

Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

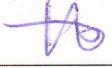
Кафедра математической
кибернетики и компьютерных наук

РАЗРАБОТКА БАЗЫ ДАННЫХ «АБИТУРИЕНТ»

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 451 группы
направления 09.03.04 – Программная инженерия
факультета КНиИТ
Евсеева Кирилла Дмитриевича

Научный руководитель
доцент, к. ф.-м. н.


10.06.17

А. С. Иванова

Заведующий кафедрой
к. ф.-м. н.


10.06.17

С. В. Миронов

Саратов 2017

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Теоретические основы и правила приема на факультет КНиИТ ФГ- БОУ ВО «СГУ имени Н. Г. Чернышевского»	4
2 Разработка элементов базы данных «Абитуриент»	6
2.1 Схема базы данных	6
2.2 Процедуры манипуляции данными	9
2.3 Вспомогательные процедуры	9
2.4 Процедуры выборок абитуриентов, поступающих по конкурсу ...	11
2.5 Черновое редактирование	14
2.6 Пользовательский интерфейс	14
ЗАКЛЮЧЕНИЕ	15
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	16

ВВЕДЕНИЕ

При приеме абитуриентов, высшее учебное заведение нуждается в учете большого количества данных. Сложная структура правил создает множество проблем, когда требуется составлять списки приказов, содержащих поступивших в состав студентов абитуриентов.

Целью бакалаврской работы является разработка базы данных для ведения отчетности по абитуриентам, поступающим на факультет КНиИТ СГУ.

Актуальность работы заключается в необходимости вести учет поступающих абитуриентов, а также сложностью сортировки абитуриентов в зависимости от их направлений, баллов ЕГЭ и правил приема в ВУЗ.

Были поставлены следующие задачи:

1. В соответствии с правилами приема разработать схему базы данных.
2. Разработать процедуры для внесения в базу новых абитуриентов (бакалавров, магистров, аспирантов).
3. Разработать процедуры для обновления абитуриентов (бакалавров, магистров, аспирантов).
4. Разработать процедуры для удаления абитуриентов (бакалавров, магистров, аспирантов).
5. Разработать процедуры для выборки студентов, используя различные сортировки.
6. Разработать процедуры для внесения в базу приказов о зачислении.
7. Разработать пользовательский интерфейс для работы с базой данных.
8. Разработать процедуры для чернового редактирования.

В структуру бакалаврской работы входят: введение, две главы («Теоретические основы технологий и правил, применяемых при разработке проекта» и «Разработка элементов базы данных «Абитуриент»: схема, процедуры, интерфейс»), заключение.

Первая глава содержит описание основных технологий, используемых при выполнении работы, а также правила приема в ВУЗ.

Вторая глава содержит описание разработанной базы данных, а также процедур, выполняющих рейтинговые выборки и манипуляцию данными, и пользовательского интерфейса.

1 Теоретические основы и правила приема на факультет КНиИТ ФГБОУ ВО «СГУ имени Н. Г. Чернышевского»

Реляционная модель данных (РМД) — логическая модель данных, прикладная теория построения баз данных, которая является приложением к задачам обработки данных таких разделов математики, как теория множеств и логика первого порядка [1, 2].

Базой данных (БД) называется организованная в соответствии с определенными правилами и поддерживаемая в памяти компьютера совокупность сведений об объектах, процессах, событиях или явлениях, относящихся к некоторой предметной области, теме или задаче. Она организована таким образом, чтобы обеспечить информационные потребности пользователей, а также удобное хранение этой совокупности данных, как в целом, так и любой ее части [3–5].

Реляционная база данных представляет собой множество взаимосвязанных таблиц, каждая из которых содержит информацию об объектах определенного вида. Каждая строка таблицы содержит данные об одном объекте (например, автомобиле, компьютере, клиенте), а столбцы таблицы содержат различные характеристики этих объектов — атрибуты (например, номер двигателя, марка процессора, телефоны фирм или клиентов) [6, 7].

Microsoft SQL Server — система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка [8–10].

В работе используется версия Microsoft SQL Server 2014.

Платформа Microsoft SQL Server 2014 обеспечивает управление данными бизнес-класса для различных приложений баз данных, предоставляя необходимые пользователям надежность, время безотказной работы, безопасность и управляемость при низкой стоимости владения [11].

При приеме абитуриентов, СГУ руководствуется правилами приема, уставом ВУЗа и опирается на лицензию на осуществление образовательной деятельности и свидетельство о государственной аккредитации вуза [12–14].

Для достижения поставленных целей в работе имеет смысл сконцентрироваться на определенных пунктах и положениях правил приема в ВУЗ, а не на всех.

1. Разделение поступающих без экзаменов, по квоте, по целевому направлению, по общему конкурсу на разные группы [15, 16].
2. Разделение очного и заочного обучения [17, 18].
3. Разделение платного и бесплатного обучения [19].
4. Абитуриент после занесения в приказ может участвовать в конкурсе далее [20].
5. При составлении приказа сортировка абитуриентов — без экзаменов, квота, целевое направление, конкурс.

2 Разработка элементов базы данных «Абитуриент»

2.1 Схема базы данных

В соответствии с поставленными целями, была разработана схема базы данных. Для хранения данных об абитуриентах используются таблицы *scc.Person*, включающая в себя характеристики присущие всем абитуриентам. Таблицы *scc.Bachelor*, *scc.Master*, *scc.PostgraduateStudent* имеют отношение один к одному с таблицей *scc.Person* и включают в себя уникальные черты присущие бакалаврам, магистрам и аспирантам соответственно.

Так как необходимо знать какую школу или университет закончил абитуриент, созданы таблицы *scc.School* и *scc.University*. Связь между таблицами *scc.School* и *scc.Bachelor*, *scc.University* и *scc.Master*, *scc.University* и *scc.PostgraduateStudent* — один к многим.

Также необходимо вести учет количества подавших документы на поступление и поступивших абитуриентов по городам и школам. Для этого создана таблица *scc.City*. Связь между таблицами *scc.City* и *scc.School*, *scc.City* и *scc.Person* — один к многим.

Таблица *scc.MethodofAdmission* создана в целях нормализации базы и содержит в себе способы поступления в ВУЗ. Связь между таблицами *scc.MethodofAdmission* и *scc.Person* — один к многим.

Зачастую необходимо иметь возможность связаться с родственниками абитуриента. Для этого создана таблица *scc.Parent*. Связь между таблицами *scc.Parent* и *scc.Person* — один к многим.

Абитуриент при поступлении может подать документы на поступление сразу на несколько направлений. С помощью таблиц *scc.Direction* и *scc.Priority* реализована связь многие к многим между таблицами *scc.Direction* и *scc.Person*.

Для вынесения приказов о зачислении созданы таблицы *scc.Decree* и *scc.Decree_Person*. Так была реализована связь многие к многим между таблицами *scc.Decree* и *scc.Person*.

Схема базы данных, представлена на рисунке 1.

Также для задач, связанных с черновым редактированием и просмотром удаленных абитуриентов создано несколько таблиц, изображенных на рисунке 2.



Рисунок 1 – Схема базы данных

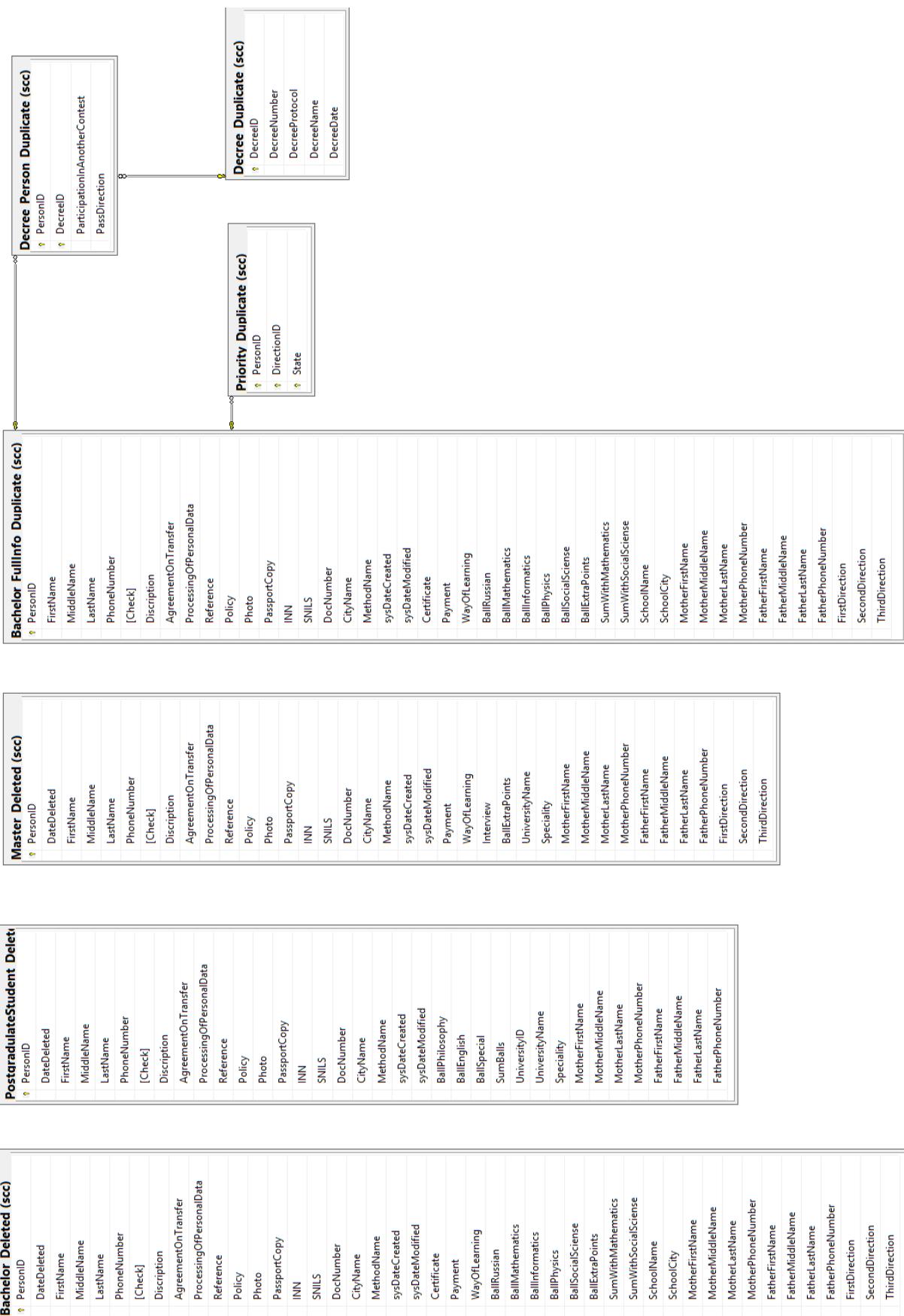


Рисунок 2 – Остальные таблицы

В целях безопасности была создана схема scc (Selection Committee CSIT).

2.2 Процедуры манипуляции данными

Процедуры *Insert*, *Update*, *Delete* предназначены для заполнения данных об абитуриентах и корректировки или удаления этих данных.

Сначала идет проверка правильности вводимых параметров. Эта часть заключена в *Begin Try*, *End Try*. Если какой-либо параметр не соответствует требованиям, то будет выведено соответствующее сообщение об ошибке.

Необходимо вести статистику абитуриентов по городам, школам, университетам. Для этого нам необходимо заносить соответствующие данные. Однако не имеет смысла хранить в базе данных сразу весь список городов, школ, университетов. Поэтому при занесении, обновлении абитуриента, происходит проверка на наличие в базе данных соответствующих городов, школ, университетов, и если их нет, то в базу вносятся в верхнем регистре и без пробелов новые данные, требуемые для данной статистики.

Так как добавление, обновление, удаление абитуриента затрагивает несколько связанных таблиц, то процедура имеет явное указание начала и конца транзакции, чтобы при возникновении ошибок происходил откат измененных данных.

Процедуры *Insert*, *Update*, *Delete* для бакалавров, магистров и аспирантов имеют идентичную структуру.

2.3 Вспомогательные процедуры

Вспомогательными процедурами, не имеющими сложную структуру, являются процедуры:

1. Алфавитный список.
2. Рейтинговая таблица.
3. Количество подавших документы на поступление и поступивших по школам.
4. Количество подавших документы на поступление и поступивших по городам.
5. Рейтинговый список абитуриентов, сдававших обществознание

Процедура *usp_Alphabet_list* производит выборку из базы данных абитуриентов в алфавитном порядке. Имеет три параметра логического типа

данных, позволяющие выбирать список из бакалавров, магистров и аспирантов соответственно.

Процедура *usp_Bachelor_Rank* имеет три параметра:

1. Количество поступающих без сдачи экзаменов (если количество абитуриентов, поступающих без экзаменов, больше чем данный параметр, люди с худшим рейтингом баллов попадают в группу поступающих по конкурсу).
2. Количество поступающих по квоте (если количество абитуриентов, поступающих по квоте, больше чем данный параметр, люди с худшим рейтингом баллов попадают в группу поступающих по конкурсу).
3. Количество поступающих по целевому направлению (если количество абитуриентов, поступающих по целевому направлению, больше чем данный параметр, люди с худшим рейтингом баллов попадают в группу поступающих по конкурсу).

Процедура производит сортировку абитуриентов (бакалавров) согласно их набранным баллам (порядок сортировки: сумма баллов, баллы математика, баллы информатика, ФИО), определяя позицию в рейтинге для каждого участника по каждому направлению, на которые подал документы абитуриент. Высшее место в списке имеют абитуриенты, поступающие без экзаменов, далее идут абитуриенты, поступающие по квоте, после них абитуриенты, поступающие по целевому направлению, а в конце оставшиеся абитуриенты, входящие в группу поступающих по конкурсу.

Процедуры *usp_Bachelor_CityPeopleCount*, *usp_Bachelor_SchoolPeopleCount* параметров не имеют и выполняют выборки количества подавших документов и поступивших по городам и школам.

Процедура *usp_Bachelor_SocialSciense* также параметров не имеет и осуществляет выборку абитуриентов-бакалавров, сдававших ЕГЭ по обществознанию.

Так как помимо поступающих по общему конкурсу абитуриентов, есть поступающие без экзаменов, по квоте, по целевому направлению требовалось создать процедуры для соответствующих выборок и заполнения приказов.

Абитуриент после занесения в приказ, может участвовать в конкурсе в дальнейшем по другому направлению.

Для этого есть отдельная процедура *usp_Bachelor_ParticipationIn*

AnotherContest. Она меняет колонку логического типа данных *Participation InAnotherContest* таблицы *scc.Decree_Person* того абитуриента, для которого применяется.

Процедуры выборок, например *usp_Bachelor_NoExams_Select*, отличаются от процедур внесения данных абитуриентов в приказ, например *usp_Bachelor_NoExams_DecreeInsert* тем, что первая просто делает выборку, а вторая еще заполняет приказы теми же абитуриентами, что выбираются в первой.

2.4 Процедуры выборок абитуриентов, поступающих по конкурсу

Данные процедуры обладают самой сложной структурой, так как нам требуется учитывать очень большое количество вещей:

- отсечь людей, поступающих не по конкурсу
- отсечь людей, присутствующих в приказах и не желающих участвовать в конкурсе в дальнейшем
- учитывать приоритеты направлений, подавших документы абитуриентов
- учитывать какие документы подал абитуриент (оригинал или копию)
- разделение очников и заочников

В начале происходит типичная проверка на корректность введенных параметров.

Далее создается временная таблица *#TmpSelectedDirection*, туда заносятся направления, которые будут участвовать в выборке.

Также заполняется таблица *#TmpKonkurs*, содержащая отсортированные данные согласно регламенту (сумма баллов с информатикой, баллы математика, баллы информатика, ФИО), исключая абитуриентов из *#TmpNoExams*, *#TmpKvota*, *#TmpCelevoe*.

Создается таблица *#CurrentRank*, которая в дальнейшем и будет содержать рейтинговый список согласно приоритетам и баллам.

Затем создается таблица *#BlackListDirection* содержащая направления, выборка по которым далее не осуществляется, так как на них более нет свободных мест, а также вставляются данные в таблицу *scc.Decree*.

Потом начинается цикл, пока согласно введенным данным есть свободные места, и есть хотя бы один абитуриент с направлениями, не содержащи-

мися в $\#BlackListDirection$, мы выбираем верхнего в списке из $\#TmpKonkurs$, согласно оставшимся местам и направлениям.

Далее мы производим выборку направлений, вставленного абитуриента в $\#CurrentRank$.

В зависимости от того, по какому направлению проходит данный абитуриент, заполняется в колонка $PassDirection$ таблицы $\#CurrentRank$, а также уменьшаем параметр, отвечающий за количество оставшихся мест на данном направлении.

Если один из параметров $@PI, @KB, @FIIT, @MOAIS, @IVT, @SAU$ становится равен 0, мы заносим их в $\#BlackListDIrection$.

В конце происходит удаление всех временных таблиц, а также удаляются записи из $scc.Decree_Person$ для тех абитуриентов, что уже присутствовали в каком-либо приказе, однако участвовали в конкурсе дальше и смогли попасть в другой приказ. А также таблица $scc.Decree_Person$ заполняется новыми данными, согласно выборке из $\#CurrentRank$, ограниченной параметром $@TotalQuantity$.

Алгоритм работы данной процедуры представлен на рисунке 3.

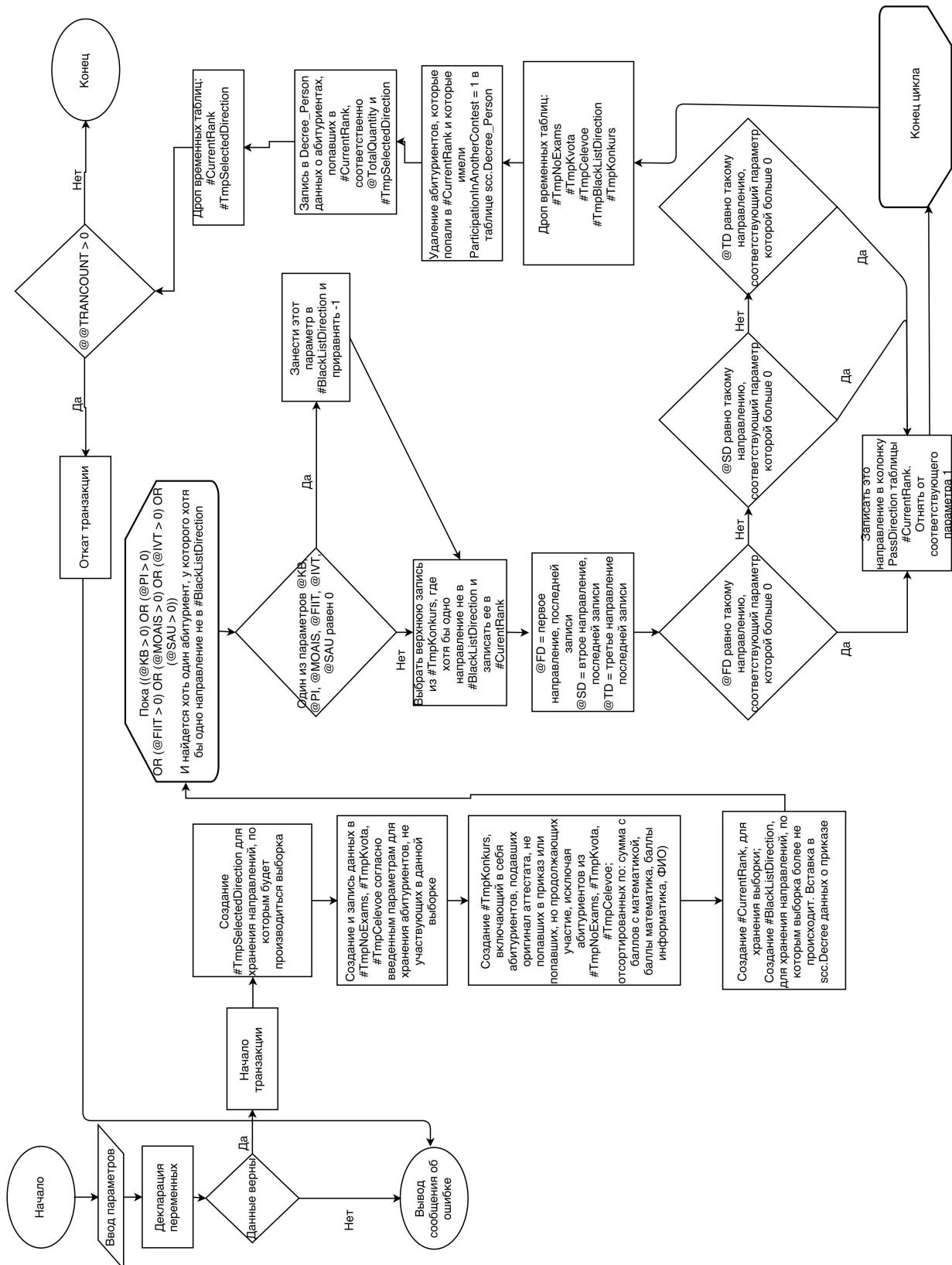


Рисунок 3 – Алгоритм работы процедуры `usp_Bachelor_CurrentRank_Konkurs_DecreeInsert`

2.5 Черновое редактирование

Процесс вынесения приказа и зачисления туда абитуриентов непростой, требует большого количества внимания, а также просчета ситуации наперед. Поэтому требовалось создать для работы с абитуриентами-бакалаврами возможность выносить приказы, заносить туда людей, менять направления поступления в режиме черновика, чтобы наглядно увидеть ситуацию при потенциальном вынесении приказа.

Для этого были созданы 4 таблицы, имеющие префикс *Duplicate*, а также набор процедур, подобных основным, также имеющие префикс *Duplicate*.

Была создана процедура, которая при запуске делает таблицы:

- *scc.Decree_Duplicate*,
- *scc.Decree_Person_Duplicate*,
- *scc.Priority_Duplicate*,
- *scc.Bachelor_FullInfo_Duplicate*

идентичными определенным основным таблицам и представлениям.

Список соответствия:

1. *scc.Decree_Duplicate* - таблица *scc.Decree*.
2. *scc.Decree_Person_Duplicate* - таблица *scc.Decree_Person*.
3. *scc.Priority_Duplicate* - таблица *scc.Priority*.
4. *scc.Bachelor_FullInfo_Duplicate* -
представление *scc.vw_Bachelor_FullInfo*.

2.6 Пользовательский интерфейс

Для удобства пользования базой разработан пользовательский веб интерфейс. Серверная часть веб-интерфейса была написана на языке *Javascript* на платформе *Node.js*. Клиентская часть была разработана на связке *js + html + css*.

ЗАКЛЮЧЕНИЕ

В ходе работы была разработана база данных для ведения учетности новых абитуриентов, поступающих на факультет КНиИТ ФГБОУ ВО «СГУ имени Н. Г. Чернышевского».

Были выполнены все поставленные задачи:

1. В соответствии с правилами приема разработана схема базы данных.
2. Разработаны процедуры для внесения в базу новых абитуриентов (бакалавров, магистров, аспирантов).
3. Разработаны процедуры для обновления абитуриентов (бакалавров, магистров, аспирантов).
4. Разработаны процедуры для удаления абитуриентов (бакалавров, магистров, аспирантов).
5. Разработаны процедуры для выборки абитуриентов согласно их приоритетам по направлениям.
6. Разработаны процедуры для выборки абитуриентов, используя различные сортировки.
7. Разработаны процедуры для внесения в базу приказов о зачислении.
8. Разработаны процедуры для чернового редактирования.
9. Разработан пользовательский интерфейс для работы с базой данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 *Дейт, К. Дж.* Введение в системы баз данных / К. Дж. Дейт. — Москва: Вильямс, 2006.
- 2 *Кузнецов, С. Д.* Основы баз данных / С. Д. Кузнецов. — Москва: Лаборатория знаний, 2007.
- 3 *Когаловский, М. Р.* Энциклопедия технологий баз данных / М. Р. Когаловский. — Москва: Финансы и статистика, 2002.
- 4 *Гарсиа-Молина, Г.* Системы баз данных. Полный курс / Г. Гарсиа-Молина, Дж. Ульман. — М.: Вильямс, 2003.
- 5 *Грофф, Дж.* Энциклопедия SQL / Дж. Грофф, П. Вайнберг. — Москва: Лори, 2003.
- 6 *Мюллер, Дж.* Базы данных и UML / Дж. Мюллер. — Москва: Лори, 2002.
- 7 *Конноли, Т.* Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Т. Конноли. — Москва: Вильямс, 2003.
- 8 *Петкович, Д.* Microsoft SQL Server 2012. Руководство для начинающих / Д. Петкович. — СПб: БХВ-Петербург, 2013.
- 9 *Оти, М.* Будущее sql server: прогноз Теда Каммерта / М. Оти, Ш. Молнар // Открытые системы. — 2011. — № 01.
- 10 Официальный сайт Microsoft [Электронный ресурс] / Microsoft. — 2017. — URL: <https://www.microsoft.com/ru-ru/> (Дата обращения 29.04.2017). Загл. с экрана. Яз. русс.
- 11 Introducing SQL Server Compact 4.0, the Next Gen Embedded Database from Microsoft [Электронный ресурс] / SQL CE Team's Blog. — Москва, 2010. — URL: <https://www.osp.ru/winitpro/2011/01/13008260/> (Дата обращения 04.05.2017). Загл. с экрана. Яз. русс.
- 12 Устав университета / СГУ. — URL: <http://www.sgu.ru/admission/priyomnaya-kampaniya-2017/pravila-priema-v-universitet-v-2017-godu> (Дата обращения 03.05.2017). Загл. с экрана. Яз. русс.
- 13 Лицензия на осуществление образовательной деятельности / СГУ. — URL: <http://www.sgu.ru/admission/priyomnaya-kampaniya-2017/pravila-priema-v-universitet-v-2017-godu>

- priema-v-universitet-v-2017-godu (Дата обращения 14.05.2017). Загл. с экран. Яз. рус.
- 14 Свидетельство о государственной аккредитации вуза / СГУ.— URL: <http://www.sgu.ru/admission/priyomnaya-kampaniya-2017/pravila-priema-v-universitet-v-2017-godu> (Дата обращения 16.05.2017). Загл. с экран. Яз. рус.
- 15 Федеральный закон «Об образовании в Российской Федерации» от 29.12.2012 n 273-ФЗ.— URL: <http://www.consultant.ru/document/cons-doc-LAW-140174/> (Дата обращения 10.05.2017). Загл. с экран. Яз. рус.
- 16 Федеральный закон «Об инновационном центре «Сколково» от 28.09.2010 n 244-ФЗ.— URL: <http://www.consultant.ru/document/cons-doc-LAW-105168/> (Дата обращения 24.05.2017). Загл. с экран. Яз. рус.
- 17 Федеральный закон «О Московском государственном университете имени М.В. Ломоносова и Санкт-Петербургском государственном университете» от 10.11.2009 n 259-ФЗ.— URL: <http://www.consultant.ru/document/cons-doc-LAW-93587/> (Дата обращения 22.05.2017). Загл. с экран. Яз. рус.
- 18 Федеральный закон «О ветеранах» от 12.01.1995 n 5-ФЗ.— URL: <http://www.consultant.ru/document/cons-doc-LAW-5490/> (Дата обращения 21.05.2017). Загл. с экран. Яз. рус.
- 19 Федеральный закон «Об особенностях правового регулирования отношений в сфере образования в связи с принятием в Российскую Федерацию Республики Крым и образованием в составе Российской Федерации новых субъектов - Республики Крым и города федерального значения Севастополя и о внесении изменений в Федеральный закон «Об образовании в Российской Федерации» от 05.05.2014 n 84-ФЗ.— URL: <http://www.consultant.ru/document/cons-doc-LAW-162566/> (Дата обращения 19.05.2017). Загл. с экран. Яз. рус.
- 20 Приказ Минобрнауки России от 14.10.2015 n 1147 (ред. от 29.07.2016) «Об утверждении Порядка приема на обучение по образовательным программам высшего образования - программам бакалавриата, программам специалитета, программам магистратуры» (Зарегистрировано в Минюсте России 30.10.2015 n 39572).—

URL: <http://www.consultant.ru/document/cons-doc-LAW-188408/> (Дата обращения 15.05.2017). Загл. с экрана. Яз. рус.

fm /Евсеев/
10.06.2017