСЕЙ ДМИТРИЕВИЧ	102600118	0

<< Назад Просмотр/Печать Сохранить

Режим формы Num Lock

Рисунок 3.8 – Окно выбора группы для проставления рейтинга

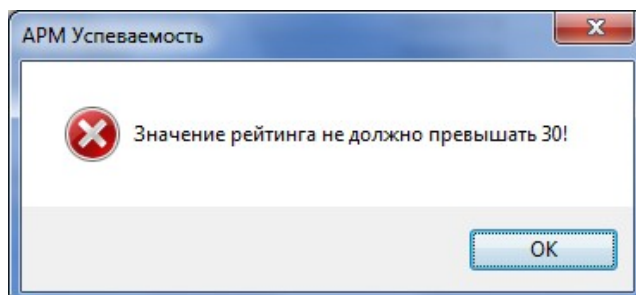


Рисунок 3.9 – Окно «Значение рейтинга не должно превышать 30!»

После ввода рейтинга необходимо нажать «Сохранить» для занесения рейтинга в базу. Система, проверив введенное значение рейтинга на пустоту и превышение, выдаст предупреждающее сообщение (Рисунки 3.10, 3.11) если что-то неправильно введено или информационное для продолжения (Рисунок 3.12). Когда система успешно сохранит данные выйдет соответствующее сообщение (Рисунок 3.13)

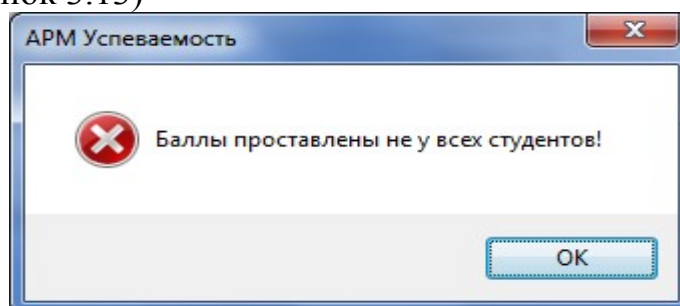


Рисунок 3.10 – Окно предупреждающее «рейтинг проставлен не у всех»

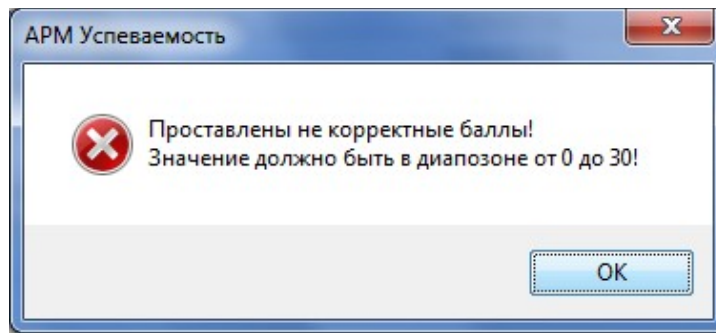


Рисунок 3.11 – Окно предупреждающее «Проставлены не корректные баллы»

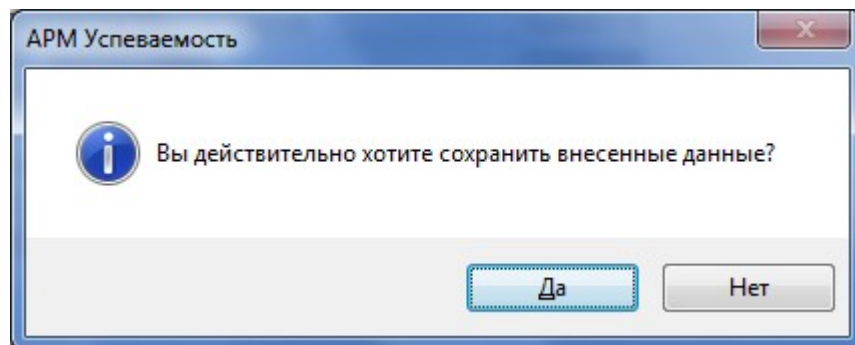


Рисунок 3.12 – Информационное окно для выбора дальнейшего действия

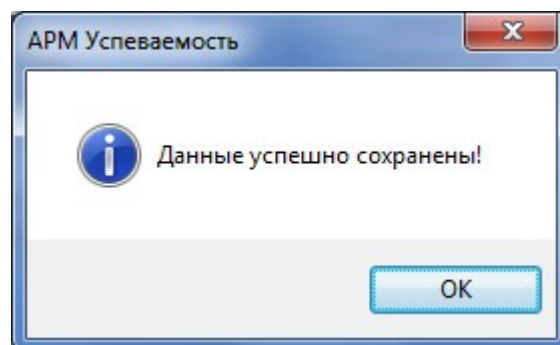


Рисунок 3.13 – Данные успешно сохранены

После успешного ввода рейтинга преподавателю станет доступен функционал по просмотру и выводу на печать рейтинговой ведомости (Рисунок 3.14).

Рейтинговая ведомость отражена в Приложение А.

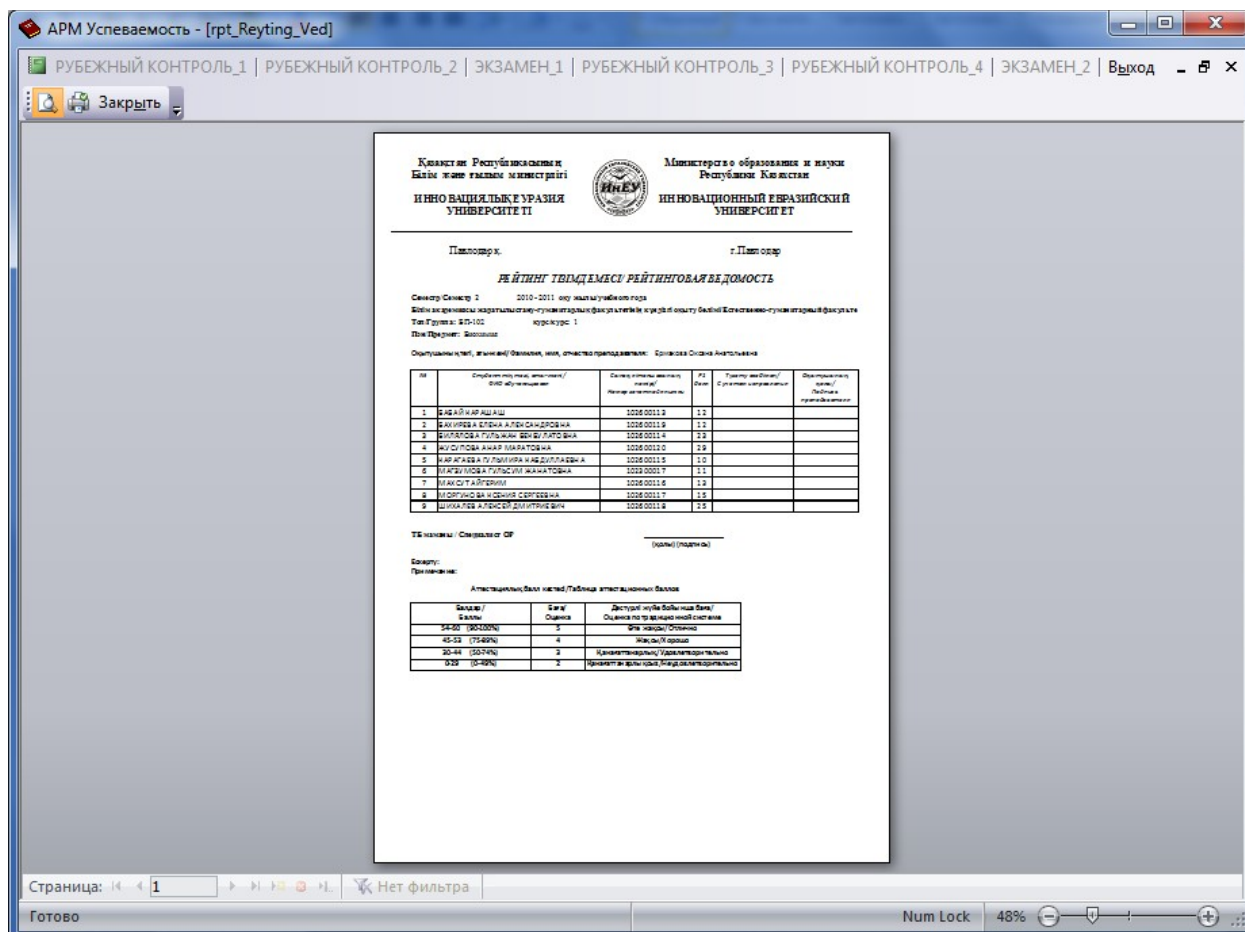


Рисунок 3.14 – Просмотр и вывод на печать рейтинговой ведомости

После того, как преподаватель ввел рейтинг, студенты имеют право по кредитной технологии повысить свой рейтинг. Для этого выбираем опцию «повышение рейтинга» (Рисунок 3.15)

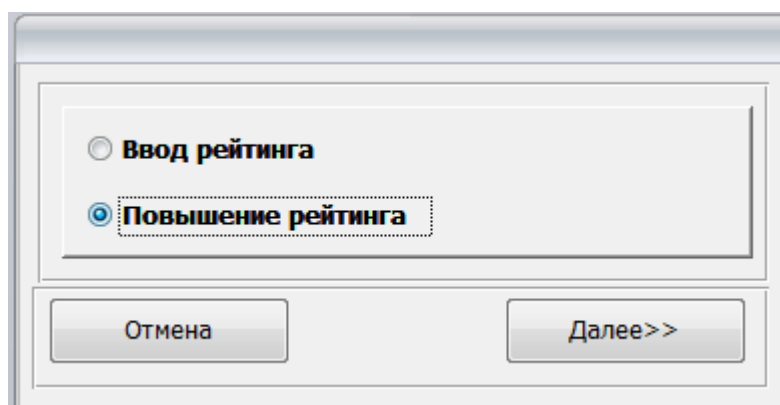


Рисунок 3.15 – Выбор опции повышения рейтинга

После этого выбираем группу, дисциплину и студента которому необходимо повысить рейтинг (Рисунок 3.16). И соответственно повышаем его (Рисунок 3.17)

Как только преподаватель повысил рейтинг, также становится доступен функционал просмотра и вывода на печать рейтинговой ведомости, но уже с учетом исправления.

Қазақстан Республикасының
Білім және ғылым министрлігі

ИННОВАЦИЯЛЫҚ ЕУАЗИЯ
УНИВЕРСИТЕТІ

Министерство образования и науки
Республики Казахстан

ИННОВАЦИОННЫЙ ЕВРАЗИЙСКИЙ
УНИВЕРСИТЕТ

Павлодар қ. г. Павлодар

РЕЙТИНГ ТІЗІМДЕМЕСІ/ РЕЙТИНГОВАЯ ВЕДОМОСТЬ

Семестр/Семестр 2010 - 2011 оқу жылы/учебного года

Білім академиясы жаратылыстану-гуманитарлық факультетінің күндізгі оқыту бөлімі/Естественно-гуманитарный факультет

Топ/Группа: БП-102 курс/курс: 1

Пән/Предмет: Анатомия человека

Оқытушының тегі, аты-жөні/Фамилия, имя, отчество преподавателя: Кривенко Галина Анатольевна

№	Студенттің тегі, аты-жөні/ ФИО обучающегося	Сынақ кітапшасының нөмірі/ Номер зачетной книжки	Р1 балл	Түзету есебімен/ С учетом исправления	Оқытушының қолы/ Подпись преподавателя
1	БАБАЙ КАРАШАШ	102600113	22		
2	БАХИРЕВА ЕЛЕНА АЛЕКСАНДРОВНА	102600119	21	25	
3	БИЛЯЛОВА ГУЛЬЖАН БЕКБУЛАТОВНА	102600114	20		
4	ЖУСУПОВА АНАР МАРАТОВНА	102600120	0		
5	КАРАГАЕВА ГУЛЬМИРА КАБДУЛЛАЕВНА	102600115	12		
6	МАГЗУМОВА ГУЛЬСУМ ЖАНАТОВНА	102300017	25		
7	МАХСУТ АЙГЕРИМ	102600116	25		

Страница: 1

Готово

Рисунок 3.18 – Рейтинговая ведомость с учетом исправления

При выборе преподавателем пункта главного меню «Экзамен» аналогично вводу «Рубежного контроля» вводятся экзаменационные баллы. Аналогично система проверяет на пустоту и превышение значения при вводе и выдает соответствующие сообщения.

Как только преподаватель ввел первый и второй рейтинг и соответственно экзаменационные баллы становится доступен функционал просмотра и вывода на печать экзаменационной ведомости. Где автоматически просчитываются и выводятся итоговый рейтинг, помимо этого высчитывается и выводится в ведомость итоговая оценка в буквенном, бальном и традиционном виде.

Экзаменационная ведомость отражена в Приложении Б

На основе введенных данных при помощи АРМ «Успеваемость», специалисты АРМ «Отдел регистрации ИнЕУ» имеют возможность выводить на печать для студентов сводно-рейтинговые ведомости в разрезе выбранного рейтинга.

Вход АРМ «Отдел регистрации ИнЕУ» осуществляется по аналогии (Рисунок 3.19), после этого выбираем в главном меню Выходные данные – Ведомости- Сводная ведомость РК(Рисунок 3.20).

После этого выбираем группу, указываем необходимый семестр и рейтинг и нажимаем «Далее»(Рисунок 3.21). Если по данной группе еще не введены даны то выйдет соответствующее сообщение, а если преподаватель не выберет семестр или рейтинг, то система выдаст ему соответствующее сообщение.

Однако, если специалист выбрал группу и система не выдал ни каких сообщений то появиться для просмотра и вывода на печать сводно-рейтинговая ведомость(Рисунок 3.22).

Сводная ведомость рейтингового контроля отображена в Приложении В.

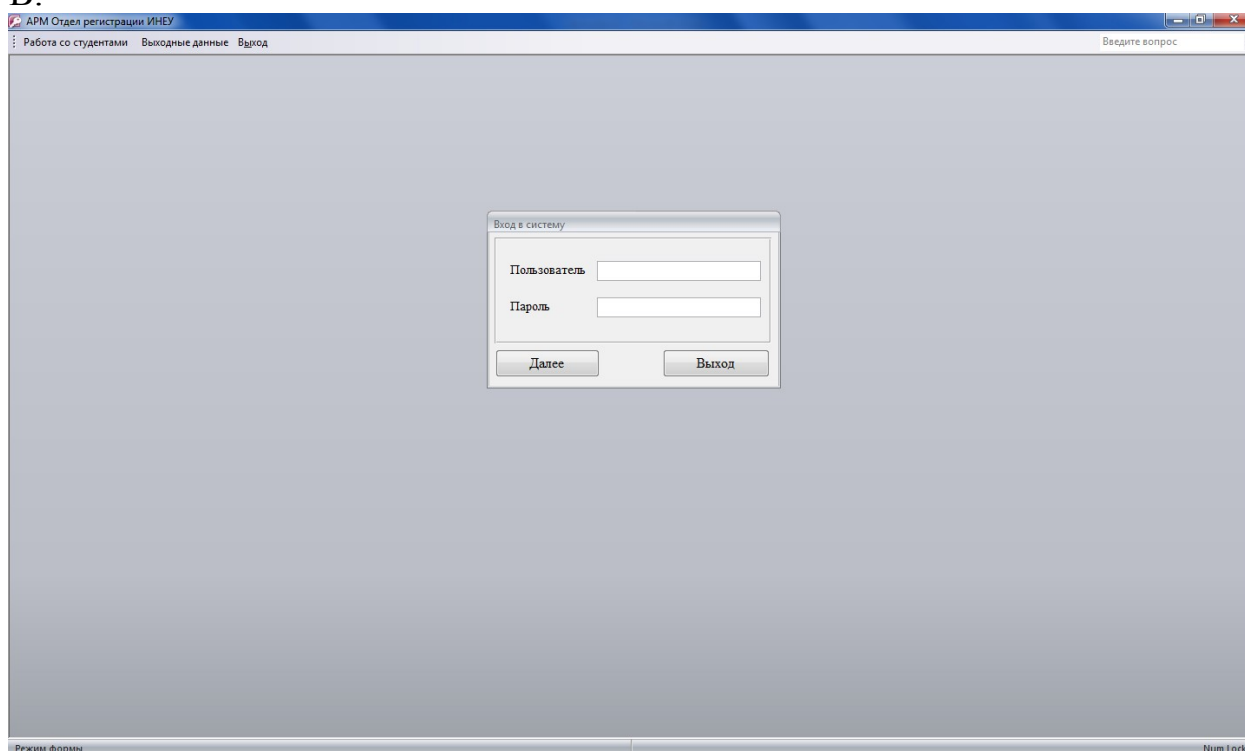


Рисунок 3.19 – Вход в АРМ «Отдел регистрации ИнЕУ»

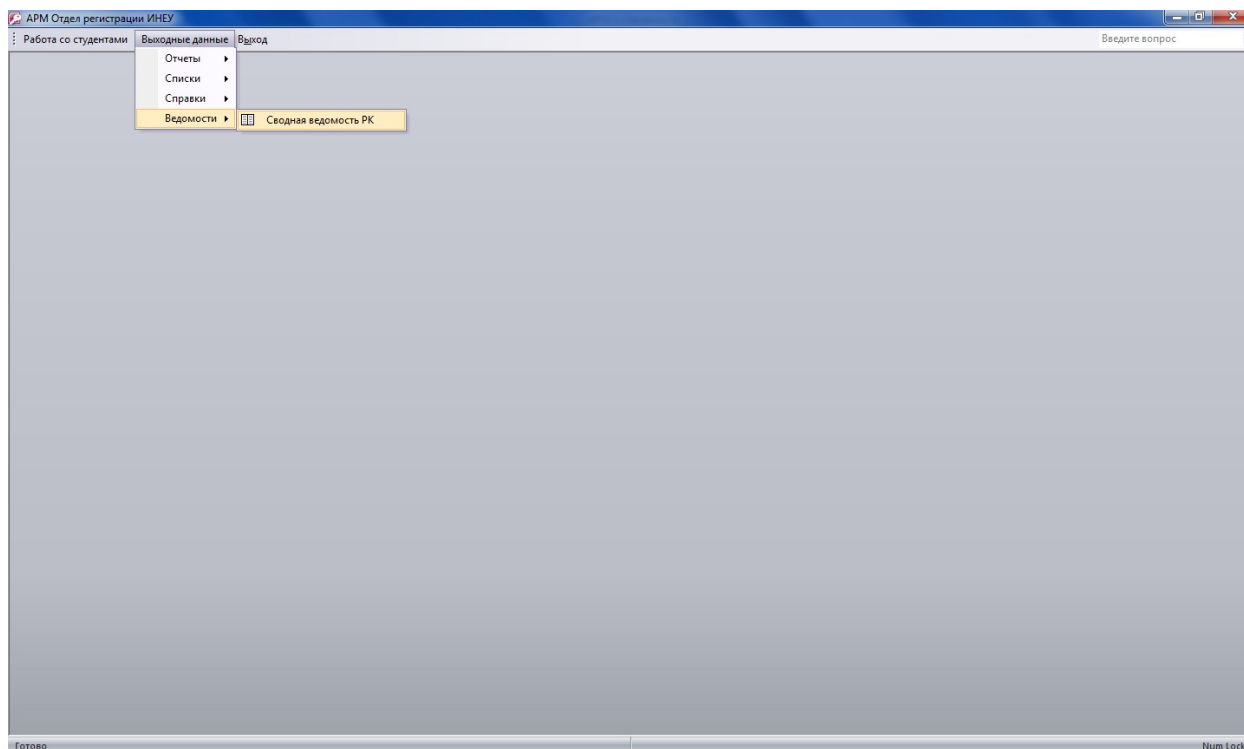


Рисунок 3.20 – Выбор сводной ведомости

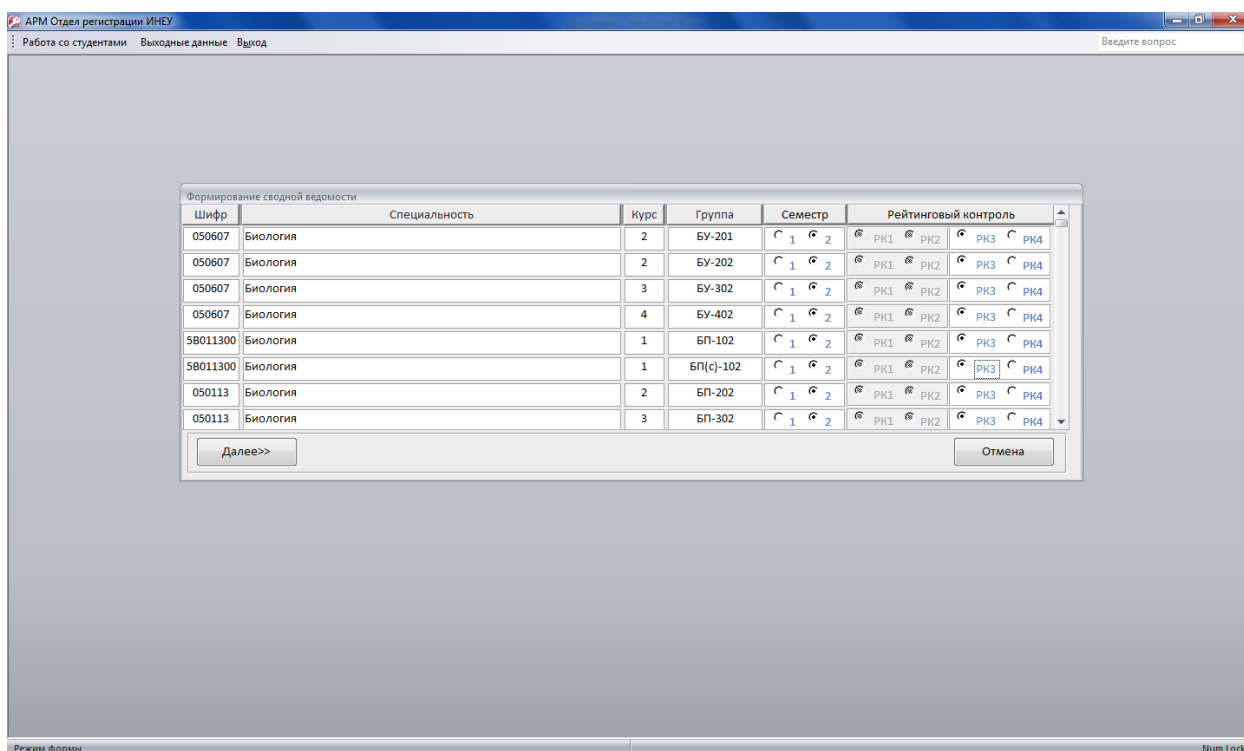


Рисунок 3.21 – Выбор рейтинга для формирования сводной ведомости

3.4 Апробация АРМ «Успеваемость»

Для внедрения результатов научно-исследовательской работы в учебный процесс были разработана автоматизация рабочего места «Успеваемость» и интегрирована в информационную систему «е-ВУЗ» Инновационного Евразийского университета.

Для оценки качества АРМ «Успеваемость» и разработанных методических указаний к лабораторным работам в Инновационном Евразийском университете была создана экспертная комиссия в составе:

1. Светличный В.А., начальник управления информационными технологиями,
2. Жанузакова Н.К., зам. начальника отдела регистрации,
3. Задинова Б.Б., специалист отдела регистрации,
4. Касембекова Д.Б., специалист отдела регистрации,
5. Жагпарова З.Е., специалист отдела регистрации.

Были проведен проведен пробный ввод рейтинга и экзаменационных баллов за 2-ой семестр 2010-2011 учебного года по студентам группы БП-102.

Экспертная комиссия постановила, что АРМ «Успеваемость» должно быть внедрено в учебный процесс, так как его применение позволяет систематизировать и автоматизировать выдачу рейтинговых и экзаменационных ведомостей.

По результатам апробации был составлен акт внедрения результатов научно-исследовательской работы в учебный процесс Инновационного Евразийского университета №154 от 16.05.2011 г.

Заключение

В результате проведенных в диссертации научных исследований были сделаны следующие выводы и разработаны предложения.

Все системы управления учебным процессом в вузе создаются, исходя из общих принципов и цели ведения учета - формирования информационной прозрачности работы вуза, поэтому независимо от предметной области учета для решения задач компьютерной поддержки ведения управления должна применяться единая технология построения компьютерных систем автоматизации.

Возможно создание собственной программной надстройки над традиционными инструментальными средствами программирования реляционных баз данных, специально предназначенной для построения компьютерных систем автоматизации управленческого учета в масштабах вуза, что позволяет повысить эффективность разработки программных продуктов в несколько раз за счет использования стандартизированных моделей данных и программных блоков для их поддержки и обработки.

Существует **функциональное ядро Microsoft Access для систем управления**, представляющее постоянную для различных программных продуктов платформу из глобальных модулей Microsoft Access, которая обеспечивает поддержку стандартной функциональности графических интерфейсов пользователя. Из факта существования функционального ядра следует, что построение компьютерных систем управления, фактически, сведено к реализации именно **информационных функций** для конкретной предметной области учета:

- а. построению модели данных (реализации сервисов данных);
- б. реализации бизнес-логики обработки данных на уровне сервера;
- с. реализации графической логики представления данных.

Это является **минимально необходимой работой**, которую требуется выполнить для построения учетной системы, что свидетельствует об оптимальности полученной технологии построения.

Технология построения учетных систем сведена к **сборочной технологии построения** из стандартных моделей данных, которые поддаются параметризации и подключаются к готовым программным модулям базовой программной платформы (функционального ядра).

Создание специализированной инструментальной среды для программной реализации компьютерных систем автоматизации управленческого учета на основе технологий Microsoft позволило сократить затраты на разработку **не** менее чем в 4 раза. Экономический эффект был достигнут за счет отказа от традиционных методов применения продуктов Microsoft как средств программирования баз данных вообще, вместо этого была реализована программная надстройка базовой функциональности ПО Microsoft, предназначенная специально для решения задач построения программных систем для ведения управления, что выступает существенной доработкой инструментальных средств Microsoft.

На сегодняшний день инструментальные средства Microsoft (Microsoft SQL Server и Microsoft Access) представляют собой одну из наиболее перспективных и технологичных платформ для построения корпоративных систем управленческого учета, по крайней мере, на уровне малого и среднего бизнеса. Среди основных преимуществ можно выделить, как практически нулевую себестоимость базовой программной платформы, в том числе сервера баз данных MS SQL Server (MSDE), гибкую функциональность с возможностью создания собственной инструментальной среды для программирования систем управления и их повсеместное распространение на территории Казахстана.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Головченко Н. Рационализация инструментов системы управленческого учета, анализа и принятия решений. // Финансовая газета. Региональный выпуск. - № 3 (631). - январь 2004. - с 8-9.
2. Глинников М. ИТ-рецепты ICN // Директор информационной службы. -2002. JSfol 1. Издательство "Открытые системы". - с. 3-4.
3. Карминский А,М,, Оленев Н,И,, Примаков А,Г,, Фалько С,Г, Контроллинг в бизнесе. Методические и практические основы построения контроллинга в организациях, - 2-е изд, - М,: Финансы и статистика, 2002, - 256с,
4. Саймон А. Р. Стратегические технологии баз данных: менеджмент на 2000 год: Пер. с англ. /Под ред. и с предисл. М. Р. Когаловского. - М.: Финансы и статистика, 1999. - 479с.
5. Ойхман Е. Г., Попов Э. В. Реинжиниринг бизнеса: Ренижиниринг организаций и информационные технологии. М.: Финансы и статистика, 1997.-336с.
6. Ролланд Д. Основные концепции баз данных.: Пер. с англ. - М.:Издательский дом "Вильямс", 2002. - 256с.
7. Бекаревич Ю. Б., Пушкина П. В., Смирнова Е. Ю. Управление базами данных: Учеб. пособие. СПб.: Изд-во Санкт-Петербургского университета. 1999.-172с.
8. Codd E. F. A Relational Model of Data for Large Shared Data Banks. //Communications of the ACM - June 1970.
9. Дэвидсон Л, Проектирование баз данных на SQL Server 2000, - М,: БИПОМ, Лаборатория знаний, 2003, - 680с,
10. Роб П., Коронел К. Системы баз данных: проектирование, реализация и управление. - 5-е изд., перераб. и доп.: Пер с англ. - СПб.: БХВ-Петербург, 2004.-1040с.
- 11.Локальные вычислительные сети: принципы построения, архитектура, коммуникационные средства /Под ред. С. В. Пазарова. — Кн. 1. — М.: Финансы и статистика, 1994. - 208с.
12. Маклаков С. Переход к технологии клиент-сервер с помощью CASE-средств Computer Associates.
13. (<http://www.interface.ru/fset.asp?Url=/ca/cs.htm>)Карпова Т. С. Базы данных: модели, разработка, реализация. - СПб.: Питер, 2001 —304с.
14. Справочное руководство пользователя к пакету Rational Rose.
15. Литвин П., Гетц К., Гунделой М. Разработка корпоративных приложений в Access 2002. Для профессионалов. - СПб.: Питер, 2003. - 848с.
16. Молчанов А. Ю. Технологии создания прикладных программ для распределенных вычислений. // Автоматизация в промышленности. - 2005. №1.-с. 13
17. Database Applications and Servers // SQL Server Books Online - справочноруководство по Microsoft SQL Server 2000.
18. Форум разработчиков баз данных (www.sql.ru).

19. Свободная энциклопедия Википедия (<http://ru.wikipedia.org/>).
20. Ресурсы по программированию Microsoft Access (<http://www.programmingmsaccess.com>).
21. Литвин П., Гетц К., Гунделой М. Разработка корпоративных приложений в Access 2002. Для профессионалов. - СПб.: Питер, 2003. - 848с.
22. MSDN Library - April 2003.
23. Бухгалтер и компьютер - 2000. JST>> 4. - с. 52.
24. Кузнецов А.И. Разработка концепции и организационно-экономических методов стратегического управления деятельностью предприятия в конкурентной среде. - 1998. (диссертация)
25. Липаев В. В., Позин Б. А. Штрик А. А. Технологии сборочного программирования. /Под ред. В. В. Липаева. - М.: Радио и связь, 1992.
26. Программирование Access 97 (http://nsa.chat.ru/Secur_About.html).
27. Программное обеспечение CASE-средств (<http://www.interface.ru/fset.asp?Url=/soft.htm>).
28. Технологии программирования в среде Microsoft Access (<http://msa.km.ru/msa/access/article.asp?id=295>).
29. Тихомиров Ю. В. Microsoft SQL Server 7.0: разработка приложений. -СПб.: БХВ-Петербург, 2000. - 352с.
30. Управленческий консалтинг и информационные технологии. //Национальная экономическая газета, №. 42 (659) от 06.06.2003. - с 4.
31. Финансовый директор - 2004. JNfol. - с. 20-24.
32. Форум по VB, VBA, MS Access (<http://am.rusimport.ru/MsAccess/forums.aspx?t= 1>).
33. Ivar Jacobson. Object-Oriented Software Engineering.- USA: Addison-Wesley Publishing Company 1994. - 528 с.
34. MSDN Library - April 2003.
35. PC Week/RE - 1999. №34. - с. 134-142.
36. SQL Server Books Online (справочное руководство по
37. Microsoft SQL Server 2000).
38. Селко Д. Деревья в SQL. // журнал DBMS, April 1996.(перевод статьи находится на сайте <http://sdm.viptop.ru/articles/sqltrees.html>)
39. Семенов А. МГТС: реалии и перспективы: Автоматизация на службе бухучета // Московский телефонист. - № 15. - 2003. - с. (<http://www.mgts.nj/>)

ЛИСТИНГ ПРОГРАММЫ

```

Option Compare Database
Private Sub cmbdis_GotFocus()
Dim rs As ADODB.Recordset
Set rs = CurrentProject.Connection.Execute("SELECT G_Code, KodStSpec, Y_Create, Kurs FROM
dbo.MAIN_Spr_dbo_sprGrp WHERE (G_Code = '' & cmbGrp.Value & '')")

Select Case Form_frmAut.tmp_par
Case 1
    Select Case rs.Fields("kurs").Value
    Case 1
        cmbdis.RowSource = "SELECT dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis,
        dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus FROM dbo.MAIN_Spr_dbo_Spr_Dis INNER JOIN
        dbo.MAIN_Spr_dbo_Spr_Dis_Caf ON dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis =
        dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Kod_Dis WHERE (dbo.MAIN_Spr_dbo_Spr_Dis.KodStSpec = '' &
        rs.Fields("KodStSpec") & '') AND (dbo.MAIN_Spr_dbo_Spr_Dis.GodPost = '' & rs.Fields("Y_Create") & '') AND
        (dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Semestr = 1) ORDER BY dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus"
    Case 2
        cmbdis.RowSource = "SELECT dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis,
        dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus FROM dbo.MAIN_Spr_dbo_Spr_Dis INNER JOIN
        dbo.MAIN_Spr_dbo_Spr_Dis_Caf ON dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis =
        dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Kod_Dis WHERE (dbo.MAIN_Spr_dbo_Spr_Dis.KodStSpec = '' &
        rs.Fields("KodStSpec") & '') AND (dbo.MAIN_Spr_dbo_Spr_Dis.GodPost = '' & rs.Fields("Y_Create") & '') AND
        (dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Semestr = 3) ORDER BY dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus"
    Case 3
        cmbdis.RowSource = "SELECT dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis,
        dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus FROM dbo.MAIN_Spr_dbo_Spr_Dis INNER JOIN
        dbo.MAIN_Spr_dbo_Spr_Dis_Caf ON dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis =
        dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Kod_Dis WHERE (dbo.MAIN_Spr_dbo_Spr_Dis.KodStSpec = '' &
        rs.Fields("KodStSpec") & '') AND (dbo.MAIN_Spr_dbo_Spr_Dis.GodPost = '' & rs.Fields("Y_Create") & '') AND
        (dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Semestr = 5) ORDER BY dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus"
    Case 4
        cmbdis.RowSource = "SELECT dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis,
        dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus FROM dbo.MAIN_Spr_dbo_Spr_Dis INNER JOIN
        dbo.MAIN_Spr_dbo_Spr_Dis_Caf ON dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis =
        dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Kod_Dis WHERE (dbo.MAIN_Spr_dbo_Spr_Dis.KodStSpec = '' &
        rs.Fields("KodStSpec") & '') AND (dbo.MAIN_Spr_dbo_Spr_Dis.GodPost = '' & rs.Fields("Y_Create") & '') AND
        (dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Semestr = 7) ORDER BY dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus"
    Case 5
        cmbdis.RowSource = "SELECT dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis,
        dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus FROM dbo.MAIN_Spr_dbo_Spr_Dis INNER JOIN
        dbo.MAIN_Spr_dbo_Spr_Dis_Caf ON dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis =
        dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Kod_Dis WHERE (dbo.MAIN_Spr_dbo_Spr_Dis.KodStSpec = '' &
        rs.Fields("KodStSpec") & '') AND (dbo.MAIN_Spr_dbo_Spr_Dis.GodPost = '' & rs.Fields("Y_Create") & '') AND
        (dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Semestr = 9) ORDER BY dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus"
    End Select
Case 2
    Select Case rs.Fields("kurs").Value
    Case 1
        cmbdis.RowSource = "SELECT dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis,
        dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus FROM dbo.MAIN_Spr_dbo_Spr_Dis INNER JOIN
        dbo.MAIN_Spr_dbo_Spr_Dis_Caf ON dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis =
        dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Kod_Dis WHERE (dbo.MAIN_Spr_dbo_Spr_Dis.KodStSpec = '' &
        rs.Fields("KodStSpec") & '') AND (dbo.MAIN_Spr_dbo_Spr_Dis.GodPost = '' & rs.Fields("Y_Create") & '') AND
        (dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Semestr = 1) ORDER BY dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus"
    Case 2
        cmbdis.RowSource = "SELECT dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis,
        dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus FROM dbo.MAIN_Spr_dbo_Spr_Dis INNER JOIN
        dbo.MAIN_Spr_dbo_Spr_Dis_Caf ON dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis =
        dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Kod_Dis WHERE (dbo.MAIN_Spr_dbo_Spr_Dis.KodStSpec = '' &

```


Case 4

Select Case rs.Fields("kurs").Value

Case 1

```
cmbdis.RowSource = "SELECT dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis,  
dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus FROM dbo.MAIN_Spr_dbo_Spr_Dis INNER JOIN  
dbo.MAIN_Spr_dbo_Spr_Dis_Caf ON dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis =  
dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Kod_Dis WHERE (dbo.MAIN_Spr_dbo_Spr_Dis.KodStSpec = '' &  
rs.Fields("KodStSpec") & '') AND (dbo.MAIN_Spr_dbo_Spr_Dis.GodPost = '' & rs.Fields("Y_Create") & '') AND  
(dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Semestr = 2) ORDER BY dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus"
```

Case 2

```
cmbdis.RowSource = "SELECT dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis,  
dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus FROM dbo.MAIN_Spr_dbo_Spr_Dis INNER JOIN  
dbo.MAIN_Spr_dbo_Spr_Dis_Caf ON dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis =  
dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Kod_Dis WHERE (dbo.MAIN_Spr_dbo_Spr_Dis.KodStSpec = '' &  
rs.Fields("KodStSpec") & '') AND (dbo.MAIN_Spr_dbo_Spr_Dis.GodPost = '' & rs.Fields("Y_Create") & '') AND  
(dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Semestr = 4) ORDER BY dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus"
```

Case 3

```
cmbdis.RowSource = "SELECT dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis,  
dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus FROM dbo.MAIN_Spr_dbo_Spr_Dis INNER JOIN  
dbo.MAIN_Spr_dbo_Spr_Dis_Caf ON dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis =  
dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Kod_Dis WHERE (dbo.MAIN_Spr_dbo_Spr_Dis.KodStSpec = '' &  
rs.Fields("KodStSpec") & '') AND (dbo.MAIN_Spr_dbo_Spr_Dis.GodPost = '' & rs.Fields("Y_Create") & '') AND  
(dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Semestr = 6) ORDER BY dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus"
```

Case 4

```
cmbdis.RowSource = "SELECT dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis,  
dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus FROM dbo.MAIN_Spr_dbo_Spr_Dis INNER JOIN  
dbo.MAIN_Spr_dbo_Spr_Dis_Caf ON dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis =  
dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Kod_Dis WHERE (dbo.MAIN_Spr_dbo_Spr_Dis.KodStSpec = '' &  
rs.Fields("KodStSpec") & '') AND (dbo.MAIN_Spr_dbo_Spr_Dis.GodPost = '' & rs.Fields("Y_Create") & '') AND  
(dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Semestr = 8) ORDER BY dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus"
```

Case 5

```
cmbdis.RowSource = "SELECT dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis,  
dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus FROM dbo.MAIN_Spr_dbo_Spr_Dis INNER JOIN  
dbo.MAIN_Spr_dbo_Spr_Dis_Caf ON dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis =  
dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Kod_Dis WHERE (dbo.MAIN_Spr_dbo_Spr_Dis.KodStSpec = '' &  
rs.Fields("KodStSpec") & '') AND (dbo.MAIN_Spr_dbo_Spr_Dis.GodPost = '' & rs.Fields("Y_Create") & '') AND  
(dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Semestr = 10) ORDER BY dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus"
```

End Select

```
"SELECT Kod_Dis, Name_Rus FROM dbo.MAIN_Spr_dbo_Spr_Dis WHERE (KodStSpec = '' &  
rs.Fields("KodStSpec") & '') AND (GodPost = '' & rs.Fields("Y_Create") & '') ORDER BY Name_Rus"
```

End Select

cmbdis.Requery

End Sub

Private Sub btnNext_Click()

If cmbGrp.Value = "" Or IsNull(cmbGrp.Value) = True Then

MsgBox "", vbInformation, ""

Exit Sub

End If

If cmbdis.Value = "" Or IsNull(cmbdis.Value) = True Then

MsgBox "Îáíáðíàèîî áúáðàòù àèñòèèèíó!", vbInformation, "ÀÐÌ Óñîáâââîîñòù"

Exit Sub

End If

If plDateProved.Value = "" Or IsNull(plDateProved.Value) = True Then

MsgBox "Îáíáðíàèîî ôèàçàòù ààòó!", vbInformation, "ÀÐÌ Óñîáâââîîñòù"

Exit Sub

End If

```

If cmbFIO.Value = "" Or IsNull(cmbFIO.Value) = True Then
MsgBox "Íáíáðíàèì àùáðàòù ñòóáíòà!", vbInformation, "ÀÐÌ Õñíááááíñòù"
Exit Sub
End If

Dim rs, RS2 As ADODB.Recordset
Dim stFIO, stDis As String
Set rs = New ADODB.Recordset
Set RS2 = New ADODB.Recordset
Set rs = CurrentProject.Connection.Execute("SELECT IKT,FIO FROM dbo.tbStud WHERE IKT="" &
cmbFIO.Value & """)
stFIO = ""
stFIO = rs.Fields("FIO").Value
Set rs = CurrentProject.Connection.Execute("SELECT Kod_Dis, Name_Rus FROM dbo.MAIN_Spr_dbo_Spr_Dis
WHERE (Kod_Dis = "" & cmbdis.Value & "")")
stDis = ""
stDis = rs.Fields("Name_Rus").Value

If Form_frmAut.tmp_par.Value = 1 Then
Set RS2 = CurrentProject.Connection.Execute("SELECT IKT, ID_PPS, ID_Dis, Date_Proved, Ball, Kol_Kr
FROM dbo.tb_RK1_Povyshenie WHERE (IKT = "" & cmbFIO.Value & "") AND (ID_Dis = "" & cmbdis.Value &
"")")
If RS2.RecordCount > 0 Then
MsgBox "Äÿ ñòóáíòà(èè) " & stFIO & " ðáéòèí ñ äèñòèèèí " & """" & stDis & """" & " óæå ñâùðàèñý!",
vbCritical, "ÀÐÌ Õñíááááíñòù"
Exit Sub
End If
Set rs = CurrentProject.Connection.Execute("SELECT IKT, ID_PPS, ID_Dis, Date_BD, Date_Proved, Ball,
Kol_Kr FROM dbo.tb_RK1 WHERE (IKT = "" & cmbFIO.Value & "") AND (ID_Dis = "" & cmbdis.Value & "")")
If rs.RecordCount = 0 Then
MsgBox "Äÿ ñòóáíòà(èè) " & stFIO & " ðáéòèí ñ äèñòèèèí " & """" & stDis & """" & " íá ââíàèèñý!",
vbCritical, "ÀÐÌ Õñíááááíñòù"
Exit Sub
End If
End If

If Form_frmAut.tmp_par.Value = 2 Then
Set RS2 = CurrentProject.Connection.Execute("SELECT IKT, ID_PPS, ID_Dis, Date_Proved, Ball, Kol_Kr
FROM dbo.tb_RK1_Povyshenie WHERE (IKT = "" & cmbFIO.Value & "") AND (ID_Dis = "" & cmbdis.Value &
"")")
If RS2.RecordCount > 0 Then
MsgBox "Äÿ ñòóáíòà(èè) " & stFIO & " ðáéòèí ñ äèñòèèèí " & """" & stDis & """" & " óæå ñâùðàèñý!",
vbCritical, "ÀÐÌ Õñíááááíñòù"
Exit Sub
End If
Set rs = CurrentProject.Connection.Execute("SELECT IKT, ID_PPS, ID_Dis, Date_BD, Date_Proved, Ball,
Kol_Kr FROM dbo.tb_RK2 WHERE (IKT = "" & cmbFIO.Value & "") AND (ID_Dis = "" & cmbdis.Value & "")")
If rs.RecordCount = 0 Then
MsgBox "Äÿ ñòóáíòà(èè) " & stFIO & " ðáéòèí ñ äèñòèèèí " & """" & stDis & """" & " íá ââíàèèñý!",
vbCritical, "ÀÐÌ Õñíááááíñòù"
Exit Sub
End If
End If

If Form_frmAut.tmp_par.Value = 3 Then
Set RS2 = CurrentProject.Connection.Execute("SELECT IKT, ID_PPS, ID_Dis, Date_Proved, Ball, Kol_Kr
FROM dbo.tb_RK3_Povyshenie WHERE (IKT = "" & cmbFIO.Value & "") AND (ID_Dis = "" & cmbdis.Value &
"")")
If RS2.RecordCount > 0 Then
MsgBox "Äÿ ñòóáíòà(èè) " & stFIO & " ðáéòèí ñ äèñòèèèí " & """" & stDis & """" & " óæå ñâùðàèñý!",
vbCritical, "ÀÐÌ Õñíááááíñòù"
Exit Sub

```

```

End If
Set rs = CurrentProject.Connection.Execute("SELECT IKT, ID_PPS, ID_Dis, Date_BD, Date_Proved, Ball, Kol_Kr FROM dbo.tb_RK3 WHERE (IKT = '' & cmbFIO.Value & '') AND (ID_Dis = '' & cmbdis.Value & '')")
If rs.RecordCount = 0 Then
    MsgBox "Äÿ ñòóääíòà(êè) " & stFIO & " ääèèíä ñ äèñöèrèèíä " & "" & stDis & "" & " íä ââîäèèñÿ!", vbCritical, "ÀÐÌ Óñîääââîñòü"
    Exit Sub
End If
End If

If Form_frmAut.tmp_par.Value = 4 Then
    Set RS2 = CurrentProject.Connection.Execute("SELECT IKT, ID_PPS, ID_Dis, Date_Proved, Ball, Kol_Kr FROM dbo.tb_RK4_Povyshenie WHERE (IKT = '' & cmbFIO.Value & '') AND (ID_Dis = '' & cmbdis.Value & '')")
    If RS2.RecordCount > 0 Then
        MsgBox "Äÿ ñòóääíòà(êè) " & stFIO & " ääèèíä ñ äèñöèrèèíä " & "" & stDis & "" & " óæâ ñâûøäèñÿ!", vbCritical, "ÀÐÌ Óñîääââîñòü"
        Exit Sub
    End If
    Set rs = CurrentProject.Connection.Execute("SELECT IKT, ID_PPS, ID_Dis, Date_BD, Date_Proved, Ball, Kol_Kr FROM dbo.tb_RK4 WHERE (IKT = '' & cmbFIO.Value & '') AND (ID_Dis = '' & cmbdis.Value & '')")
    If rs.RecordCount = 0 Then
        MsgBox "Äÿ ñòóääíòà(êè) " & stFIO & " ääèèíä ñ äèñöèrèèíä " & "" & stDis & "" & " íä ââîäèèñÿ!", vbCritical, "ÀÐÌ Óñîääââîñòü"
        Exit Sub
    End If
End If

Dim rst As ADODB.Recordset
Set rst = New ADODB.Recordset
Set rst = CurrentProject.Connection.Execute("SELECT G_Code, G_Name, Kurs FROM MAIN_Spr_dbo_sprGrp WHERE ( G_Code = '' & cmbGrp.Value & '')")

Select Case Form_frmAut.tmp_par.Value
Case 1:
    If plDateProved > Date Then
        MsgBox "Óèàçàíà ÍÄ ÊÎÐÄÄÊÓÍÄâ äàòà!", vbCritical, "ÀÐÌ Óñîääââîñòü"
        Exit Sub
    End If
Case 2:
    If plDateProved > Date Then
        MsgBox "Óèàçàíà ÍÄ ÊÎÐÄÄÊÓÍÄâ äàòà!", vbCritical, "ÀÐÌ Óñîääââîñòü"
        Exit Sub
    End If
Case 3:
    If plDateProved.Value < CDate("01.02." & Year(Date)) Or plDateProved.Value > Date Then
        MsgBox "Óèàçàíà ÍÄ ÊÎÐÄÄÊÓÍÄâ äàòà!", vbCritical, "ÀÐÌ Óñîääââîñòü"
        Exit Sub
    End If
Case 4:
    If plDateProved > Date Then
        MsgBox "Óèàçàíà ÍÄ ÊÎÐÄÄÊÓÍÄâ äàòà!", vbCritical, "ÀÐÌ Óñîääââîñòü"
        Exit Sub
    End If
End Select

'Set rs = CurrentProject.Connection.Execute("Update dbo.tmp_pk SET Date_Proved = '' & plDateProved.Value &

```

```

",ID_PPS = "" & tmpUnicode.Value & "", ID_DIS = "" & cmbdis.Value & "", Kol_kr = 0 WHERE (G_Code = "" &
cmbGrp.Value & ""))

```

```

DoCmd.OpenForm "frmRK_Povyshenie"

```

```

End Sub

```

```

Private Sub cmbFIO_GotFocus()

```

```

cmbFIO.RowSource = "SELECT IKT AS Expr2, FIO FROM dbo.tbStud WHERE (G_CODE = "" & cmbGrp.Value
& "") ORDER BY FIO"

```

```

cmbFIO.Requery

```

```

End Sub

```

```

Private Sub cmbGrp_Change()

```

```

cmbdis.Value = ""

```

```

Dim rs As ADODB.Recordset

```

```

Set rs = New ADODB.Recordset

```

```

Set rs = CurrentProject.Connection.Execute("select ikt from dbo.tbStud where G_Code = "" & cmbGrp.Value & """)

```

```

If rs.RecordCount = 0 Then

```

```

MsgBox "À âúáðáííé ãðóñâ íàò ñòóääíðíâ!", vbInformation, "ÀÐÌ Óñíââââíñòù"

```

```

Exit Sub

```

```

End If

```

```

Set rs = CurrentProject.Connection.Execute("SELECT G_Code, KodStSpec, Y_Create FROM
dbo.MAIN_Spr_dbo_sprGrp WHERE (G_Code = "" & cmbGrp.Value & ""))"

```

```

Set rs = CurrentProject.Connection.Execute("SELECT Kod_Dis, Name_Rus FROM dbo.MAIN_Spr_dbo_Spr_Dis
WHERE (KodStSpec = "" & rs.Fields("KodStSpec") & "") AND (GodPost = "" & rs.Fields("Y_Create") & "")
ORDER BY Name_Rus")

```

```

If rs.RecordCount = 0 Then

```

```

MsgBox "Äÿ ãðóñù " & cmbGrp.Text & " ñíèñíê àèñòèèèí ÍÁ ÄÄÄÄÄÍ!" & vbCrLf & "Íáðàòèòâñù â ìòââè
ðââèñòðàòèè!", vbInformation, "ÀÐÌ Óñíââââíñòù"

```

```

Exit Sub

```

```

End If

```

```

cmbdis.Enabled = True

```

```

End Sub

```

```

Private Sub Form_Load()

```

```

'ñíâíâ ðâñêèääèè íà èèðèèèèó

```

```

If CurrentLayout <> Language.IngRussian Then

```

```

    ChangeLayout Language.IngRussian

```

```

End If

```

```

Dim rs As ADODB.Recordset

```

```

Set rs = New ADODB.Recordset

```

```

Set rs = CurrentProject.Connection.Execute("SELECT Kod, UniCode, Fio, Imy, Otch FROM
dbo.MAIN_Kadri_Vuz_dbo_BD WHERE ( UniCode= "" & Form_frmAut.tmp_unicode.Value & ""))"

```

```

Dim param_fio As String

```

```

Dim param_imy As String

```

```

Dim param_otch As String

```

```

Dim param_unic As String

```

```

param_unic = rs.Fields("UniCode").Value

```

```

param_fio = rs.Fields("Fio").Value

```

```

param_imy = rs.Fields("Imy").Value

```

```

param_otch = rs.Fields("Otch").Value

```

```

Form_frm_Povyshenie.Form.Caption = "İÐÄİİÄÄÄÀÖÄËÜ: " & UCase(param_fio) & " " & UCase(param_imy)

```

```

& " " & UCase(param_otch) & "!"
Form_frm_Povyshenie.tmp_fio.Value = param_fio & " " & param_imy & " " & param_otch
Form_frm_Povyshenie.tmpUnicode.Value = param_unic

End Sub

Private Sub otmena_Click()

CommandBars("glMenu").Controls("ΔΟΑΑΑΕΙΥΕ ΕΙΙΟΔΙΕÜ_1").Enabled = False
CommandBars("glMenu").Controls("ΔΟΑΑΑΕΙΥΕ ΕΙΙΟΔΙΕÜ_2").Enabled = False
CommandBars("glMenu").Controls("ΔΟΑΑΑΕΙΥΕ ΕΙΙΟΔΙΕÜ_3").Enabled = False
CommandBars("glMenu").Controls("ΔΟΑΑΑΕΙΥΕ ΕΙΙΟΔΙΕÜ_4").Enabled = False
CommandBars("glMenu").Controls("ΥΕÇÀÌÁ_1").Enabled = False
CommandBars("glMenu").Controls("ΥΕÇÀÌÁ_2").Enabled = False

DoCmd.Close
DoCmd.OpenForm "frm_Vybor"
End Sub

Private Sub plDateProved_GotFocus()
MsgBox "Άúáâðèòά çíà÷áíèά ààòù èññëüçóý êàëáíäðü!", vbInformation, "ÀÐÌ Óñíääáâíñòü"
End Sub

Private Sub plDateProved_KeyPress(KeyAscii As Integer)
KeyAscii = 0
MsgBox "Άúáâðèòά çíà÷áíèά ààòù èññëüçóý êàëáíäðü!", vbInformation, "ÀÐÌ Óñíääáâíñòü"
Exit Sub
End Sub

Option Compare Database

Private Sub btnNext_Click()
Select Case grpReit.Value
Case 1:
Form_frm_Vybor.Visible = False
DoCmd.OpenForm "frmPriv"
Case 2:
Form_frm_Vybor.Visible = False
DoCmd.OpenForm "frm_Povyshenie"
End Select
End Sub

Private Sub Êñíèà6_Click()
CommandBars("glMenu").Controls("ΔΟΑΑΑΕΙΥΕ ΕΙΙΟΔΙΕÜ_1").Enabled = False
CommandBars("glMenu").Controls("ΔΟΑΑΑΕΙΥΕ ΕΙΙΟΔΙΕÜ_2").Enabled = False
CommandBars("glMenu").Controls("ΔΟΑΑΑΕΙΥΕ ΕΙΙΟΔΙΕÜ_3").Enabled = True
CommandBars("glMenu").Controls("ΔΟΑΑΑΕΙΥΕ ΕΙΙΟΔΙΕÜ_4").Enabled = False
CommandBars("glMenu").Controls("ΥΕÇÀÌÁ_1").Enabled = False
CommandBars("glMenu").Controls("ΥΕÇÀÌÁ_2").Enabled = True
DoCmd.Close
End Sub

Option Compare Database

Private Sub Form_Activate()
Dim stDocName As String
stDocName = "frmAut"
DoCmd.OpenForm stDocName
Form_frmAut.tmp_par.Value = 1
Form_frm1.Visible = False

```

End Sub

Option Compare Database

Private Sub Form_Activate()

```
Dim stDocName As String
stDocName = "frmAut"
DoCmd.OpenForm stDocName
Form_frmAut.tmp_par.Value = 2
Form_frm2.Visible = False
```

End Sub

Option Compare Database

Private Sub Form_Activate()

```
Dim stDocName As String
stDocName = "frmAut"
DoCmd.OpenForm stDocName
Form_frmAut.tmp_par.Value = 3
Form_frm3.Visible = False
```

End Sub

Option Compare Database

Private Sub Form_Activate()

```
Dim stDocName As String
stDocName = "frmAut"
DoCmd.OpenForm stDocName
Form_frmAut.tmp_par.Value = 4
Form_frm4.Visible = False
```

End Sub

Option Compare Database

Private Sub Form_Activate()

```
Dim stDocName As String
stDocName = "frmAut"
DoCmd.OpenForm stDocName
Form_frmAut.tmp_par.Value = 5
Form_frm5.Visible = False
```

End Sub

Option Compare Database

Private Sub Form_Activate()

```
Dim stDocName As String
stDocName = "frmAut"
DoCmd.OpenForm stDocName
Form_frmAut.tmp_par.Value = 6
Form_frm6.Visible = False
```

End Sub

Option Compare Database

Private Sub Form_Activate()

```
CommandBars("glMenu").Controls("ΔΟΑΑΑΕΙΥΕ ΕΙΙΟΔΙΕÜ_1").Enabled = False
CommandBars("glMenu").Controls("ΔΟΑΑΑΕΙΥΕ ΕΙΙΟΔΙΕÜ_2").Enabled = False
CommandBars("glMenu").Controls("ΔΟΑΑΑΕΙΥΕ ΕΙΙΟΔΙΕÜ_3").Enabled = False
CommandBars("glMenu").Controls("ΔΟΑΑΑΕΙΥΕ ΕΙΙΟΔΙΕÜ_4").Enabled = False
CommandBars("glMenu").Controls("ΥΕÇÀÌÁ_1").Enabled = False
CommandBars("glMenu").Controls("ΥΕÇÀÌÁ_2").Enabled = False
```

```
If tmp_par.Value = 1 Or tmp_par.Value = 2 Or tmp_par.Value = 3 Or tmp_par.Value = 4 Then
Form_frmAut.Caption = "ΑÕÎÄ Â ÑÈÑÒÀÌÓ ÄËß ÂÂÎÄÀ ΔΟΑΑΑΕΙΥΕ ΕΙΙΟΔΙΕß '" & tmp_par.Value
End If
```

```
If tmp_par.Value = 5 Then
Form_frmAut.Caption = "ΑÕÎÄ Â ÑÈÑÒÀÌÓ ÄËß ÂÂÎÄÀ ΥΕÇÀÌÁÄ"
End If
```

```
If tmp_par.Value = 6 Then
Form_frmAut.Caption = "ΑÕÎÄ Â ÑÈÑÒÀÌÓ ÄËß ÂÂÎÄÀ ΥΕÇÀÌÁÄ"
End If
```

```
pl_login.Value = ""
pl_pas.Value = ""
End Sub
```

```
Private Sub Form_Close()
CommandBars("glMenu").Controls("ΔΟΑΑΑΕΙΥΕ ΕΙΙΟΔΙΕÜ_1").Enabled = False
CommandBars("glMenu").Controls("ΔΟΑΑΑΕΙΥΕ ΕΙΙΟΔΙΕÜ_2").Enabled = False
CommandBars("glMenu").Controls("ΔΟΑΑΑΕΙΥΕ ΕΙΙΟΔΙΕÜ_3").Enabled = True
CommandBars("glMenu").Controls("ΔΟΑΑΑΕΙΥΕ ΕΙΙΟΔΙΕÜ_4").Enabled = False
CommandBars("glMenu").Controls("ΥΕÇÀÌÁ_1").Enabled = False
CommandBars("glMenu").Controls("ΥΕÇÀÌÁ_2").Enabled = True
End Sub
```

Private Sub Form_Open(Cancel As Integer)

```
pl_login.Value = ""
pl_pas.Value = ""
End Sub
```

Private Sub εἰîèà_âîéòè_Click()

```
If pl_login.Value = "" Then
MsgBox "Ââââèòâ εἰâεί!", vbCritical, "ÀΔÌ Óñîââââîîñòü"
Exit Sub
End If
```

```
If pl_pas.Value = "" Then
MsgBox "Ââââèòâ îâðîëü!", vbCritical, "ÀΔÌ Óñîââââîîñòü"
Exit Sub
End If
```

```
Dim rs As ADODB.Recordset
Set rs = New ADODB.Recordset
```

```
Set rs = CurrentProject.Connection.Execute("SELECT KodSotr, Login, Passw FROM
dbo.MAIN_usersINEU_dbo_tbUsersPGPI WHERE (Login = '' & pl_login.Value & '') and (Passw = '' &
pl_pas.Value & '')")
Dim param1 As String
'-----if
```



```

If rs.RecordCount > 0 Then
    param1 = rs.Fields("KodSotr").Value
    tmp_unicode.Value = param1
    If tmp_par = 5 Or tmp_par = 6 Then
        Form_frmAut.Visible = False
        DoCmd.OpenForm "frmPriv"
    Else
        Form_frmAut.Visible = False
        DoCmd.OpenForm "frm_Vybor"
    End If
Else
    MsgBox "Ëîæï/îðîëü åååâî ÍÁ ÂÄÍÍ!", vbCritical, "ÀÐÌ Óñîååâîñîù"
Exit Sub

End If
'-----end if

End Sub

Option Compare Database
Private Sub btnAdd_Click()

Dim rs As ADODB.Recordset
Set rs = New ADODB.Recordset

'-----add
stMsg = MsgBox("Âû ääéñðàèòäëüî ðîèèðâ ñîððàîèòü âîññâîîü äâîîü?", vbYesNo + vbInformation, "ÀÐÌ Óñîååâîñîù")
Select Case stMsg
    Case vbYes

Set rs = CurrentProject.Connection.Execute("SELECT IKT, G_CODE, ID_PPS, Ball, ID_DIS FROM dbo.tmp_ekz
WHERE (Ball IS NULL) AND (G_CODE = '' & Form_frmPriv.cmbGrp & '') and (ID_DIS = '' &
Form_frmPriv.cmbdis.Value & '') and (ID_PPS = '' & Form_frmPriv.tmpUnicode.Value & '')")

If rs.RecordCount > 0 Then

MsgBox "Áäëëü îðîñðàèèâîü íá ó âñâð ñòóâîîîâ!", vbCritical, "ÀÐÌ Óñîååâîñîù"
Exit Sub
End If

Set rs = CurrentProject.Connection.Execute("SELECT IKT, G_CODE, ID_PPS, Ball, ID_DIS FROM dbo.tmp_ekz
WHERE (Ball > 40) AND (G_CODE = '' & Form_frmPriv.cmbGrp & '') and (ID_DIS = '' &
Form_frmPriv.cmbdis.Value & '') and (ID_PPS = '' & Form_frmPriv.tmpUnicode.Value & '')")
If rs.RecordCount > 0 Then

MsgBox "Ïðîñðàèèâîü íâ èððàèèòüâ ääëëü!" & vbCrLf & "Çîâ-âîèâ âîëæîü áùòü â àèàñçîâ îò 0 âî 40!", vbCritical,
"ÀÐÌ Óñîååâîñîù"
Exit Sub
End If

Set rs = CurrentProject.Connection.Execute("SELECT IKT, G_CODE, Ball, ID_PPS, ID_DIS FROM dbo.tmp_ekz
WHERE (Ball <> 0) AND (G_CODE = '' & Form_frmPriv.cmbGrp & '') and (ID_DIS = '' &
Form_frmPriv.cmbdis.Value & '') and (ID_PPS = '' & Form_frmPriv.tmpUnicode.Value & '')")

If rs.RecordCount = 0 Then

stMsg = MsgBox("Ó âñâð ñòóâîîîâ èîèè-âñâîâ ääëëâ äââîü ""0""!", vbOKCancel + vbInformation, "ÀÐÌ Óñîååâîñîù")
Select Case stMsg
    Case vbOK

```

```

stMsg = MsgBox("Àû ääéñòäèòäëüí òìòèòà ïñòääèòü äääâáíúâ ääëëü?", vbYesNo + vbInformation, "ÀÐÌ Óñíääâáíñòü")
Select Case stMsg
    Case vbYes
        Blsave = True
        btnPrint.Enabled = True

If Form_frmAut.tmp_par.Value = 5 Then
Set rs = CurrentProject.Connection.Execute("INSERT INTO dbo.tb_Ekzamen1 (IKT, Ball, ID_PPS, ID_Dis, Date_Proved, Kol_Kr, Num_Semestr) SELECT IKT, Ball, ID_PPS, ID_DIS, Date_Proved, Kol_kr, Num_Semestr FROM dbo.tmp_ekz WHERE (G_CODE = '' & Form_frmPriv.cmbGrp & '') and (ID_DIS = '' & Form_frmPriv.cmbdis.Value & '') and (ID_PPS = '' & Form_frmPriv.tmpUnicode.Value & '')")
End If

If Form_frmAut.tmp_par.Value = 6 Then
Set rs = CurrentProject.Connection.Execute("INSERT INTO dbo.tb_Ekzamen2 (IKT, Ball, ID_PPS, ID_Dis, Date_Proved, Kol_Kr, Num_Semestr) SELECT IKT, Ball, ID_PPS, ID_DIS, Date_Proved, Kol_kr, Num_Semestr FROM dbo.tmp_ekz WHERE (G_CODE = '' & Form_frmPriv.cmbGrp & '') and (ID_DIS = '' & Form_frmPriv.cmbdis.Value & '') and (ID_PPS = '' & Form_frmPriv.tmpUnicode.Value & '')")
End If

'-----delete
Set rs = CurrentProject.Connection.Execute("DELETE FROM dbo.tmp_ekz WHERE (ID_DIS = '' & Form_frmPriv.cmbdis.Value & '') and (G_CODE = '' & Form_frmPriv.cmbGrp.Value & '') and (ID_PPS = '' & Form_frmPriv.tmpUnicode.Value & '')")

MsgBox "Àáííúâ óñíâøíî ñîððáíúí!", vbInformation, "ÀÐÌ Óñíääâáíñòü"
btnPrint.SetFocus
btnAdd.Enabled = False
'DoCmd.Close
Exit Sub

    Case vbNo
        Exit Sub
    End Select

Case vbCancel
Exit Sub
End Select

End If

Blsave = True
btnPrint.Enabled = True

If Form_frmAut.tmp_par.Value = 5 Then
Set rs = CurrentProject.Connection.Execute("INSERT INTO dbo.tb_Ekzamen1 (IKT, Ball, ID_PPS, ID_Dis, Date_Proved, Kol_Kr, Num_Semestr) SELECT IKT, Ball, ID_PPS, ID_DIS, Date_Proved, Kol_kr, Num_Semestr FROM dbo.tmp_ekz WHERE (G_CODE = '' & Form_frmPriv.cmbGrp & '') and (ID_DIS = '' & Form_frmPriv.cmbdis.Value & '') and (ID_PPS = '' & Form_frmPriv.tmpUnicode.Value & '')")
End If

If Form_frmAut.tmp_par.Value = 6 Then
Set rs = CurrentProject.Connection.Execute("INSERT INTO dbo.tb_Ekzamen2 (IKT, Ball, ID_PPS, ID_Dis, Date_Proved, Kol_Kr, Num_Semestr) SELECT IKT, Ball, ID_PPS, ID_DIS, Date_Proved, Kol_kr, Num_Semestr FROM dbo.tmp_ekz WHERE (G_CODE = '' & Form_frmPriv.cmbGrp & '') and (ID_DIS = '' & Form_frmPriv.cmbdis.Value & '') and (ID_PPS = '' & Form_frmPriv.tmpUnicode.Value & '')")
End If

```

```

'-----delete
Set rs = CurrentProject.Connection.Execute("DELETE FROM dbo.tmp_ekz WHERE (ID_DIS = "" &
Form_frmPriv.cmbdis.Value & "") and (G_CODE = "" & Form_frmPriv.cmbGrp.Value & "") and (ID_PPS = "" &
Form_frmPriv.tmpUnicode.Value & "")")

MsgBox "Àáíúâ òñĩãøíî ñîððáíáíú!", vbInformation, "ÀÐÌ Óñĩããããîñòù"
btnPrint.SetFocus
btnAdd.Enabled = False
'DoCmd.Close

Case vbNo

End Select

End Sub

Private Sub btnNazad_Click()
If Blsave = False Then
stMsg = MsgBox("Áû ääéñòâèðäëüíî ðìòèðâ ìòìáíèòù äâîä äáííûð?", vbYesNo + vbInformation, "ÀÐÌ Óñĩããããîñòù")
Select Case stMsg
    Case vbYes

'-----delete
Set rs = CurrentProject.Connection.Execute("DELETE FROM dbo.tmp_ekz WHERE (ID_DIS = "" &
Form_frmPriv.cmbdis.Value & "") and (G_CODE = "" & Form_frmPriv.cmbGrp.Value & "") and (ID_PPS = "" &
Form_frmPriv.tmpUnicode.Value & "")")
Form_frmPriv.cmbGrp.Value = ""
Form_frmPriv.cmbdis.Value = ""
Form_frmPriv.cmbdis.Enabled = False
Form_frmPriv.plDateProved = Date
DoCmd.Close

Case vbNo

End Select
Else

Form_frmPriv.cmbGrp.Value = ""
Form_frmPriv.cmbdis.Value = ""
Form_frmPriv.cmbdis.Enabled = False
Form_frmPriv.plDateProved = Date
DoCmd.Close

End If
End Sub

Private Sub btnPrint_Click()
On Error Resume Next
DoCmd.OpenReport "rpt_Ekz_Ved_Part_2", acViewPreview
End Sub

Private Sub Form_Load()

Dim rs As ADODB.Recordset
Set rs = New ADODB.Recordset
Set rs = CurrentProject.Connection.Execute("SELECT G_Code, G_Name, Kurs FROM MAIN_Spr_dbo_sprGrp
WHERE ( G_Code="" & Form_frmPriv.cmbGrp.Value & "")")

If Form_frmAut.tmp_par.Value = 5 Then
Form_frmEKZ.Caption = "ÝÊÇÀÌÁÌÀÖÈÎÎÍÀß ÂÄÄÎÎÎÑÒÛ"

```

```

If rs.Fields("Kurs").Value = 1 Then Form_frmEKZ.İfëâ37.Value = " 1"
If rs.Fields("Kurs").Value = 2 Then Form_frmEKZ.İfëâ37.Value = " 3"
If rs.Fields("Kurs").Value = 3 Then Form_frmEKZ.İfëâ37.Value = " 5"
If rs.Fields("Kurs").Value = 4 Then Form_frmEKZ.İfëâ37.Value = " 7"
If rs.Fields("Kurs").Value = 5 Then Form_frmEKZ.İfëâ37.Value = " 9"

```

End If

```

If Form_frmAut.tmp_par.Value = 6 Then
Form_frmEKZ.Caption = "ÝÊÇÀÌÁÌÀÖÈÎÍÀß ÅÅÄÎÎÑÒÜ"

```

```

If rs.Fields("Kurs").Value = 1 Then Form_frmEKZ.İfëâ37.Value = " 2"
If rs.Fields("Kurs").Value = 2 Then Form_frmEKZ.İfëâ37.Value = " 4"
If rs.Fields("Kurs").Value = 3 Then Form_frmEKZ.İfëâ37.Value = " 6"
If rs.Fields("Kurs").Value = 4 Then Form_frmEKZ.İfëâ37.Value = " 8"
If rs.Fields("Kurs").Value = 5 Then Form_frmEKZ.İfëâ37.Value = " 10"

```

End If

```

'-----grp

```

```

Dim param_grp As String
Dim param_kurs As String

```

```

param_grp = rs.Fields("G_Name").Value
param_kurs = rs.Fields("Kurs").Value

```

```

Form_frmEKZ.plGrp = param_grp
Form_frmEKZ.plkurs = param_kurs

```

```

'-----dis

```

```

Dim rst As ADODB.Recordset
Set rst = New ADODB.Recordset
Set rst = CurrentProject.Connection.Execute("SELECT Kod_Dis, Name_Rus FROM dbo.MAIN_Spr_dbo_Spr_Dis
WHERE ( Kod_Dis= " & Form_frmPriv.cmbdis.Value & ")")

```

```

Dim param_dis As String

```

```

param_dis = rst.Fields("Name_Rus").Value

```

```

Form_frmEKZ.pldis = param_dis

```

End Sub

Option Compare Database

Private Sub cmbdis_GotFocus()

Dim rs As ADODB.Recordset

Set rs = New ADODB.Recordset

```

Set rs = CurrentProject.Connection.Execute("SELECT G_Code, KodStSpec, Y_Create, Kurs FROM
dbo.MAIN_Spr_dbo_sprGrp WHERE (G_Code = " & cmbGrp.Value & ")")

```

Select Case Form_frmAut.tmp_par

Case 1

Select Case rs.Fields("kurs").Value

Case 1

```

cmbdis.RowSource = "SELECT dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis,
dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus FROM dbo.MAIN_Spr_dbo_Spr_Dis INNER JOIN
dbo.MAIN_Spr_dbo_Spr_Dis_Caf ON dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis =
dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Kod_Dis WHERE (dbo.MAIN_Spr_dbo_Spr_Dis.KodStSpec = " &
rs.Fields("KodStSpec") & ") AND (dbo.MAIN_Spr_dbo_Spr_Dis.GodPost = " & rs.Fields("Y_Create") & ") AND
(dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Semestr = 1) ORDER BY dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus"

```



```

dbo.MAIN_Spr_dbo_Spr_Dis_Caf ON dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis =
dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Kod_Dis WHERE (dbo.MAIN_Spr_dbo_Spr_Dis.KodStSpec = "" &
rs.Fields("KodStSpec") & "") AND (dbo.MAIN_Spr_dbo_Spr_Dis.GodPost = "" & rs.Fields("Y_Create") & "") AND
(dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Semestr = 2) ORDER BY dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus"

```

Case 2

```

cmbdis.RowSource = "SELECT dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis,
dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus FROM dbo.MAIN_Spr_dbo_Spr_Dis INNER JOIN
dbo.MAIN_Spr_dbo_Spr_Dis_Caf ON dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis =
dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Kod_Dis WHERE (dbo.MAIN_Spr_dbo_Spr_Dis.KodStSpec = "" &
rs.Fields("KodStSpec") & "") AND (dbo.MAIN_Spr_dbo_Spr_Dis.GodPost = "" & rs.Fields("Y_Create") & "") AND
(dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Semestr = 4) ORDER BY dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus"

```

Case 3

```

cmbdis.RowSource = "SELECT dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis,
dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus FROM dbo.MAIN_Spr_dbo_Spr_Dis INNER JOIN
dbo.MAIN_Spr_dbo_Spr_Dis_Caf ON dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis =
dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Kod_Dis WHERE (dbo.MAIN_Spr_dbo_Spr_Dis.KodStSpec = "" &
rs.Fields("KodStSpec") & "") AND (dbo.MAIN_Spr_dbo_Spr_Dis.GodPost = "" & rs.Fields("Y_Create") & "") AND
(dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Semestr = 6) ORDER BY dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus"

```

Case 4

```

cmbdis.RowSource = "SELECT dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis,
dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus FROM dbo.MAIN_Spr_dbo_Spr_Dis INNER JOIN
dbo.MAIN_Spr_dbo_Spr_Dis_Caf ON dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis =
dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Kod_Dis WHERE (dbo.MAIN_Spr_dbo_Spr_Dis.KodStSpec = "" &
rs.Fields("KodStSpec") & "") AND (dbo.MAIN_Spr_dbo_Spr_Dis.GodPost = "" & rs.Fields("Y_Create") & "") AND
(dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Semestr = 8) ORDER BY dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus"

```

Case 5

```

cmbdis.RowSource = "SELECT dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis,
dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus FROM dbo.MAIN_Spr_dbo_Spr_Dis INNER JOIN
dbo.MAIN_Spr_dbo_Spr_Dis_Caf ON dbo.MAIN_Spr_dbo_Spr_Dis.Kod_Dis =
dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Kod_Dis WHERE (dbo.MAIN_Spr_dbo_Spr_Dis.KodStSpec = "" &
rs.Fields("KodStSpec") & "") AND (dbo.MAIN_Spr_dbo_Spr_Dis.GodPost = "" & rs.Fields("Y_Create") & "") AND
(dbo.MAIN_Spr_dbo_Spr_Dis_Caf.Semestr = 10) ORDER BY dbo.MAIN_Spr_dbo_Spr_Dis.Name_Rus"

```

End Select

End Select

cmbdis.Requery

End Sub

Private Sub btnNext_Click()

If cmbGrp.Value = "" Then

MsgBox "Íáíáðíäèî ãûáðàòù äðóó!", vbInformation, "ÀÐÌ Óñíääââîñòù"

Exit Sub

End If

If cmbdis.Value = "" Then

MsgBox "Íáíáðíäèî ãûáðàòù äèñòèèéó!", vbInformation, "ÀÐÌ Óñíääââîñòù"

Exit Sub

End If

If IsNull(plDateProved.Value) = True Then

MsgBox "Íáíáðíäèî óèâçàòù äàòó!", vbInformation, "ÀÐÌ Óñíääââîñòù"

Exit Sub

End If

Dim rs As ADODB.Recordset

Set rs = New ADODB.Recordset

If Form_frmAut.tmp_par.Value = 1 Then

Set rs = CurrentProject.Connection.Execute(" SELECT dbo.tb_RK1.ID_Dis, dbo.tbStud.G_Code, dbo.tb_RK1.IKT


```
FROM dbo.tb_RK1 INNER JOIN dbo.tbStud ON dbo.tb_RK1.IKT = dbo.tbStud.IKT WHERE
(dbo.tb_RK1.ID_Dis = "" & cmbdis.Value & "") AND (dbo.tbStud.G_Code = "" & cmbGrp.Value & "")
```

```
If rs.RecordCount > 0 Then
```

```
MsgBox "Đăéòèiă ñ ààííé ãđóñă óăă âââââí!", vbInformation, "ÀĐÌ Óñiăâââiñnòü"
```

```
Exit Sub
```

```
End If
```

```
End If
```

```
If Form_frmAut.tmp_par.Value = 2 Then
```

```
Set rs = CurrentProject.Connection.Execute(" SELECT dbo.tb_RK2.ID_Dis, dbo.tbStud.G_Code, dbo.tb_RK2.IKT
FROM dbo.tb_RK2 INNER JOIN dbo.tbStud ON dbo.tb_RK2.IKT = dbo.tbStud.IKT WHERE
(dbo.tb_RK2.ID_Dis = "" & cmbdis.Value & "") AND (dbo.tbStud.G_Code = "" & cmbGrp.Value & "")")
```

```
If rs.RecordCount > 0 Then
```

```
MsgBox "Đăéòèiă ñ ààííé ãđóñă óăă âââââí!", vbInformation, "ÀĐÌ Óñiăâââiñnòü"
```

```
Exit Sub
```

```
End If
```

```
End If
```

```
If Form_frmAut.tmp_par.Value = 3 Then
```

```
Set rs = CurrentProject.Connection.Execute(" SELECT dbo.tb_RK3.ID_Dis, dbo.tbStud.G_Code, dbo.tb_RK3.IKT
FROM dbo.tb_RK3 INNER JOIN dbo.tbStud ON dbo.tb_RK3.IKT = dbo.tbStud.IKT WHERE
(dbo.tb_RK3.ID_Dis = "" & cmbdis.Value & "") AND (dbo.tbStud.G_Code = "" & cmbGrp.Value & "")")
```

```
If rs.RecordCount > 0 Then
```

```
MsgBox "Đăéòèiă ñ ààííé ãđóñă óăă âââââí!", vbInformation, "ÀĐÌ Óñiăâââiñnòü"
```

```
Exit Sub
```

```
End If
```

```
End If
```

```
If Form_frmAut.tmp_par.Value = 4 Then
```

```
Set rs = CurrentProject.Connection.Execute(" SELECT dbo.tb_RK4.ID_Dis, dbo.tbStud.G_Code, dbo.tb_RK4.IKT
FROM dbo.tb_RK4 INNER JOIN dbo.tbStud ON dbo.tb_RK4.IKT = dbo.tbStud.IKT WHERE
(dbo.tb_RK4.ID_Dis = "" & cmbdis.Value & "") AND (dbo.tbStud.G_Code = "" & cmbGrp.Value & "")")
```

```
If rs.RecordCount > 0 Then
```

```
MsgBox "Đăéòèiă ñ ààííé ãđóñă óăă âââââí!", vbInformation, "ÀĐÌ Óñiăâââiñnòü"
```

```
Exit Sub
```

```
End If
```

```
End If
```

```
Dim rst As ADODB.Recordset
```

```
Set rst = New ADODB.Recordset
```

```
Set rst = CurrentProject.Connection.Execute("SELECT G_Code, G_Name, Kurs FROM MAIN_Spr_dbo_sprGrp
WHERE ( G_Code= "" & Form_frmPriv.cmbGrp.Value & "")")
```

```
If Form_frmAut.tmp_par.Value = 5 Then
```

```
Set rs = CurrentProject.Connection.Execute(" SELECT dbo.tb_Ekzamen1.ID_Dis, dbo.tbStud.G_Code,
dbo.tb_Ekzamen1.IKT FROM dbo.tb_Ekzamen1 INNER JOIN dbo.tbStud ON dbo.tb_Ekzamen1.IKT =
dbo.tbStud.IKT WHERE (dbo.tb_Ekzamen1.ID_Dis = "" & cmbdis.Value & "") AND (dbo.tbStud.G_Code = "" &
cmbGrp.Value & "")")
```

```
If rs.RecordCount > 0 Then
```

```
MsgBox "Ýêçàîâòèííûă áăëëü ñ ààííé ãđóñă óăă âââââí!", vbInformation, "ÀĐÌ Óñiăâââiñnòü"
```

```
Exit Sub
```

```
End If
```

```
Set rs = CurrentProject.Connection.Execute("SELECT MAIN_Spr_dbo_sprGrp.G_Code,
```

```

MAIN_Spr_dbo_sprGrp.G_Name, tb_RK1.ID_Dis, tb_RK1.Ball, tb_RK2.Ball AS Expr1 FROM tb_RK2 INNER
JOIN tb_RK1 ON tb_RK2.ID_Dis = tb_RK1.ID_Dis AND tb_RK2.IKT = tb_RK1.IKT INNER JOIN tbStud ON
tb_RK2.IKT = tbStud.IKT INNER JOIN MAIN_Spr_dbo_sprGrp ON tbStud.G_Code =
MAIN_Spr_dbo_sprGrp.G_Code WHERE (MAIN_Spr_dbo_sprGrp.G_CODE = "" & cmbGrp.Value & "") And
(tb_RK1.ID_DIS = "" & cmbdis.Value & "") ORDER BY MAIN_Spr_dbo_sprGrp.G_Name")
If rs.RecordCount = 0 Then
    MsgBox "Ñíà÷àèà íáíáôíäèíí áíãñòè ðáéòèíäè!", vbCritical, "ÀÐÌ Óñíääáíñòù"
    Exit Sub
End If
'-----
If rst.Fields("Kurs").Value = 1 Then
    Form_frmPriv.tmp_kurs.Value = 1
End If
If rst.Fields("Kurs").Value = 2 Then
    Form_frmPriv.tmp_kurs.Value = 3
End If
If rst.Fields("Kurs").Value = 3 Then
    Form_frmPriv.tmp_kurs.Value = 5
End If
If rst.Fields("Kurs").Value = 4 Then
    Form_frmPriv.tmp_kurs.Value = 7
End If
If rst.Fields("Kurs").Value = 5 Then
    Form_frmPriv.tmp_kurs.Value = 9
End If

End If

If Form_frmAut.tmp_par.Value = 6 Then
Set rs = CurrentProject.Connection.Execute(" SELECT dbo.tb_Ekzamen2.ID_Dis, dbo.tbStud.G_Code,
dbo.tb_Ekzamen2.IKT FROM dbo.tb_Ekzamen2 INNER JOIN dbo.tbStud ON dbo.tb_Ekzamen2.IKT =
dbo.tbStud.IKT WHERE (dbo.tb_Ekzamen2.ID_Dis = "" & cmbdis.Value & "") AND (dbo.tbStud.G_Code = "" &
cmbGrp.Value & "")")
If rs.RecordCount > 0 Then
    MsgBox "Ýêçàíáíàòèíííúâ áäëü ñ äàííé äðóñâ óæâ ââääáíú!", vbInformation, "ÀÐÌ Óñíääáíñòù"
    Exit Sub
End If

Set rs = CurrentProject.Connection.Execute("SELECT MAIN_Spr_dbo_sprGrp.G_Code,
MAIN_Spr_dbo_sprGrp.G_Name, tb_RK3.ID_Dis, tb_RK3.Ball, tb_RK4.Ball AS Expr1 FROM tb_RK4 INNER
JOIN tb_RK3 ON tb_RK4.ID_Dis = tb_RK3.ID_Dis AND tb_RK4.IKT = tb_RK3.IKT INNER JOIN tbStud ON
tb_RK4.IKT = tbStud.IKT INNER JOIN MAIN_Spr_dbo_sprGrp ON tbStud.G_Code =
MAIN_Spr_dbo_sprGrp.G_Code WHERE (MAIN_Spr_dbo_sprGrp.G_CODE = "" & cmbGrp.Value & "") And
(tb_RK3.ID_DIS = "" & cmbdis.Value & "") ORDER BY MAIN_Spr_dbo_sprGrp.G_Name")
If rs.RecordCount = 0 Then
    MsgBox "Ñíà÷àèà íáíáôíäèíí áíãñòè ðáéòèíäè!", vbCritical, "ÀÐÌ Óñíääáíñòù"
    Exit Sub
End If

If rst.Fields("Kurs").Value = 1 Then
    Form_frmPriv.tmp_kurs.Value = 2
End If
If rst.Fields("Kurs").Value = 2 Then
    Form_frmPriv.tmp_kurs.Value = 4
End If
If rst.Fields("Kurs").Value = 3 Then
    Form_frmPriv.tmp_kurs.Value = 6
End If
If rst.Fields("Kurs").Value = 4 Then
    Form_frmPriv.tmp_kurs.Value = 8
End If
If rst.Fields("Kurs").Value = 5 Then

```

```
Form_frmPriv.tmp_kurs.Value = 10
End If
```

```
End If
```

```
Select Case Form_frmAut.tmp_par.Value
```

```
Case 1:
```

```
If plDateProved > Date Then
```

```
MsgBox "Óêàçàíà ÍÃ ÊÎÐÐÃÊÒÍÀß äàòà!", vbCritical, "ÀÐÌ Óñðåäåëèíîó"
```

```
Exit Sub
```

```
End If
```

```
Case 2:
```

```
If plDateProved > Date Then
```

```
MsgBox "Óêàçàíà ÍÃ ÊÎÐÐÃÊÒÍÀß äàòà!", vbCritical, "ÀÐÌ Óñðåäåëèíîó"
```

```
Exit Sub
```

```
End If
```

```
Case 3:
```

```
If plDateProved.Value < CDate("01.02." & Year(Date)) Or plDateProved.Value > Date Then
```

```
MsgBox "Óêàçàíà ÍÃ ÊÎÐÐÃÊÒÍÀß äàòà!", vbCritical, "ÀÐÌ Óñðåäåëèíîó"
```

```
Exit Sub
```

```
End If
```

```
Case 4:
```

```
If plDateProved > Date Then
```

```
MsgBox "Óêàçàíà ÍÃ ÊÎÐÐÃÊÒÍÀß äàòà!", vbCritical, "ÀÐÌ Óñðåäåëèíîó"
```

```
Exit Sub
```

```
End If
```

```
Case 5:
```

```
If plDateProved.Value > Date Then
```

```
MsgBox "Óêàçàíà ÍÃ ÊÎÐÐÃÊÒÍÀß äàòà!", vbCritical, "ÀÐÌ Óñðåäåëèíîó"
```

```
Exit Sub
```

```
End If
```

```
Case 6:
```

```
If plDateProved.Value < CDate("01.02." & Year(Date)) Or plDateProved.Value > Date Then
```

```
MsgBox "Óêàçàíà ÍÃ ÊÎÐÐÃÊÒÍÀß äàòà!", vbCritical, "ÀÐÌ Óñðåäåëèíîó"
```

```
Exit Sub
```

```
End If
```

```
End Select
```

```
If (Form_frmAut.tmp_par.Value = 1) Or (Form_frmAut.tmp_par.Value = 2) Or (Form_frmAut.tmp_par.Value = 3)
Or (Form_frmAut.tmp_par.Value = 4) Then
```

```
Set rs = CurrentProject.Connection.Execute("INSERT INTO dbo.tmp_pk (IKT, FIO, G_CODE, NZACH) SELECT
IKT, FIO, G_Code, NZach FROM dbo.tbStud WHERE (G_Code = " & cmbGrp.Value & ") ")
```

```
Set rs = CurrentProject.Connection.Execute("Update dbo.tmp_pk SET Date_Proved = " &
Form_frmPriv.plDateProved.Value & ", ID_PPS = " & Form_frmPriv.tmpUnicode.Value & ", ID_DIS = " &
Form_frmPriv.cmbdis.Value & ", Kol_kr = 0 WHERE (G_Code = " & cmbGrp.Value & ")")
```

```
DoCmd.OpenForm "frmRK"
```

```
Form_frmPriv.Visible = False
```

```
End If
```

```
If (Form_frmAut.tmp_par.Value = 5) Or (Form_frmAut.tmp_par.Value = 6) Then
```

```
Set rs = CurrentProject.Connection.Execute("INSERT INTO dbo.tmp_ekz (IKT, FIO, G_CODE, NZACH) SELECT
```

```
IKT, FIO, G_Code, NZach FROM dbo.tbStud WHERE (G_Code = "" & cmbGrp.Value & ""))
```

```
Set rs = CurrentProject.Connection.Execute("Update dbo.tmp_ekz SET Date_Proved = "" &  
Form_frmPriv.plDateProved.Value & "", ID_PPS = "" & Form_frmPriv.tmpUnicode.Value & "", ID_DIS = "" &  
Form_frmPriv.cmbdis.Value & "", Num_Semestr = "" & Form_frmPriv.tmp_kurs.Value & "", Kol_kr = 0 WHERE  
(G_Code = "" & cmbGrp.Value & ""))")
```

```
DoCmd.OpenForm "frmEKZ"  
Form_frmPriv.Visible = False  
End If
```

```
End Sub  
Private Sub cmbGrp_Change()  
cmbdis.Value = ""  
Dim rs As ADODB.Recordset  
Set rs = New ADODB.Recordset
```

```
Set rs = CurrentProject.Connection.Execute("select ikt from dbo.tbStud where G_Code = "" & cmbGrp.Value & """)  
If rs.RecordCount = 0 Then  
MsgBox "À áâãäåííë ãðóñâ íàò ñòóääíòíâ!", vbInformation, "ÀÐÌ Óñíääââíñòü"  
Exit Sub  
End If
```

```
Set rs = CurrentProject.Connection.Execute("SELECT G_Code, KodStSpec, Y_Create FROM  
dbo.MAIN_Spr_dbo_sprGrp WHERE (G_Code = "" & cmbGrp.Value & ""))")  
Set rs = CurrentProject.Connection.Execute("SELECT Kod_Dis, Name_Rus FROM dbo.MAIN_Spr_dbo_Spr_Dis  
WHERE (KodStSpec = "" & rs.Fields("KodStSpec") & "") AND (GodPost = "" & rs.Fields("Y_Create") & "")  
ORDER BY Name_Rus")  
If rs.RecordCount = 0 Then  
MsgBox "Äÿ ãðóñü " & cmbGrp.Text & " ññññ ãèñòèèèè ÍÄ ÄÄÄÄÄÍ!" & vbCrLf & "Íáðàòèòâñü â ìòääè  
ðääèñòðàòèè!", vbInformation, "ÀÐÌ Óñíääââíñòü"  
Exit Sub  
End If
```

```
cmbdis.Enabled = True  
End Sub  
Private Sub Form_Load()  
'ñíâíà ðàññèèääèè ìà èèðèèèèè  
If CurrentLayout <> Language.lngRussian Then  
ChangeLayout Language.lngRussian  
End If
```

```
Dim rs As ADODB.Recordset  
Set rs = New ADODB.Recordset  
Set rs = CurrentProject.Connection.Execute("SELECT Kod, UniCode, Fio, Imy, Otch FROM  
dbo.MAIN_Kadri_Vuz_dbo_BD WHERE ( UniCode="" & Form_frmAut.tmp_unicode.Value & ""))")
```

```
Dim param_fio As String  
Dim param_imy As String  
Dim param_otch As String  
Dim param_unic As String
```

```
param_unic = rs.Fields("UniCode").Value  
param_fio = rs.Fields("Fio").Value  
param_imy = rs.Fields("Imy").Value  
param_otch = rs.Fields("Otch").Value
```

```
Form_frmPriv.Form.Caption = "İÄÄİİÄÄÄÄÖÄËÜ: " & UCase(param_fio) & " " & UCase(param_imy) & " " &  
UCase(param_otch) & "!"  
Form_frmPriv.tmp_fio.Value = param_fio & " " & param_imy & " " & param_otch  
Form_frmPriv.tmpUnicode.Value = param_unic
```

```

End Sub
Private Sub otmena_Click()

If Form_frmAut.tmp_par.Value = 1 Or Form_frmAut.tmp_par.Value = 2 Or Form_frmAut.tmp_par.Value = 3 Or
Form_frmAut.tmp_par.Value = 4 Then

CommandBars("glMenu").Controls("ΔΟΑΑΑΕΙΥΕ ΕΙΙΟΔΙΕÜ_1").Enabled = False
CommandBars("glMenu").Controls("ΔΟΑΑΑΕΙΥΕ ΕΙΙΟΔΙΕÜ_2").Enabled = False
CommandBars("glMenu").Controls("ΔΟΑΑΑΕΙΥΕ ΕΙΙΟΔΙΕÜ_3").Enabled = False
CommandBars("glMenu").Controls("ΔΟΑΑΑΕΙΥΕ ΕΙΙΟΔΙΕÜ_4").Enabled = False
CommandBars("glMenu").Controls("ΥΕÇÀÌÁ_1").Enabled = False
CommandBars("glMenu").Controls("ΥΕÇÀÌÁ_2").Enabled = False
DoCmd.Close
Form_frm_Vybor.Visible = True
Else
CommandBars("glMenu").Controls("ΔΟΑΑΑΕΙΥΕ ΕΙΙΟΔΙΕÜ_1").Enabled = False
CommandBars("glMenu").Controls("ΔΟΑΑΑΕΙΥΕ ΕΙΙΟΔΙΕÜ_2").Enabled = False
CommandBars("glMenu").Controls("ΔΟΑΑΑΕΙΥΕ ΕΙΙΟΔΙΕÜ_3").Enabled = True
CommandBars("glMenu").Controls("ΔΟΑΑΑΕΙΥΕ ΕΙΙΟΔΙΕÜ_4").Enabled = False
CommandBars("glMenu").Controls("ΥΕÇÀÌÁ_1").Enabled = False
CommandBars("glMenu").Controls("ΥΕÇÀÌÁ_2").Enabled = True
DoCmd.Close
End If

End Sub

Private Sub plDateProved_GotFocus()
MsgBox "Αúáâðèòà çíá÷áíèâ ààòù èññëüçóý èàèáíàäü!", vbInformation, "ÀÐÌ Óñíáááâíñòù"
End Sub

Private Sub plDateProved_KeyPress(KeyAscii As Integer)
KeyAscii = 0
MsgBox "Αúáâðèòà çíá÷áíèâ ààòù èññëüçóý èàèáíàäü!", vbInformation, "ÀÐÌ Óñíáááâíñòù"
Exit Sub

'MsgBox Asc(plDateProved.Value)
'Select Case Len(plDateProved.Text)
'Case 0: If (KeyAscii <> 8) Or (KeyAscii < 48) Or (KeyAscii > 57) Then KeyAscii = 0 'backspace = 8
'Case 1: If (KeyAscii <> 8) Or (KeyAscii < 48) Or (KeyAscii > 57) Then KeyAscii = 0 'backspace = 8
'Case 2: If (KeyAscii <> 8) And (KeyAscii <> 46) Then KeyAscii = 0 'backspace = 8 ". "=46
'Case 3: If (KeyAscii <> 8) Or (KeyAscii < 48) Or (KeyAscii > 57) Then KeyAscii = 0 'backspace = 8
'Case 4: If (KeyAscii <> 8) Or (KeyAscii < 48) Or (KeyAscii > 57) Then KeyAscii = 0 'backspace = 8
'Case 5: If (KeyAscii <> 8) And (KeyAscii <> 46) Then KeyAscii = 0 'backspace = 8 ". "=46
'Case 6 To 10: If (KeyAscii <> 8) Or (KeyAscii < 48) Or (KeyAscii > 57) Then KeyAscii = 0 'backspace = 8
'Case Else: KeyAscii = 0
'End Select
End Sub

Option Compare Database
Private Sub btnAdd_Click()
Dim rs As ADODB.Recordset
Set rs = New ADODB.Recordset
'-----add
stMsg = MsgBox("Αú áäèñòàèòàëüíí ðíòèòà ñíððáíèòù áíáñáííúâ äáííúâ?", vbYesNo + vbInformation, "ÀÐÌ
Óñíáááâíñòù")
Select Case stMsg
Case vbYes

Set rs = CurrentProject.Connection.Execute("SELECT IKT, G_CODE, ID_PPS, Ball, ID_DIS FROM dbo.tmp_pk
WHERE (Ball IS NULL) AND (G_CODE = '' & Form_frmPriv.cmbGrp & '') and (ID_DIS = '' &
Form_frmPriv.cmbdis.Value & '') and (ID_PPS = '' & Form_frmPriv.tmpUnicode.Value & '')")

```

If rs.RecordCount > 0 Then

MsgBox "Áäëü ïðñòääëäíú íá ó ãñãð ñòóääíòíâ!", vbCritical, "ÀÐÌ Õñíääääíñòù"

Exit Sub

End If

Set rs = CurrentProject.Connection.Execute("SELECT IKT, G_CODE, ID_PPS, Ball, ID_DIS FROM dbo.tmp_pk WHERE (Ball > 30) AND (G_CODE = "" & Form_frmPriv.cmbGrp & "") and (ID_DIS = "" & Form_frmPriv.cmbdis.Value & "") and (ID_PPS = "" & Form_frmPriv.tmpUnicode.Value & "")")

If rs.RecordCount > 0 Then

MsgBox "Ìðñòääëäíú íá êððääëòíúâ áäëü!" & vbCrLf & "Çíà÷áèà äîëæíí áúòù â àèàñçíâ ìò 0 âí 30!", vbCritical, "ÀÐÌ Õñíääääíñòù"

Exit Sub

End If

Set rs = CurrentProject.Connection.Execute("SELECT IKT, G_CODE, Ball, ID_PPS, ID_DIS FROM dbo.tmp_pk WHERE (Ball <> 0) AND (G_CODE = "" & Form_frmPriv.cmbGrp & "") and (ID_DIS = "" & Form_frmPriv.cmbdis.Value & "") and (ID_PPS = "" & Form_frmPriv.tmpUnicode.Value & "")")

If rs.RecordCount = 0 Then

stMsg = MsgBox("Ó ãñãð ñòóääíòíâ êîë÷ãñòâí ääëëâ ðääí ""0""!", vbOKCancel + vbInformation, "ÀÐÌ Õñíääääíñòù")

Select Case stMsg

Case vbOK

stMsg = MsgBox("Â ãäëñòäèòäëüí òìòèòâ ñòóääëòù äääääíúé ðäéòèâ?", vbYesNo + vbInformation, "ÀÐÌ Õñíääääíñòù")

Select Case stMsg

Case vbYes

Blsave = True

btnPrint.Enabled = True

btnPrint.SetFocus

btnAdd.Enabled = False

If Form_frmAut.tmp_par.Value = 1 Then

Set rs = CurrentProject.Connection.Execute("INSERT INTO dbo.tb_RK1 (IKT, Ball, ID_PPS, ID_Dis, Date_Proved, Kol_Kr) SELECT IKT, Ball, ID_PPS, ID_DIS, Date_Proved, Kol_kr FROM dbo.tmp_pk WHERE (G_CODE = "" & Form_frmPriv.cmbGrp & "") and (ID_DIS = "" & Form_frmPriv.cmbdis.Value & "") and (ID_PPS = "" & Form_frmPriv.tmpUnicode.Value & "")")

End If

If Form_frmAut.tmp_par.Value = 2 Then

Set rs = CurrentProject.Connection.Execute("INSERT INTO dbo.tb_RK2 (IKT, Ball, ID_PPS, ID_Dis, Date_Proved, Kol_Kr) SELECT IKT, Ball, ID_PPS, ID_DIS, Date_Proved, Kol_kr FROM dbo.tmp_pk WHERE (G_CODE = "" & Form_frmPriv.cmbGrp & "") and (ID_DIS = "" & Form_frmPriv.cmbdis.Value & "") and (ID_PPS = "" & Form_frmPriv.tmpUnicode.Value & "")")

End If

If Form_frmAut.tmp_par.Value = 3 Then

Set rs = CurrentProject.Connection.Execute("INSERT INTO dbo.tb_RK3 (IKT, Ball, ID_PPS, ID_Dis, Date_Proved, Kol_Kr) SELECT IKT, Ball, ID_PPS, ID_DIS, Date_Proved, Kol_kr FROM dbo.tmp_pk WHERE (G_CODE = "" & Form_frmPriv.cmbGrp & "") and (ID_DIS = "" & Form_frmPriv.cmbdis.Value & "") and (ID_PPS = "" & Form_frmPriv.tmpUnicode.Value & "")")

End If

If Form_frmAut.tmp_par.Value = 4 Then

Set rs = CurrentProject.Connection.Execute("INSERT INTO dbo.tb_RK4 (IKT, Ball, ID_PPS, ID_Dis, Date_Proved, Kol_Kr) SELECT IKT, Ball, ID_PPS, ID_DIS, Date_Proved, Kol_kr FROM dbo.tmp_pk WHERE (G_CODE = "" & Form_frmPriv.cmbGrp & "") and (ID_DIS = "" & Form_frmPriv.cmbdis.Value & "") and (ID_PPS = "" & Form_frmPriv.tmpUnicode.Value & "")")

```

End If

'-----delete
Set rs = CurrentProject.Connection.Execute("DELETE FROM dbo.tmp_pk WHERE (ID_DIS = "" &
Form_frmPriv.cmbdis.Value & "") and (G_CODE = "" & Form_frmPriv.cmbGrp.Value & "") and (ID_PPS = "" &
Form_frmPriv.tmpUnicode.Value & "")")

MsgBox "Ààííúâ òñîâøíî ñîððàíáíú!", vbInformation, "ÀÐÌ Óñîââââîñîù"

'DoCmd.Close
Exit Sub

Case vbNo
Exit Sub
End Select

Case vbCancel
Exit Sub
End Select

End If
Blsave = True
    btnPrint.Enabled = True
    btnPrint.SetFocus
    btnAdd.Enabled = False

If Form_frmAut.tmp_par.Value = 1 Then
Set rs = CurrentProject.Connection.Execute("INSERT INTO dbo.tb_RK1 (IKT, Ball, ID_PPS, ID_Dis,
Date_Proved, Kol_Kr) SELECT IKT, Ball, ID_PPS, ID_DIS, Date_Proved, Kol_kr FROM dbo.tmp_pk WHERE
(G_CODE = "" & Form_frmPriv.cmbGrp & "") and (ID_DIS = "" & Form_frmPriv.cmbdis.Value & "") and (ID_PPS
= "" & Form_frmPriv.tmpUnicode.Value & "")")
End If

If Form_frmAut.tmp_par.Value = 2 Then
Set rs = CurrentProject.Connection.Execute("INSERT INTO dbo.tb_RK2 (IKT, Ball, ID_PPS, ID_Dis,
Date_Proved, Kol_Kr) SELECT IKT, Ball, ID_PPS, ID_DIS, Date_Proved, Kol_kr FROM dbo.tmp_pk WHERE
(G_CODE = "" & Form_frmPriv.cmbGrp & "") and (ID_DIS = "" & Form_frmPriv.cmbdis.Value & "") and (ID_PPS
= "" & Form_frmPriv.tmpUnicode.Value & "")")
End If

If Form_frmAut.tmp_par.Value = 3 Then
Set rs = CurrentProject.Connection.Execute("INSERT INTO dbo.tb_RK3 (IKT, Ball, ID_PPS, ID_Dis,
Date_Proved, Kol_Kr) SELECT IKT, Ball, ID_PPS, ID_DIS, Date_Proved, Kol_kr FROM dbo.tmp_pk WHERE
(G_CODE = "" & Form_frmPriv.cmbGrp & "") and (ID_DIS = "" & Form_frmPriv.cmbdis.Value & "") and (ID_PPS
= "" & Form_frmPriv.tmpUnicode.Value & "")")
End If

If Form_frmAut.tmp_par.Value = 4 Then
Set rs = CurrentProject.Connection.Execute("INSERT INTO dbo.tb_RK4 (IKT, Ball, ID_PPS, ID_Dis,
Date_Proved, Kol_Kr) SELECT IKT, Ball, ID_PPS, ID_DIS, Date_Proved, Kol_kr FROM dbo.tmp_pk WHERE
(G_CODE = "" & Form_frmPriv.cmbGrp & "") and (ID_DIS = "" & Form_frmPriv.cmbdis.Value & "") and (ID_PPS
= "" & Form_frmPriv.tmpUnicode.Value & "")")
End If

'-----delete
Set rs = CurrentProject.Connection.Execute("DELETE FROM dbo.tmp_pk WHERE (ID_DIS = "" &
Form_frmPriv.cmbdis.Value & "") and (G_CODE = "" & Form_frmPriv.cmbGrp.Value & "") and (ID_PPS = "" &
Form_frmPriv.tmpUnicode.Value & "")")

MsgBox "Ààííúâ òñîâøíî ñîððàíáíú!", vbInformation, "ÀÐÌ Óñîââââîñîù"

```

```

'DoCmd.Close

Case vbNo

End Select

End Sub

Private Sub btnNazad_Click()
If Blsave = False Then
stMsg = MsgBox("Άύ άάέñòääèòääüí ðìòèòå ìòìáíèòü ââîâ äâííûð?", vbYesNo + vbInformation, "ÀÐÌ Óñîååââîñòü")
Select Case stMsg
    Case vbYes

'-----delete
Set rs = CurrentProject.Connection.Execute("DELETE FROM dbo.tmp_pk WHERE (ID_DIS = "" &
Form_frmPriv.cmbdis.Value & "") and (G_CODE = "" & Form_frmPriv.cmbGrp.Value & "") and (ID_PPS = "" &
Form_frmPriv.tmpUnicode.Value & "")")

DoCmd.Close
Case vbNo

End Select
Else
Form_frmPriv.cmbGrp.Value = ""
Form_frmPriv.cmbdis.Value = ""
Form_frmPriv.cmbdis.Enabled = False
Form_frmPriv.plDateProved = Date
DoCmd.Close
End If

End Sub

Private Sub btnPrint_Click()
On Error Resume Next
DoCmd.OpenReport "rpt_Reyting_Ved", acViewPreview
End Sub

Private Sub Form_Load()
Blsave = False

Dim rs As ADODB.Recordset
Set rs = New ADODB.Recordset
Set rs = CurrentProject.Connection.Execute("SELECT G_Code, G_Name, Kurs FROM MAIN_Spr_dbo_sprGrp
WHERE ( G_Code= "" & Form_frmPriv.cmbGrp.Value & "")")

If Form_frmAut.tmp_par.Value = 1 Or Form_frmAut.tmp_par.Value = 2 Then
Form_frmRK.Caption = "ÐÀÉÒÈÍÁÎÀß ÃÃÃÎÎÑÒÛ "" & Form_frmAut.tmp_par.Value
If rs.Fields("Kurs").Value = 1 Then Form_frmRK.Îëå37.Value = " 1"
If rs.Fields("Kurs").Value = 2 Then Form_frmRK.Îëå37.Value = " 3"
If rs.Fields("Kurs").Value = 3 Then Form_frmRK.Îëå37.Value = " 5"
If rs.Fields("Kurs").Value = 4 Then Form_frmRK.Îëå37.Value = " 7"
If rs.Fields("Kurs").Value = 5 Then Form_frmRK.Îëå37.Value = " 9"
End If

If Form_frmAut.tmp_par.Value = 3 Or Form_frmAut.tmp_par.Value = 4 Then
Form_frmRK.Caption = "ÐÀÉÒÈÍÁÎÀß ÃÃÃÎÎÑÒÛ "" & Form_frmAut.tmp_par.Value
If rs.Fields("Kurs").Value = 1 Then Form_frmRK.Îëå37.Value = " 2"
If rs.Fields("Kurs").Value = 2 Then Form_frmRK.Îëå37.Value = " 4"
If rs.Fields("Kurs").Value = 3 Then Form_frmRK.Îëå37.Value = " 6"
If rs.Fields("Kurs").Value = 4 Then Form_frmRK.Îëå37.Value = " 8"
If rs.Fields("Kurs").Value = 5 Then Form_frmRK.Îëå37.Value = " 10"

```


End If

```
'-----grp
Dim param_grp As String
Dim param_kurs As String
param_grp = rs.Fields("G_Name").Value
param_kurs = rs.Fields("Kurs").Value
Form_frmRK.plGrp = param_grp
Form_frmRK.plkurs = param_kurs
rs.Close
'-----dis
Dim rst As ADODB.Recordset
Dim param_dis As String
Set rst = New ADODB.Recordset
Set rst = CurrentProject.Connection.Execute("SELECT Kod_Dis, Name_Rus FROM dbo.MAIN_Spr_dbo_Spr_Dis
WHERE ( Kod_Dis= '" & Form_frmPriv.cmbdis.Value & "'")
param_dis = rst.Fields("Name_Rus").Value
Form_frmRK.pldis = param_dis
rst.Close
```

End Sub

Option Compare Database
Private Sub btnAdd_Click()

Dim rs As ADODB.Recordset
Set rs = New ADODB.Recordset

```
'-----add
stMsg = MsgBox("Âú ääéñòäèòäëüíí òíòèòä ñíððäíèòü äíâñáííúâ äàííúâ?", vbYesNo + vbInformation, "ÀÐÌ
Óñíââââíñòü")
Select Case stMsg
    Case vbYes
If CInt(Form_pfrm_RK_Povyshenie.txtBall_Povyshenie.Value) > 30 Then
    MsgBox "Çíà÷áíèä ðäéòèíââ íâ äíëæíí íðââúòàòü 30!", vbCritical, "ÀÐÌ Óñíââââíñòü"
    Form_pfrm_RK_Povyshenie.txtBall_Povyshenie.Value = 0
    Exit Sub
End If
```

```
""ÍÁ ÊÎÐÐÄÊÒÍÁ çíà÷áíèä ðäéòèíââ!"
If CInt(Form_pfrm_RK_Povyshenie.txtBall_Povyshenie.Value) < Form_pfrm_RK_Povyshenie.Ball.Value Then
    MsgBox "Íðíòââèâíú íâ èíððäèòíúâ ääëëú!" & vbCrLf & "Çíà÷áíèä äíëæíí áúòü â äèàñçíâ íò " &
Form_pfrm_RK_Povyshenie.Ball.Value & " äí 30!", vbCritical, "ÀÐÌ Óñíââââíñòü"
    Form_pfrm_RK_Povyshenie.txtBall_Povyshenie.Value = 0
    Exit Sub
End If
```

```
BlSave_Pov = True
btnPrint.Enabled = True
If Form_frmAut.tmp_par.Value = 1 Then
Set rs = CurrentProject.Connection.Execute("INSERT INTO dbo.tb_RK1_Povyshenie (IKT, ID_PPS, ID_Dis,
Date_BD, Date_Proved, Ball, Kol_Kr) SELECT IKT, ID_PPS, ID_Dis, Date_BD, Date_Proved, Ball, Kol_Kr
FROM dbo.tb_RK1 WHERE (IKT = '" & Form_frm_Povyshenie.cmbFIO.Value & "') AND (ID_Dis = '" &
Form_frm_Povyshenie.cmbdis.Value & "'")
Set rs = CurrentProject.Connection.Execute("UPDATE dbo.tb_RK1 SET ID_PPS = '" &
Form_frm_Povyshenie.tmpUnicode.Value & "', Ball = '" & Form_pfrm_RK_Povyshenie.txtBall_Povyshenie.Value
& "', Date_Proved = '" & Form_frm_Povyshenie.plDateProved.Value & "', Date_BD=''" & Now & "' WHERE (IKT
= '" & Form_frm_Povyshenie.cmbFIO.Value & "') AND (ID_Dis = '" & Form_frm_Povyshenie.cmbdis.Value &
"')")
End If
```

```

If Form_frmAut.tmp_par.Value = 2 Then
Set rs = CurrentProject.Connection.Execute("INSERT INTO dbo.tb_RK2_Povyshenie (IKT, ID_PPS, ID_Dis,
Date_BD, Date_Proved, Ball, Kol_Kr) SELECT IKT, ID_PPS, ID_Dis, Date_BD, Date_Proved, Ball, Kol_Kr
FROM dbo.tb_RK2 WHERE (IKT = "" & Form_frm_Povyshenie.cmbFIO.Value & "") AND (ID_Dis = "" &
Form_frm_Povyshenie.cmbdis.Value & "")")
Set rs = CurrentProject.Connection.Execute("UPDATE dbo.tb_RK2 SET ID_PPS = "" &
Form_frm_Povyshenie.tmpUnicode.Value & "", Ball = "" & Form_pfrm_RK_Povyshenie.txtBall_Povyshenie.Value
& "", Date_Proved = "" & Form_frm_Povyshenie.plDateProved.Value & "", Date_BD="" & Now & "" WHERE (IKT
= "" & Form_frm_Povyshenie.cmbFIO.Value & "") AND (ID_Dis = "" & Form_frm_Povyshenie.cmbdis.Value &
""))")
End If

```

```

If Form_frmAut.tmp_par.Value = 3 Then
Set rs = CurrentProject.Connection.Execute("INSERT INTO dbo.tb_RK3_Povyshenie (IKT, ID_PPS, ID_Dis,
Date_BD, Date_Proved, Ball, Kol_Kr) SELECT IKT, ID_PPS, ID_Dis, Date_BD, Date_Proved, Ball, Kol_Kr
FROM dbo.tb_RK3 WHERE (IKT = "" & Form_frm_Povyshenie.cmbFIO.Value & "") AND (ID_Dis = "" &
Form_frm_Povyshenie.cmbdis.Value & "")")
Set rs = CurrentProject.Connection.Execute("UPDATE dbo.tb_RK3 SET ID_PPS = "" &
Form_frm_Povyshenie.tmpUnicode.Value & "", Ball = "" & Form_pfrm_RK_Povyshenie.txtBall_Povyshenie.Value
& "", Date_Proved = "" & Form_frm_Povyshenie.plDateProved.Value & "", Date_BD="" & Now & "" WHERE (IKT
= "" & Form_frm_Povyshenie.cmbFIO.Value & "") AND (ID_Dis = "" & Form_frm_Povyshenie.cmbdis.Value &
""))")
End If

```

```

If Form_frmAut.tmp_par.Value = 4 Then
Set rs = CurrentProject.Connection.Execute("INSERT INTO dbo.tb_RK4_Povyshenie (IKT, ID_PPS, ID_Dis,
Date_BD, Date_Proved, Ball, Kol_Kr) SELECT IKT, ID_PPS, ID_Dis, Date_BD, Date_Proved, Ball, Kol_Kr
FROM dbo.tb_RK4 WHERE (IKT = "" & Form_frm_Povyshenie.cmbFIO.Value & "") AND (ID_Dis = "" &
Form_frm_Povyshenie.cmbdis.Value & "")")
Set rs = CurrentProject.Connection.Execute("UPDATE dbo.tb_RK4 SET ID_PPS = "" &
Form_frm_Povyshenie.tmpUnicode.Value & "", Ball = "" & Form_pfrm_RK_Povyshenie.txtBall_Povyshenie.Value
& "", Date_Proved = "" & Form_frm_Povyshenie.plDateProved.Value & "", Date_BD="" & Now & "" WHERE (IKT
= "" & Form_frm_Povyshenie.cmbFIO.Value & "") AND (ID_Dis = "" & Form_frm_Povyshenie.cmbdis.Value &
""))")
End If

```

```

'-----delete
'Set rs = CurrentProject.Connection.Execute("DELETE FROM dbo.tmp_pk WHERE (ID_DIS = "" &
Form_frm_Povyshenie.cmbdis.Value & "") and (G_CODE = "" & Form_frm_Povyshenie.cmbGrp.Value & "") and
(ID_PPS = "" & Form_frm_Povyshenie.tmpUnicode.Value & "")")

```

```

MsgBox "Άάίίúά όñĩãøíî ñîððáíáíú!", vbInformation, "ÀÐÌ Óñĩãääááîñòü"
btnPrint.SetFocus
btnAdd.Enabled = False
'DoCmd.Close
Exit Sub

```

```

Case vbNo
Exit Sub
End Select

```

```

If Form_frmAut.tmp_par.Value = 1 Then
Set rs = CurrentProject.Connection.Execute("INSERT INTO dbo.tb_RK1 (IKT, Ball, ID_PPS, ID_Dis,
Date_Proved, Kol_Kr) SELECT IKT, Ball, ID_PPS, ID_DIS, Date_Proved, Kol_kr FROM dbo.tmp_pk WHERE
(G_CODE = "" & Form_frm_Povyshenie.cmbGrp & "") and (ID_DIS = "" & Form_frm_Povyshenie.cmbdis.Value &
"")) and (ID_PPS = "" & Form_frm_Povyshenie.tmpUnicode.Value & "")")
End If

```

```

If Form_frmAut.tmp_par.Value = 2 Then
Set rs = CurrentProject.Connection.Execute("INSERT INTO dbo.tb_RK2 (IKT, Ball, ID_PPS, ID_Dis,
Date_Proved, Kol_Kr) SELECT IKT, Ball, ID_PPS, ID_DIS, Date_Proved, Kol_kr FROM dbo.tmp_pk WHERE

```

```
(G_CODE = "" & Form_frm_Povyshenie.cmbGrp & "") and (ID_DIS = "" & Form_frm_Povyshenie.cmbdis.Value &
"") and (ID_PPS = "" & Form_frm_Povyshenie.tmpUnicode.Value & ""))
End If
```

```
If Form_frmAut.tmp_par.Value = 3 Then
Set rs = CurrentProject.Connection.Execute("INSERT INTO dbo.tb_RK3 (IKT, Ball, ID_PPS, ID_Dis,
Date_Proved, Kol_Kr) SELECT IKT, Ball, ID_PPS, ID_DIS, Date_Proved, Kol_kr FROM dbo.tmp_pk WHERE
(G_CODE = "" & Form_frm_Povyshenie.cmbGrp & "") and (ID_DIS = "" & Form_frm_Povyshenie.cmbdis.Value &
"") and (ID_PPS = "" & Form_frm_Povyshenie.tmpUnicode.Value & ""))
End If
```

```
If Form_frmAut.tmp_par.Value = 4 Then
Set rs = CurrentProject.Connection.Execute("INSERT INTO dbo.tb_RK4 (IKT, Ball, ID_PPS, ID_Dis,
Date_Proved, Kol_Kr) SELECT IKT, Ball, ID_PPS, ID_DIS, Date_Proved, Kol_kr FROM dbo.tmp_pk WHERE
(G_CODE = "" & Form_frm_Povyshenie.cmbGrp & "") and (ID_DIS = "" & Form_frm_Povyshenie.cmbdis.Value &
"") and (ID_PPS = "" & Form_frm_Povyshenie.tmpUnicode.Value & ""))
End If
```

```
'-----delete
Set rs = CurrentProject.Connection.Execute("DELETE FROM dbo.tmp_pk WHERE (ID_DIS = "" &
Form_frm_Povyshenie.cmbdis.Value & "") and (G_CODE = "" & Form_frm_Povyshenie.cmbGrp.Value & "") and
(ID_PPS = "" & Form_frm_Povyshenie.tmpUnicode.Value & ""))")
```

```
MsgBox "Ääíüâ òñîåøí ñîðäíáíü!", vbInformation, "ÀÐÌ Óñîääâîñîù"
btnPrint.SetFocus
btnAdd.Enabled = False
```

```
'DoCmd.Close
```

```
End Sub
```

```
Private Sub btnNazad_Click()
If BISave_Pov = False Then
stMsg = MsgBox("Äü ääéñòâèðäëüí ðìðèðà ìòìáíèòù äââä ääíüð?", vbYesNo + vbInformation, "ÀÐÌ Óñîääâîñîù")
Select Case stMsg
Case vbYes
```

```
'-----delete
Set rs = CurrentProject.Connection.Execute("DELETE FROM dbo.tmp_pk WHERE (ID_DIS = "" &
Form_frm_Povyshenie.cmbdis.Value & "") and (G_CODE = "" & Form_frm_Povyshenie.cmbGrp.Value & "") and
(ID_PPS = "" & Form_frm_Povyshenie.tmpUnicode.Value & ""))")
Form_frm_Povyshenie.cmbGrp.Value = ""
Form_frm_Povyshenie.cmbdis.Value = ""
Form_frm_Povyshenie.cmbFIO.Value = ""
Form_frm_Povyshenie.cmbdis.Enabled = False
Form_frm_Povyshenie.plDateProved = Date
DoCmd.Close
Case vbNo
```

```
End Select
Else
Form_frm_Povyshenie.cmbGrp.Value = ""
Form_frm_Povyshenie.cmbdis.Value = ""
Form_frm_Povyshenie.cmbFIO.Value = ""
Form_frm_Povyshenie.cmbdis.Enabled = False
Form_frm_Povyshenie.plDateProved = Date
DoCmd.Close
End If
End Sub
```

```
Private Sub btnPrint_Click()
```

```

On Error Resume Next
DoCmd.OpenReport "rpt_Reyting_Ved_Povyshenie", acViewPreview
End Sub

Private Sub Form_Load()

Dim rs As ADODB.Recordset
Set rs = New ADODB.Recordset
Set rs = CurrentProject.Connection.Execute("SELECT G_Code, G_Name, Kurs FROM MAIN_Spr_dbo_sprGrp
WHERE ( G_Code=" & Form_frm_Povyshenie.cmbGrp.Value & ")")

If Form_frmAut.tmp_par.Value = 1 Then
Form_frmRK_Povyshenie.Caption = "ÐÀÉÒÈÍÀÎÀß ÂÄÏÎÑÒÜ '1"

If rs.Fields("Kurs").Value = 1 Then Form_frmRK_Povyshenie.Îëå37.Value = " 1"
If rs.Fields("Kurs").Value = 2 Then Form_frmRK_Povyshenie.Îëå37.Value = " 3"
If rs.Fields("Kurs").Value = 3 Then Form_frmRK_Povyshenie.Îëå37.Value = " 5"
If rs.Fields("Kurs").Value = 4 Then Form_frmRK_Povyshenie.Îëå37.Value = " 7"
If rs.Fields("Kurs").Value = 5 Then Form_frmRK_Povyshenie.Îëå37.Value = " 9"

End If

If Form_frmAut.tmp_par.Value = 2 Then
Form_frmRK_Povyshenie.Caption = "ÐÀÉÒÈÍÀÎÀß ÂÄÏÎÑÒÜ '2"

If rs.Fields("Kurs").Value = 1 Then Form_frmRK_Povyshenie.Îëå37.Value = " 1"
If rs.Fields("Kurs").Value = 2 Then Form_frmRK_Povyshenie.Îëå37.Value = " 3"
If rs.Fields("Kurs").Value = 3 Then Form_frmRK_Povyshenie.Îëå37.Value = " 5"
If rs.Fields("Kurs").Value = 4 Then Form_frmRK_Povyshenie.Îëå37.Value = " 7"
If rs.Fields("Kurs").Value = 5 Then Form_frmRK_Povyshenie.Îëå37.Value = " 9"

End If

If Form_frmAut.tmp_par.Value = 3 Then
Form_frmRK_Povyshenie.Caption = "ÐÀÉÒÈÍÀÎÀß ÂÄÏÎÑÒÜ '3"

If rs.Fields("Kurs").Value = 1 Then Form_frmRK_Povyshenie.Îëå37.Value = " 2"
If rs.Fields("Kurs").Value = 2 Then Form_frmRK_Povyshenie.Îëå37.Value = " 4"
If rs.Fields("Kurs").Value = 3 Then Form_frmRK_Povyshenie.Îëå37.Value = " 6"
If rs.Fields("Kurs").Value = 4 Then Form_frmRK_Povyshenie.Îëå37.Value = " 8"
If rs.Fields("Kurs").Value = 5 Then Form_frmRK_Povyshenie.Îëå37.Value = " 10"

End If

If Form_frmAut.tmp_par.Value = 4 Then
Form_frmRK_Povyshenie.Caption = "ÐÀÉÒÈÍÀÎÀß ÂÄÏÎÑÒÜ '4"

If rs.Fields("Kurs").Value = 1 Then Form_frmRK_Povyshenie.Îëå37.Value = " 2"
If rs.Fields("Kurs").Value = 2 Then Form_frmRK_Povyshenie.Îëå37.Value = " 4"
If rs.Fields("Kurs").Value = 3 Then Form_frmRK_Povyshenie.Îëå37.Value = " 6"
If rs.Fields("Kurs").Value = 4 Then Form_frmRK_Povyshenie.Îëå37.Value = " 8"
If rs.Fields("Kurs").Value = 5 Then Form_frmRK_Povyshenie.Îëå37.Value = " 10"

End If

'-----grp

Dim param_grp As String
Dim param_kurs As String

param_grp = rs.Fields("G_Name").Value

```

```

param_kurs = rs.Fields("Kurs").Value

Form_frmRK_Povyshenie.plGrp = param_grp
Form_frmRK_Povyshenie.plkurs = param_kurs

rs.Close
'-----dis
Dim rst As ADODB.Recordset
Set rst = New ADODB.Recordset
Set rst = CurrentProject.Connection.Execute("SELECT Kod_Dis, Name_Rus FROM dbo.MAIN_Spr_dbo_Spr_Dis
WHERE ( Kod_Dis= '" & Form_frm_Povyshenie.cmbdis.Value & "'")

Dim param_dis As String

param_dis = rst.Fields("Name_Rus").Value

Form_frmRK_Povyshenie.pldis = param_dis

rst.Close

'With Form_pfrm_RK_Povyshenie
' .RecordSource = ""
' .InputParameters = "@IKT=FORMS!frm_Povyshenie!cmbFIO.value,@ID_Dis=FORMS!frm_Povyshenie!
cmbDis.value"
' Select Case Form_frmAut.tmp_par
' Case 1:
' .RecordSource = "sp_RK1_Povyshenie"
' Case 2:
' .RecordSource = "sp_RK2_Povyshenie"
' Case 3:
' .InputParameters = "@IKT=FORMS!frm_Povyshenie!cmbFIO.value,@ID_Dis=FORMS!frm_Povyshenie!
cmbDis.value"
' .RecordSource = "sp_RK3_Povyshenie"
' Case 4:
' .RecordSource = "sp_RK4_Povyshenie"
' End Select
' .Requery
'End With
End Sub

Option Compare Database

Private Sub Ball_Change()
Dim rs As ADODB.Recordset
Set rs = New ADODB.Recordset

If Form_frmAut.tmp_par.Value = 5 Then
Set rs = CurrentProject.Connection.Execute("SELECT dbo.tb_RK1.Ball + dbo.tb_RK2.Ball AS Expr1 FROM
dbo.tb_RK1 INNER JOIN dbo.tb_RK2 ON dbo.tb_RK1.IKT = dbo.tb_RK2.IKT WHERE (dbo.tb_RK1.Ball +
dbo.tb_RK2.Ball < 30) AND (dbo.tb_RK1.ID_Dis = '" & Form_frmPriv.cmbdis.Value & "') AND (dbo.tb_RK1.IKT
= '" & IKT.Value & "')")
If rs.RecordCount > 0 Then
MsgBox "Νόóääíò " & FIO.Value & " íâ äîróúâí ê ýêçàíáíó!", vbCritical, "ÀÐÌ Óñîääääîñòù"
Ball.Value = 0
End If
End If

If Form_frmAut.tmp_par.Value = 6 Then
Set rs = CurrentProject.Connection.Execute("SELECT dbo.tb_RK3.Ball + dbo.tb_RK4.Ball AS Expr1 FROM
dbo.tb_RK3 INNER JOIN dbo.tb_RK4 ON dbo.tb_RK3.IKT = dbo.tb_RK4.IKT WHERE (dbo.tb_RK3.Ball +
dbo.tb_RK4.Ball < 30) AND (dbo.tb_RK3.ID_Dis = '" & Form_frmPriv.cmbdis.Value & "') AND (dbo.tb_RK3.IKT

```

```

= "" & IKT.Value & "" )")
    If rs.RecordCount > 0 Then
        MsgBox "Ñòóääíò " & FIO.Value & " íâ ãñíóùáí ê ýêçàíáíó!", vbCritical, "ÀÐÌ Óñíáääáíññòù"
        Ball.Value = 0
    End If
End If

End Sub

Private Sub Ball_GotFocus()
    Dim rs As ADODB.Recordset
    Set rs = New ADODB.Recordset

    If Form_frmAut.tmp_par.Value = 5 Then
        Set rs = CurrentProject.Connection.Execute("SELECT dbo.tb_RK1.Ball + dbo.tb_RK2.Ball AS Expr1 FROM
        dbo.tb_RK1 INNER JOIN dbo.tb_RK2 ON dbo.tb_RK1.IKT = dbo.tb_RK2.IKT WHERE (dbo.tb_RK1.Ball +
        dbo.tb_RK2.Ball < 30) AND (dbo.tb_RK1.ID_Dis = "" & Form_frmPriv.cmbdis.Value & "") AND (dbo.tb_RK1.IKT
        = "" & IKT.Value & "" )")
        If rs.RecordCount > 0 Then
            MsgBox "Ñòóääíò " & FIO.Value & " íâ ãñíóùáí ê ýêçàíáíó!", vbCritical, "ÀÐÌ Óñíáääáíññòù"
            Ball.Value = 0
        End If
    End If

    If Form_frmAut.tmp_par.Value = 6 Then
        Set rs = CurrentProject.Connection.Execute("SELECT dbo.tb_RK3.Ball + dbo.tb_RK4.Ball AS Expr1 FROM
        dbo.tb_RK3 INNER JOIN dbo.tb_RK4 ON dbo.tb_RK3.IKT = dbo.tb_RK4.IKT WHERE (dbo.tb_RK3.Ball +
        dbo.tb_RK4.Ball < 30) AND (dbo.tb_RK3.ID_Dis = "" & Form_frmPriv.cmbdis.Value & "") AND (dbo.tb_RK3.IKT
        = "" & IKT.Value & "" )")
        If rs.RecordCount > 0 Then
            MsgBox "Ñòóääíò " & FIO.Value & " íâ ãñíóùáí ê ýêçàíáíó!", vbCritical, "ÀÐÌ Óñíáääáíññòù"
            Ball.Value = 0
        End If
    End If

End Sub

Private Sub Ball_KeyPress(KeyAscii As Integer)
    If ((KeyAscii <> 8) And (KeyAscii < 48)) Or (KeyAscii > 57) Then KeyAscii = 0 'backspase = 8
End Sub
Private Sub Ball_LostFocus()
    If Ball.Value > 40 Then
        Ball.Text = ""

        MsgBox "Çíà÷áíèâ ýêçàíáíàöèíííâí áàèèâ íâ ãñíóùáí ðääâùøàòù 40!", vbCritical, "ÀÐÌ Óñíáääáíññòù"
        Ball.Value = 0

    End Sub

Exit Sub

End If

End Sub

```