

Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЯДЕРНЫЙ УНИВЕРСИТЕТ «МИФИ»
ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ
КАФЕДРА «КОМПЬЮТЕРНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ»

На правах рукописи
УДК 004.75

ПРОКОФЬЕВ СЕРГЕЙ АЛЕКСАНДРОВИЧ

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ПРИНЯТИЯ РЕШЕНИЯ ПО
КРЕДИТНОЙ ЗАЯВКЕ

Выпускная квалификационная работа бакалавра

Направление подготовки 09.03.01 Информатика и вычислительная техника

Выпускная квалификационная
работа защищена

«__ __»_____2020 г.

Оценка _____

Секретарь ГЭК _____

г. Москва
2020

Студент-дипломник	_____	/ Прокофьев С.А. /
Руководитель работы	_____	/ Овчаренко Е.С. /
Рецензент	_____	/ Бушина К.С. /
Заведующий кафедрой №12	_____	/ Иванов М.А. /

АННОТАЦИЯ

Данная выпускная квалификационная работа посвящена разработке автоматизированной системы, позволяющей автоматизировать, повысить точность и значительно ускорить процесс принятия решения по кредитной заявке в банке.

Работа изложена на 167 страницах; состоит из введения, четырех глав, заключения; включает 36 рисунков, 70 таблиц, библиографический список из 41 источника литературы, 5 приложений.

Во введении обосновывается актуальность работы, ставится цель и задачи работы.

В первой главе проводятся обзор предметной области, исследование технологий обработки информации, анализ и сравнение аналогов и прототипов.

Вторая глава посвящена вопросам разработки структуры автоматизированной системы, построению архитектуры, проектированию инфологической и даталогической моделей.

Третья глава содержит экспериментальные результаты и их обсуждение.

В четвертой главе приводится экономическое обоснование разработки автоматизированной системы.

В заключении подводится итог проделанной работы.

Ознакомление с работой будет полезно специалистам, работающим в банковской сфере, а именно – в сферах автоматизации банковских процессов, минимизации кредитных рисков и построения автоматизированных систем.

ОГЛАВЛЕНИЕ

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ	6
ВВЕДЕНИЕ	8
1 ОБЗОРНАЯ ЧАСТЬ.....	11
1.1 Введение.....	11
1.2 Описание предметной области.	11
1.2.1 Естественнo-языковая модель предметной области	11
1.2.2 Сущности и отношения между ними	13
1.3 Исследование технологий обработки информации	15
1.3.1 Существующие технологии обработки информации и принятия управленческих решений	15
1.3.2 Перечень функций, подлежащих автоматизации.....	15
1.3.3 Выбор и обоснование критериев качества.....	15
1.3.4 Графическая модель предметной области.	16
1.3.5 Анализ аналогов и прототипов.	16
1.3.6 Сравнение аналогов и прототипов.....	24
1.3.7 Вывод.....	28
2 РАСЧЕТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ	29
2.1 Введение.....	29
2.2 Разработка структуры АСОИУ	29
2.3 Разработка архитектуры АСОИУ	31
2.3.1 Выбор средств разработки.....	31
2.3.2 Описание архитектуры	39
2.4 Проектирование инфологической схемы базы данных.....	42
2.5 Проектирование даталогической схемы базы данных	56
2.6 Оптимизация базы данных.....	74
2.6.1 Схема MARTS	74
2.6.2 Схема DICTS.....	78

2.6.3 Схема MAIN.....	82
2.7 Разработка системы передачи данных	90
2.8 Разработка интерфейса взаимодействия пользователя с системой.....	91
3 ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ	100
3.1 Введение.....	100
3.2 Разработка алгоритмов обработки данных.....	100
3.3 Тестирование и отладка.....	105
3.4 Оценка качества функционирования.	107
3.4.1 Описание модели. Входные и выходные данные.....	107
3.4.2 GPSS-моделирование	110
3.4.3 Результаты моделирования	112
3.4.4 Вывод.....	114
4 ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ.....	115
4.1 Введение.....	115
4.2 Определение затрат на разработку	115
4.2.1 Расчет стоимости производимых работ	115
4.2.2 Определение затрат на программное обеспечение	116
4.2.3 Определение затрат на оборудование	117
4.2.4 Определение затрат на оплату труда.....	117
4.2.5 Определение затрат на накладные расходы	118
4.2.6 Расчет себестоимости	118
4.2.7 Расчет прибыли.....	119
4.2.8 Расчет цены	119
4.3 Определение затрат на внедрение системы.....	119
4.4 Затраты на эксплуатацию системы	120
4.5 Расчет срока окупаемости системы после внедрения	122
4.6 Вывод	126
ЗАКЛЮЧЕНИЕ.....	127

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....	128
ПРИЛОЖЕНИЕ А. Описание используемых технологий.....	133
A.1 СУБД Oracle 10g.....	133
A.2 Утилита cURL.....	133
A.3 Red Hat JBOSS Enterprise Application Platform.....	134
A.4 VPN.....	134
A.5 Oracle DataGuard.....	135
A.6 RAID10	136
ПРИЛОЖЕНИЕ Б. Оптимизация схемы MAIN.....	138
ПРИЛОЖЕНИЕ В. Расчет минимальной пропускной способности	151
ПРИЛОЖЕНИЕ Г. Листинг модуля проверки взаимодействия с БД.....	153
ПРИЛОЖЕНИЕ Д. Листинг модуля проверки правил и стоп-листов.....	156

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

АБС – автоматизированная банковская система.

АРМ – автоматизированное рабочее место.

АС – автоматизированная система.

АСОИУ – автоматизированная система обработки информации.

База данных – организованная в соответствии с определёнными правилами и поддерживаемая в памяти компьютера совокупность данных, характеризующая актуальное состояние некоторой предметной области и используемая для удовлетворения информационных потребностей пользователей.

БД – база данных.

БКИ – бюро кредитных историй.

КИ – кредитная история.

Кредитный скоринг – система оценки кредитоспособности лица, основанная на численных методах.

Локальная вычислительная сеть – система связи компьютеров или вычислительного оборудования, покрывающая обычно небольшую территорию или небольшую группу зданий.

ЛВС, LAN – локальная вычислительная сеть.

МБКИ – межбанковское бюро кредитных историй.

НБКИ – национальное бюро кредитных историй.

ОКБ – объединенное кредитное бюро.

ПО – программное обеспечение.

Протокол – набор соглашений интерфейса логического уровня, которые определяют обмен данными между различными программами. Эти соглашения задают единообразный способ передачи сообщений и обработки ошибок при взаимодействии программного обеспечения разнесённой в пространстве аппаратуры, соединённой тем или иным интерфейсом.

Репликация – механизм синхронизации содержимого нескольких копий объекта.

Ручной андеррайтинг – ручная проверка кредитной заявки сотрудником или сотрудниками банка. Окончательное решение о выдаче кредита принимается на основе их опыта, знаний, анализа документов заемщика и проверки предоставленных заемщиком сведений.

Скоринг – кредитный скоринг.

Скорбалл – оценка кредитоспособности лица.

СОЗ – система обработки заявок.

Система принятия решений – разрабатываемая система.

СПР, Система – система принятия решений.

СУБД – система управления базой данных.

УНП – условно неизбыточное покрытие.

ФЗ – функциональная зависимость.

Функциональная зависимость – бинарное отношение между множествами атрибутов данного отношения и является, по сути, связью типа «один ко многим».

ФМС – федеральная миграционная служба.

Система противодействия мошенничеству – сервис, предназначенный для решения задач предотвращения мошенничества и нарушений при кредитовании физических лиц в кредитных организациях.

FPS (Fraud Prevention Service) – система противодействия мошенничеству от Equifax.

NH (National Hunter) – система противодействия мошенничеству от ОКБ.

SAS – американская частная компания, разработчик программного обеспечения в области бизнес-аналитики.

UI – пользовательский интерфейс.

ВВЕДЕНИЕ

Информационные технологии в современном мире инновационных разработок фактически выступают ключевым средством повышения эффективности любой коммерческой деятельности. И банковская деятельность здесь не является исключением.

Большая часть руководства банков признает пользу применения современных достижений области ИТУ (информационных технологий управления), и того, насколько они меняют их бизнес в лучшую сторону, приводя его на достаточно высокие позиции на рынке банковских услуг. Важнейшим из факторов, определяющим общее направление развития организации, а также функционирование и совершенствование информационных технологий управления – является полная интеграция с направлением бизнеса, составляющими его процессами и нацеленностью на достижение поставленных целей.

Согласно последним отчетам по развитию банковского дела, а также статистическим данным, прогнозируется существенный рост рисков, связанных с банковской деятельностью, включая риски розничного и корпоративного кредитования. По состоянию на 01.07.2019 общая сумма дефолтов по кредитам, выданным индивидуальным предпринимателям и юридическим лицам, стремится к 36,0 трлн рублей. Общая же сумма дефолтов по кредитам физическим лицам по состоянию на 01.07.2019 составляет 15,5 трлн рублей [1,2].

Исходя из приведенной выше статистики, очевидно, что банкам необходимо использовать более современные средства мониторинга и снижения рисков в своем рабочем процессе, например, можно назвать информационные системы для расчета, оценки, контроля и управления рисками. Главной проблемой, стоящей перед банками, является снижение показателей кредитного риска.

Задача кредитования сильно зависима от имеющихся исходных данных, а также от условий выдачи кредита, поэтому к ней могут применяться различные классы задач принятия решений, которые, в свою очередь, требуют различных методов исполнения. Следует иметь в виду, что исходные данные задачи подразумевают не только количественные параметры (возраст, средний доход за последние 3 месяца и т.д.), но и качественные (семейное положение, пол, гражданство и т.д.), обеспечивающие прогнозирование возврата кредита с большей точностью [3]. Кроме того, немало важно обеспечить возможность внедрения и тестирования кредитных стратегий в максимально короткие сроки, а также гибкость в разработке этих самых стратегий.

Актуальность данной работы заключается в разработке системы принятия решений в реальном времени, основанной на оценке рисков розничного кредитования. Актуальность подтверждается, с одной стороны, статистикой продолжительного роста дефолтов по кредитам, выданным на основе существующих методов оценки кредитных рисков, а, с другой стороны, существующей необходимостью сохранения кредитной деятельности, минимизируя при этом воздействие кризиса на банковский сектор. Кроме того, у банков существует потребность в снижении операционных издержек процессов розничного кредитования и их автоматизации.

Целью данной работы является создание единой платформы для настройки скорингового процесса, позволяющей ускорить и повысить гибкость разработки, внедрения и тестирования кредитных стратегий (минимизация программирования), а также обеспечить возможность аудита жизненного цикла и маршрута кредитной заявки. В соответствии с поставленной целью, работа над АС (автоматизированной системой) была разделена на несколько этапов, в рамках которых решались следующие задачи:

- анализ предметной области и составление ее естественно-языковой модели;
- выделение базовых сущностей и отношений между ними;
- обзор и сравнение современных технологий обработки информации и принятия управленческих решений;
- выделение перечня функций, подлежащих автоматизации;
- разработка структуры АСОИУ (автоматизированная система обработки информации) и ее архитектуры;
- проектирование инфологической и даталогической моделей БД (базы данных);
- оптимизация полученной модели БД;
- разработка системы передачи данных;
- разработка алгоритмов обработки данных;
- проектирование и последующая разработка интерфейса взаимодействия пользователя с АСОИУ;
- тестирование разработанной АСОИУ;
- оценка качества функционирования.

1 ОБЗОРНАЯ ЧАСТЬ

1.1 Введение

В данной главе представлены исследование и анализ предметной области, описывающие существующие технологии и процессы получения клиентом кредита и оценки банком его кредитоспособности.

1.2 Описание предметной области.

1.2.1 Естественнo-языковая модель предметной области

Бизнес-процесс получения клиентом кредита и оценки банком его кредитоспособности представлен на рисунке 1.1.

Клиент, желающий получить кредит заполняет анкету клиента, указывая в ней требуемые данные, после чего на основе заполненной анкеты сотрудник банка вводит данные в систему обработки заявок и запускает заявку в обработку.

Для передачи заявки в обработку формируется сообщение в формате XML, которое передается в систему принятия решения.

Система принятия решения, получив сообщение производит его разбор в соответствующие поля БД. Затем система принятия решения обрабатывает заявку и формирует решение по заявке.

В процессе обработки заявки проводится ряд проверок, в том числе и проверки списков, позволяющих уже на первых этапах принять решение по кредиту и не проводить дальнейших проверок. В случае прохождения первичных проверок, производится запрос в различные БКИ, проводится скоринг заявки, рассчитываются параметры предложения и проводятся полный комплекс проверок, после чего рассчитывается итоговое решение по кредитной заявке.

На основе результата обработки заявки формируется сообщение в формате XML, которое передается в систему обработки заявок.

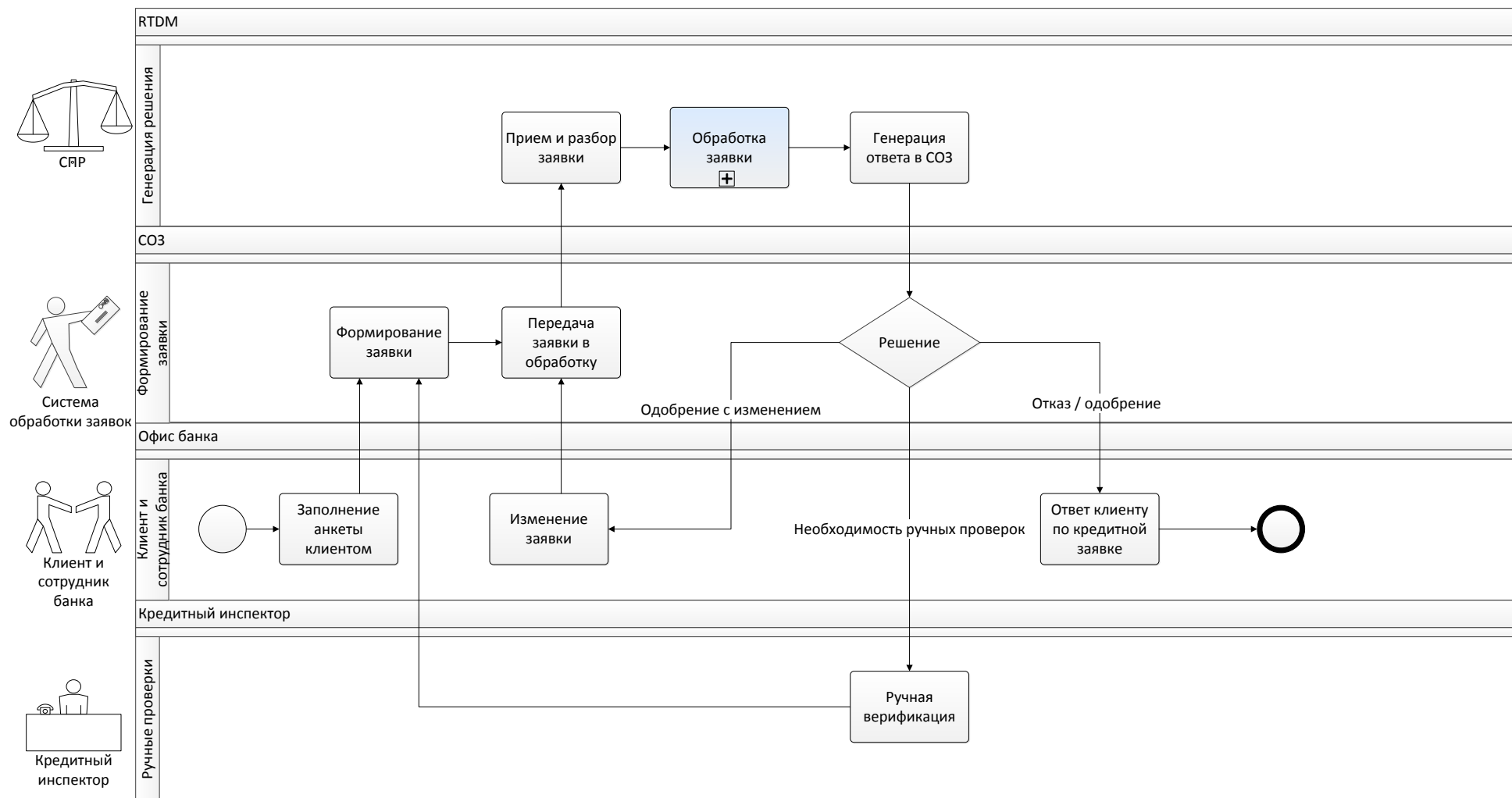


Рисунок 1.1 - Общий алгоритм выдачи кредита

В зависимости от решения, принятого по кредитной заявке возможны следующие варианты:

- Если по заявке принято решение об отказе или одобрении кредита – то данное решение сообщается клиенту, после чего клиенту либо отказывают в получении кредита, либо заключают договор на предоставление кредита, соответственно.
- Если по заявке принято решение «Одобрение с изменением» - то данное решение сообщается клиенту и ему предлагается изменить условия запрашиваемого кредита, после чего производится повторная передача заявки в обработку.
- Если по заявке принято решение «Необходимость ручных проверок» – то данное решение сообщается кредитному инспектору, который должен провести ряд ручных проверок заявки и заполнить соответствующие поля заявки. После проведения ручной верификации заявка повторно отправляется на обработку.

1.2.2 Сущности и отношения между ними

В процессе анализа предметной области выделены следующие сущности:

- «Заявки»;
- «Расчеты»;
- «Ответы»;
- «Расчеты БКИ»;
- «Журнал отказов»;
- «Правила»;
- справочник «Ставка»;
- справочник «Решение»;
- справочник «Продукты»;
- справочник «Правила»;

- справочник «Регионы»;
- справочник «Параметры»;
- справочник «Сектор»;
- «Витрина заявок»;
- «Витрина зарплатных клиентов»;
- «Черный список телефонов»;
- «Черный список клиентов»;
- «Белый список клиентов»;
- «Список террористов».

При каждом запуске СПР для обработки заявки формируется запись сущности «Расчеты».

Обработка проводится по заявке. Одна и та же заявка может быть запущена на обработку несколько раз.

В процессе обработки производятся запросы в БКИ и расчет переменных на основе результата запроса. В расчетах фиксируется лишь последний успешный (без ошибок) запрос.

В процессе обработки данные заявки проверяются по заданным бизнес-правилам, результаты проверки которых для данного запуска СПР фиксируются. В случае возникновения ошибок в процессе обработки все отказы фиксируются.

По окончании обработки заявки формируется ответ по результатам расчётов.

Сущности справочников, витрин и списков представляют собой независимые друг от друга сущности, каждая из которых в отдельности используется при расчётах или проверках правил в процессе обработки заявки.

1.3 Исследование технологий обработки информации

1.3.1 Существующие технологии обработки информации и принятия управленческих решений

Система представляет собой, разработанную на ПО SAS, диаграмму принятия решения, которая позволяет автоматизировать процесс обработки кредитной заявки.

До внедрения системы используются разрозненные приложения различных систем банка и ручные проверки андеррайтером.

1.3.2 Перечень функций, подлежащих автоматизации

В ходе анализа было выявлено, что автоматизации подлежат следующие функции:

- 1) ввод данных кредитной заявки;
- 2) получение и анализ внутренней кредитной истории клиента;
- 3) получение и анализ внешней кредитной истории клиента;
- 4) поиск клиента в стоп-списках;
- 5) расчет скорингового балла и других риск-переменных;
- 6) расчет предложения по кредиту (лимит суммы, срок, ставка);
- 7) анализ кредитоспособности клиента и принятие решения по кредитной заявке;
- 8) предоставление данных по клиенту для анализа кредитным инспектором;
- 9) предоставление результатов решения по кредитной заявке клиенту.

1.3.3 Выбор и обоснование критериев качества.

Для проведения сравнительного анализа аналогов и прототипов выбраны следующие критерии:

- 1) трудозатраты по взаимодействию с БКИ;
- 2) настраиваемость системы;
- 3) достоверность результата;

- 4) потребность в дополнительном ПО;
- 5) степень интеграции с другими системами банка.

Критерий «Трудозатраты по взаимодействию с БКИ» определяет уровень трудозатрат для реализации вызова сервисов БКИ.

Критерий «Настраиваемость системы» определяет уровень трудозатрат, требуемый для изменения существующего алгоритма обработки и настройки работы системы.

Критерий «Достоверность результата» определяет степень достоверности полученного результата обработки заявки.

Критерий «Потребность в дополнительном ПО» определяет объем дополнительного ПО для реализации полного цикла обработки заявки.

Критерий «Степень интеграции с другими системами банка» определяет критерий, описывающий количество систем, с которыми возможно интегрироваться и объем трудозатрат для реализации данных интеграций [4].

1.3.4 Графическая модель предметной области.

Графическая модель предметной области представлена на рисунке 1.2.

1.3.5 Анализ аналогов и прототипов.

Рассмотрим аналоги, прототипы и СПР с точки зрения выбранных критериев качества.

1.3.5.1 Deductor.Loans

Deductor.Loans представляет собой готовое решение, построенное на базе аналитической платформы Deductor и web-технологий, автоматизирующее всю последовательность действий от получения заявки на кредит в удаленной торговой точке до принятия решения о его выдаче и формировании необходимого пакета документов.

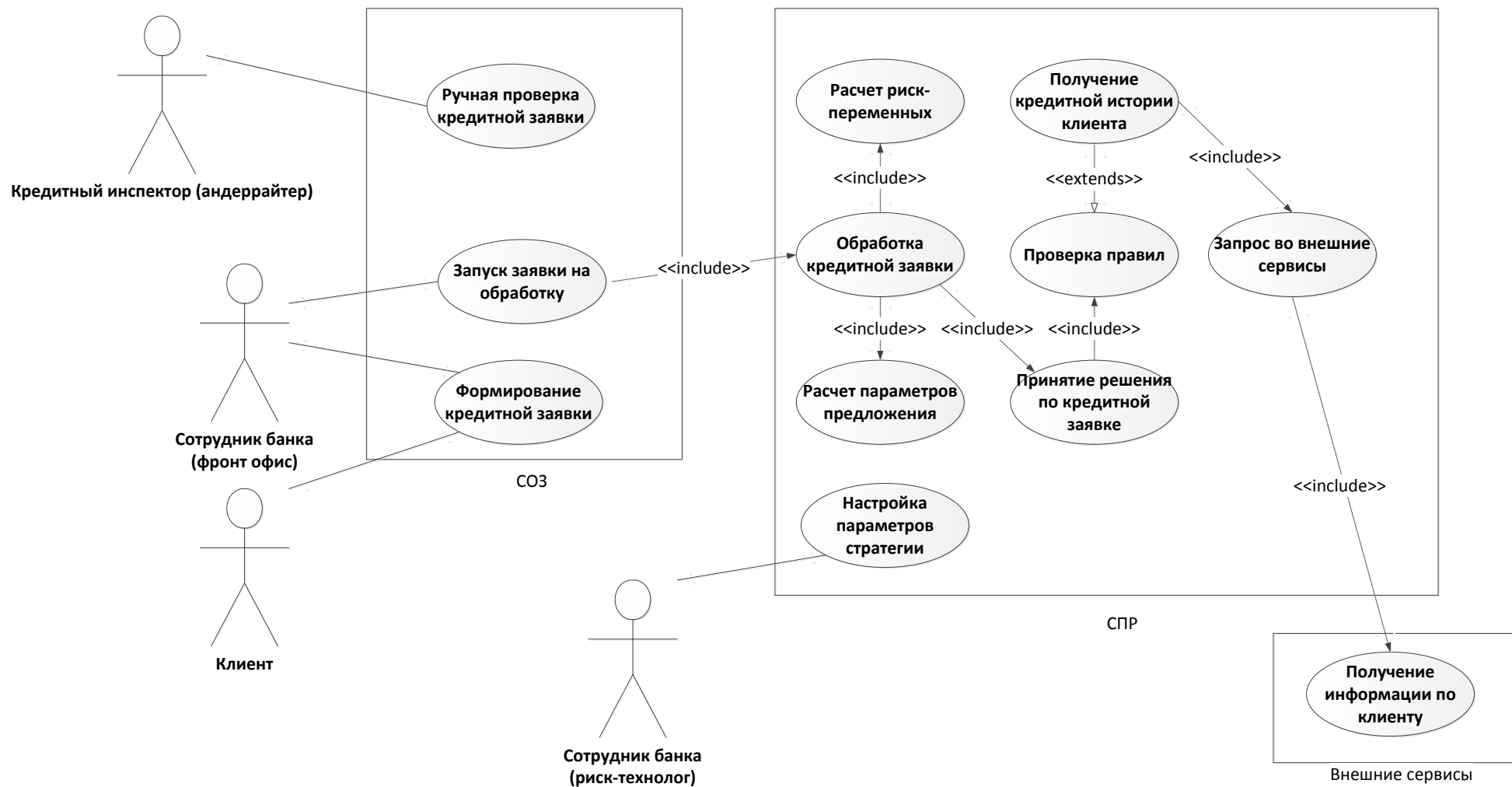


Рисунок 1.1 - Графическая модель предметной области (диаграмма прецедентов)

При этом в процессе задействованы все звенья – оператор торговой точки, служба безопасности, кредитный инспектор банка, адаптируемая скоринговая модель, используемая автоматизированная банковская система.

Платформа Deductor работает с сервером БКИ «Эквифакс Кредит Сервисиз» в режиме On-line (данный способ позволяет банку автоматически запрашивать и получать кредитные отчеты в формате xml-файла), одновременно получая отчеты по целой группе заемщиков. Для обращения к другим внешним сервисам требуется разработка дополнительных модулей.

Работникам банка не нужно вручную заполнять форму запроса и переносить полученную информацию в хранилище данных кредитной организации, как при интерактивном режиме получения кредитных историй. Пользователям не нужно ждать продолжительное время для получения отчетов, и отпадает необходимость в генерации и обработке xml-файла отдельным специализированным ПО. Информация из кредитного отчета уже интегрирована в автоматизированный процесс принятия решения, поэтому банку не нужно разрабатывать отдельный модуль интеграции с «Equifax Credit Services» [5].

Также присутствует интеграция с другими продуктами Deductor и АБС.

Автоматизирован процесс поиска в списках. Автоматизация этого процесса позволяет существенно ускорить соответствующую процедуру проверки заемщика. Возможности аналитической платформы позволяют дополнительно организовывать нечеткий поиск, то есть, когда возвращаемые записи не полностью совпадают с запрошенной информацией, а только лишь характеризуются существенной близостью по отношению к интересующим сведениям.

Во взаимодействии с другими продуктами Deductor можно обеспечить полный цикл обработки заявки от момента подачи заявки до момента выдачи/не выдачи кредита.

Основные возможности решения на базе Deductor:

- консолидация данных из десятков разнородных источников;
- очистка, систематизация и обогащение собранной информации;
- отчетность, визуализация, OLAP-анализ, расчет KPI;
- моделирование, прогнозирование, оптимизация;
- самообучение на новых данных и адаптация моделей.

Процесс принятия решения настраивается вводом определенного набора параметров и алгоритм принятия решения фиксирован. Принятие решение производится на основе созданной в системе фиксированной схемы БД.

Достоинства:

- интеграция с «Эквифакс Кредит Сервисиз» позволяет использовать встроенный модуль для взаимодействия с БКИ и настраивать лишь параметры подключения к нему;
- в сочетании с другими продуктами Deductor можно обеспечить полный цикл обработки заявки;
- можно интегрироваться с АБС банка, списками и другими продуктами Deductor.

Недостатки:

- отсутствие возможности взаимодействия с другими БКИ (не «Эквифакс Кредит Сервисиз»);
- настройка бизнес-процесса осуществляется лишь установкой значений заданных параметров в экранных формах и нет возможности изменять ход бизнес-процесса;
- достоверность оценки определяется заданным в продукте алгоритмом обработки и не может изменяться [5].

1.3.5.2 DiaSoft Flextera

Прикладное решение Flextera «Кредитный конвейер» – это комплекс программных компонентов, предназначенных для автоматизации процессов консультирования потенциальных заёмщиков, предварительного подбора

банковских предложений, приёма, хранения и обработки заявок на приобретение кредитных продуктов, для проверки, верификации и анализа данных, принятия и согласования решений о возможности выдачи кредита, а также для поддержки последующего мониторинга и обслуживания кредитных договоров

Модуль «Взаимодействие с БКИ» – обеспечивает получение информации о кредитных историях, хранящихся в БКИ, позволяет оценить уровень дисциплинированности заёмщика (своевременность исполнения обязательств по кредитным договорам) и минимизировать риски кредитования. Модуль взаимодействия уже сформирован и может настраиваться под различные сервисы [6].

Бизнес-процесс обработки заявки настраивается в визуальном интерфейсе в виде настраиваемой блок-схемы процесса.

Прикладное решение «Кредитный конвейер» интегрируется со следующими системами банка:

- CRM – система управления работы с клиентами;
- стоп-листы;
- системы проверки данных;
- бюро кредитных историй;
- системы анализа банковских операций ПОД/ФТ;
- управление лимитами;
- хранилище данных;
- системы учёта кредитных сделок и обеспечения;
- главная книга.

Прикладное решение «Кредитный конвейер» представляет собой многозвенное JavaEE-приложение (Java Enterprise Edition), построенное в соответствии с принципами SOA (сервис-ориентированной архитектуры) и EDA (архитектуры, управляемой событиями). Таким образом есть возможность разрабатывать дополнительные модули, которые можно

интегрировать с решением «Кредитный конвейер» и тем самым обеспечить полный цикл обработки кредитной заявки.

Достоинства:

- модуль «Взаимодействие с БКИ» позволяет использовать сервисы различных БКИ без необходимости дополнительной разработки;
- бизнес-процесс обработки заявки настраивается в визуальном интерфейсе в виде настраиваемой блок-схемы процесса;
- так как является Java-приложением, то можно разработать дополнительные модули, с помощью которых обеспечить полный цикл обработки кредитной заявки;
- можно интегрироваться с большим количеством банковских систем.

Недостатки:

- дополнительные модули и интеграции требуют написания программного кода;
- возможны проблемы с реализацией возможных алгоритмов обработки заявки на используемой программной платформе.

1.3.5.3 Ручные проверки

При ручном андеррайтинге сотруднику банка требуется большое количество разнородного программного обеспечения для взаимодействия с различными системами, такими как АБС, БКИ, списки и т.д. Возможности взаимодействия с другими системами зависят лишь от соответствующего программного обеспечения.

Алгоритм проверки андеррайтером зависит лишь от требований банка и не требует каких-либо трудозатрат по настройке, как при автоматизированной обработке [7].

Схема ручного андеррайтинга представлена на рисунке 1.3.

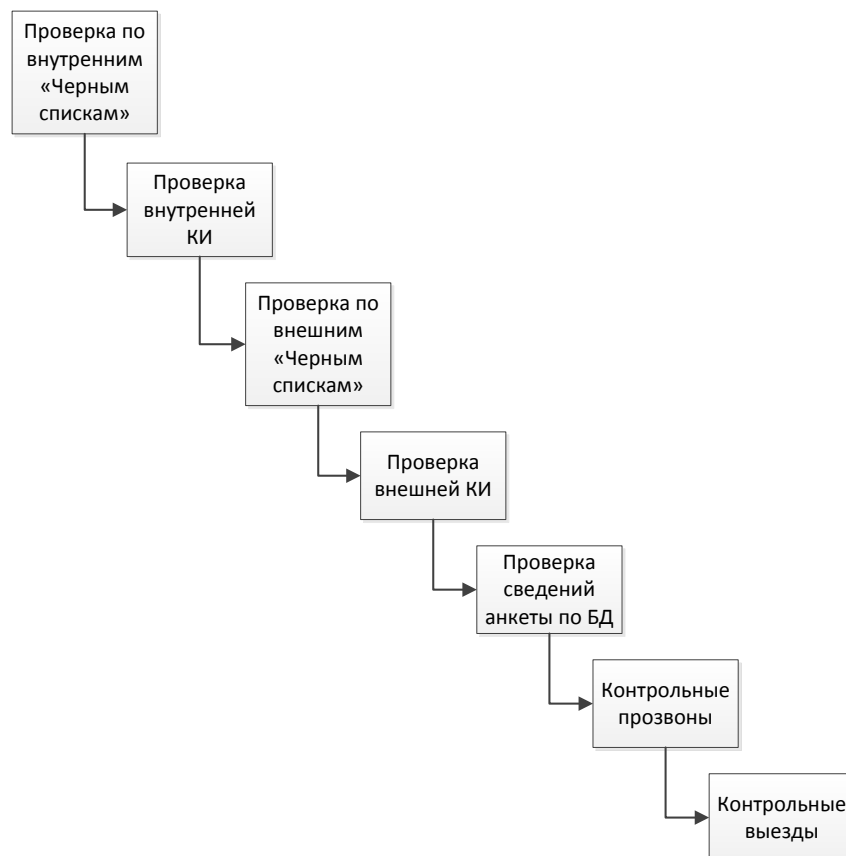


Рисунок 1.2 - Схема ручного андеррайтинга

Достоинства:

- алгоритм обработки назначается банком и однозначно выполняется андеррайтером;
- отсутствует необходимость подстройки процесса под возможности программного продукта.

Недостатки:

- низкая скорость обработки заявки;
- для взаимодействия со всеми системами требуется большое количество различного ПО (имеющегося от поставщика системы или разработанного самостоятельно), которое может и не интегрироваться друг с другом;
- человеческий фактор.

1.3.5.4 Система принятия решений (СПР)

Вызов служб БКИ осуществляется посредством использования внешнего модуля взаимодействия с внешними системами, созданного на той же программной платформе, что и СПР. Вызов любого внешнего веб-сервиса осуществляется вызовом определенной функции в скрипте кода.

Ход бизнес-алгоритма настраивается как с помощью пользовательского интерфейса в виде блок-схемы процесса, так и, при необходимости реализации функционала, превосходящего возможности SAS Decision Management, с помощью программного кода. Таким образом обеспечен высокий уровень гибкости и настраиваемости процесса обработки [8,9].

Для реализации полного цикла обработки заявки может потребоваться дополнительное программное обеспечение. В качестве данного программного обеспечения могут использоваться другие продукты SAS. Таким образом пропадает необходимость трудоёмкой настройки интеграции между этими дополнительными продуктами SAS и СПР.

Взаимодействиями с другими системами не требует больших трудозатрат и осуществляется посредством использования веб-сервисов, интеграционных областей и предоставления витрин данных.

Достоинства:

- для интеграции с БКИ используется единый модуль взаимодействия с внешними системами;
- есть возможность вызова любого веб-сервиса;
- полный цикл обработки заявки можно обеспечить во взаимодействии с другими продуктами используемой платформы SAS;
- можно комбинировать достоинства программного кода и диаграммной логики (описания бизнес-процесса с помощью блок-схемы);
- интеграция с различными источниками данных при помощи возможностей платформы SAS.

Недостатки:

- для реализации функционала взаимодействия с внешними сервисами и построчной обработки данных требуется использование программного кода;
- возможны проблемы с реализацией возможных алгоритмов обработки заявки на используемой программной платформе.

1.3.6 Сравнение аналогов и прототипов

При сравнении используется метод упрощенной аналитической иерархии [10]. Упорядочим критерии по убыванию важности (Таблица 1.1).

Таблица 1.1 – Критерии сравнения, упорядоченные по убыванию важности

Критерий	Обозначение
Настраиваемость системы	K1
Трудозатраты по взаимодействию с БКИ	K2
Достоверность результата	K3
Степень интеграции с другими системами банка	K4
Потребность в дополнительном ПО	K5

Составим матрицу сравнений критериев. При построении будем использовать вербально-числовую шкалу превосходств, описанную в таблице 1.2. Матрица сравнений критериев представлена в таблице 1.3.

Таблица 1.2 – Вербально-числовая шкала превосходств

Равная	1
Умеренное превосходство	3
Существенное превосходство	5
Значительное превосходство	7
Абсолютное превосходство	9

Таблица 1.3 – Матрица сравнения критериев

	K1	K2	K3	K4	K5	Собственное значение	Весовой коэф. α_i
K1	1	3	3	5	7	$\sqrt[5]{315} = 3,159$	$\frac{3,159}{6,325} = 0,5$
K2	$\frac{1}{3}$	1	1	3	5	$\sqrt[5]{5} = 1,149$	$\frac{1,149}{6,325} = 0,181$
K3	$\frac{1}{3}$	1	1	3	5	$\sqrt[5]{5} = 1,149$	$\frac{1,149}{6,325} = 0,181$
K4	$\frac{1}{5}$	$\frac{1}{3}$	$\frac{1}{3}$	1	3	$\sqrt[5]{\frac{1}{15}} = 0,582$	$\frac{0,582}{6,325} = 0,092$
K5	$\frac{1}{7}$	$\frac{1}{5}$	$\frac{1}{5}$	$\frac{1}{3}$	1	$\sqrt[5]{\frac{1}{525}} = 0,286$	$\frac{0,286}{6,325} = 0,046$

Оценим степень согласованности таблицы парных сравнений критериев по формуле (1.1):

$$OC = \frac{\lambda_{max} - m_f}{(m_f - 1) * R}; \quad (1.1)$$

$$m_f = 5 \Rightarrow R = 1,12;$$

$$\lambda_{max} = \left(1 + \frac{1}{3} + \frac{1}{3} + \frac{1}{5} + \frac{1}{7}\right) * 0,5 + \left(3 + 1 + 1 + \frac{1}{3} + \frac{1}{5}\right) * 0,181 +$$

$$+ \left(3 + 1 + 1 + \frac{1}{3} + \frac{1}{5}\right) * 0,181 + \left(5 + 3 + 3 + 1 + \frac{1}{3}\right) * 0,092 +$$

$$+ (7 + 5 + 5 + 3 + 1) * 0,046 =$$

$$= 2,007 * 0,5 + 5,533 * 0,181 + 5,533 * 0,181 + 12,333 * 0,092 + 21 * 0,046 =$$

$$= 5,107$$

$$OC = \frac{5,107 - 5}{(5 - 1) * 1,12} = 0,024$$

Т.к. $OC < 0.1$, то матрица парных сравнений критериев – согласованная.

Соответствие рассматриваемых аналогов указанным критерием представлено в таблице 1.4.

Таблица 1.4 – Качественные характеристики аналогов

	Deductor.Loans	Flextera	Ручные проверки	СПР
Настраиваемость системы	Удовлетворительно	Хорошо	Отлично	Отлично
Трудозатраты по взаимодействию с БКИ	Очень хорошо	Отлично	Очень плохо	Хорошо
Достоверность результата	Удовлетворительно	Очень хорошо	Отлично	Очень хорошо
Степень интеграции с другими системами	Хорошо	Очень хорошо	Плохо	Очень хорошо
Потребность в дополнительном ПО	Отлично	Удовлетворительно	Очень плохо	Очень хорошо

Для перехода к количественным показателям используется шкала, представленная в таблице 1.5.

Таблица 1.5 – Шкала перевода качественных характеристик в количественные

Качественное значение	Очень плохо	Плохо	Удовлетворительно	Хорошо	Очень хорошо	Отлично
Количественное значение	0,1	0,3	0,5	0,7	0,9	1

Результат перевода качественных характеристик в количественные представлен в таблице 1.6.

Таблица 1.6 – Количественные характеристики аналогов

	α	Deductor.Loans	Flextera	Ручные проверки	СПР
Настраиваемость системы	0,5	0,5	0,7	1	1
Трудозатраты по взаимодействию с БКИ	0,181	0,9	1	0,1	0,7
Достоверность результата	0,181	0,5	0,9	1	0,9
Степень интеграции с другими системами	0,092	0,7	0,9	0,3	0,9
Потребность в дополнительном ПО	0,046	1	0,5	0,1	0,9

Для оценки требуется нормировать значения количественных характеристик рассматриваемых аналогов. Нормированные характеристики представлены в таблице 1.7.

Таблица 1.7 – Нормированные количественные характеристики аналогов

	α	Deductor.Loans	Flextera	Ручные проверки	СПР
Настраиваемость системы	0,5	0,5	0,7	1	1
Трудозатраты по взаимодействию с БКИ	0,181	0,9	1	0,1	0,7
Достоверность результата	0,181	0,5	0,9	1	0,9
Степень интеграции с другими системами	0,092	0,778	0,9	0,333	0,9
Потребность в дополнительном ПО	0,046	1	0,5	0,1	0,9
Σ		0,621	0,8	0,734	0,914

1.3.7 Вывод

По результатам сравнения аналогов видно, что СПР имеет наивысший балл и соответственно разработка СПР обоснована.

2 РАСЧЕТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ

2.1 Введение

В данной главе представлена поэтапная разработка структуры и архитектуры АСОИУ. Также описываются проектирование и оптимизация модели данных, разработка системы передачи данных и UI.

2.2 Разработка структуры АСОИУ

Структура взаимодействия систем выглядит следующим образом:

СОЗ передает заявку на обработку в СПР. По окончании обработки результаты обработки возвращаются в СОЗ.

При обработке кредитной заявки СПР производит запрос во внешние сервисы и получает от них результаты.

При обработке кредитной заявки СПР взаимодействует с БД.

В основную схему (MAIN) СПР записывает данные журнализации работы системы и хода обработки заявки. Также СПР записывает и считывает данные заявки, рассчитанные переменные, сработавшие бизнес-правила и данные, полученные от внешних сервисов.

Из схемы справочников (DICTS) СПР считывает данные необходимых справочников и параметры стратегии.

Из схемы витрин (MARTS) СПР считывает данные, предоставленных банком витрин данных и списков.

АРМ клиента (риск-технолога) взаимодействует с СПР для создания стратегии и внесения изменений в алгоритм обработки заявки. Также АРМ клиента взаимодействует с БД.

В основную схему (MAIN) клиент записывает и считывает из неё данные заявки, рассчитанные переменные, сработавшие бизнес-правила. Также считывает данные, полученные от внешних сервисов, и данные журналов.

В схему справочников (DICTS) клиент записывает и считывает из неё данные необходимых справочников и параметры стратегии.

Из схемы витрин (MARTS) клиент считывает данные, предоставленных банком витрин данных и списков.

Структура и схема взаимодействия систем представлена на рисунке 2.1.

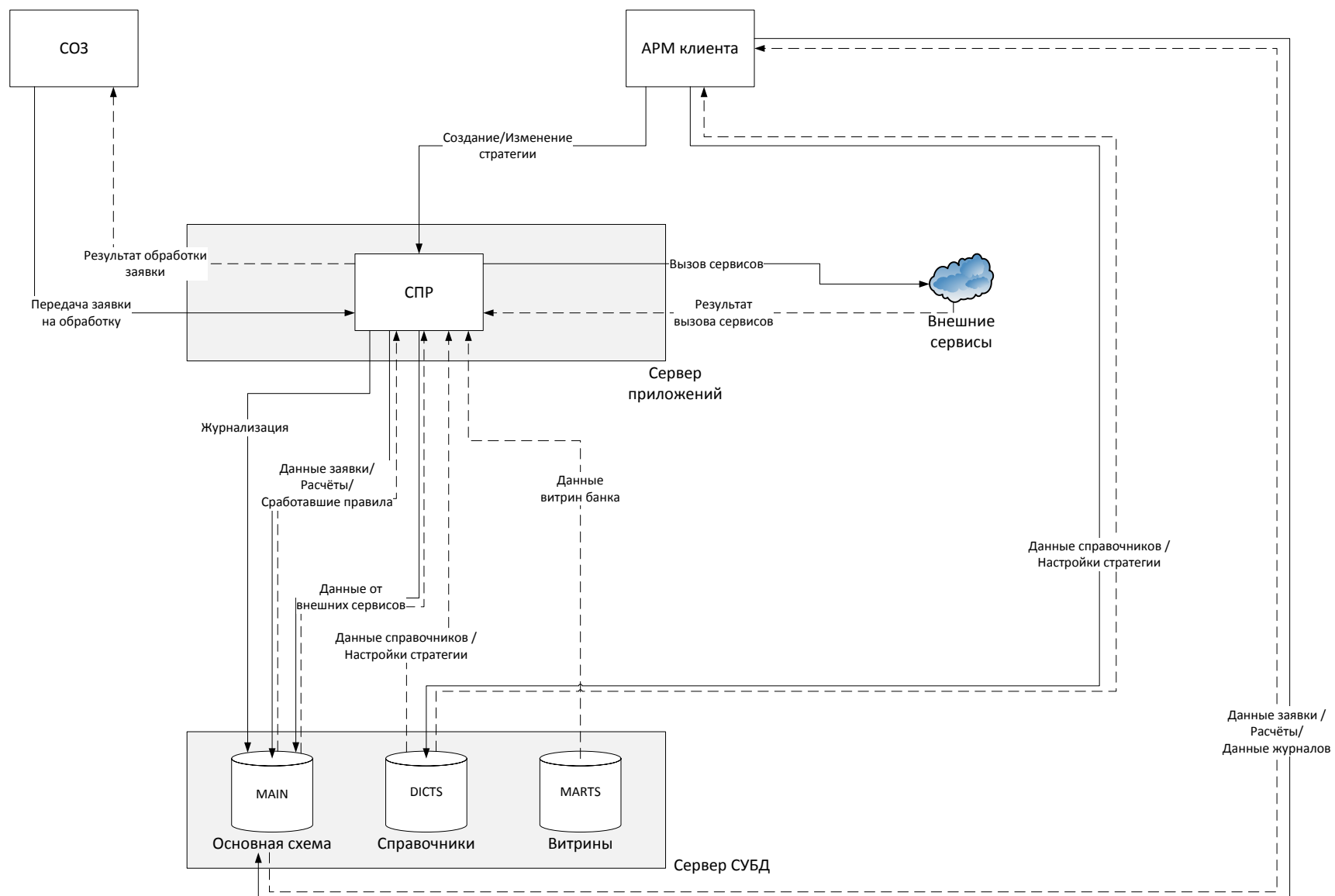


Рисунок 2.1 - Схема взаимодействия систем

2.3 Разработка архитектуры АСОИУ

2.3.1 Выбор средств разработки

2.3.1.1 Критерии сравнения

Для проведения сравнительного анализа выбраны следующие критерии:

1. Функционал;
2. Требования к квалификации оператора;
3. Возможность вызова внешних сервисов;
4. Тестирование и отладка;
5. Сопровождение.

Критерий «Функционал» определяет степень полноты требуемого функционала, реализуемого данным продуктом, без использования дополнительных программных средств.

Критерий «Требования к квалификации оператора» определяет сложность требований, предъявляемых к оператору, для взаимодействия с продуктом. Чем сложность меньше, тем лучше.

Критерий «Возможность вызова внешних сервисов» определяет возможность вызова внешних веб-сервисов.

Критерий «Тестирование и отладка» определяет степень трудоёмкости проведения тестирования и отладки проекта при использовании данного продукта. Чем трудоёмкость меньше, тем лучше.

Критерий «Сопровождение» определяет степень трудоёмкости внесения изменений и сопровождения проекта при помощи выбранного продукта. Чем трудоёмкость меньше, тем лучше [4].

2.3.1.2 SAS RTDM

SAS Real-Time Decision Manager (RTDM) представляет собой законченное решение, реализующее возможности по принятию решения в части работы с клиентами и маркетинга. Используемые технологии позволяют

собирать данные из различных баз данных организации и проводить скоринг клиента для дальнейшего принятия решения [9,13].

Время ответа на запрос составляет меньше секунды и позволяет обрабатывать порядка тысячи транзакций в минуту.

SAS RTDM имеет возможность интеграции с другими приложениями платформы SAS и позволяет использовать скрипты кода и прогнозные модели, построенные с помощью SAS Enterprise и SAS Model Manager.

Основными блоками, из которых формируется диаграмма принятия решения являются: узлы ветвления, процессы работы с данными (чтение, изменение, добавление), процессы вызова веб-сервисов, узлы вызова скриптов кода.

Бизнес-пользователи могут разрабатывать процессы принятия решения в интерактивной, визуальной среде, используя при этом различные источники данных и аналитические инструменты, не используя при этом написание какого-либо кода. Также можно создавать вычисляемые переменные, не используя написание кода [14].

SAS RTDM позволяет видеть информацию, на основании которой принимается решение, в процессе разработки диаграммы.

Имеется возможность использования матрицы решений для снижения числа узлов на схеме. Матрица решений задает зависимость значений одной переменной от одной или двух других. Все диаграммы, блоки, вычисляемые переменные и тест-кейсы могут быть повторно использованы в других диаграммах [11].

SAS RTDM позволяет обращаться к любым внешним сервисам.

Имеется возможность пакетного тестирования, в котором можно использовать исторические и смоделированные данные. По результатам тестирования формируются графики и отчеты по результатам.

Бизнес-пользователь может, используя интерфейс тестирования, указать входные и выходные параметры и сравнить теоретические и практические результаты.

Имеется набор утилит для упрощения интеграции с другими приложениями на уровне данных [14].

SAS RTDM имеет возможность автоматического разбора и генерации XML сообщений, при помощи набора JAVA-классов, входящего в SAS Decision Services Client Application Programming Interface for Java [9,12].

Достоинства:

- возможность создания вычисляемых переменных без необходимости написания кода;
- возможность администрирования кампаний бизнес-пользователями, без участия сотрудников ИТ;
- возможность видеть информацию, на основании которой принимается решение, в процессе разработки диаграммы;
- возможность повторного использования блоков, вычисляемых переменных, тест-кейсов и поддиаграм в других диаграммах;
- возможность строить алгоритмы практически любой сложности и разветвленности;
- возможность тестирования не только всей кампании принятия решения, но и поддиаграм и отдельных путей обработки информации;
- возможность пакетного тестирования;
- вызов любых внешних сервисов.

Недостатки:

- нельзя организовывать циклы в диаграмме;
- в отсутствии блоков логирования процесс отладки является проблематичным;
- для сложных операций требуется использование кода [15]

2.3.1.3 IBM SPSS Decision Management

IBM SPSS Decision Management автоматизирует и оптимизирует процесс принятия решения во фронт-офисе организации для максимизации

результатов. IBM SPSS Decision Management предоставляет возможности обработки в реальном времени.

Процесс создания стратегии для обработки кредитной заявки включает 5 основных шагов:

- определение данных;
- выборка;
- правила;
- комбинация;
- разворачивание (deploy).

Можно объединять данные различных источников. В качестве источника данных можно использовать таблицу, текстовый файл.

При необходимости использования агрегатов, используется SPSS Modeler. Затем созданная модель подключается в качестве источника данных. С помощью пользовательского интерфейса можно создавать правила по проверке значений входных данных. При анализе можно использовать правила, скоринговую карту и прогнозную модель [16].

Отсутствует возможность строить разветвленные и вложенные алгоритмы. Стратегию можно обрабатывать лишь следующим путем: правила, скоринг, прогнозная модель, получение результата. Нет возможности вернуться к предыдущим шагам. Нельзя создавать циклы.

Настройка среды разработки выполняется через специальные xml-файлы. Готовится тестовый набор заявок в качестве источника и прогоняется через стратегию, после чего строится отчет с анализом количества заявок, прошедших по каждой ветке, правилу и блоку процесса. Можно менять матрицу решений и сверять результаты разных тестовых прогонов [17].

Преимущества:

- имеется версионность. Есть возможность восстанавливать старую версию проекта;

- можно объединять данные различных источников. В качестве источника данных можно использовать таблицу, текстовый файл и другие источники данных;
- возможность использования правил скоринга и прогнозных моделей;
- возможность получать итоговое решение на основе всех перечисленных методов;
- возможность пакетного тестирования.

Недостатки:

- не может вызывать внешние веб-сервисы;
- отсутствует возможность строить сложные нелинейные алгоритмы;
- трудность настройки среды разработки.

2.3.1.4 Написание кода

В качестве средства разработки можно использовать написание программного кода на одном из языков программирования. В качестве языка программирования можно использовать тот, который способен обеспечить необходимый функционал, например: C++, Java, SAS.

Использование программного кода позволяет реализовать алгоритмы любой сложности. Таким образом, это позволяет максимизировать функциональные возможности разрабатываемого продукта.

Для работы с разработанным программным продуктом и его сопровождения пользователю требуется взаимодействовать с программным кодом. Этот факт требует от пользователя системы должного уровня знания языка и методов программирования. При внесении возможных изменений в систему предъявляются повышенные требования к документированию кода при разработке.

Для тестирования и отладки требуется применение дополнительного программного обеспечения или использования специальных модулей среды разработки. В случае обнаружения ошибки возможны трудности её изменения, связанные со сложной структурой программного кода.

Преимущества:

- возможность реализации алгоритмов высокой сложности;
- преимущества конкретного языка программирования;
- возможность вызова внешних систем.

Основные недостатки:

- высокие требования к квалификации оператора при работе с системой и сопровождении;
- трудность отладки и тестирования;
- трудность внесения изменений в систему.

2.3.1.5 Сравнение средств разработки

При сравнении используется метод упрощенной аналитической иерархии [10]. Упорядоченные по убыванию важности критерии представлены в таблице 2.1.

Таблица 2.1 – Критерии сравнения, упорядоченные по убыванию важности

Критерий	Обозначение
Функционал	K1
Возможность вызова внешних сервисов	K2
Сопровождение	K3
Требования к квалификации оператора	K4
Тестирование и отладка	K5

Составим матрицу сравнений критериев. При построении воспользуемся вербально-числовой шкалой превосходств, описанной в таблице 2.2.

Таблица 2.2 – Вербально-числовая шкала превосходств

Равная	1
Умеренное превосходство	3
Существенное превосходство	5
Значительное превосходство	7
Абсолютное превосходство	9

Матрица сравнений критериев представлена в таблице 2.3.

Таблица 2.3 – Матрица сравнения критериев

	K1	K2	K3	K4	K5	Собственное значение	Весовой коэф. α_i
K1	1	3	5	5	7	$\sqrt[5]{525} = 3,5$	$\frac{3,5}{6,985} = 0,501$
K2	$\frac{1}{3}$	1	3	3	5	$\sqrt[5]{15} = 1,719$	$\frac{1,719}{6,985} = 0,246$
K3	$\frac{1}{5}$	$\frac{1}{3}$	1	1	3	$\sqrt[5]{\frac{1}{5}} = 0,725$	$\frac{0,725}{6,985} = 0,104$
K4	$\frac{1}{5}$	$\frac{1}{3}$	1	1	3	$\sqrt[5]{\frac{1}{5}} = 0,725$	$\frac{0,725}{6,985} = 0,104$
K5	$\frac{1}{7}$	$\frac{1}{5}$	$\frac{1}{3}$	$\frac{1}{3}$	1	$\sqrt[5]{\frac{1}{315}} = 0,316$	$\frac{0,316}{6,985} = 0,045$

Оценим степень согласованности таблицы парных сравнений критериев по формуле (2.1):

$$OC = \frac{\lambda_{max} - m_f}{(m_f - 1) * R}; \quad (2.1)$$

$$m_f = 5 \Rightarrow R = 1,12;$$

$$\begin{aligned} \lambda_{max} &= \left(1 + \frac{1}{3} + \frac{1}{5} + \frac{1}{5} + \frac{1}{7}\right) * 0,501 + \left(3 + 1 + \frac{1}{3} + \frac{1}{3} + \frac{1}{5}\right) * 0,246 + \\ &+ \left(5 + 3 + 1 + 1 + \frac{1}{3}\right) * 0,104 + \left(5 + 3 + 1 + 1 + \frac{1}{3}\right) * 0,104 + \\ &+ (7 + 5 + 3 + 3 + 1) * 0,045 = \\ &= 1,876 * 0,501 + 4,866 * 0,246 + 10,333 * 0,104 + 10,333 * 0,104 + 19 * 0,045 = \\ &= 5,141 \end{aligned}$$

$$OC = \frac{5,141 - 5}{(5 - 1) * 1,12} = 0,031$$

Т.к. $OC < 0.1$, то матрица парных сравнений критериев – согласованная.

Соответствие рассматриваемых аналогов указанным критерием представлено в таблице 2.4.

Таблица 2.4 – Качественные характеристики аналогов

	SAS RTDM	IBM SPSS Decision Management	Программный код
Функционал	Очень хорошо	Удовлетворительно	Отлично
Возможность вызова внешних сервисов	Да	Нет	Да
Сопровождение	Очень хорошо	Отлично	Плохо
Требования к квалификации оператора	Хорошо	Отлично	Плохо
Тестирование и отладка	Хорошо	Отлично	Плохо

Для перехода к количественным показателям используется шкала, представленная в таблице 2.5.

Таблица 2.5 – Шкала перевода качественных характеристик в количественные

Качественное значение	Нет	Плохо	Удовлетворительно	Хорошо	Очень хорошо	Отлично / Да
Количественное значение	0,1	0,3	0,5	0,7	0,9	1

Результат перевода качественных характеристик в количественные представлен в таблице 2.6.

Таблица 2.6 – Количественные характеристики аналогов

	α	SAS RTDM	IBM SPSS Decision Management	Программный код
Функционал	0,501	0,9	0,5	1
Возможность вызова внешних сервисов	0,246	1	0,1	1
Сопровождение	0,104	0,9	1	0,3
Требования к квалификации оператора	0,104	0,7	1	0,3
Тестирование и отладка	0,045	0,7	1	0,3

Для оценки требуется нормировать значения количественных характеристик рассматриваемых аналогов. Нормированные характеристики представлены в таблице 2.7.

Таблица 2.7 – Нормированные количественные характеристики аналогов

	α	SAS RTDM	IBM SPSS Decision Management	Программный код
Функционал	0,501	0,9	0,5	1
Возможность вызова внешних сервисов	0,246	1	0,1	1
Сопровождение	0,104	0,9	1	0,3
Требования к квалификации оператора	0,104	0,7	1	0,3
Тестирование и отладка	0,045	0,7	1	0,3
Σ		0,895	0,528	0,8229

По результатам сравнения аналогов видно, что SAS RTDM имеет наивысший балл и соответственно при разработке СПР будет использован этот продукт.

2.3.2 Описание архитектуры

Система построена с использованием двухуровневой и трёхуровневой архитектур клиент-сервер. Данная архитектура выбрана по причине низкого объёма трафика, передаваемого между компонентами системы и с целью снижения требований к клиентским машинам и обеспечения безопасности хранимых данных.

При обработке заявки и работы непосредственно со стратегией используется трёхуровневая модель архитектуры с использованием сервера приложений и сервера СУБД. Оба сервера являются виртуальными и размещаются на одном физическом сервере [18].

При работе пользователя с данными БД с использованием специализированного ПО используется двухуровневая модель архитектуры (модель сервера СУБД).

В качестве клиентов выступают: система обработки заявок (СОЗ) и АРМ риск-технологов.

На сервере СУБД установлена СУБД Oracle 11g (обоснование выбора и описание технологии представлено в Приложении А). В экземпляре БД созданы 3 схемы (подробное описание и обоснование представлено в пункте 2.1.4):

- основная схема (MAIN);
- справочники (DICTS);
- витрины (MARTS).

На сервере приложений установлено ПО SAS и Red Hat JBoss Enterprise Application Platform (описание технологии представлено в Приложении А). Также в операционной системе, установленной на сервере приложений, установлена утилита cURL (описание утилиты представлено в Приложении А).

На АРМ риск-технологов установлены приложения SAS Customer Intelligence Studio и SAS Data Integration Studio. Также установлено приложение для взаимодействия с БД.

SAS Customer Intelligence Studio предоставляет визуальную среду для разработки, изменения и тестирования стратегии [14].

SAS Data Integration Studio предоставляет визуальную среду для разработки, изменения и тестирования скриптов кода по расчёту риск-переменных на основе данных, полученных из БД. Созданные скрипты затем используются при создании стратегии.

Приложение для взаимодействия с БД предоставляет визуальный интерфейс взаимодействия с БД. Это приложение используется для установки и изменения параметров стратегии, просмотра журналов, результатов обработки заявки.

Для обработки кредитной заявки СОЗ осуществляет вызов соответствующего веб-сервиса системы, используя протокол SOAP. Пакет

ПО JBoss Enterprise Application Platform включает в себя веб-сервер Apache, который предоставляет клиенту интерфейс веб-сервиса СПР [19]. При вызове в систему передается xml-файл с данными заявки. Вызов веб-сервиса системой обработки заявок осуществляется по сети VPN (описание технологии представлено в Приложении А).

Веб-сервис вызывает компонент SAS RTDM (управление принятием решения в реальном времени), который запускает СПР. Алгоритм обработки заявки, реализуемый кредитной стратегией выполняется в SAS RTDM.

При обращении к данным БД компонент SAS RTDM использует JDBC-драйвер. При необходимости запуска SAS-кода из стратегии осуществляется вызов компонента SAS Activity Wrapper с указанием имени файла скрипта кода, после чего код передаётся на SAS Object Spawner.

Компонент SAS Object Spawner прослушивает соответствующие порты на наличие запроса на выполнение и при получении запроса запускает процесс на сервере хранимых процессов (SAS Stored Process Server). SAS Object Spawner также может выполнять роль балансировщика нагрузки в случае использования нескольких серверов. Запущенный на сервере хранимых процессов процесс исполняет код скрипта SAS-кода [13].

Для обращения в скрипте к БД используется модуль SAS/ACCESS, который обеспечивает для SAS прозрачный доступ к данным, находящимся вне SAS. Для получения данных из СУБД Oracle используется SAS/ACCESStoOracle.

При обращении в скрипте к внешним сервисам используется утилита cURL, которая вызывает внешний веб-сервис посредством SOAP-запроса [20].

Для работы в приложениях SAS Customer Intelligence и SAS Data Integration Studio пользователь сперва подключается к серверу метаданных SAS, который проводит аутентификацию и идентификацию пользователя. После успешного прохождения аутентификации процессы, запускаемые этими приложениями, передаются в Object Spawner, который затем передаёт их на сервер хранимых процессов [13].

Приложение для взаимодействия с БД реализовано на языке программирования C# с использованием технологии Windows Forms. Для запуска приложения используется .NetFramework 2.0. Для взаимодействия с БД используется ODBC драйвер. Обмен данными с сервером СУБД осуществляется посредством протокола TCP/IP.

Архитектура системы представлена на рисунке 2.2.

2.3.2.1 Организация информационной безопасности

СПР имеет три точки входа в систему:

- вызов веб-сервиса обработки заявок;
- взаимодействие со стратегией через клиентские приложения SAS;
- взаимодействие с СУБД через клиентские приложения SAS и специализированное ПО для взаимодействия с БД.

Клиентская и серверные части находятся в единой VPN сети, что блокирует возможность доступа к СПР извне сети банка. Используется протокол IPSec [21].

Безопасность взаимодействия со стратегией и серверным ПО SAS обеспечена средствами аутентификации SAS. При использовании любого клиентского приложения SAS требуется пройти аутентификацию на сервере метаданных SAS.

Безопасность взаимодействия с данными СУБД организована на уровне аутентификации подключения к серверу СУБД и различных схем БД.

2.4 Проектирование инфологической схемы базы данных

Основой базы данных является модель данных. Информационно-логическая (инфологическая) модель предметной области отражает предметную область в виде совокупности информационных объектов и их структурных связей [22].

Описание сущностей инфологической модели представлено в таблице 2.8.

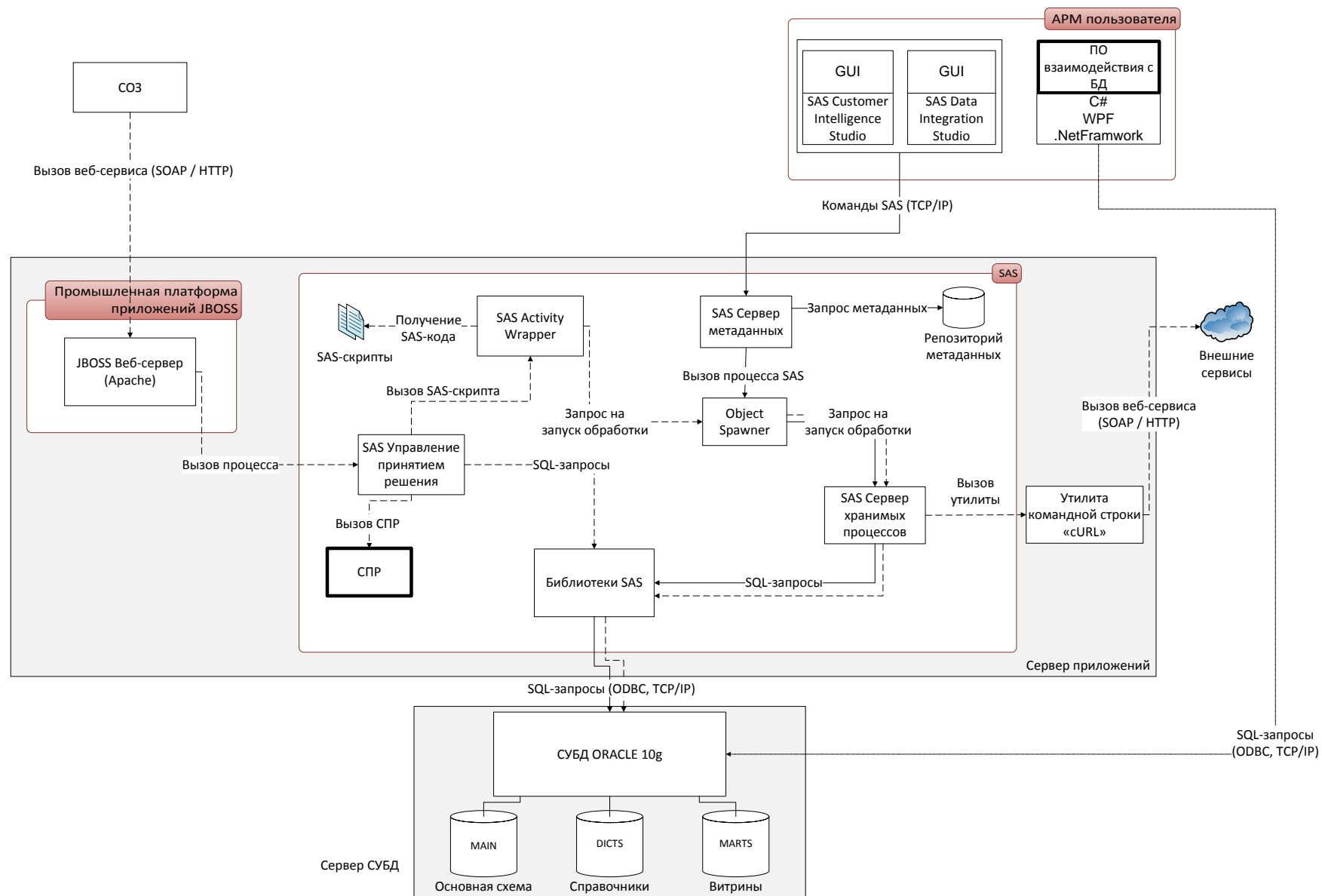


Рисунок 2.2 - Архитектура системы

Таблица 2.8 – Описание сущностей инфологической модели данных

Сущность	Описание	Ключ сущности
Справочник необходимости проверки правил	Справочник содержит в себе флаги необходимости проверки правил.	Код продукта, статус клиента, сегмент, ID правила
Справочник параметров кредитной стратегии	Справочник, содержащий параметры стратегии для проверки правил и расчета риск-переменных	Стратегия, Дата актуальности стратегии
Справочник сегмента	Справочник зависимости сегмента от скорбаллов	Максимальное значение анкетного скорбалла, минимальное значение анкетного скорбалла, максимальное значение скорбалла БКИ, минимальное значение скорбалла БКИ
Справочник регионов присутствия	Таблица содержит список регионов, в которых производится выдача при наличии места работы или места постоянной регистрации	Код региона
Справочник определения процентной ставки	Справочник определения процентной ставки	Бакет, сегмент
Справочник для принятия решения	Справочник определения промежуточного решения по заявке	Отказ, одобрение, одобрение с изменением, необходимость верификации

Сущность	Описание	Ключ сущности
Справочник параметров кредитных продуктов	Справочник содержит параметры для каждого продукта кредитования	Код продукта
Заявка	Данные по заявке, полученные из СОЗ	ID заявки
Результаты проверок правил	Таблица содержит результат проверки правил	ID правила, время срабатывания правила
Результаты расчётов	Таблица содержит промежуточные и окончательные результаты вычислений (параметры для лимита, риск-переменные и прочие расчетные показатели)	ID запуска
Журнал отказов	Таблица содержит ошибки, возникшие при обработке заявки	Время отказа
Расчетные атрибуты по обязательствам БКИ	Таблица содержит расчетные атрибуты по обязательствам, полученным из БКИ	ID запроса
Ответ в СОЗ	Таблица, на основании полей которой будет генерироваться ответ в СОЗ	
Скор балл для клиентов с заработной платой	Витрина содержит рассчитанный скоринговый балл и информацию о поступлениях на зарплатную карту за последние 6 месяцев	Имя, фамилия, отчество, дата рождения

Сущность	Описание	Ключ сущности
Витрина заявок банка	Витрина содержит данные по заявкам и договорам в банке	ID заявки
Черный список клиентов банка	Список клиентов банка, которым будет отказано в выдаче кредита	Имя, фамилия, отчество, дата рождения
Белый список клиентов банка	Список клиентов банка, которым будет одобрена выдача кредита	Имя, фамилия, отчество, дата рождения
Список террористов	Список клиентов банка, которые признаны террористами и которым будет отказано в получении кредита. Данный список банк получает от федеральной службы по финансовому мониторингу.	Имя, фамилия, отчество, дата рождения
Черный список телефонов	Номера телефонов, владельцам которых будет отказано в выдаче кредита	Номер телефона, префикс телефона

Выделенные в процессе анализа предметной области связи инфологической модели:

- проводятся для: Расчеты – Заявки (М:1);
- формируется по: Ответы – Расчёты (1:1);
- запрашиваются по: Расчеты БКИ – Расчёты (1:1);
- ведётся для: Журнал отказов – Расчёты (М:1);
- проверяются по: Правила – Расчёты (М:1);
- вычисляется по: Ставка – Расчёты (1:М);
- вычисляется по: Решение – Расчёты (1:М).

Атрибуты инфологической модели представлены в таблицах 2.9–2.27.

Таблица 2.9 – Атрибуты инфологической модели данных сущности «Витрина заявок»

Атрибут	Тип переменной
Номер заявки СОЗ	Строка (20)
Аннуитетный платеж	Число
Код продукта	Число
Одобренная ставка	Число
Одобренный срок	Число
Доход заемщика(основной)	Число
Валюта основного дохода	Число
Количество дней просрочки	Число
Решение	Строка (50)
Дата решения	Дата/время
Дата открытия договора	Дата/время
Номер домашнего телефона	Строка(300)
Префикс домашнего телефона	Число
Сумма задолженности (баланс + просрочка)	Число
Номер мобильного телефона	Строка(300)
Префикс мобильного телефона	Число
Сумма на просрочке	Число
Номер паспорта	Строка(100)
Серия паспорта	Строка(100)
Кем выдан паспорт	Строка(300)
Дата выдачи паспорта	Дата/время
Код решения (стратегия)	Строка(300)
Код УБ (ручной верификации)	Строка(1000)
Номер рабочего телефона	Строка(300)
Префикс рабочего телефона	Число

Таблица 2.10 – Атрибуты инфологической модели данных сущности «Результаты расчетов»

Атрибут	Тип переменной
Возраст	Число
Максимально допустимый ежемесячный платеж по кредиту	Число
Одобренная сумма	Число
Одобренный срок	Число
Наличие в списках (террористы, ФМС, черные списки и белые списки банка)	Строка(100)
Максимальное количество дней просрочки по кредиту в банке	Число
Дата принятия решения	Дата/время
Есть одобрение	Число
Есть одобрение с изменениями	Число
Есть отказ	Число
Есть необходимость верификации	Число
Вектор, содержащий данные о найденных пересекающихся заявках	Строка(3000)
Id заявки	Строка(300)
Доход увеличился более чем на 100% за 3 месяца	Число
Доход увеличился более чем на 20% за 3 месяца	Число
Сумма всей текущей задолженности	Число
Идентификатор хорошего клиента	Число
Идентификатор плохого клиента	Число
Бакет	Число
Флаг предложения кредитной карты	Число
Плохая кредитная история у других заемщиков с тем же мобильным или домашним телефоном	Число

Атрибут	Тип переменной
Количество отказов за последние дни	Число
Id запуска	Число
Скоринговый балл по карте 1	Число
Скоринговый балл по карте 2	Число
Сегмент	Число
Список сработавших правил	Строка(2000)
Идентификатор запроса во внешний сервис	Число

Таблица 2.11 – Атрибуты инфологической модели данных сущности «Заявки»

Атрибут	Тип переменной
Дата заявления	Дата/время
Запрашиваемая ставка	Число
Доход заёмщика	Число
Валюта дохода	Число
Гражданство	Число
Наличие судимостей	Число
Код продукта	Число
Статус клиента	Число
Дата рождения	Дата/время
Образование	Число
Семейное положение	Число
Имя	Строка(300)
Пол	Число
Домашний телефон (номер)	Строка(300)
Домашний телефон(префикс)	Число
Id заявки	Строка(300)
Фамилия	Строка(300)
Код фейс-контроля	Число
Отчество	Строка(300)
Мобильный телефон(номер)	Строка(300)
Мобильный телефон(префикс)	Число

Атрибут	Тип переменной
Количество детей	Число
Кем выдан паспорт	Строка(300)
Дата выдачи паспорта	Дата/время
Номер паспорта	Строка(100)
Серия паспорта	Строка(100)
Стаж на текущей работе	Число
Наличие поручителя	Число
Код точки приема заявки	Строка(100)
Регион регистрации	Число
Запрашиваемая сумма кредита	Число
Запрашиваемый срок кредита	Число
Шаг процесса	Число
Код андеррайтинга	Число
Рабочий телефон(номер)	Строка(300)
Рабочий телефон(префикс)	Число
Регион работодателя	Число

Таблица 2.12 – Атрибуты инфологической модели данных сущности «Расчеты по БКИ»

Атрибут	Тип переменной
ID запроса	Число
Общее количество закрытых кредитных обязательств	Число
Число ошибок при запросе в БКИ	Число
Количество кредитов с текущей просрочкой	Число
Общее количество кредитных обязательств	Число
Ехреgian скоринговый бал	Число
FICO скоринговый бал	Число
Количество последних платежей без просрочки	Число
Максимальный объем просроченной задолженности	Число

Атрибут	Тип переменной
Количество открытых кредитов за последний месяц	Число
Количество просрочек 60-90 по кредитной истории БКИ	Число
Количество просрочек 90+ по кредитной истории БКИ	Число
ФМС фрод-статус по паспорту	Число
НН фрод-статус по адресу регистрации	Число
Ежемесячный платеж	Число
Количество непогашенных кредитов свыше 100000 рублей	Число

Таблица 2.13 – Атрибуты инфологической модели данных сущности «Справочник «Решение»»

Атрибут	Тип переменной
Сработало правило отказа	Число
Сработало правило одобрения	Число
Сработало правило одобрения с изменением	Число
Сработало правило необходимости верификации	Число
Решение	Строка(32)

Таблица 2.14 – Атрибуты инфологической модели данных сущности «Справочник «Ставка»»

Атрибут	Тип переменной
Ставка	Число
Бакет	Число
Сегмент	Число

Таблица 2.15 – Атрибуты инфологической модели данных сущности «Справочник «Параметры»»

Атрибут	Тип переменной
Стратегия	Строка(64)
Дата актуальности стратегии	Дата/время
Курс евро к рублю для БКИ	Число
Курс доллара к рублю для БКИ	Число
Период, за который рассчитывается увеличение дохода	Число
Период, за который рассчитывается увеличение дохода	Число
Максимально допустимая длительность просрочки	Число
Максимально допустимая сумма просрочки	Число
Период, за который учитывается отказ по тому же номеру телефона	Число
Период за который вычисляется количество отказов	Число
Глубина поиска БКИ в днях	Строка(64)

Таблица 2.16 – Атрибуты инфологической модели данных сущности «Справочник «Продукты»»

Атрибут	Тип переменной
Код продукта	Число
Максимальный срок по продукту	Число
Минимальный срок по продукту	Число
Минимальный общий стаж по продукту	Число
Максимальный возраст по продукту	Число
Минимальный возраст по продукту	Число
Максимальный лимит кредитования для продукта	Число
Минимальный лимит кредитования для продукта	Число

Таблица 2.17 – Атрибуты инфологической модели данных сущности «Справочник «Регионы»»

Атрибут	Тип переменной
Проверка адреса регистрации	Число
Проверка адреса работы	Число
Код региона	Число

Таблица 2.18 – Атрибуты инфологической модели данных сущности «Справочник «Правила»»

Атрибут	Тип переменной
Программа кредитования	Число
Статус клиента	Строка(64)
Необходимость проверки правила	Число
Код правила	Строка(64)
Тип правила	Строка(64)
Сегмент	Строка(64)

Таблица 2.19 – Атрибуты инфологической модели данных сущности «Справочник «Сегмент»»

Атрибут	Тип переменной
Максимальное значение анкетного скорбалла	Число
Минимальное значение анкетного скорбалла	Число
Максимальное значение скорбалла БКИ	Число
Минимальное значение скорбалла БКИ	Число
Сегмент	Число

Таблица 2.20 – Атрибуты инфологической модели данных сущности «Черный список клиентов»

Атрибут	Тип переменной
Имя	Строка(300)
Фамилия	Строка(300)
Отчество	Строка(300)
Дата рождения	Дата/время
Срок нахождения в списке	Дата/время

Таблица 2.21 – Атрибуты инфологической модели данных сущности «Черный список телефонов»

Атрибут	Тип переменной
Номер телефона	Строка(300)
Префикс	Число
Срок нахождения в списке	Дата/время

Таблица 2.22 – Атрибуты инфологической модели данных сущности «Список террористов»

Атрибут	Тип переменной
Имя	Строка(300)
Фамилия	Строка(300)
Отчество	Строка(300)
Дата рождения	Дата/время
Срок нахождения в списке	Дата/время

Таблица 2.23 – Атрибуты инфологической модели данных сущности «Белый список клиентов»

Атрибут	Тип переменной
Имя	Строка(300)
Фамилия	Строка(300)
Отчество	Строка(300)
Дата рождения	Дата/время
Срок нахождения в списке	Дата/время

Таблица 2.24 – Атрибуты инфологической модели данных сущности «Витрина зарплатных клиентов»

Атрибут	Тип переменной
Имя	Строка(300)
Фамилия	Строка(300)
Отчество	Строка(300)
Дата рождения	Дата/время
Доход	Число
Предрассчитанный скорбалл	Число

Таблица 2.25 – Атрибуты инфологической модели данных сущности «Ответ»

Атрибут	Тип переменной
Максимально допустимый ежемесячный платеж по кредиту	Число
Дата заявления	Дата/время
Одобренная сумма	Число
Одобренный срок	Число
Запрашиваемая ставка	Число
Наличие в списках (террористы, ФМС, черные списки и белые списки банка)	Строка(100)
Код продукта	Число
Дата принятия решения	Дата/время
Ехреgianскоринговый бал	Число
FICO скоринговый бал	Число
Вектор, содержащий данные о найденных пересекающихся заявках	Строка(3000)
Id заявки	Строка(300)
Код фейс-контроля	Число
Идентификатор хорошего клиента	Число
Идентификатор плохого клиента	Число
Флаг предложения кредитной карты	Число
Кем выдан паспорт	Строка(300)
Дата выдачи паспорта	Дата/время
Номер паспорта	Строка(100)
Серия паспорта	Строка(100)
ФМС фрод-статус по паспорту	Число
НН фрод-статус по адресу регистрации	Число
Запрашиваемая сумма кредита	Число
Запрашиваемый срок кредита	Число
Id запуска	Число
Скоринговый балл по карте 1	Число
Скоринговый балл по карте 2	Число
Ежемесячный платеж	Число
Код андеррайтинга	Число
Список сработавших правил	Строка(2000)

Таблица 2.26 – Атрибуты инфологической модели данных сущности «Сработавшие правила»

Атрибут	Тип переменной
Комментарий	Строка(300)
Результат проверки правила	Число
Id правила	Строка(64)
Id запуска	Число
Текущие дата и время	Дата/время

Таблица 2.27 – Атрибуты инфологической модели данных сущности «Журнал отказов»

Атрибут	Тип переменной
Код ошибки	Число
Текст ошибки	Строка(300)
Id узла	Строка(20)
Id запуска	Число
Текущие дата и время	Дата/время

Инфологическая модель данных представлена на рисунках 2.3, 2.4, 2.5.

2.5 Проектирование даталогической схемы базы данных

Даталогическое проектирование сводится к представлению инфологической модели в терминах выбранной системы управления базами данных (СУБД), т.е. даталогическая модель описывает собственно данные и связи между данными [23].

В экземпляре БД созданы 3 схемы:

- основная схема (MAIN);
- справочники (DICTS);
- витрины (MARTS).

В СУБД заведены 3 группы пользователей:

- СПР;
- риск-технологии;
- банк.

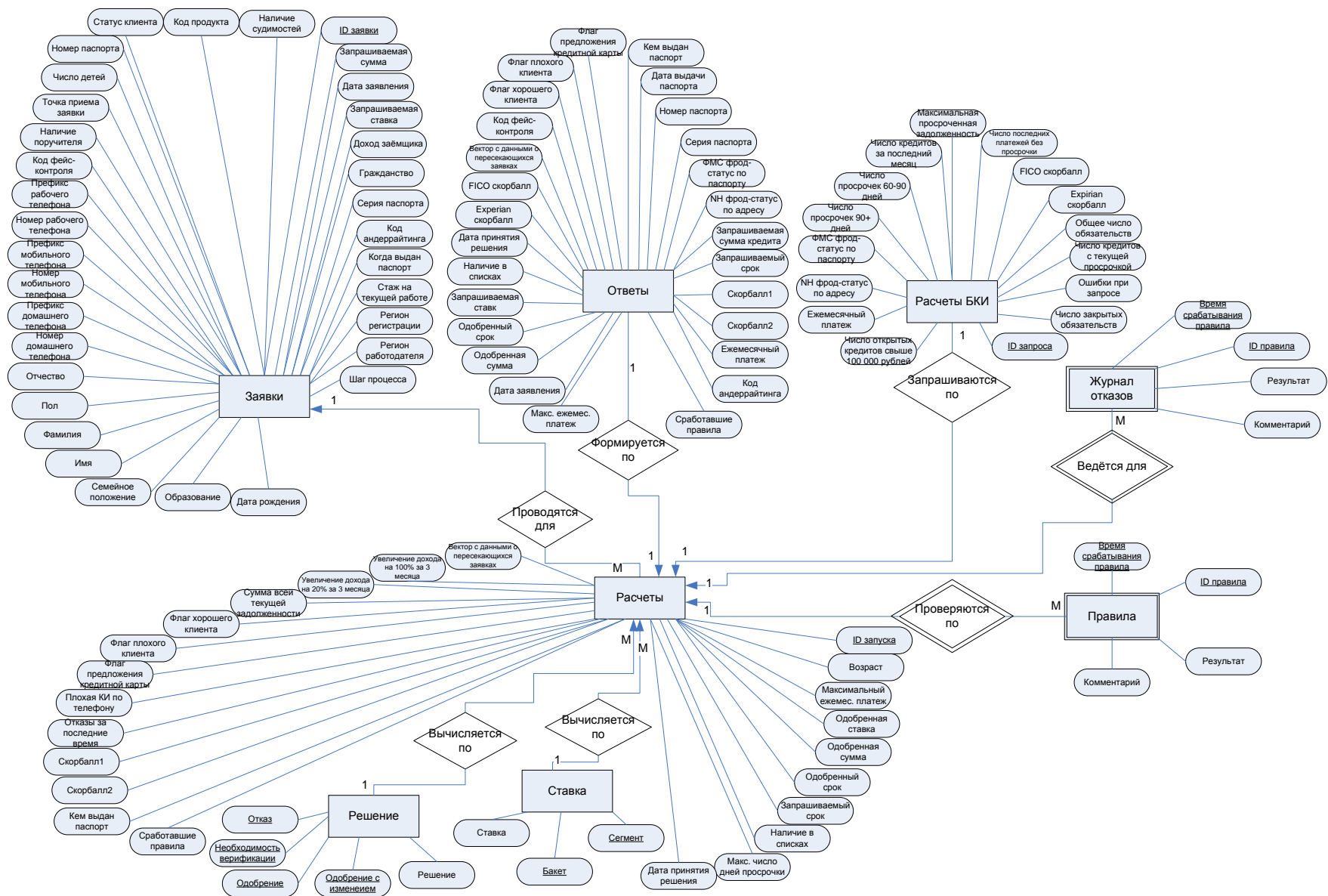


Рисунок 2.3 – Инфологическая модель данных

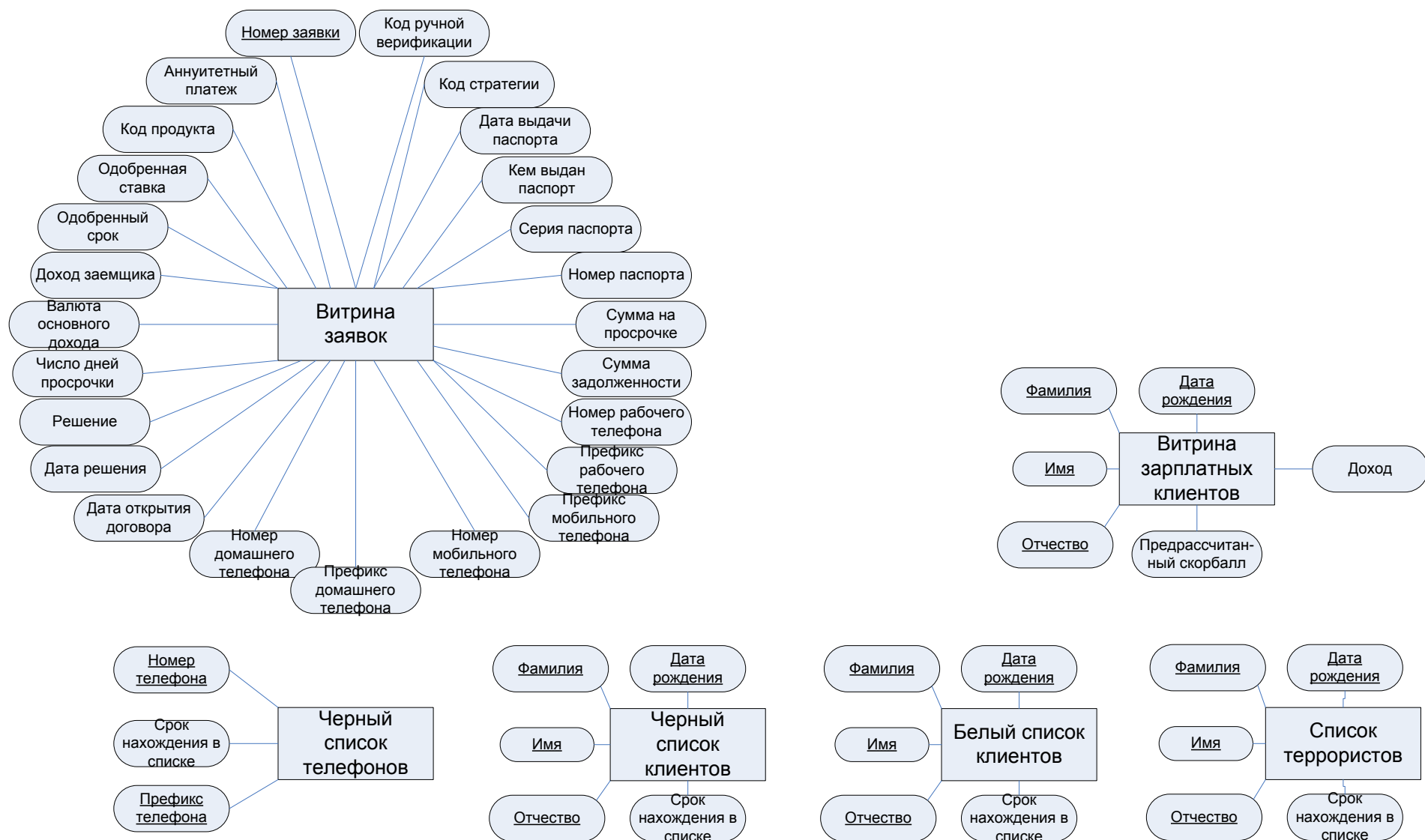


Рисунок 2.4 – Инфологическая модель данных (продолжение)

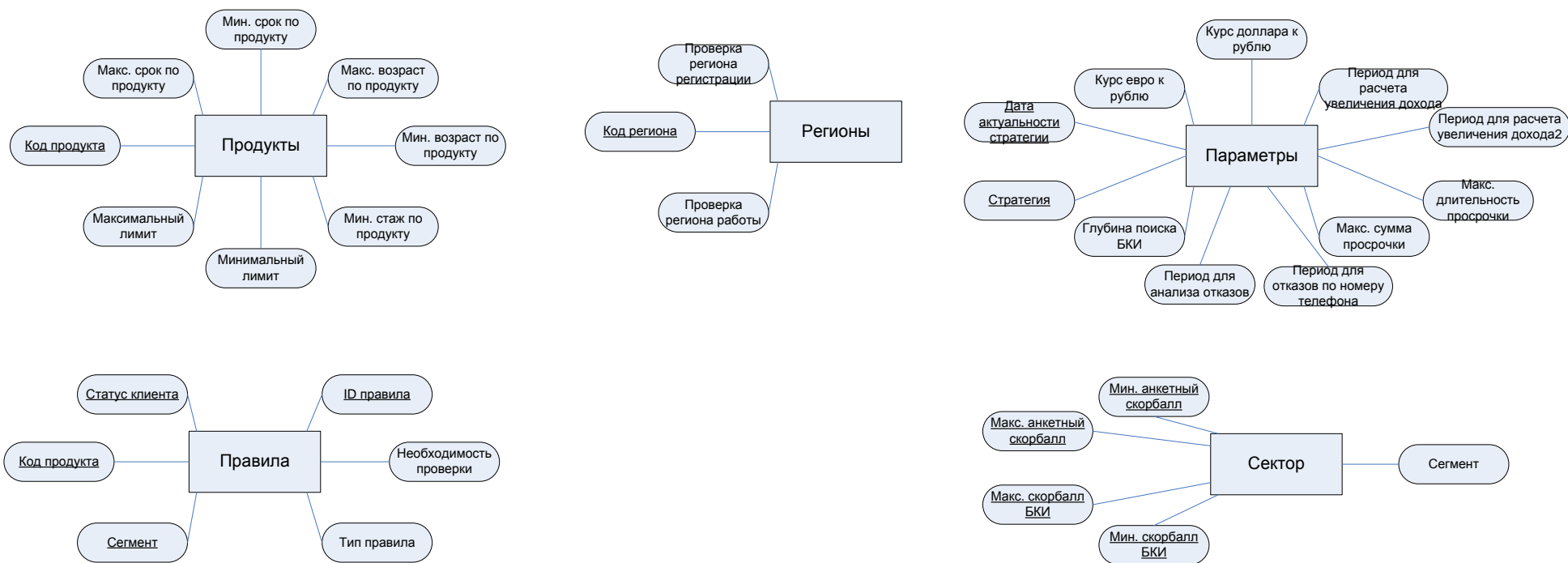


Рисунок 2.5 – Инфологическая модель данных (продолжение)

Схема MAIN является основной операционной схемой СПР. В ней хранятся таблицы заявок, расчётов, результатов проверок правил и журнализации. Для пользователя СПР данная схема предоставлена на чтение, изменение и добавление; для пользователей группы «риск-технологии» схема предоставлена на чтение, изменение, добавление и удаление; для пользователей группы «банк» схема предоставлена только на чтение.

Схема DICTS хранит таблицы справочников и настроек стратегии. Для пользователя СПР данная схема предоставлена только на чтение; для пользователей группы «риск-технологии» схема предоставлена на чтение, изменение, добавление и удаление; для пользователей группы «банк» схема предоставлена только на чтение.

Схема MARTS хранит таблицы витрины данных банка и списки банка. Для пользователя СПР данная схема предоставлена только на чтение; для пользователей группы «риск-технологии» схема предоставлена только на чтение; для пользователей группы «банк» схема предоставлена на чтение, изменение, добавление и удаление. Данные в таблицы данной схемы загружаются сотрудниками банка в рамках регламентного ETL-процесса.

Сущности БД представлены в таблице 2.28. Атрибуты данных таблиц представлены в таблицах 2.29–2.46.

Таблица 2.28 – Описание сущностей даталогической модели

Сущность	Описание	Ключ сущности (PK)	Физическое название таблицы	Схема БД
Справочник необходимо сти проверки правил	Справочник содержит в себе флаги необходимости проверки правил.	CreditProgramCode, CustomerStatus, Sector, Rule_id	DICT_RULE_CHECK	DICTS

Сущность	Описание	Ключ сущности (PK)	Физическое название таблицы	Схема БД
Справочник параметров кредитной стратегии	Справочник, содержащий параметры стратегии для проверки правил и расчета риск-переменных	Strategy, StrategyDate	DICT _PARAMETERS	DICTS
Справочник сегмента	Справочник зависимости сегмента от скорбаллов	BKI_SCORE_MIN, BKI_SCORE_MAX, APP_SCORE_MIN, APP_SCORE_MAX	DICT_SECTOR	DICTS
Справочник регионов присутствия	Таблица содержит список регионов, в которых производится выдача при наличии места работы или места постоянной регистрации	Region_code	DICT_REGION	DICTS
Справочник определения процентной ставки	Справочник определения процентной ставки	Node, sector	DICT _INTEREST _RATE	DICTS
Справочник для принятия решения	Справочник определения промежуточного решения по заявке	Decline, Approve, Approve_Change, Refer	DICT _DECISION	DICTS
Справочник параметров кредитных продуктов	Справочник содержит параметры для каждого продукта кредитования	CreditProgramCode	DICT _PRODUCT	DICTS

Сущность	Описание	Ключ сущности (PK)	Физическое название таблицы	Схема БД
Заявка	Данные по заявке, полученные из СОЗ	APP_ID	APPLICATIONS	MAIN
Результаты проверок правил	Таблица содержит результат проверки правил	Run_id, Rule_id, RuleTime	RULE_CHECK	MAIN
Результаты расчётов	Таблица содержит промежуточные и окончательные результаты вычислений (параметры для лимита, риск- переменные и прочие расчетные показатели)	Run_id	APP_RESULTS	MAIN
Журнал отказов	Таблица содержит ошибки, возникшие при обработке заявки	Run_id, FaultTime	SAS_LOG	MAIN
Расчетные атрибуты по обязательствам ам БКИ	Таблица содержит расчетные атрибуты по обязательствам. полученным из БКИ	Request_ID	BKI_RESULTS	MAIN
Ответ в СОЗ	Таблица, на основании полей которой будет генерироваться ответ в СОЗ	RUN_ID	RESPONSE	MAIN

Сущность	Описание	Ключ сущности (PK)	Физическое название таблицы	Схема БД
Скор балл для клиентов с заработной платой	Витрина содержит рассчитанный скоринговый балл и информацию о поступлениях на зарплатную карту за последние 6 месяцев	Name, Sirname, Patronymic, BirthDate	PAYROLL _MART	MART S
Витрина заявок банка	Витрина содержит данные по заявкам и договорам в банке	APP_ID	APP_MART	MART S
Черный список клиентов банка	Список клиентов банка, которым будет отказано в выдаче кредита	NameBl, SirnameBl, PatronymicBl, BirthDateBl	LIST_BLACK	MART S
Белый список клиентов банка	Список клиентов банка, которым будет одобрена выдача кредита	NameWh, SirnameWh, PatronymicWh, BirthDateWh	LIST_WHITE	MART S
Список террористов	Список клиентов банка, которые признаны террористами и которым будет отказано в получении кредита	NameTerr, SirnameTerr, PatronymicTerr, BirthDateTerr	LIST_TERR	MART S
Черный список телефонов	Номер телефонов, владельцам которых будет отказано в выдаче кредита	PhoneNumberBl, PhonePrefixBl	LIST_BLACK _PHONES	MART S

Таблица 2.29 – Атрибуты даталогической модели данных таблицы «Витрина заявок»

Атрибут	Физическое имя	Тип переменной
Номер заявки СОЗ (РК)	App_ID	Varchar2(20)
Аннуитетный платеж	Payment	Number
Код продукта	CreditProgramCode	Number
Одобренная ставка	Rate	Number
Одобренный срок	Term	Number
Доход заемщика (основной)	Income	Number
Валюта основного дохода	IncomeCurrency	Number
Количество дней просрочки	DaysPastDue	Number
Решение	Decision	Varchar2(50)
Дата решения	DecisionDate	date
Дата открытия договора	AppDate	date
Номер домашнего телефона	HomePhoneNumber	Varchar2(300)
Префикс домашнего телефона	HomePhonePrefix	Number
Сумма задолженности (баланс + просрочка)	Loan	Number
Номер мобильного телефона	MobPhoneNumber	Varchar2(300)
Префикс мобильного телефона	MobPhonePrefix	Number
Сумма на просрочке	Overdueamt	Number
Номер паспорта	PasportNumber	Varchar2(100)
Серия паспорта	PasportSeries	Varchar2(100)
Кем выдан паспорт	PasportGivenBy	Varchar2(300)
Дата выдачи паспорта	PasportIssueDate	Date
Код решения (стратегия)	ScoringCode	Varchar2(300)
Код УБ (ручной верификации)	SecurityCode	Varchar2(1000)
Номер рабочего телефона	WorkPhoneNumber	Varchar2(300)
Префикс рабочего телефона	WorkPhonePrefix	Number

Таблица 2.30 – Атрибуты даталогической модели данных таблицы «Результаты расчетов»

Атрибут	Физическое имя	Тип переменной
Возраст	Age	Number
Максимально допустимый ежемесячный платеж по кредиту	AnnuityMax	Number
Одобренная сумма	ApprovedAmount	Number
Одобренный срок	ApprovedTerm	Number
Наличие в списках (террористы, ФМС, черные списки и белые списки банка)	Lists	Varchar2(100)
Максимальное количество дней просрочки по кредиту в банке	MaxDue	Number
Дата принятия решения	DecisionDate	Date
Есть одобрение	Has_approve	Number
Есть одобрение с изменениями	Has_approve_change	Number
Есть отказ	Has_decline	Number
Есть необходимость верификации	Has_refer	Number
Вектор, содержащий данные о найденных пересекающихся заявках	PrevApplications	Varchar2(3000)
Id заявки (FK)	APP_ID	Varchar2(300)
Доход увеличился более чем на 100% за 3 месяца	IncomeGrow1	Number
Доход увеличился более чем на 20% за 3 месяца	IncomeGrow2	Number
Сумма всей текущей задолженности	SumArrear	Number
Идентификатор хорошего клиента	GoodClient	Number
Идентификатор плохого клиента	BadClient	Number
Бакет	Node	Number
Флаг предложения кредитной карты	OfferCreditCard	Number

Атрибут	Физическое имя	Тип переменной
Плохая кредитная история у других заемщиков с тем же мобильным или домашним телефоном	BadPhoneHist	Number
Количество отказов за последние DICT_PARAMETERS.Prev_decl_period дней	PrevDeclines	Number
Id запуска (PK)	Run_id	Number
Скоринговый балл по карте 1	APP_Score1	Number
Скоринговый балл по карте 2	APP_Score2	Number
Сегмент	Sector	Number
Список сработавших правил	ScoringCode	Varchar(2000)
Идентификатор запроса во внешний сервис (FK)	Request_ID	Number

Таблица 2.31 – Атрибуты даталогической модели данных таблицы «Заявки»

Атрибут	Физическое имя	Тип переменной
Дата заявления	ApplicationDate	Date
Запрашиваемая ставка	ReqRate	Number
Доход заёмщика	Income	Number
Валюта дохода	Currency	Number
Гражданство	Citizenship	Number
Наличие судимостей	Convictions	Number
Код продукта	CreditProgramCode	Number
Статус клиента	CustomerStatus	Number
Дата рождения	BirthDate	Date
Образование	Education	Number
Семейное положение	FamilyStatus	Number
Имя	Name	Varchar2(300)
Пол	Gender	Number
Домашний телефон (номер)	HomePhoneNumber	Varchar2(300)
Домашний телефон(префикс)	HomePhonePrefix	Number
Id заявки (PK)	APP_ID	Varchar2(300)

Атрибут	Физическое имя	Тип переменной
Фамилия	Sirname	Varchar2(300)
Код фейс-контроля	ManagerCode	Number
Отчество	Patronymic	Varchar2(300)
Мобильный телефон(номер)	MobPhoneNumber	Varchar2(300)
Мобильные телефон(префикс)	MobPhonePrefix	Number
Количество детей	Children	Number
Кем выдан паспорт	PasportGivenBy	Varchar2(300)
Дата выдачи паспорта	PasportIssueDate	Date
Номер паспорта	PasportNumber	Varchar2(100)
Серия паспорта	PasportSeries	Varchar2(100)
Стаж на текущей работе	WorkExperience	Number
Наличие поручителя	Poruchit	Number
Код точки приема заявки	POSCode	Varchar2(100)
Регион регистрации	AdressRegion	Number
Запрашиваемая сумма кредита	ReqAmount	Number
Запрашиваемый срок кредита	ReqTerm	Number
Шаг процесса	Stage	Number
Код андеррайтинга	UWcode	Number
Рабочий телефон(номер)	WorkPhoneNumber	Varchar2(300)
Рабочий телефон(префикс)	WorkPhonePrefix	Number
Регион работодателя	WorkRegion	Number

Таблица 2.32 – Атрибуты даталогической модели данных таблицы «Расчеты по БКИ»

Атрибут	Физическое имя	Тип переменной
ID запроса (PK)	Request_ID	Number
Общее количество закрытых кредитных обязательств	ClosedCount	Number
Число ошибок при запросе в БКИ	BKIRequestErrors	Number
Количество кредитов с текущей просрочкой	CurOverdueCount	Number

Атрибут	Физическое имя	Тип переменной
Общее количество кредитных обязательств	DealCount	Number
Experian скоринговый бал	ExperianScore	Number
FICO скоринговый бал	FICOscore	Number
Количество последних платежей без просрочки	GoodPlat	Number
Максимальный объем просроченной задолженности	MaxOverdueAmount	Number
Количество открытых кредитов за последний месяц	Overdue32Count	Number
Количество просрочек 60-90 по кредитной истории БКИ	Overdue60Count	Number
Количество просрочек 90+ по кредитной истории БКИ	Overdue90PlusCount	Number
ФМС фрод-статус по паспорту	PaspIdStatus	Number
НН фрод-статус по адресу регистрации	RegAddrStatus	Number
Ежемесячный платеж	SummaryPayment	Number
Количество непогашенных кредитов свыше 100000рублей	WOCCge100	Number

Таблица 2.33 – Атрибуты даталогической модели данных таблицы «Справочник «Решение»

Атрибут	Физическое имя	Тип переменной
Сработало правило отказа (PK)	Decline	Number
Сработало правило одобрения (PK)	Approve	Number
Сработало правило одобрения с изменением (PK)	Approve_Change	Number
Сработало правило необходимости верификации (PK)	Refer	Number
Решение	Decision	Varchar(32)

Таблица 2.34 – Атрибуты даталогической модели данных таблицы «Справочник «Ставка»

Атрибут	Физическое имя	Тип переменной
Ставка	ApprovedRate	Number
Бакет (PK)	Node	Number
Сегмент (PK)	Sector	Number

Таблица 2.35 – Атрибуты даталогической модели данных таблицы «Справочник «Параметры системы»

Атрибут	Физическое имя	Тип переменной
Стратегия (PK)	Strategy	Varchar2(64)
Дата актуальности стратегии (PK)	StrategyDate	Date
Курс евро к рублю для БКИ	EUR_Rate	Number
Курс доллара к рублю для БКИ	USD_Rate	Number
Период, за который рассчитывается увеличение дохода	Inc_grow_period	Number
Период, за который рассчитывается увеличение дохода	Inc_grow_period_02	Number
Максимально допустимая длительность просрочки	Overdue_count	Number
Максимально допустимая сумма просрочки	Overdue_sum	Number
Период, за который учитывается отказ по тому же номеру телефона	Ph_badhist_period	Number
Период за который вычисляется количество отказов	Prev_decl_period	Number
Глубина поиска БКИ в днях	BKISearchDepth	Varchar2(64)

Таблица 2.36 – Атрибуты даталогической модели данных таблицы «Справочник «Продукты»

Атрибут	Физическое имя	Тип переменной
Код продукта (PK)	CreditProgramCode	Number
Максимальный срок по продукту	MaxTerm	Number
Минимальный срок по продукту	MinTerm	Number

Атрибут	Физическое имя	Тип переменной
Минимальный общий стаж по продукту	MinExper	Number
Максимальный возраст по продукту	MaxAge	Number
Минимальный возраст по продукту	MinAge	Number
Максимальный лимит кредитования для продукта	MaxCreditLimit	Number
Минимальный лимит кредитования для продукта	MinCreditLimit	Number

Таблица 2.37 – Атрибуты даталогической модели данных таблицы «Справочник «Регионы»

Атрибут	Физическое имя	Тип переменной
Проверка адреса регистрации	CHECK_AR	Number
Проверка адреса работы	CHECK_WORK	Number
Код региона (РК)	RegionCode	Number

Таблица 2.38 – Атрибуты даталогической модели данных таблицы «Справочник «Правила»

Атрибут	Физическое имя	Тип переменной
Программа кредитования (РК)	CreditProgram_code	Number
Статус клиента (РК)	CustomerStatus	Varchar2(64)
Необходимость проверки правила	Is_check	Number
Код правила (РК)	Rule_id	Varchar2(64)
Тип правила	Rule_type	Varchar2(64)
Сегмент (РК)	Sector	Varchar2(64)

Таблица 2.39 – Атрибуты даталогической модели данных таблицы «Справочник «Сектор»

Атрибут	Физическое имя	Тип переменной
Максимальное значение анкетного скорбалла (РК)	APP_SCORE_MAX	Number
Минимальное значение анкетного скорбалла (РК)	APP_SCORE_MIN	Number
Максимальное значение скорбалла БКИ (РК)	BKI_SCORE_MAX	Number
Минимальное значение скорбалла БКИ (РК)	BKI_SCORE_MIN	Number
Сегмент	Sector	Number

Таблица 2.40 – Атрибуты даталогической модели данных таблицы «Черный список клиентов»

Атрибут	Физическое имя	Тип переменной
Имя (РК)	NameBl	Varchar2(300)
Фамилия (РК)	SirnameBl	Varchar2(300)
Отчество (РК)	PatronymicBl	Varchar2(300)
Дата рождения (РК)	BirthDateBl	Date
Срок нахождения в списке	ListDate	Date

Таблица 2.41 – Атрибуты даталогической модели данных таблицы «Черный список телефонов»

Атрибут	Физическое имя	Тип переменной
Номер телефона (РК)	PhoneNumberBl	Varchar2(300)
Префикс (РК)	PhonePrefixBl	Number
Срок нахождения в списке	ListDate	Date

Таблица 2.42 – Атрибуты даталогической модели данных таблицы «Список террористов»

Атрибут	Физическое имя	Тип переменной
Имя (РК)	NameTerr	Varchar2(300)
Фамилия (РК)	SirnameTerr	Varchar2(300)
Отчество (РК)	PatronymicTerr	Varchar2(300)

Атрибут	Физическое имя	Тип переменной
Дата рождения (PK)	BirthDateTerr	Date
Срок нахождения в списке	ListDate	Date

Таблица 2.42 – Атрибуты даталогической модели данных таблицы «Белый список клиентов»

Атрибут	Физическое имя	Тип переменной
Имя (PK)	NameWh	Varchar2(300)
Фамилия (PK)	SirnameWh	Varchar2(300)
Отчество (PK)	PatronymicWh	Varchar2(300)
Дата рождения (PK)	BirthDateWh	Date
Срок нахождения в списке	ListDate	Date

Таблица 2.43 – Атрибуты даталогической модели данных таблицы «Витрина зарплатных клиентов»

Атрибут	Физическое имя	Тип переменной
Имя (PK)	Name	Varchar2(300)
Фамилия (PK)	Sirname	Varchar2(300)
Отчество (PK)	Patronymic	Varchar2(300)
Дата рождения (PK)	BirthDate	Date
Доход	IncomePayroll	Number
Предрасчитанный скорбалл	PayrollScore	Number

Таблица 2.44 – Атрибуты даталогической модели данных таблицы «Ответ»

Атрибут	Физическое имя	Тип переменной
Максимально допустимый ежемесячный платеж по кредиту	AnnuityMax	Number
Дата заявления	ApplicationDate	Date
Одобренная сумма	ApprovedAmount	Number
Одобренный срок	ApprovedTerm	Number
Запрашиваемая ставка	ReqRate	Number
Наличие в списках (террористы, ФМС, черные списки и белые списки банка)	Lists	Varchar2(100)

Атрибут	Физическое имя	Тип переменной
Код продукта	CreditProgramCode	Number
Дата принятия решения	DecisionDate	Date
Experian скоринговый бал	ExperianScore	Number
FICO скоринговый бал	FICOscore	Number
Вектор, содержащий данные о найденных пересекающихся заявках	PrevApplications	Varchar2(3000)
Id заявки	APP_ID	Varchar2(300)
Код фейс-контроля	ManagerCode	Number
Идентификатор хорошего клиента	GoodClient	Number
Идентификатор плохого клиента	BadClient	Number
Флаг предложения кредитной карты	OfferCreditCard	Number
Кем выдан паспорт	PasportGivenBy	Varchar2(300)
Дата выдачи паспорта	PasportIssueDate	Date
Номер паспорта	PasportNumber	Varchar2(100)
Серия паспорта	PasportSeries	Varchar2(100)
ФМС фрод-статус по паспорту	PaspIdStatus	Number
НН фрод-статус по адресу регистрации	RegAddrStatus	Number
Запрашиваемая сумма кредита	ReqAmount	Number
Запрашиваемый срок кредита	ReqTerm	Number
Id запуска (PK) (FK)	Run_ID	Number
Скоринговый балл по карте 1	APP_Score1	Number
Скоринговый балл по карте 2	APP_Score2	Number
Ежемесячный платеж	SummaryPayment	Number
Код андеррайтинга	UWcode	Number
Список сработавших правил	ScoringCode	Varchar2(2000)

Таблица 2.45 – Атрибуты даталогической модели данных таблицы «Сработавшие правила»

Атрибут	Физическое имя	Тип переменной
Комментарий	Description	Varchar2(300)
Результат проверки правила	Result	Number
Id правила (PK)	Rule_id	Varchar2(64)
Id запуска (PK) (FK)	Run_id	Number
Дата и время срабатывания правила (PK)	RuleTime	Timestamp

Таблица 2.46 – Атрибуты даталогической модели данных таблицы «Журнал отказов»

Атрибут	Физическое имя	Тип переменной
Код ошибки	ErrorCode	Number
Текст ошибки	ErrorText	Varchar2(300)
Id узла	NodeId	Varchar2(20)
Id запуска (PK) (FK)	Run_id	Number
Текущие дата и время (PK)	FaultTime	Timestamp

2.6 Оптимизация базы данных

2.6.1 Схема MARTS

Рассмотрим схему MARTS и покажем, что каждое отношение находится в 3НФи НФБК.

$$\rho_{MARTS} = (R_1, R_2, R_3, R_4, R_5, R_6)$$

$$R_1 = G_1 G_2 G_3;$$

$$R_2 = H_1 H_2 H_3 H_4 H_5;$$

$$R_3 = I_1 I_2 I_3 I_4 I_5;$$

$$R_4 = J_1 J_2 J_3 J_4 J_5;$$

$$R_5 = K_1 K_2 K_3 K_4 K_5 K_6 K_7 K_8 K_9 K_{10} K_{11} K_{12} K_{13} K_{14} K_{15} K_{16} K_{17} K_{18} K_{19} K_{20} K_{21} K_{22} K_{23} K_{24};$$

$$R_6 = L_1 L_2 L_3 L_4 L_5 L_6;$$

Модель данных представлены в таблице 2.47.

Таблица 2.47 – Модель данных схемы MARTS

R₁ (Черный список телефонов)	
Номер телефона	G ₁
Префикс	G ₂
Срок нахождения в списке	G ₃
R₂ (Список террористов)	
Имя	H ₁
Фамилия	H ₂
Отчество	H ₃
Дата рождения	H ₄
Срок нахождения в списке	H ₅
R₃ (Черный список клиентов)	
Имя	I ₁
Фамилия	I ₂
Отчество	I ₃
Дата рождения	I ₄
Срок нахождения в списке	I ₅
R₄ (Белый список клиентов)	
Имя	J ₁
Фамилия	J ₂
Отчество	J ₃
Дата рождения	J ₄
Срок нахождения в списке	J ₅
R₅ (Витрина заявок)	
Номер заявки СОЗ	K ₁
Аннуитетный платеж	K ₂
Код продукта	K ₃
Одобренная ставка	K ₄
Одобренный срок	K ₅
Доход заемщика(основной)	K ₆
Валюта основного дохода	K ₇
Количество дней просрочки	K ₈
Решение	K ₉
Дата решения	K ₁₀
Дата открытия договора	K ₁₁
Номер домашнего телефона	K ₁₂
Префикс домашнего телефона	K ₁₃

Сумма задолженности (баланс+просрочка)	K ₁₄
Номер мобильного телефона	K ₁₅
Префикс мобильного телефона	K ₁₆
Сумма на просрочке	K ₁₇
Номер паспорта	K ₁₈
Серия паспорта	K ₁₉
Кем выдан паспорт	K ₂₀
Дата выдачи паспорта	K ₂₁
Код решения (стратегия)	K ₂₂
Код УБ (ручной верификации)	K ₂₃
Номер рабочего телефона	K ₂₄
R₆ (Витрина зарплатных клиентов)	
Имя	L ₁
Фамилия	L ₂
Отчество	L ₃
Дата рождения	L ₄
Доход	L ₅
Предрасчитанный скорбалл	L ₆

Используемые ФЗ: $F = (G_1 G_2 \rightarrow G_3;$

$H_1 H_2 H_3 H_4 \rightarrow H_5;$

$I_1 I_2 I_3 I_4 \rightarrow I_5;$

$J_1 J_2 J_3 J_4 \rightarrow J_5;$

$K_1 \rightarrow K_2 K_3 K_4 K_5 K_6 K_7 K_8 K_9 K_{10} K_{11} K_{12} K_{13} K_{14} K_{15} K_{16} K_{17} K_{18} K_{19} K_{20} K_{21} K_{22} K_{23} K_{24};$

$L_1 L_2 L_3 L_4 \rightarrow L_5 L_6).$

Отношение находится в 3НФ, если нельзя найти атрибуты X, Y, H, где X – ключ, Y – подмножество атрибутов отношения, H – подмножество не ключевых атрибутов, такие, что выполняются следующие условия:

- $X \rightarrow Y$
- $Y \nrightarrow X$
- $Y \rightarrow H$

Рассмотрим отношение R₁: $F_{R1} = (G_1 G_2 \rightarrow G_3)$

$X = G_1 G_2$

$H = G_3$

$Y \rightarrow H \Rightarrow Y = G_1 G_2$, но в этом случае: $Y \rightarrow X$, таким образом, отношение находится в 3НФ.

Так как в данном отношении присутствует лишь одна ФЗ, где в левой части стоит ключ, то отношение также находится в НФБК.

Рассмотрим отношение R₂: $F_{R2} = (H_1H_2H_3H_4 \rightarrow H_5)$

$$X = H_1H_2H_3H_4$$

$$H = H_5$$

$Y \rightarrow H \Rightarrow Y = H_1H_2H_3H_4$, но в этом случае: $Y \rightarrow X$, таким образом, отношение находится в 3НФ.

Так как в данном отношении присутствует лишь одна ФЗ, где в левой части стоит ключ, то отношение также находится в НФБК.

Рассмотрим отношение R₃: $F_{R3} = (I_1I_2I_3I_4 \rightarrow I_5)$

$$X = I_1I_2I_3I_4$$

$$H = I_5$$

$Y \rightarrow H \Rightarrow Y = I_1I_2I_3I_4$, но в этом случае: $Y \rightarrow X$, таким образом, отношение находится в 3НФ.

Так как в данном отношении присутствует лишь одна ФЗ, где в левой части стоит ключ, то отношение также находится в НФБК.

Рассмотрим отношение R₄: $F_{R4} = (J_1J_2J_3J_4 \rightarrow J_5)$

$$X = J_1J_2J_3J_4$$

$$H = J_5$$

$Y \rightarrow H \Rightarrow Y = J_1J_2J_3J_4$, но в этом случае: $Y \rightarrow X$, таким образом, отношение находится в 3НФ.

Так как в данном отношении присутствует лишь одна ФЗ, где в левой части стоит ключ, то отношение также находится в НФБК.

Рассмотрим отношение R₅: $F_{R5} = (K_1 \rightarrow K_2K_3K_4K_5K_6K_7K_8K_9K_{10}K_{11}K_{12}K_{13}K_{14}K_{15}K_{16}K_{17}K_{18}K_{19}K_{20}K_{21}K_{22}K_{23}K_{24})$

$$X = K_1$$

Для любого подмножества неключевых атрибутов H, $Y \rightarrow H \Rightarrow Y = K_1$, но в этом случае: $Y \rightarrow X$, таким образом, отношение находится в 3НФ.

Так как в данном отношении присутствует лишь одна ФЗ, где в левой части стоит ключ, то отношение также находится в НФБК.

Рассмотрим отношение $R_6: F_{R_6} = (L_1L_2L_3L_4 \rightarrow L_5L_6)$

$$X = L_1L_2L_3L_4$$

Для любого подмножества неключевых атрибутов H , $Y \rightarrow H \Rightarrow Y = L_1L_2L_3L_4$, но в этом случае: $Y \rightarrow X$, таким образом, отношение находится в 3НФ.

Так как в данном отношении присутствует лишь одна ФЗ, где в левой части стоит ключ, то отношение также находится в НФБК.

Итоговая схема отношения представлена на рисунке 2.6.

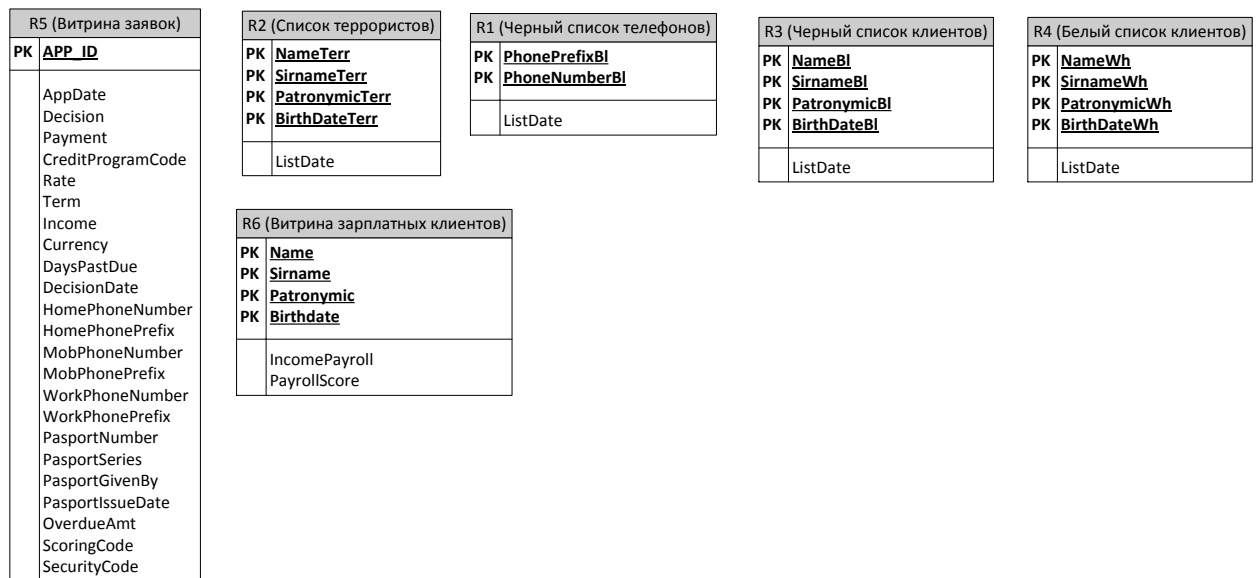


Рисунок 2.6 – Схема отношения MARTS

2.6.2 Схема DICTS

Рассмотрим схему DICTS и покажем, что каждое отношение находится в 3НФи НФБК.

$$\rho_{DICTS} = (R_1, R_2, R_3, R_4, R_5, R_6, R_7)$$

$$R_1 = A_7B_6B_7;$$

$$R_2 = B_1B_2B_3B_4B_5;$$

$$R_3 = B_8B_9B_{10};$$

$$R_4 = A_8A_9B_8C_1C_2C_3;$$

$$R_5 = D_1D_2D_3D_4B_8;$$

$$R_6 = E_1E_2E_3E_4E_5E_6E_7E_8E_9E_{10}E_{11};$$

$$R_7 = A_8M_1M_2M_3M_4M_5M_6M_7;$$

Модель данных представлена в таблице 2.48.

Таблица 2.48 – Модель данных схемы DICTS

R₁(Справочник «Регионы»)	
Код региона	A ₇
Проверка адреса регистрации	B ₆
Проверка адреса работы	B ₇
R₂ (Справочник «Решение»)	
Отказа	B ₁
Одобрение	B ₂
Одобрение с изменением	B ₃
Необходимость верификации	B ₄
Решение	B ₅
R₃ (Справочник «Ставка»)	
Ставка	B ₈
Бакет	B ₉
Сегмент	B ₁₀
R₄ (Справочник «Правила»)	
Код правила	C ₁
Программа кредитования	A ₈
Статус клиента	A ₉
Сегмент	B ₈
Тип правила	C ₂
Необходимость проверки правила	C ₃
R₅ (Справочник «Сектор»)	
Макс. значение анкетного скорбалла	D ₁
Мин. значение анкетного скорбалла	D ₂
Макс. значение скорбаллабки	D ₃
Мин. значение скорбаллабки	D ₄
Сегмент	B ₈
R₆ (Справочник «Параметры»)	
Стратегия	E ₁
Дата актуальности стратегии	E ₂
Курс евро к рублю для бки	E ₃
Курс доллара к рублю для бки	E ₄
Период, за который рассчитывается увеличение дохода	E ₅

Период, за который рассчитывается увеличение дохода ²	E ₆
Максимально допустимая длительность просрочки	E ₇
Максимально допустимая сумма просрочки	E ₈
Период, за который учитывается отказ по тому же номеру телефона	E ₉
Период за который анализируется количество отказов	E ₁₀
Глубина поиска БКИ в днях	E ₁₁
R₇ (Справочник «Продукты»)	
Код продукта	A ₈
Max срок по продукту	M ₁
Min срок по продукту	M ₂
Min общий стаж по продукту	M ₃
Max возраст по продукту	M ₄
Min возраст по продукту	M ₅
Максимальный лимит кредитования для продукта	M ₆
Минимальный лимит кредитования для продукта	M ₇

Используемые ФЗ: $F = (A_7 \rightarrow B_6 B_7;$

$A_8 A_9 B_8 C_1 \rightarrow C_2 C_3;$

$B_8 B_9 \rightarrow B_{10};$

$B_1 B_2 B_3 B_4 \rightarrow B_5;$

$D_1 D_2 D_3 D_4 \rightarrow B_8;$

$E_1 E_2 \rightarrow E_3 E_4 E_5 E_6 E_7 E_8 E_9 E_{10} E_{11};$

$A_8 \rightarrow M_1 M_2 M_3 M_4 M_5 M_6 M_7).$

Отношение находится в 3НФ, если нельзя найти атрибуты X, Y, H, где X – ключ, Y – подмножество атрибутов отношения, H – подмножество не ключевых атрибутов, такие, что выполняются следующие условия:

- $X \rightarrow Y$
- $Y \nrightarrow X$
- $Y \rightarrow H$

Рассмотрим отношение R₁: $F_{R1} = (A_7 \rightarrow B_6 B_7)$

$X = A_7$

Для любого подмножества неключевых атрибутов H , $Y \rightarrow H \Rightarrow Y = A_7$, но в этом случае: $Y \rightarrow X$, таким образом, отношение находится в 3НФ.

Так как в данном отношении присутствует лишь одна ФЗ, где в левой части стоит ключ, то отношение также находится в НФБК.

Рассмотрим отношение R_2 : $F_{R2} = (A_8 A_9 B_8 C_1 \rightarrow C_2 C_3)$

$$X = A_8 A_9 B_8 C_1$$

Для любого подмножества неключевых атрибутов H , $Y \rightarrow H \Rightarrow Y = A_8 A_9 B_8 C_1$, но в этом случае: $Y \rightarrow X$, таким образом, отношение находится в 3НФ.

Так как в данном отношении присутствует лишь одна ФЗ, где в левой части стоит ключ, то отношение также находится в НФБК.

Рассмотрим отношение R_3 : $F_{R3} = (B_8 B_9 \rightarrow B_{10})$

$$X = B_8 B_9$$

$$H = B_{10}$$

$Y \rightarrow H \Rightarrow Y = B_8 B_9$, но в этом случае: $Y \rightarrow X$, таким образом, отношение находится в 3НФ.

Так как в данном отношении присутствует лишь одна ФЗ, где в левой части стоит ключ, то отношение также находится в НФБК.

Рассмотрим отношение R_4 : $F_{R4} = (B_1 B_2 B_3 B_4 \rightarrow B_5)$

$$X = B_1 B_2 B_3 B_4$$

$$H = B_5$$

$Y \rightarrow H \Rightarrow Y = B_1 B_2 B_3 B_4$, но в этом случае: $Y \rightarrow X$, таким образом, отношение находится в 3НФ.

Так как в данном отношении присутствует лишь одна ФЗ, где в левой части стоит ключ, то отношение также находится в НФБК.

Рассмотрим отношение R_5 : $F_{R5} = (D_1 D_2 D_3 D_4 \rightarrow B_8)$

$$X = D_1 D_2 D_3 D_4$$

$$H = B_8$$

$Y \rightarrow H \Rightarrow Y = D_1 D_2 D_3 D_4$, но в этом случае: $Y \rightarrow X$, таким образом, отношение находится в 3НФ.

Так как в данном отношении присутствует лишь одна ФЗ, где в левой части стоит ключ, то отношение также находится в НФБК.

Рассмотрим отношение R_6 : $F_{R6} = (E_1E_2 \rightarrow E_3E_4E_5E_6E_7E_8E_9E_{10}E_{11})$

$$X = E_1E_2$$

Для любого подмножества неключевых атрибутов H , $Y \rightarrow H \Rightarrow Y = A_8$, но в этом случае: $Y \rightarrow X$, таким образом, отношение находится в 3НФ.

Так как в данном отношении присутствует лишь одна ФЗ, где в левой части стоит ключ, то отношение также находится в НФБК.

Рассмотрим отношение R_7 : $F_{R7} = (A_8 \rightarrow M_1M_2M_3M_4M_5M_6M_7)$

$$X = A_8$$

Для любого подмножества неключевых атрибутов H , $Y \rightarrow H \Rightarrow Y = A_8$, но в этом случае: $Y \rightarrow X$, таким образом, отношение находится в 3НФ.

Так как в данном отношении присутствует лишь одна ФЗ, где в левой части стоит ключ, то отношение также находится в НФБК.

Итоговая схема отношения представлена на рисунке 2.7.

R7 (Справочник "Продукты") PK CreditProgramCode MaxTerm MinTerm MinExper MaxAge MinAge MaxCreditLimit MinCreditLimit	R4 (Справочник "Правила") PK CustomerStatus PK Sector PK RULE_ID RULE_TYPE IS_CHECK R1 (Справочник "Регионы") PK RegionCode CHECK_AR CHECK_WORK	R6 (Справочник "Параметры") PK Strategy PK StrategyDate BKISearchDepth EUR_Rate USD_Rate Inc_grow_period Inc_grow_period_02 Overdue_count Overdue_sum Ph_badhist_period Prev_decl_period	R2 (Справочник "Решение") PK DECLINE PK APPROVE PK APPROVE_CHANGE PK REFER Decision	R3 (Справочник "Ставка") PK Node PK Sector ApprovedRate	R5 (Справочник "Сектор") PK BKI_SCORE_MIN PK BKI_SCORE_MAX PK APP_SCORE_MIN PK APP_SCORE_MAX Sector
--	--	--	---	---	---

Рисунок 2.7 - Схема отношения DICTS

2.6.3 Схема MAIN

Модель данных, используемая в схеме MAIN представлена в таблице 2.49.

Таблица 2.49 – Модель данных схемы MAIN

Id запуска	A ₁
Номер заявки СОЗ	A ₂
Имя	A ₃
Фамилия	A ₄
Отчество	A ₅
Дата рождения	A ₆
Регион регистрации	A ₇
Код продукта	A ₈
Статус клиента	A ₉

Серия паспорта	A ₁₀
Номер паспорта	A ₁₁
Дата выдачи паспорта	A ₁₂
Регион работодателя	A ₁₃
Кем выдан	A ₁₄
Дата заявления	A ₁₅
Доход заемщика(основной)	A ₁₆
Валюта дохода	A ₁₇
Гражданство	A ₁₈
Наличие судимостей	A ₁₉
Дата рождения	A ₂₀
Образование	A ₂₁
Семейное положение	A ₂₂
Количество детей	A ₂₃
Номер домашнего телефона	A ₂₄
Префикс домашнего телефона	A ₂₅
Номер мобильного телефона	A ₂₆
Префикс мобильного телефона	A ₂₇
Номер рабочего телефона	A ₂₈
Префикс рабочего телефона	A ₂₉
Код фейс-контроля	A ₃₀
Стаж на текущей работе	A ₃₁
Наличие поручителя	A ₃₂
Код точки приема заявки	A ₃₃
Запрашиваемая сумма кредита	A ₃₄
Запрашиваемый срок кредита	A ₃₅
Запрашиваемая ставка	A ₃₆
Шаг процесса	A ₃₇
Код андеррайтинга	A ₃₈
Пол	A ₃₉
Есть отказ	B ₁
Есть одобрение	B ₂
Есть одобрение с изменениями	B ₃
Есть необходимость верификации	B ₄

Решение	B ₅
Проверка адреса регистрации	B ₆
Проверка адреса работы	B ₇
Сегмент, Актуальное состояние,	B ₈
Бакет	B ₉
Ставка	B ₁₀
Скор балл БКИ	B ₁₁
Скоринговый балл по карте 1	B ₁₂
Скоринговый балл по карте 2	B ₁₃
Возраст	B ₁₄
Одобренная сумма	B ₁₅
Одобренный срок	B ₁₆
Дата решения	B ₁₇
Вектор, содержащий данные о найденных пересекающихся заявках	B ₁₈
Доход увеличился более чем на 100% за 3 месяца	B ₁₉
Доход увеличился более чем на 20% за 3 мес.	B ₂₀
Макс. допустимый ежемесячный платеж по кредиту	B ₂₁
Доход заемщика(расчётный)	B ₂₂
Максимальное количество дней просрочки	B ₂₃
Сумма всей текущей задолженности	B ₂₄
Идентификатор хорошего клиента	B ₂₅
Идентификатор плохого клиента	B ₂₆
Флаг предложения кредитной карты	B ₂₇
Плохая кредитная история (по номеру телефона)	B ₂₈
Количество отказов за последние x дней	B ₂₉
Наличие в списках	B ₃₀
Код решения (стратегия)	B ₃₁
Код правила	C ₁
Тип правила	C ₂
Необходимость проверки правила	C ₃
Результат проверки правила	C ₄
Текущие дата и время	C ₅

Комментарий	C ₆
Текущие дата и время	F ₁
Id узла	F ₂
Код ошибки	F ₃
Текст ошибки	F ₄
Мах срок по продукту	M ₁
Min срок по продукту	M ₂
Min общий стаж по продукту	M ₃
Мах возраст по продукту	M ₄
Min возраст по продукту	M ₅
Максимальный лимит кредитования для продукта	M ₆
Минимальный лимит кредитования для продукта	M ₇
Идентификатор запроса во внешний сервис	N ₁
Общее количество закрытых кредитных обязательств	N ₂
Число ошибок при запросе в БКИ	N ₃
Количество кредитов с текущей просрочкой	N ₄
Общее количество кредитных обязательств	N ₅
Experian скоринговый бал	N ₆
FICO скоринговый бал	N ₇
Количество последних платежей без просрочки	N ₈
Максимальный объем просроченной задолженности	N ₉
Количество открытых кредитов за последний месяц	N ₁₀
Количество просрочек 60-90 по кредитной истории БКИ	N ₁₁
Количество просрочек 90+ по кредитной истории БКИ	N ₁₂
ФМС фрод-статус по паспорту	N ₁₃
НН фрод-статус по адресу регистрации	N ₁₄
Ежемесячный платеж	N ₁₅
Количество непогашенных кредитов свыше 100тыс. руб.	N ₁₆

Используемые ФЗ: $F(A_1A_2B_1B_2B_3B_4B_6B_7B_8B_9B_{11}B_{12}B_{13}B_{14}B_{15}B_{16}$
 $B_{17}B_{18}B_{19}B_{20}B_{21}B_{22}B_{23}B_{24}B_{25}B_{26}B_{27}B_{28}B_{29}B_{30}B_{31}N_1;$
 $A_2 \rightarrow A_7A_8A_9A_{10}A_{11}A_{12}A_{13}A_{14}A_{15}A_{16}A_{17}A_{18}A_{19}A_{20}A_{21}A_{22}A_{23}A_{24}A_{25}$
 $A_{26}A_{27}A_{28}A_{29}A_{30}A_{31}A_{32}A_{33}A_{34}A_{35}A_{36}A_{37}A_{38}A_{39};$
 $A_7 \rightarrow B_6;$
 $A_{13} \rightarrow B_7;$
 $A_{10}A_{11}A_{12}A_{14} \rightarrow A_3A_4A_5A_6A_{39};$
 $A_1F_1 \rightarrow F_2F_3F_4;$
 $A_1C_1C_5 \rightarrow C_2C_3C_4C_6;$
 $A_8A_9B_8C_1 \rightarrow C_2C_3;$
 $B_8B_9 \rightarrow B_{10};$
 $B_1B_2B_3B_4 \rightarrow B_5;$
 $A_8 \rightarrow M_1M_2M_3M_4M_5M_6M_7;$
 $N_1 \rightarrow N_2N_3N_4N_5N_6N_7N_8N_9N_{10}N_{11}N_{12}N_{13}N_{14}N_{15}N_{16}).$

Для оптимизации схемы MAIN воспользуемся алгоритмом GDBSS (ввиду большого объема выкладок, описание и реализация алгоритма вынесены в Приложение Б).

Итоговая схема отношения представлена на рисунке 2.9.

Сущности R_{12} (Справочник «Продукты»), R_8 (Справочник «Решение»), R_7 (Справочник «Ставка»), R_6 (Справочник «Правила») уже представлены в схеме DICTS, рассмотренной ранее. Сущности R_{10} (Справочник «Регион проживания») и R_{11} (Справочник «Регион работы») также уже представлены сущностью «Справочник «Регионы» схемы DICTS, рассмотренной ранее.

Так как сущность R_9 (Сущность соединения), используемая для связи журнала отказов и сработавших при обработке заявки бизнес-правил, не несёт в себе практического смысла, то она исключена из итоговой схемы БД.

Платформа SASRTDM для возможности автоматического принятия запроса в обработку требует, чтобы тело XML-файла запроса имело линейную структуры (без вложенности тегов). Также в БД должна иметься таблица, имена атрибутов которой полностью совпадают с именами тегов входного

XML-файла. В таком случае SAS RTDM автоматически сохраняет данные входного XML-файла в таблицу.

Таким образом, из-за требования к линейности тела XML-файла запроса нет возможности сохранять информацию о паспорте в отдельную таблицу «Паспорта». Вся информация по заявке будет храниться в таблице «Заявки».

Для автоматического формирования ответа платформе SASRTDM требуется отдельная таблица, на основании строки и имён атрибутов которой будет формироваться выходной XML-файл. Таким образом, добавится сущность «Ответ», в которой будут храниться значения тех атрибутов, которые необходимо возвращать в результате обработки.

С учётом указанных требований и замечаний схема БД имеет вид, представленный на рисунке 2.10.

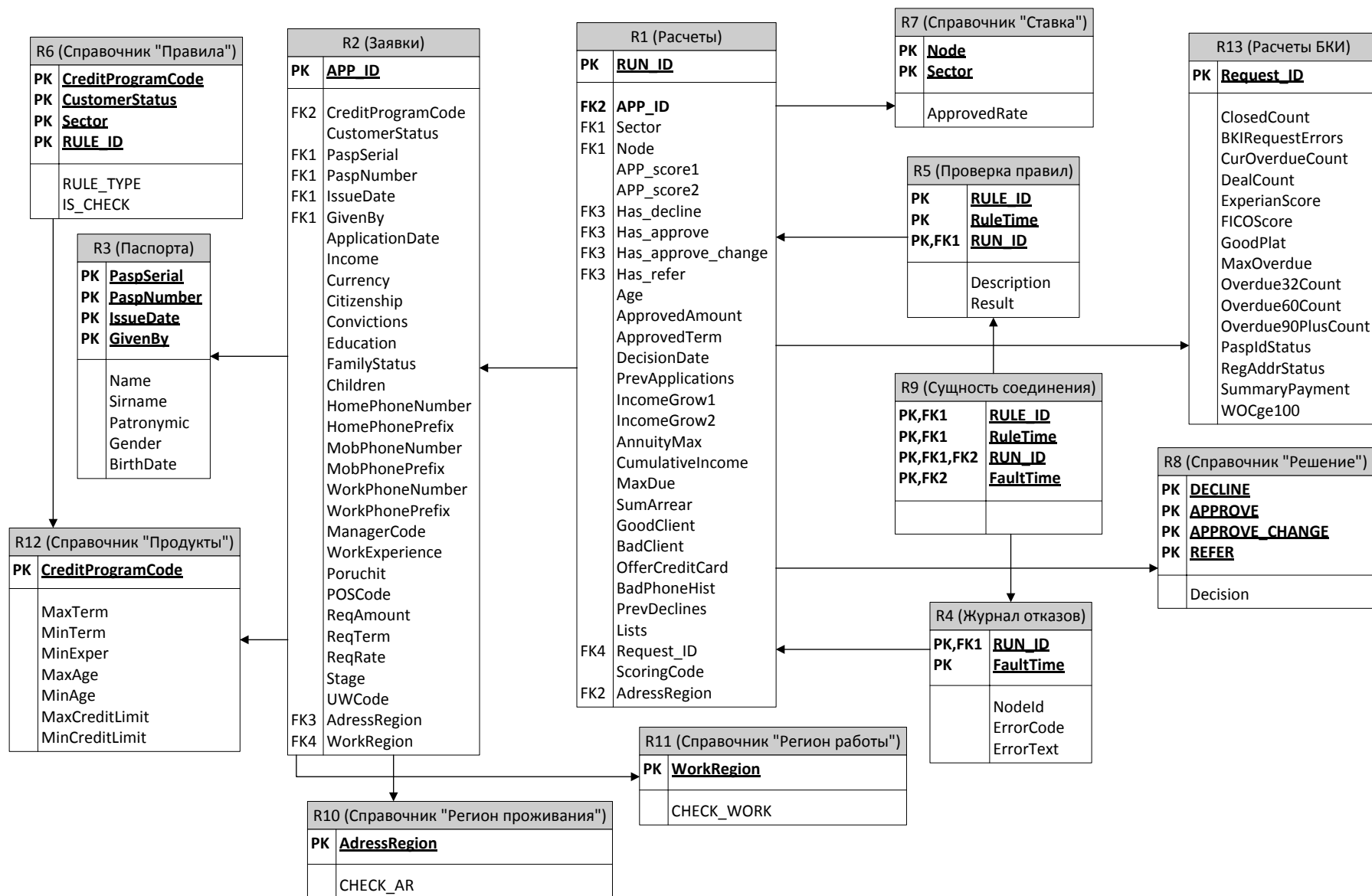


Рисунок 2.9 – Теоретическая схема отношения (MAIN)

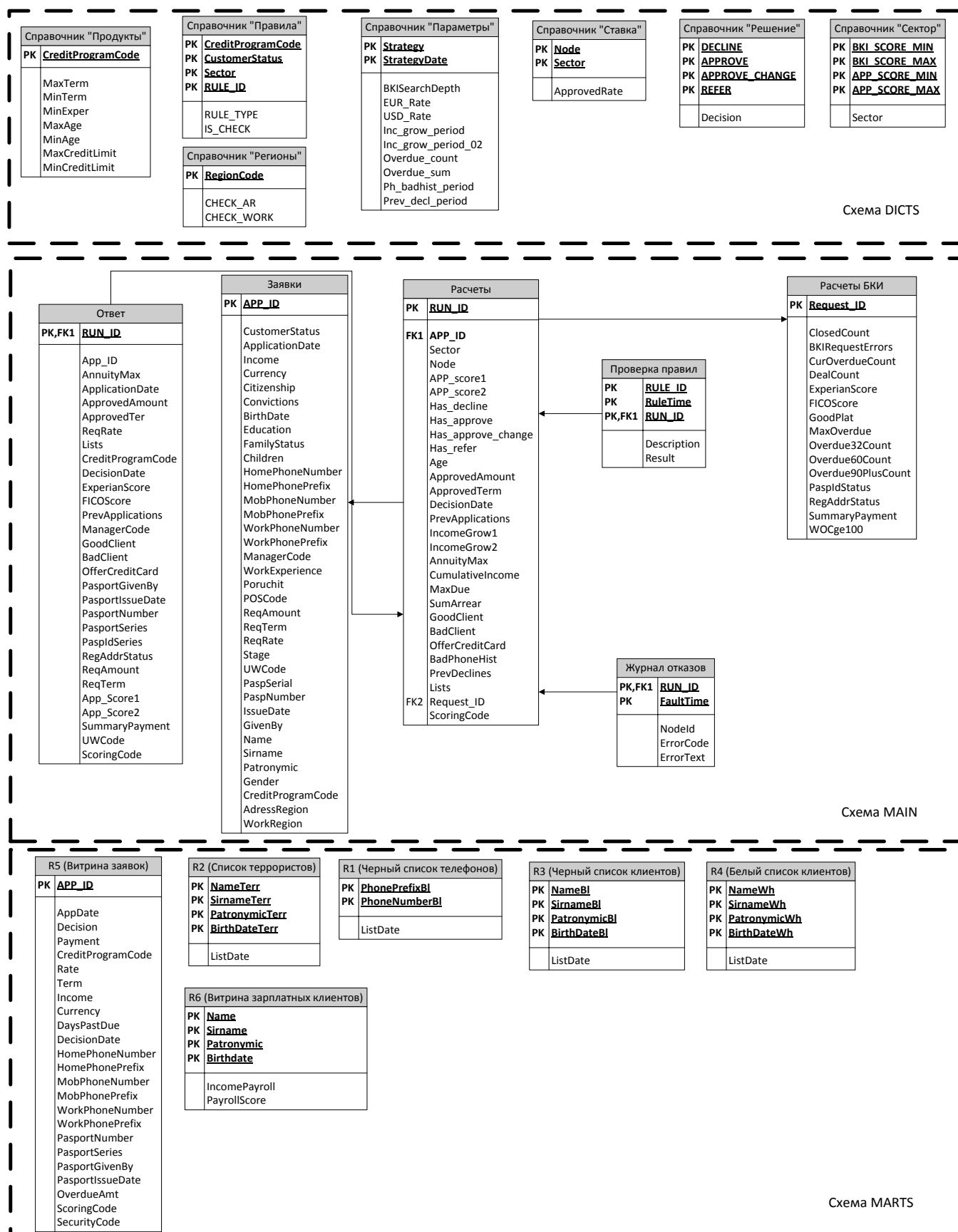


Рисунок 2.10 – Итоговая схема БД

2.7 Разработка системы передачи данных

Структура распределенной сети обработки данных представлена на рисунке 2.11.

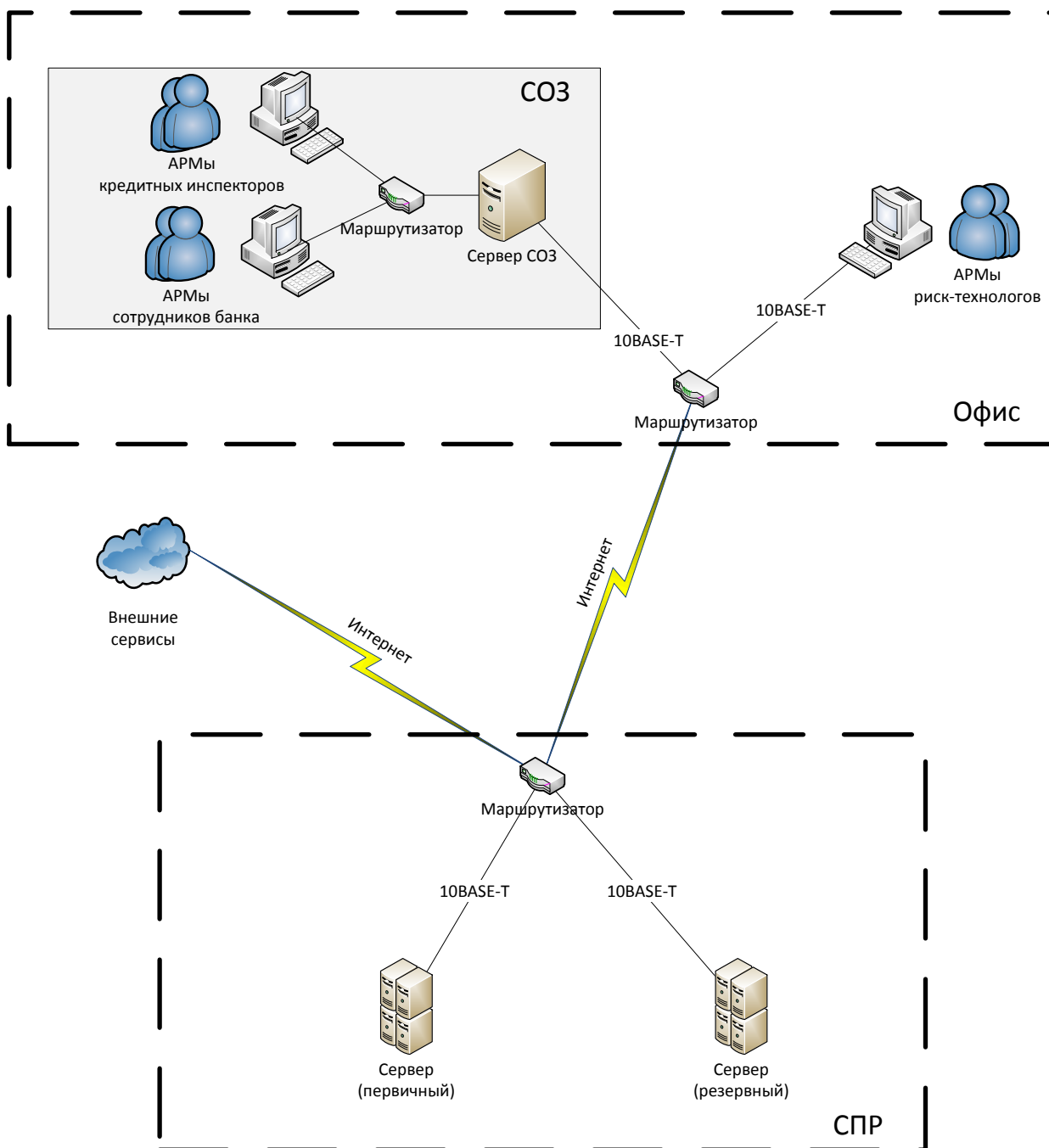


Рисунок 2.11 – Структура распределенной сети обработки данных

Распределенная система обработки данных состоит из двух основных ЛВС:

- ЛВС офиса банка;
- ЛВС, где находится СПР.

ЛВС офиса банка включает систему обработки заявок (СОЗ) и её клиентов, а также АРМ риск-технологов, которые работают непосредственно с СПР.

ЛВС, где находится СПР, состоит из двух серверов: первичный и резервный, в которых поддерживается репликация данных. Резервирование и репликация данных реализована с использованием технологии Oracle Data Guard (описание технологии представлено в Приложении А). Дисковая система организована с использованием технологии RAID-10 (описание технологии представлено в Приложении А).

Каждая ЛВС построена с использованием топологии «звезда» и технологии 10Base-T.

Данные между ЛВС передаются по сети интернет. Пограничными устройствами для ЛВС являются маршрутизаторы [24].

Технология 10Base-T выбрана в связи с используемой средой передачи данных (витая пара) и требованием к минимальной пропускной способности сети (подробный расчет приведен в Приложении В).

2.8 Разработка интерфейса взаимодействия пользователя с системой

Граф диалога пользователя приложения для взаимодействия с БД представлен на рисунке 2.12.

При запуске приложения пользователь проходит аутентификацию в СУБД Oracle. При нажатии кнопки «Выход» приложение прекращает работу и закрывается.

В случае не успешной авторизации пользователь возвращается к форме авторизации. В случае успешной авторизации пользователь переходит в основное меню. При нажатии кнопки «Выход» приложение прекращает работу и закрывается.

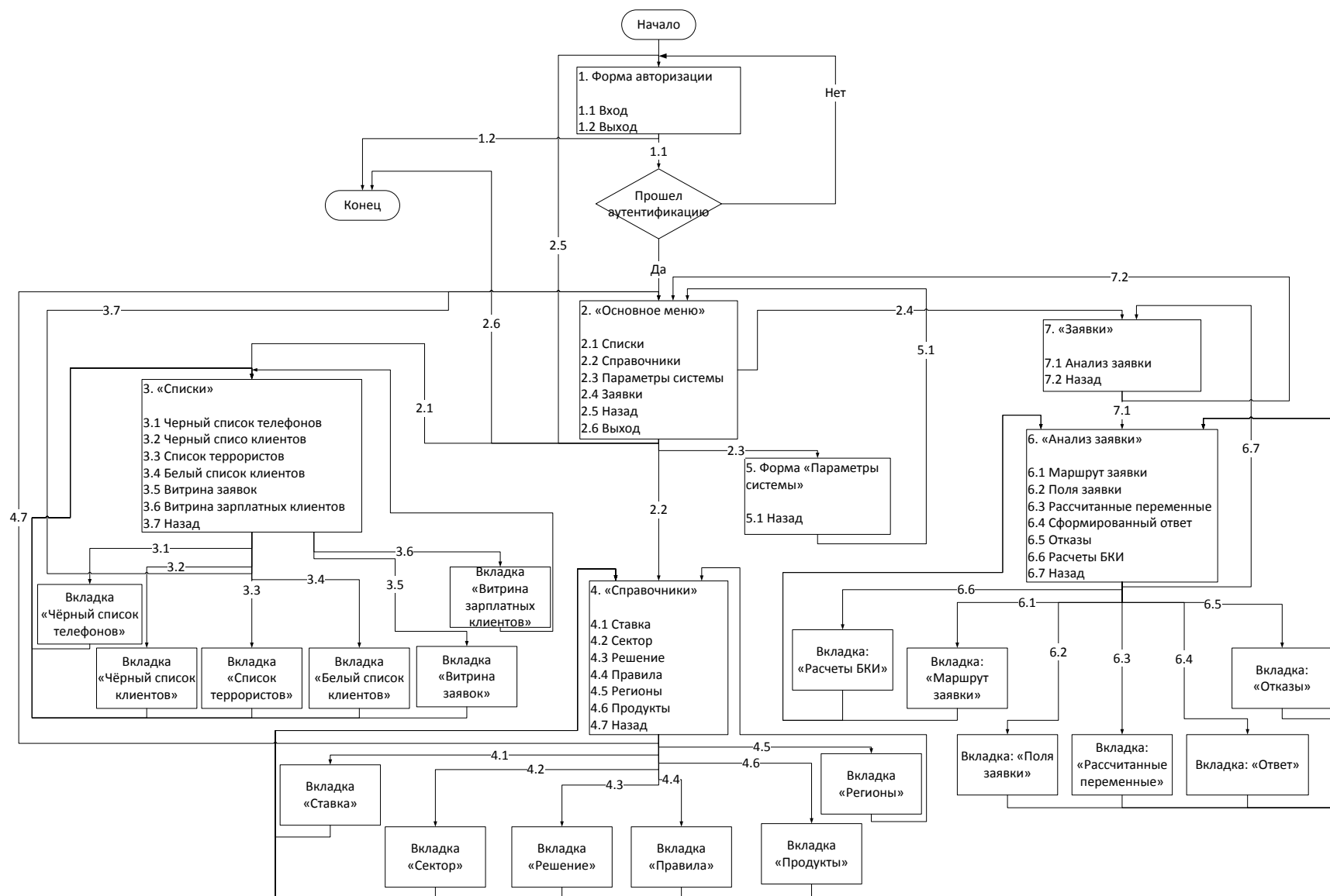


Рисунок 2.12 – Граф диалога пользователя

Вид экранной формы аутентификации представлен на рисунке 2.13.

Рисунок 2.13 – Форма авторизации

Вид основного меню представлен на рисунке 2.14.

Рисунок 2.14 – Основное меню

Из основного меню пользователь может перейти к следующим формам:

- «Списки»;
- «Справочники»;
- «Параметры системы»;
- «Заявки».

Форма «Списки» состоит из шести вкладок:

- «Черный список телефонов»;
- «Черный список клиентов»;
- «Список террористов»;
- «Белый список клиентов»;
- «Витрина заявок»;
- «Витрина зарплатных клиентов».

Вид экранной формы «Черный список телефонов» представлен на рисунке 2.15.

Номер телефона	Префикс	Срок нахождения в списке
7984568	7986	05.06.2016
1356879	7986	29.01.2015
12354777	7916	15.04.2015
12366477	3804	15.05.2015
98763122	194	11.05.2016
12313112	497	30.07.2014
77891916	7916	01.09.2014
74487423	1235	11.11.2017
15687985	4563	15.06.2015
79845623	458	22.12.2016
12132872	4123	16.01.2014
78913134	916	16.01.2014
12123198	1568	11.04.2015
55261488	7894	30.05.2015
49795663	123	01.10.2015

Рисунок 2.15 – Форма «Списки». Вкладка «Черный список телефонов»

Форма «Справочники» состоит из шести вкладок:

- «Решение»;
- «Ставка»;
- «Сектор»;
- «Правила»;
- «Регионы»;
- «Продукты».

Вид экранной формы «Решение» представлен на рисунке 2.16.

Отказ	Одобрение	Одобрение с изменением	Верификация	Решение
0	0	0	1	REFER
0	0	1	0	APPROVE
0	0	0	0	REFER
0	0	1	1	APPROVE
0	1	0	0	APPROVE
0	1	0	1	APPROVE
0	1	1	0	APPROVE
0	1	1	1	APPROVE
1	0	0	0	DECLINE
1	0	0	1	REFER
1	0	1	0	NULL
1	0	1	1	NULL
1	1	0	0	APPROVE
1	1	0	1	APPROVE
1	1	1	0	NULL
1	1	1	1	NULL

Рисунок 2.16 – Форма «Справочники». Вкладка «Решение»

Вид экранной формы «Ставка» представлен на рисунке 2.17.

Ставка	Бакет	Сегмент
23.9	10	2
23.9	11	2
23.9	12	2
23.9	13	2
23.9	14	2
23.9	15	2
23.9	16	2
23.9	17	2
23.9	18	2
23.9	19	2
23.9	20	2
22.5	21	2
23.9	22	2
23.9	23	2
23.9	24	2
23.9	25	2

Рисунок 2.17 – Форма «Справочники». Вкладка «Ставка»

Вид экранной формы «Сектор» представлен на рисунке 2.18.

Макс. значение анкетного скор. бал	Мин. значение анкетного скор. балл	Макс. значение скор. балла БКИ	Мин. значение скор. балла БКИ	Сегмент
1.6538	-999	1.5084	-999	6
2.1525	1.6538	1.5084	-999	6
2.6885	2.1525	1.5084	-999	6
999	2.6885	1.5084	-999	5
1.6538	-999	1.9074	1.5084	6
2.1525	1.6538	1.9074	1.5084	6
3.2432	2.1525	1.9074	1.5084	5
999	3.2432	1.9074	1.5084	4
1.6538	-999	2.194	1.9074	6
2.328	1.6538	2.194	1.9074	5
3.2432	2.328	2.194	1.9074	4
999	3.2432	2.194	1.9074	2
1.9397	-999	2.4248	2.194	6
2.1525	1.9397	2.4248	2.194	5
2.5037	2.1525	2.4248	2.194	4

Рисунок 2.18 – Форма «Справочники». Вкладка «Сектор»

Вид экранной формы «Правила» представлен на рисунке 2.19.

Программа кредитования	статус клиента	Необходимость проверки правила	Код правила	Тип прае
70	OTHER	1	A6	INF
70	OTHER	1	A6	INF
70	88	1	A6	DECLINE
70	1	1	A6	DECLINE
70	3	1	A6	DECLINE
70	4	1	A6	DECLINE
70	5	1	A6	DECLINE
70	6	1	A6	DECLINE
70	7	1	A6	DECLINE
70	8	1	A6	DECLINE
70	9	1	A6	DECLINE
70	10	1	A6	DECLINE
70	11	1	A6	DECLINE

Рисунок 2.19 – Форма «Справочники». Вкладка «Правила»

Вид экранной формы «Регионы» представлен на рисунке 2.20.

Справочники

Регион

	Код региона	Проверка адреса регистрации	Проверка адреса работы
▶	1	0	1
	2	0	0
	3	0	1
	4	0	1
	5	1	1
	6	1	1
	7	0	0
	8	1	1
	9	1	1
	10	0	0
	11	0	1
	12	0	0
	13	0	0
	14	1	1
	15	1	1

Назад

Рисунок 2.20 – Форма «Справочники». Вкладка «Регионы»

Вид экранной формы «Продукты» представлен на рисунке 2.21.

Справочники

Продукт 70

	Код продукта	мак срок по продукту	min срок по продукту	min общий стаж по продукту	мак возраст по продукту
▶	70	36	12	12	60
*					

Назад

Рисунок 2.21 – Форма «Справочники». Вкладка «Продукты»

Вид экранной формы «Параметры системы» представлен на рисунке 2.22.

Параметры системы

Нижняя граница даты актуальности: 23 мая 2014 г.

	Стратегия	Дата актуальности стратегии	Курс евро к рублю для БКИ	Курс доллора к рублю для БКИ	Период, за который рассчитывается у	Период, за который рассчитывается у1	Максимально допустимая длительно	Максимально допустимая сумма прос	Период, за который учитывается отк	Период, за который учитывается отк
▶	POS	09.05.2014	49,80	35,32	30	60	60	100000	14	30
	CASH	15.09.2014	50,16	34,40	60	90	30	20000	14	30
*										

Назад

Рисунок 2.22 – Форма «Параметры системы»

Вид экранной формы «Параметры системы» представлен на рисунке 2.12. При выборе конкретной заявки и нажатии кнопки «Анализ заявки» пользователь переходит к форме «Анализ заявки».

Заявки

Фамилия: Петров, Имя: Петр

Параметр: Значение

	Id заявки	дата заявления	Фамилия	Имя	Отчество
▶	144729	01.03.2012	Петров	Петр	Петрович
	1488	05.05.2012	Петров	Петр	Иванович
*					

Анализ заявки

Назад

Рисунок 2.23 – Форма «Заявки»

Форма «Справочники» состоит из шести вкладок:

- «Поля заявки»;
- «Рассчитанные переменные»;
- «Расчёты БКИ»;
- «Ответ»;
- «Отказы».

Вид экранной формы «Отказы» представлен на рисунке 2.24.

Анализ заявки

Заявка № 144729

Поля заявки | Рассчитанные переменные | Расчеты БКИ | Ответ | Маршрут заявки | **Отказы**

	Дата и время ошибки	Код ошибки	Текст ошибки	Id узла
▶	12.05.2014 12:10	1	Значение скорб...	SC1
	12.05.2014 12:10	153	Неверный форм...	L4
*				

Назад

Рисунок 2.24 – Форма «Анализ заявки». Вкладка «Отказы»

3 ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ

3.1 Введение

В данной главе представлена разработка алгоритмов обработки данных, проведена оценка качества функционирования и описаны чек-листы тестирования АСОИУ.

3.2 Разработка алгоритмов обработки данных

После приёма xml-файла заявки система производит разбор xml-файла в таблицу (APPLICATIONS) БД. После чего начинается обработка заявки. Граф алгоритма обработки заявки в системе представлен на рисунке 1.8.

Процесс обработки состоит из следующих основных элементов:

- чтение/запись данных;
- вызова SAS-кода;
- узел ветвления;
- бизнес-правило.

Блоки «Чтения/записи данных» производят чтение, вставку или изменение одной записи в БД.

Блоки «Вызова SAS-кода» запускают выполнение соответствующих скриптов SAS-кода.

«Узлы ветвления» обеспечивают переход на тот или иной блок в зависимости от результата выполнения заданного условия.

Блоки «Бизнес-правил» представляют собой блоки добавления записи в таблицу проверки правил (RULE_CHECK) с записью результата проверки заданного правила в поле результата проверки правила (Result) и указанием кода правила (Rule_ID). Большинство блоков бизнес-правил независимы друг от друга, что существенно упрощает разработку и тестирование новых правил [9].

В начале обработки заявки производится добавление пустой записи (отсутствуют расчётные поля) для данной заявки в таблицу результатов расчётов. Затем проводится корректность полученной заявки – наличие в справочниках региона (DICT_REGION) представленных в заявке региона

работы и регистрации, а также наличие в справочнике кредитных продуктов программы кредитования, указанной в заявке. В случае некорректности полученной заявки, производится переход на генерацию и отправку ответа в СОЗ, где указывается причина прерывания обработки заявки. Иначе производится переход на этап внутренних проверок.

На этапе внутренних проверок производится расчёт риск-переменных, необходимых для проверки минимальных требований таких как возраст, количество отказов за последний период (период устанавливается в параметрах стратегии), максимальное количество дней просрочки по кредиту в банке, резкое увеличение дохода и другие.

На основе рассчитанных переменных производится проверка правил минимальных требований, на основе результатов которой рассчитывается решение по минимальным требованиям. В случае срабатывания хотя бы одного правила отказа система переходит на этап формирования решения, иначе – переходит к проверке списков.

Проверка срабатывания правил по заявке производится следующим образом: используется представление, которое соединяет таблицы проверки правил (RULE_CHECK) и справочник необходимости проверки правил (DICT_RULE_CHECK) по идентификатору правила, статусу клиента, сегменту и программе кредитования. На основе полученного набора данных подсчитывается количество правил, которые имеют положительный результат (Result=1), необходимость проверки (Is_check=1) и требуемый тип правила (Rule_type) [11].

На этапе проверки списков проводятся проверки наличия соответствующих данных кредитной заявки в соответствующих списках с учётом проверки даты актуальности присутствия этих данных в списке. Проверяются следующие списки:

Чёрный список телефонов – перечень номеров телефонов, владельцам которых банк отказывает в предоставлении кредита.

Чёрный список клиентов – перечень клиентов, которым банк отказывает в предоставлении кредита.

Белый список клиентов – перечень клиентов, которым банк принял решение предоставлять кредит, вне зависимости от результатов дальнейших проверок.

Список террористов – перечень физических лиц, в отношении которых имеются сведения об их причастности к экстремистской деятельности или терроризму. Данный список банк получает от федеральной службы по финансовому мониторингу [18].

В случае обнаружения данных заявки в списках банка система переходит на этап формирования решения, иначе – переходит к этапу получения и обработки внешних данных.

На этапе работы с внешними данными система производит запрос во внешние сервисы:

- НБКИ – национальное бюро кредитных историй;
- МБКИ – межбанковское бюро кредитных историй;
- Equifax – бюро кредитных историй «Эквифакс кредит сервисиз»;
- ОКБ – объединенное кредитное бюро;
- NationalHunter – сервис противодействия мошенничеству от ОКБ;
- FPS – сервис противодействия мошенничеству от Equifax;
- ФМС – федеральная миграционная служба.

После получения данных от внешних сервисов производится расчёт риск-переменных для анализа внешних данных, например, количество кредитов с текущей просрочкой, количество непогашенных кредитов свыше 100 000 рублей, количество просрочек с длительностью свыше 90 дней и другие.

На этапе расчёта лимита производятся скоринг, расчёт сегмента, расчёт лимита и проверки правил в зависимости от типа программы кредитования.

Для новых и повторных клиентов проводится скоринг по соответствующим картам, подсчет сегмента, затем проверяются правила по данным, полученным от БКИ (например, наличие просрочки длительностью более 90 дней). Для новых клиентов также проводятся проверки правил FPS. Затем производится расчет лимита (одобренная ставка, одобренный срок кредитования, одобренная сумма, максимально допустимый ежемесячный

платеж по кредиту и другие). После расчета лимита проверяются правила по лимиту и скорингу (например – проверка значений скорингового балла и сегмента, процентная ставка меньше минимальной для данной кредитной программы).

Для зарплатных клиентов проводятся расчёт сегмента, проверки правил по данным, полученным от БКИ, затем проверяются правила по доходу (соответствие данных в анкете данным в витрине зарплатных клиентов (PAYROLL_MART)). Затем производится расчёт лимита (одобренная ставка, одобренный срок кредитования, одобренная сумма, максимально допустимый ежемесячный платеж по кредиту и другие) [18].

После расчёта лимита производятся проверки правил верификации (например, срок работы на текущем месте работы не удовлетворяет условиям банка; мошенничество с паспортом) и общие правила (например, недопустимая кредитная нагрузка; отсутствие места работы или регистрации в регионе присутствия банка). Затем проводится проверка срабатывания правил отказа и переход на этап формирования решения.

На этапе принятия решения производятся проверки правил назначения верификации (например – отличие дохода в заявке от дохода любой из предыдущих заявок за последние 3 месяца более, чем на 20%), затем проверка срабатывания одного из правил верификации. Производится вычисление промежуточных переменных, на основе сработавших ранее правил, затем проверка правил назначения решения. Затем проводится проверка срабатывания правил одобрения и одобрения с изменением, после чего принимается промежуточное решение по заявке. После принятия решения проводится еще ряд проверок, которые устанавливают итоговое решение по кредитной заявке.

После вычисления итогового решения по кредитной заявке производится формирование записи в таблице ответа (RESPONSE), затем генерация на её основе xml-файла для ответа и отправка ответа в СОЗ.

UML-диаграмма полного цикла обработки заявки в системе представлена на рисунке 3.1.

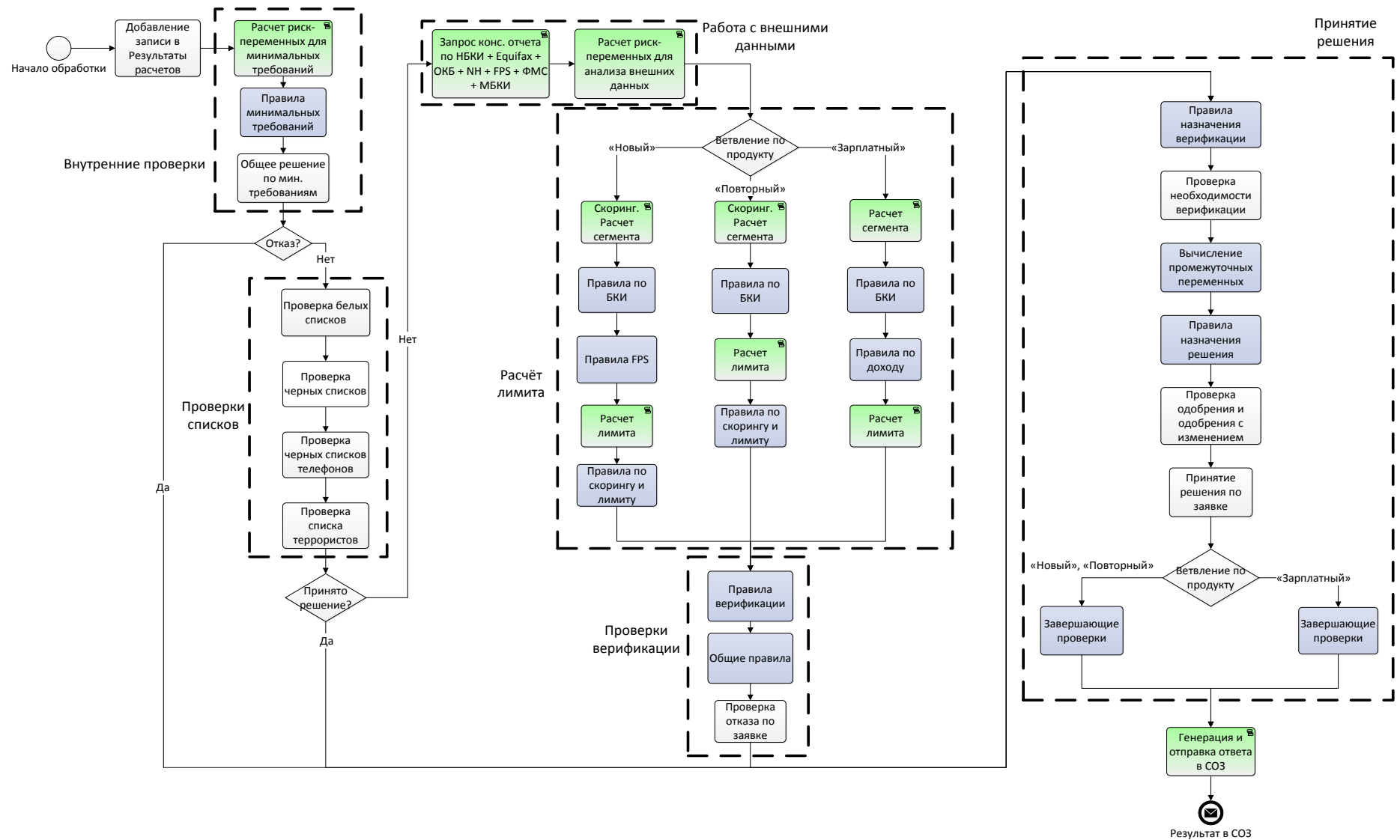


Рисунок 3.1 – UML-диаграмма обработки заявки

3.3 Тестирование и отладка

Для отладки каждого компонента АСОИУ была проведена разработка тестового чек-листа в соответствии с требованиями и реализацией. Также для проверки консистентности данных на сервере «stand by» был создан ряд сверочных витрин данных. Для проверки корректности срабатывания бизнес-правил и стоп-листов были подготовлен специальный загрузочный файл с тест-кейсами. В таблице 3.1 приведен чек-лист для всех частей автоматизированной системы. В ряде случаев тест-кейсы для нескольких проверок однотипны, поэтому конкретные описания тест-кейсов для них не приведены.

Таблица 3.1 – Чек-лист для проверки хранилища

Тестируемый функционал	Ожидаемый результат
DDL (структура таблиц)	Атрибутный состав таблиц соответствует спроектированной модели
DDL (именование и типы атрибутов)	Название и типы атрибутов таблиц соответствуют писанным в модели
PK, FK	Все таблицы имеют корректный внешний/первичный ключ
DML (insert, update, delete)	При внесении изменений в БД данные в ней обновляются (создание, удаление и редактирование записей)
DML (select)	Запросы в БД выполняются без каких-либо сбоев
Время ответа	Время ответа сервера БД в рамках допустимого
NOT NULL	При попытке установки значения NULL (в атрибутах, на которые стоит запрет установки NULL) БД выдает ошибку
Работа сервера ID	Пользователь не может войти в базу данных без аутентификации

Тестируемый функционал	Ожидаемый результат
Идентификация пользователя	Пользователь успешно вошел в систему под своей учетной записью (включая ТУЗ)
Корректная работа с разными часовыми зонами	Если клиент и сервер расположены в разных часовых зонах, не возникает проблем с синхронизацией
Хранимые процедуры	Функциональность хранимых процедур БД (название, параметры, возвращаемые значения, синтаксис и т.д.) соответствует описанию
Count	Кол-во записей в таблицах соответствует кол-ву записей на сервере Stand by
Контроль корректности завершения транзакции	Изменения, сделанные пользователем, сохраняются только по завершении процесса (если в процессе работы произошел сбой, изменения должны быть проигнорированы)
Консистентность данных	Данные в БД соответствуют данным в сверочных витринах
Вход по ТУЗ из SAS RTDM	Успешный вход по ТУЗ из SAS RTDM
Вызов блоков SAS-кода	При вызове внутренних процедур в SAS RTDM не возникает ошибок
Доступность таблиц для чтения из SAS RTDM	Все таблицы доступны для чтения из SAS RTDM (модуль для проверки вынесен в Приложение Г)
Доступность таблиц для записи/апдейта из SAS RTDM	Все таблицы доступны для записи и апдейта из SAS RTDM (модуль для проверки вынесен в Приложение Г)

Тестируемый функционал	Ожидаемый результат
Доступность хранимых процедур из SAS RTDM	Хранимые процедуры успешно запускаются из под ТУЗа из SAS RTDM
Узлы ветвления	Узлы ветвления срабатывают корректно (модуль для проверки вынесен в Приложение Д)
Срабатывание стоп-листов	Корректное срабатывание стоп-листов в предусмотренных тест-кейсами случаях (модуль для проверки вынесен в Приложение Д)
Срабатывание бизнес-правил	Корректное срабатывание бизнес-правил в предусмотренных тест-кейсами случаях (модуль для проверки вынесен в Приложение Д)
Ручные проверки	На ручные проверки отправляется заданный процент заявок

3.4 Оценка качества функционирования.

3.4.1 Описание модели. Входные и выходные данные

Для оценки характеристик функционирования системы применяется имитационное моделирование. В качестве системы моделирования используем GPSS [25]. Формализованная схема PCOD представлена на рисунке 3.2.

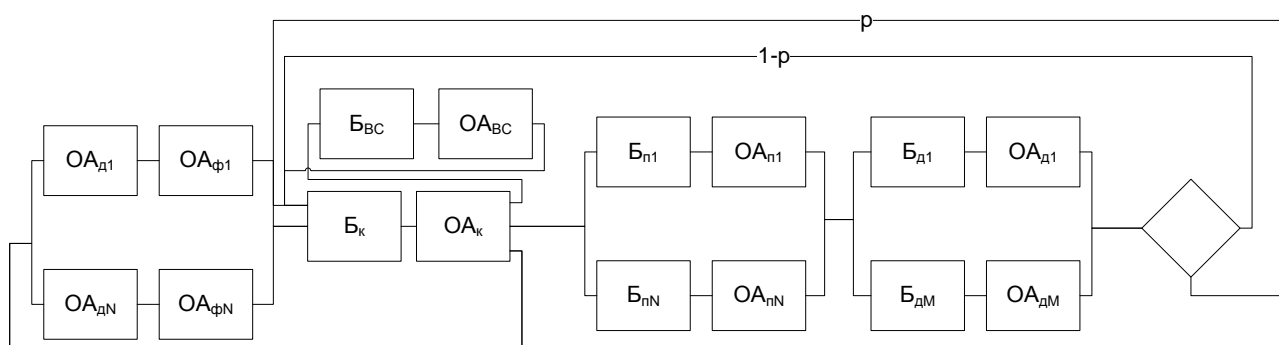


Рисунок 3.2 – Формализованная схема

В схеме используются следующие обозначения:

- OA_{Di} – обслуживающий аппарат, имитирующий дообработку запроса на i -том источнике заявок после обработки запроса на сервере.
- $OA_{\Phi i}$ – обслуживающий аппарат, имитирующий формирование запроса от i -том источнике заявок; ($i = \overline{1 \dots N}$);
- B_k – буфер, имитирующий очередь запросов к каналу;
- OA_K – обслуживающий аппарат, имитирующий задержку при передаче данных через канал;
- B_n – буфер, имитирующий очередь заявок к процессорам;
- OA_n – обслуживающие аппараты, имитирующие работу процессоров;
- B_{di} – буфер, имитирующий очередь заявок к i -му диску;
- OA_{di} – обслуживающий аппарат, имитирующий работу i -го диска;
- B_{BC} – буфер, имитирующий очередь заявок к внешним сервисам;
- OA_{BC} – обслуживающие аппараты, имитирующие работу внешних сервисов.

Обслуживание заявок во всех ОА подчиняется экспоненциальному закону.

Путь прохождения заявки по схеме следующий: после формирования заявки на источнике заявок заявка через канал поступает на обработку в один из процессоров, после чего переходит на обработку в один из свободных дисков. После обработки на диске по вероятности p заявка переходит через канал на обработку во внешние сервисы, иначе (по вероятности $1-p$) переходит через канал на источник заявок. После обработки во внешних сервисах заявка через канал переходит на источник заявок.

Входные параметры рассматриваемой модели описаны в таблице 3.2.

Таблица 3.2 – Исходные данные модели.

Обозначение	Описание
N	число потоков обработки заявок
T_0	среднее значение времени дообработки заявки от источника к серверу (с)
T_p	среднее значение времени формирования заявки от источника заявок к серверу (с)
t_k	среднее значение времени передачи заявки по каналу (с)
C	число процессоров сервера
$t_{пр}$	среднее значение времени обработки заявки в ЦП сервера (с)
$t_{дi}$	среднее значение времени обработки заявки в диске сервера (с)
m	количество дисков в сервере, при условии, что все они одинаковые
$t_{вс}$	среднее значение времени обработки заявки во внешних сервисах (с)
p	вероятность обращения во внешние сервисы

Выходные параметры рассматриваемой модели описаны в таблице 3.3.

Таблица 3.3 – Выходные характеристики модели.

Обозначение	Описание
$T_{реак}$	среднее значение времени реакции системы (с)
$T_{внутр}$	среднее время обработки заявки в системе (с)
ρ_k	коэффициент загрузки ОА, имитирующего работу канала передачи данных
$\rho_{пр}$	коэффициент загрузки ОА, имитирующего работу процессора сервера
$\rho_{дi}$	коэффициент загрузки ОА, имитирующего работу i -ого диска сервера
T_k	время пребывания заявки в канале (с)
T_d	время пребывания заявки в диске (с)
$T_{пр}$	время пребывания заявки в процессоре (с)

Время обработки заявки в системе не включает время обработки заявки во внешних сервисах (3.1):

$$T_{\text{внутр}} = T_{\text{реак}} - t_{\text{вс}} \quad (3.1)$$

Так как используется канал передачи данных с полосой пропускания 10Мб/с, средний размер заявки 30 КБ, то время передачи через канал составляет

$$t_{\text{к}} = \frac{30 * 8}{10 * 1024} = 0,023 \text{ с.}$$

Вероятность обращения во внешние сервисы составляет 80%.

3.4.2 GPSS-моделирование

В нотации GPSS модель выглядит следующим образом [26]:

INITIAL	X\$STATION_N, 35	;число потоков обработки заявок
INITIAL	X\$STATION_TD, 60	;время формирования заявки в СОЗ
INITIAL	X\$STATION_TF, 60	;время обработки ответа по заявке
INITIAL	X\$CANAL_T, 0.008	;время передачи через канал
INITIAL	X\$SERVER_T, 4	;время обработки на процессоре
INITIAL	X\$DISK_N, 2	;количество дисков
INITIAL	X\$DISK_T, 0.01	;время обработки на диске
INITIAL	X\$BKI_T, 30	;время запроса во внешние сервисы

WORKSTATION_D	STORAGE	100
WORKSTATION_F	STORAGE	100

BKI_N	STORAGE	999	
SERVER	STORAGE	2	;количество процессоров сервера
DISK_N	FUNCTION	RN1, D2	
0.5, 1/1, 2			

EXPON	FUNCTION	RN1, C23
0, 0/.1, .104/.2, .222/.3, .355/.4, .510/.5, .69/.6, .915/.7, 1.2/.75, 1.37/.8, 1.5/.84, 1.83/.88, 2.12/.9, 2.3/.92, 2.52/.94, 2.82/.95, 2.98/.96, 3.2/.97, 3.5/.98, 3.9/.995, 5.3/.998, 6.2/.9995, 7/1, 8		
GENERATE , , , X\$STATION_N		

WOSF	QUEUE	QSYSTEM	;формирование заявки
	ENTER	WORKSTATION_F, 1	
	ADVANCE	X\$STATION_TF, FN\$EXPON	
	LEAVE	WORKSTATION_F, 1	
	QUEUE	QINNSYSTEM	
	ASSIGN	3, SVR	

CAN	QUEUE	QCANAL	;канал
	SEIZE	CANAL	
	DEPART	QCANAL	
	ADVANCE	X\$CANAL_T, FN\$EXPON	
	RELEASE	CANAL	

	TRANSFER	, P3	
SVR	QUEUE	QPROCESSOR	; сервер
	ENTER	SERVER, 1	
	ADVANCE	X\$SERVER_T, FN\$EXPON	
	LEAVE	SERVER, 1	
	DEPART	QPROCESSOR	
	ASSIGN	5, FN\$DISK_N	
	QUEUE	P5	
	SEIZE	P5	
	DEPART	P5	
	ADVANCE	X\$DISK_T, FN\$EXPON	
	RELEASE	P5	
	TRANSFER	0.8, PER, CBKI	
CBKI	ASSIGN	3, BKI	
	TRANSFER	, CAN	
PER	ASSIGN	3, WOSD	
	TRANSFER	, CAN	
WOSD	DEPART	QINNSYSTEM	; приём и обработка заявки
	ENTER	WORKSTATION_D, 1	
	ADVANCE	X\$STATION_TD, FN\$EXPON	
	LEAVE	WORKSTATION_D, 1	
	DEPART	QSYSTEM	
	TRANSFER	, WOSF	
BKI	ENTER	BKI_N, 1	; внешние сервисы
	ADVANCE	X\$BKI_T, FN\$EXPON	
	LEAVE	BKI_N, 1	
	QUEUE	QCANAL	; канал
	SEIZE	CANAL	
	DEPART	QCANAL	
	ADVANCE	X\$CANAL_T, FN\$EXPON	
	RELEASE	CANAL	
	ENTER	SERVER, 1	
	ADVANCE	X\$SERVER_T, FN\$EXPON	
	LEAVE	SERVER, 1	
	ASSIGN	5, FN\$DISK_N	
	QUEUE	P5	
	SEIZE	P5	
	DEPART	P5	
	ADVANCE	X\$DISK_T, FN\$EXPON	
	RELEASE	P5	
	TRANSFER	, PER	
	GENERATE	100000	
	TERMINATE	1	
START	1		

3.4.3 Результаты моделирования

Результаты аналитического моделирования представлены в таблице 3.4.

Таблица 3.4 – Результаты аналитического моделирования

Параметр	Номер расчета				
	1	2	3	4	5
Входные данные					
N	20	25	30	35	40
T_0	60	60	60	60	60
T_p	60	60	60	60	60
t_k	0,008	0,008	0,008	0,008	0,008
C	2	2	2	2	2
$t_{пр}$	5	5	5	5	5
$t_{дi}$	0,01	0,01	0,01	0,01	0,01
m	2	2	2	2	2
t_{BC}	30	30	30	30	30
p	0,8	0,8	0,8	0,8	0,8
Выходные данные					
$T_{реак}$	34,872	35,743	36,828	39,328	44,826
$T_{внутр}$	4,872	5,734	6,828	9,328	14,826
ρ_k	0,004	0,005	0,005	0,006	0,007
$\rho_{пр}$	0,467	0,586	0,692	0,791	0,884
$\rho_{дi}$	0,001	0,001	0,002	0,002	0,002
T_k	0,008	0,008	0,008	0,008	0,008
T_d	0,010	0,010	0,010	0,010	0,010
$T_{пр}$	4,841	5,572	6,726	8,433	11,582

Графики зависимостей времени обработки заявки, коэффициентов загрузки и времени пребывания в узлах представлены на рисунках 3.3, 3.4, 3.5, соответственно.

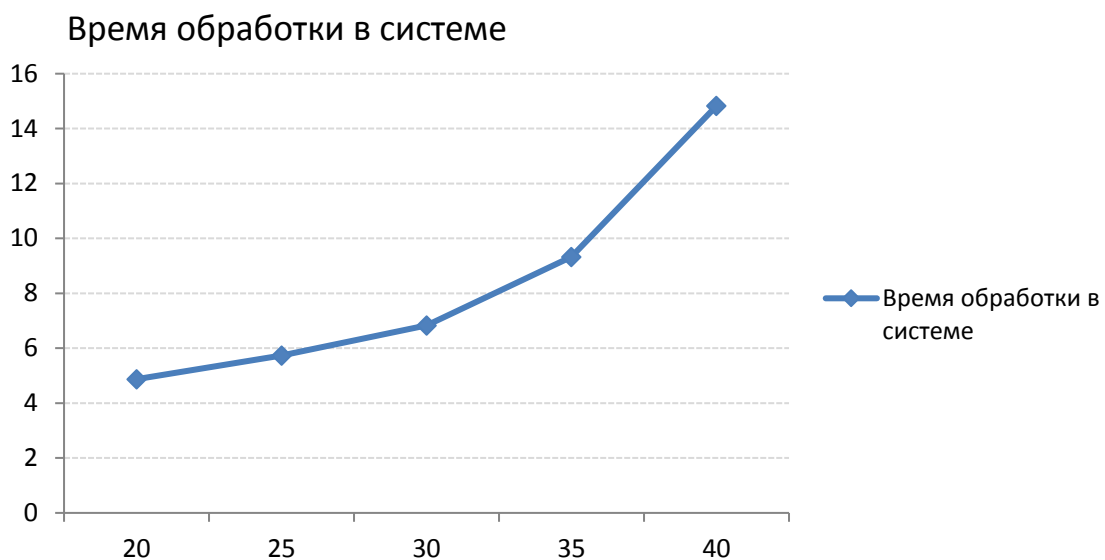


Рисунок 3.3 – Зависимость времени обработки в системе от числа источников заявок

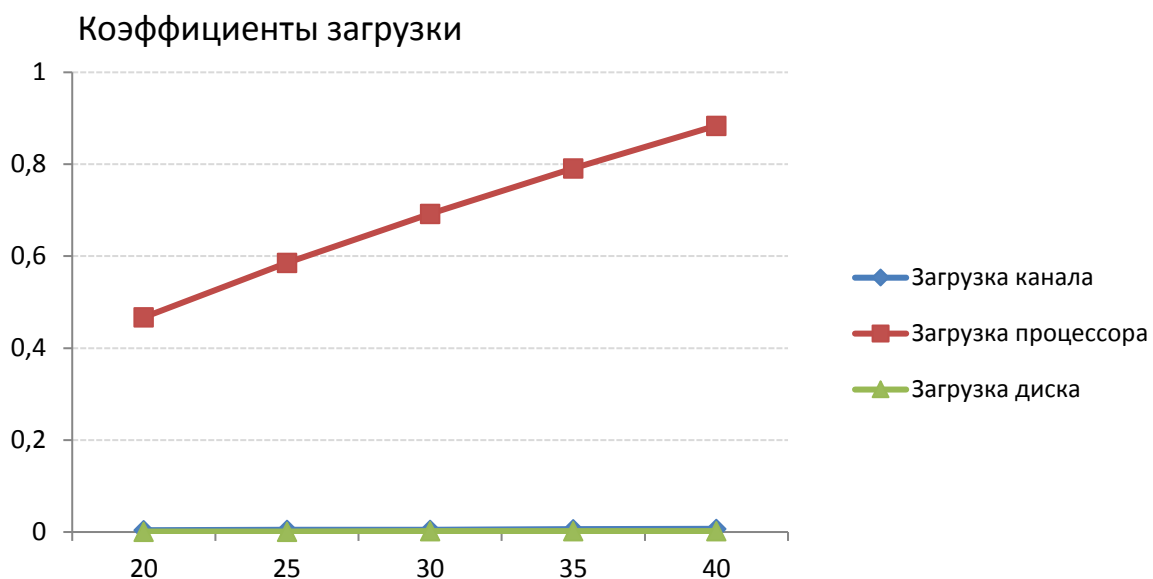


Рисунок 3.4 – Зависимость коэффициентов загрузки от числа источников заявок

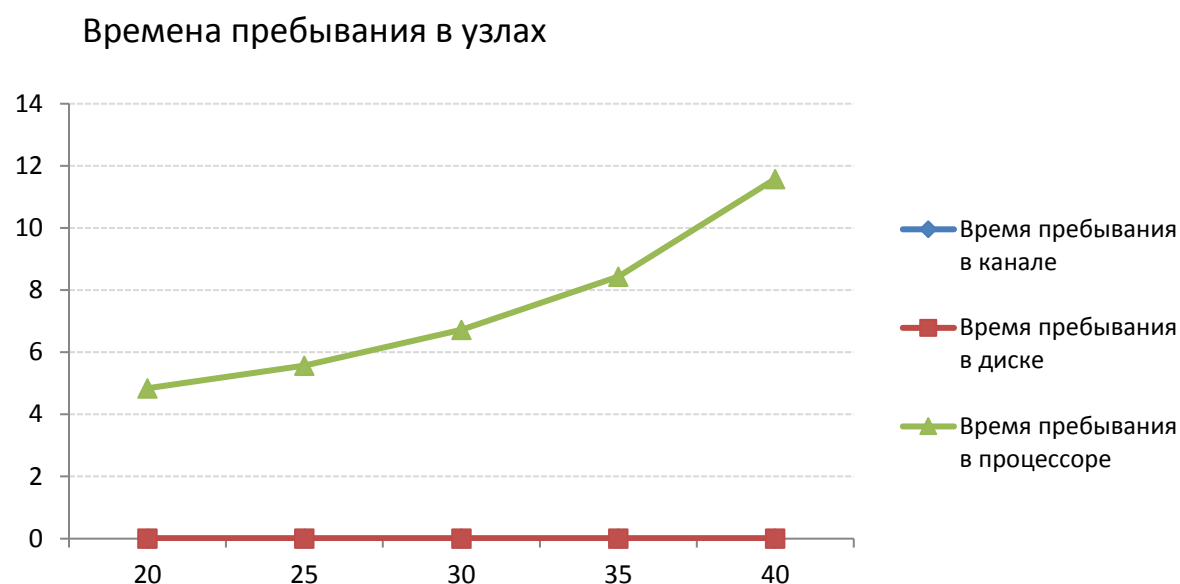


Рисунок 3.5 – Зависимость времени пребывания в узлах от числа источников заявок

3.4.4 Вывод

Наименьшую загрузку имеет канал передачи данных, так как канал имеет пропускную способность 10 Мб/с, а данные, передаваемые через канал имеют средний размер в 10 КБ.

Узким местом системы является процессор, так как на него приходится основная нагрузка при принятии решения.

Объем данных, записываемых в БД и файлы журналов работы системы, составляет менее 1 МБ, а средняя скорость чтения/записи диска составляет порядка 100 МБ/с. Диски имеют малую загрузку в связи с тем, что при обработке заявки почти все операции производятся «на лету» и не требуют взаимодействия с диском, а обращение к дискам – минимально.

При параллельной обработке заявок не более, чем в 35 потоков система удовлетворяет требованиям, предъявляемым к времени обработки заявки системой и загрузка всех узлов системы принимает приемлемые значения.

При параллельной обработке в более, чем 35 потоков загрузка процессора превышает 80% и время обработки заявки в системе превышает 10 секунд.

4 ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ

4.1 Введение

В данном разделе производится расчет затрат на разработку и внедрение системы СПР, определение экономического эффекта от внедрения системы и обоснование разработки и внедрения системы.

4.2 Определение затрат на разработку

4.2.1 Расчет стоимости производимых работ

Используя значения стоимости одного человеко-дня для различных специалистов из таблицы 4.1, получим распределение затрат по этапам разработки, представленное в таблице 4.2.

Таблица 4.1 – Стоимость человеко-дня специалистов

Исполнитель	Стоимость 1 ч/д, руб.
Аналитик	6000
Разработчик	5000
Архитектор	15000
Руководитель проекта	12000

Таблица 4.2 – Распределение затрат на разработку

№	Название задачи	Длительность	Трудозатраты	Исполнитель	Затраты
1	Анализ и формализация требований	15 дней	15 дней		90 000,00 Р
1.1	Аналитика по универсальной компоненте	6 дней	6 дней	Аналитик	36 000,00 Р
1.2	Аналитика по стратегии	9 дней	9 дней	Аналитик	54 000,00 Р
2	Реализация	26 дней	27 дней		141 000,00 Р
2.1	Универсальная компонента	18 дней	18 дней		90 000,00 Р
2.1.1	Обработка данных БКИ	2 дней	2 дней	Разработчик	10 000,00 Р
2.1.2	Консолидация обязательств из БКИ (для расчета лимита)	3 дней	3 дней	Разработчик	15 000,00 Р
2.1.3	Прескоринг расчет	2 дней	2 дней	Разработчик	10 000,00 Р

№	Название задачи	Длительность	Трудозатраты	Исполнитель	Затраты
2.1.4	Расчет по анализу КИ	3 дней	3 дней	Разработчик	15 000,00 Р
2.1.5	Расчет по анти-фроду	3 дней	3 дней	Разработчик	15 000,00 Р
2.1.6	Расчет для скоринга	2 дней	2 дней	Разработчик	10 000,00 Р
2.1.7	Скоринговая карта	3 дней	3 дней	Разработчик	15 000,00 Р
2.2	Стратегия	13 дней	9 дней		51 000,00 Р
2.2.1	Расчет для лимитов	3 дней	3 дней	Разработчик	15 000,00 Р
2.2.2	Уровень риска	1 день	1 день	Аналитик	6 000,00 Р
2.2.3	Правила прескоринга	2 дней	1 день	Аналитик	6 000,00 Р
2.2.4	Правила анализа КИ	2 дней	1 день	Аналитик	6 000,00 Р
2.2.5	Расчет лимита по продуктам	1 день	1 день	Аналитик	6 000,00 Р
2.2.6	Описание использования справочника	2 дней	1 день	Аналитик	6 000,00 Р
2.2.7	Принятие решения по заявке	2 дней	1 день	Аналитик	6 000,00 Р
3	Тестирование	21 дней	21 дней		123 000,00 Р
3.1	Приемочное тестирование	18 дней	18 дней	Аналитик	108 000,00 Р
3.2	Нагрузочное тестирование	3 дней	3 дней	Разработчик	15 000,00 Р
4	Архитектурный контроль	50 дней	6 дней	Архитектор	90 000,00 Р
5	Управление проектом	50 дней	8 дней	Руководитель проекта	96 000,00 Р
	Внедрение СПР	53 дней	77 дней		540 000,00 Р

4.2.2 Определение затрат на программное обеспечение

Расчет стоимости программного обеспечения приведен в таблице 4.3.

Таблица 4.3 – Стоимость программного обеспечения

Программное обеспечение	Стоимость, руб.
Windows Server 2016 (2 экземпляра)	2*42 000 = 84 000
Microsoft Office 2016	11 000
Oracle 10g (2 экземпляра)	2*60 000 = 120 000
Пакет ПО SAS	400 000
Итого	615 000

4.2.3 Определение затрат на оборудование

Затраты на использование оборудования можно рассчитать по формуле (4.1):

$$C_{\text{ЭВМ}} = \frac{C_{\text{ЭВМ}} * T_{\text{ЭВМ}}}{T_{\text{ам}}} \quad (4.1)$$

Где $C_{\text{ЭВМ}}$ – затраты на использование, покупку оборудования.

Затраты на покупку оборудования:

2 сервера HP Pro Liant BL460c G8: $2 * 172\,000 = 344\,000$ руб.

Затраты на покупку 2 ПЭВМ для разработки: $2 * 65\,000 = 130\,000$ руб.

Таким образом, $C_{\text{ЭВМ}} = 474\,000$ руб.

$T_{\text{ам}}$ – срок амортизации вычислительной техники, составляет 3 года (36 месяцев).

$T_{\text{ЭВМ}}$ – время использования оборудования для разработки системы.

$$T_{\text{ЭВМ}} = \frac{53}{30} = 1,77 \text{ мес}$$

Затраты на ремонт вычислительной техники составляют 5% от стоимости её использования $C_{\text{РЕМ}} = 0,05 * C_{\text{ЭВМ}}$

Затраты на использование оборудования с учётом ремонта составляют:

$$C_{\text{об}} = C_{\text{ЭВМ}} + C_{\text{РЕМ}} + C_{\text{ЭВМ}} = \frac{C_{\text{ЭВМ}} * T_{\text{ЭВМ}}}{T_{\text{ам}}} * 1,05 + 474\,000 = 521\,005,00 \text{ руб.}$$

4.2.4 Определение затрат на оплату труда

В соответствии с главой 23 НКРФ доходы физических лиц за вычетом некоторых льгот подлежат обязательному налогообложению (налог на доходы физических лиц). Для компенсации выплат размер месячного оклада увеличивается по формуле (4.2):

$$C_{\text{зп осн}} = \frac{C_{\text{зп осн (бн)}}}{1 - H_{\text{нфл}}} \quad (4.2)$$
$$C_{\text{зп осн}} = \frac{C_{\text{зп осн (бн)}}}{1 - H_{\text{нфл}}} = \frac{540\,000}{0,87} = 620\,689,66 \text{ руб.}$$

Расходы на дополнительную заработанную плату учитывают все выплаты непосредственно исполнителям за время, не проработанное на

производстве, но предусмотренное законодательством, в том числе: оплата очередных отпусков, компенсация за недоиспользованный отпуск, и др. Величина этих выплат составляет 20% от размера основной заработной платы.

$$C_{\text{зп доп}} = 0,2 * C_{\text{зп осн (бн)}} = 0,2 * 540\,000 = 108\,000 \text{ руб.}$$

Отчисления в фонды:

- 22% – отчисления в ПФР (пенсионный фонд), страховая и накопительная части;
- 0,2% – отчисления в ФСС (фонд социального страхования) по травматизму;
- 2,9% – отчисления в ФСС по временной нетрудоспособности и в связи с материнством;
- 5,1% – отчисления в ФФОМС (федеральный фонд обязательного медицинского страхования).

$$C_{\text{отч}} = 0,302 * C_{\text{зп осн}} = 0,302 * 620\,689,66 = 187\,448 \text{ руб}$$

В результате затраты на оплату труда составляют:

$$C_{\text{опл тр}} = C_{\text{зп осн}} + C_{\text{за доп}} + C_{\text{отч}} = 916\,137,93 \text{ руб.}$$

4.2.5 Определение затрат на накладные расходы

В накладные расходы включены затраты на общехозяйственные расходы (аренда здания, где ведется разработка, ремонт, электроэнергия).

Накладные расходы составляют 12,5% от затрат, рассмотренных ранее:

$$\begin{aligned} C_{\text{НР}} &= 0,125 * (C_{\text{ПО}} + C_{\text{об}} + C_{\text{опл тр}}) = \\ &= 0,125 * (615\,000 + 521\,005 + 916\,137,93) = \\ &= 256\,517,87 \text{ руб.} \end{aligned}$$

4.2.6 Расчет себестоимости

Себестоимость рассчитывается как сумма по всем перечисленным статьям затрат:

$$S = 615\,000 + 521\,005 + 916\,137,93 + 256\,517,87 = 2\,308\,660,80 \text{ руб.}$$

4.2.7 Расчет прибыли

Расчет прибыли произведем исходя из нормы прибыли. Норма прибыли $H_{\text{пр}} = 15\%$.

Чистую прибыль можно определить по формуле (4.4):

$$P_{\text{ч}} = S * H_{\text{пр}} \quad (4.4)$$

$$P_{\text{ч}} = S * H_{\text{пр}} = 346\,299,12 \text{ руб.}$$

Налог на прибыль (20%): $H = 0,2 * 346\,299,12 = 86\,574,78 \text{ руб.}$

Итого: $P = P_{\text{ч}} + H = 432\,873,90 \text{ руб.}$

4.2.8 Расчет цены

Цена без учета налога на добавленную стоимость (НДС) рассчитывается как сумма себестоимости и прибыли:

$$C = S + P = 2\,741\,534,70 \text{ руб.}$$

Отпускная цена (сумма ценна и НДС (18%)):

$$C_{\text{итог}} = \frac{C}{1 - 0,18} = \frac{2\,741\,534,70}{0,82} = 3\,343\,335,00 \text{ руб.}$$

Итоговая схема затрат на разработку продукта приведена в таблице 4.4:

Таблица 4.4 – Итоговая смета затрат на разработку

Статья затрат	Сумма (руб.)
Затраты на ПО	615 000,00
Затраты на оборудование	521 005,00
Затраты на оплату труда	916 137,93
Затраты на накладные расходы	256 517,87
Себестоимость	2 208 660,80
Прибыль	432 873,90
Цена	2 741 534,70
Продажная цена	3 343 335,00

4.3 Определение затрат на внедрение системы

Затраты на внедрение системы включают:

- продажная цена программного продукта;
- затраты на обучение сотрудников.

Планируется взаимодействие с системой двух сотрудников.

Стоимость прохождения курса обучения по работе с SAS RTDM для одного человека составляет 35 600 руб.

Итоговые затраты на внедрение определяются по формуле (4.5):

$$C_{\text{внедр}} = C_{\text{итог}} + C_{\text{обуч}} \quad (4.5)$$

$$C_{\text{внедр}} = C_{\text{итог}} + C_{\text{обуч}} = 3\,343\,335 + 71\,200 = 3\,414\,535 \text{ руб.}$$

4.4 Затраты на эксплуатацию системы

Рассмотрим затраты на эксплуатацию системы за месяц.

Затраты на эксплуатацию включают:

- Затраты на оплату труда сотрудника;
- Затраты на содержание и эксплуатацию системы.

Затраты на оплату труда администратора включают:

- основная заработная плата;
- дополнительная заработная плата (20%);
- отчисления в фонды:
 - 22% – отчисления в ПФР (пенсионный фонд), страховая и накопительная части;
 - 0,2% – отчисления в ФСС (фонд социального страхования) по травматизму;
 - 2,9% – отчисления в ФСС по временной нетрудоспособности и в связи с материнством;
 - 5,1% – отчисления в ФФОМС (федеральный фонд обязательного медицинского страхования).

$$C_{\text{опл тр}} = 60\,000 + 60\,000 * 0,2 + 60\,000 * 0,302 = 90\,120 \text{ руб.}$$

Расходы на содержание и эксплуатацию оборудования состоят из затрат на амортизацию оборудования, его ремонт, электроэнергию и материалы.

В качестве оборудования используется вся вычислительная техника, включая персональный компьютер с монитором, принтером и т.д. В затраты входит не сама полная стоимость оборудования, а только амортизационные отчисления, затраты на техническое обслуживание и ремонт оборудования.

Амортизируемым имуществом в соответствии со статьей 256 Налогового кодекса признается имущество со сроком полезного использования более 12 месяцев и первоначальной стоимостью более 20 000 руб.

ПЭВМ и сервера относятся к третьей группе амортизируемого имущества со сроком полезного использования свыше 3 лет до 5 лет включительно. В расчетах принимается 3 года (36 месяцев). Амортизационные отчисления за месяц находятся по следующей (4.6) формуле (линейный метод амортизации без использования ускоряющих коэффициентов):

$$A = \frac{C_{об}}{T_{ам}} \quad (4.6)$$

$$A = \frac{C_{об}}{T_{ам}} = \frac{512\,005}{36} = 14\,472,36 \text{ руб.}$$

где $C_{об}$ - стоимость оборудования;

$T_{ам}$ - период амортизации оборудования (в месяцах).

Затраты на ремонт оборудования составляют примерно 10% от суммы амортизационных отчислений.

$$C_{рем} = 0,1 * A = 1\,447,23 \text{ руб.}$$

Расходы на электроэнергию в месяц составляют:

$$\mathcal{E} = P * n * 8 * \frac{365}{12} * C_{кВт}$$

где P – средняя потребляемая мощность техники (0,3 кВт*час);

n – число используемых единиц техники (2 сервера и 2 АРМ пользователей);

$C_{кВт}$ – цена за кВт*ч электроэнергии (7,07 руб.).

$$\mathcal{E} = 0,3 * 4 * 8 * \frac{365}{12} * 7,07 = 2\,065,85 \text{ руб.}$$

Итоговая схема затрат на эксплуатацию системы в месяц приведена в таблице 4.5.

Таблица 4.5 – Затраты на эксплуатацию

№	Статья затрат	Сумма (руб.)
1	Затраты на оплату труда	90 120
2	Расходы на содержание и эксплуатацию техники, в том числе:	17 985,45
2.1	Амортизация	14 472,36
2.2	Ремонт	1 447,23
2.3	Электроэнергия	2 065,85
Итого		108 105,45

4.5 Расчет срока окупаемости системы после внедрения

Разрабатываемая система предназначена для использования в кредитных организациях.

Система предназначена для повышения эффективности работ операторов, выдающих кредиты, сотрудников риск-отдела, разрабатывающих кредитные стратегии, кредитных инспекторов, проводящих анализ кредитных заявок.

Расчет сроков окупаемости ведется на основе сравнения затрат на выполнение определенного объема работ для решения аналогичных задач без использования разрабатываемого программного комплекса и с его использованием и приведен в таблицах 4.6–4.7.

Внедрение системы даёт следующие преимущества:

- снижение времени обслуживания клиента;
- высвобождение операторов фронт-офиса;
- снижение времени разработки кредитной стратегии;
- высвобождение кредитных инспекторов.

Если до внедрения системы для обслуживания потока клиентов требовалось 6 операторов фронт-офиса, то с увеличением скорости принятия решения по заявке и автоматизацией процесса ввода заявки число операторов может быть сокращено до 2.

Так как решение «Необходимость ручных проверок» принимается в среднем менее, чем в 20% заявок, то число кредитных инспекторов, занимающихся ручными проверками можно сократить с 5 до 1.

Таблица 4.6 – Затраты на оплату труда до внедрения

Сотрудник	Заработная зарплата	Дополнительная плата (20%)	Взносы (30,2%)	Затраты на сотрудника	Число сотрудников
Оператор фронт- офиса	40 000	40 000*0,2 = 8 000	40 000*0,302 = 12 080	60 080	6
Кредитный инспектор	50 000	50 000*0,2 = 10 000	50 000*0,302 = 15 100	75 100	5
Итого: 735 980 руб.					

Таблица 4.7 – Затраты на оплату труда после внедрения

Сотрудник	Заработная зарплата	Дополнительная плата (20%)	Взносы (30,2%)	Затраты на сотрудника	Число сотрудников
Оператор фронт- офиса	40 000	40 000*0,2 = 8 000	40 000*0,302 = 12 080	60 080	2
Кредитный инспектор	50 000	50 000*0,2 = 10 000	50 000*0,302 = 15 100	75 100	1
Итого: 195 260 руб.					

Таким образом в месяц организация будет экономить:

$$Pr = 735\,980 - 108\,318,08 - 195\,260 = 432\,614,55 \text{ руб.}$$

Затраты на внедрение окупятся через: $t_{\text{ок}} = \frac{3\,414\,353,00}{432\,614,55} = 7,89 \text{ мес.}$

Итоговая смета затрат представлена в таблице 4.8.

Таблица 4.8 – Итоговая смета затрат

Затраты на разработку, руб.	
Затраты на ПО	615 000
Затраты на оборудование	521 005
Затраты на оплату труда	916 137,93
Затраты на накладные расходы	256 517,87
Себестоимость	2 308 660,80
Прибыль	432 873,90
Цена	2 741 534,70
Продажная цена	3 343 335,00

Затраты на внедрение, руб.		
Продажная цена	3 343 335,00	
Затраты на обучения персонала	71 200	
Общая сумма затрат на внедрение	3 414 535,00	
Затраты на эксплуатацию (руб. в месяц)		
Затраты на оплату труда администратора	90 120	
Затраты на содержание и эксплуатацию техники	17 985,45	
Общая сумма затрат на эксплуатацию	108 105,45	
Затраты на оплату труда (руб. в месяц)		
Сотрудник	До внедрения	После внедрения
Оператор фронт-офиса	360 480	120 160
Кредитный инспектор	375 500	75 100
Общая сумма затрат на оплату труда	735 980	195 260
Срок окупаемости (мес.)	7,89	

График затрат до и после внедрения системы представлен на рисунке 4.1.

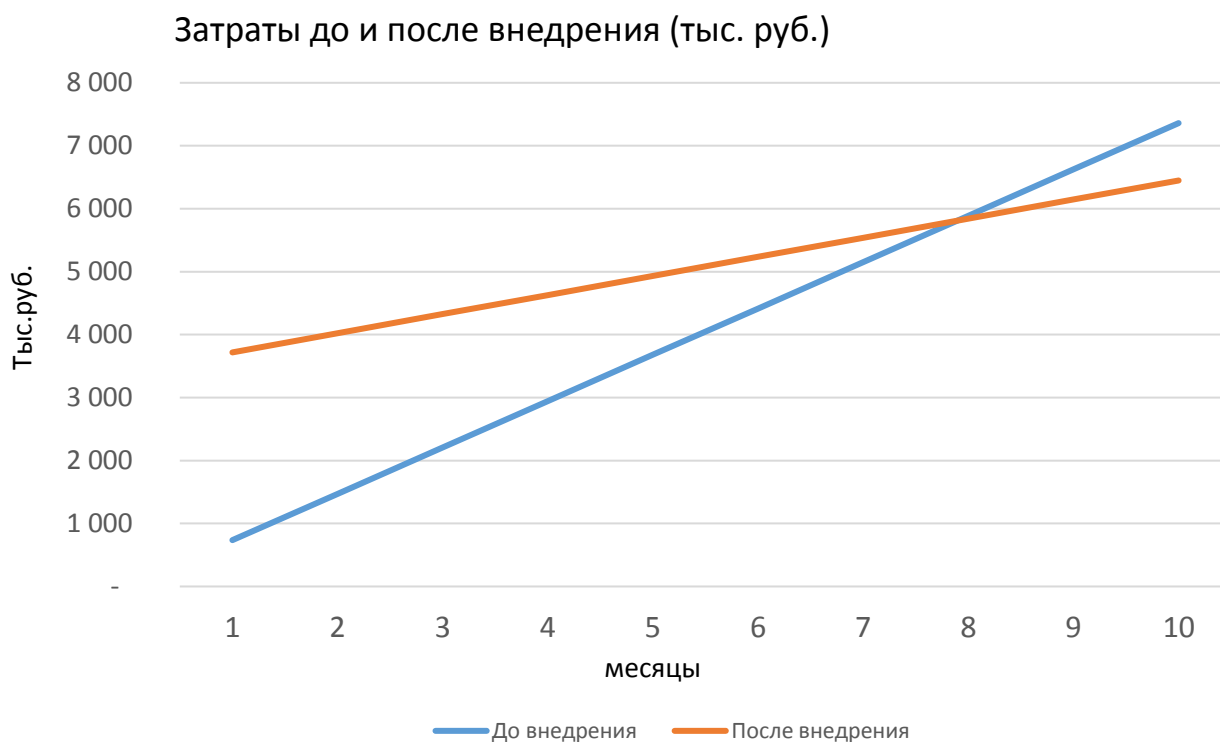


Рисунок 4.1 – Затраты до и после внедрения системы

Диаграмма Ганта проекта внедрения СПР представлена на рисунке 4.2

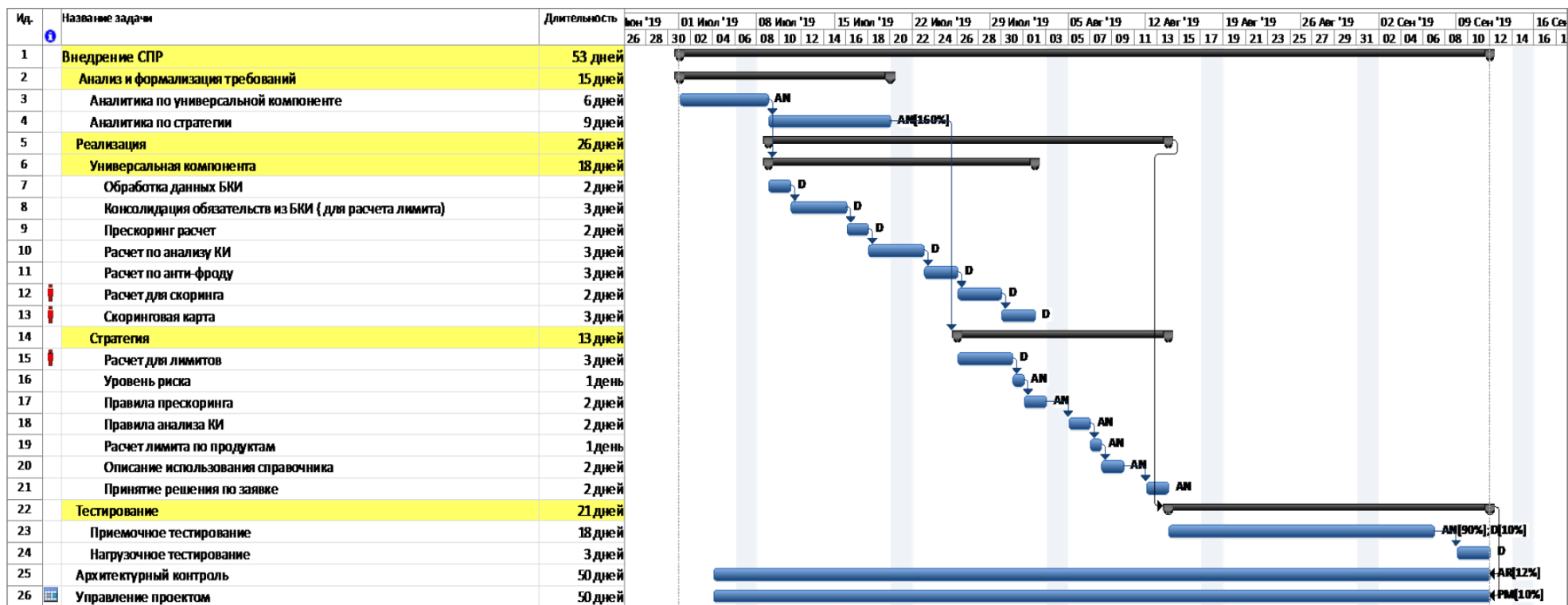


Рисунок 4.2 – Диаграмма Ганта

4.6 Вывод

Система предназначена для повышения эффективности работ операторов, выдающих кредиты, сотрудников риск-отдела, разрабатывающих кредитные стратегии, кредитных инспекторов, проводящих анализ кредитных заявок.

Внедрение системы даёт следующие преимущества:

- Снижение времени обслуживания клиента;
- Высвобождение операторов фронт-офиса;
- Снижение времени разработки кредитной стратегии;
- Высвобождение кредитных инспекторов.

Общая трудоёмкость разработки системы составила: 77 чел/д.

Внедрение системы позволяет экономить средства организации и затраты на внедрение окупаются через 7,89 месяцев. Таким образом, внедрение системы обосновано.

ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы была выполнена разработка системы, позволяющей автоматизировать процесс принятия решения по кредитной заявке в банке.

В ходе выполнения работы проведено комплексное исследование, включавшее в себя анализ предметной области, а также обзор современных технологий обработки информации и принятия управленческих решений. В результате данного исследования был выделен перечень функций, подлежащих автоматизации и выбрано технологическое решение для их реализации.

Разработаны структура АСОИУ и ее архитектура. Спроектированы инфологическая и даталогическая схемы базы данных и проведена оптимизация с использованием GDBSS алгоритма. Так же разработана структура распределенной сети обработки данных.

Для оценки характеристик функционирования системы проведено имитационное моделирование. В качестве системы моделирования была использована GPSS.

Сформированы сверочные витрины для проверки корректности загрузки, укладки и чтения данных. Также проведено тестирование корректности срабатывания бизнес-правил.

Разработано вспомогательное приложение, позволяющее имитировать работу пользователя с системой, что позволило провести комплексное тестирование и оценить качество разработанных алгоритмов обработки информации.

Приведено организационно-экономическое обоснование разработки.

Таким образом, работа выполнена в полном объеме. Описанная система удовлетворяет всем требованиям технического задания и пригодна для функционирования в банке.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Балясова К., Кожекина Л., Сараев А. Обзор рынка потребительского кредитования по итогам 1-го полугодия 2019 года / К. Балясова, Л. Кожекина, А. Сараев // Expert | RA. – 2019.
2. Петухова М.В. Рейтинговая методика оценки кредитного риска физических лиц / М.В. Петухова // Институт экономики и промышленного производства СО РАН: Вестник НГУ. – 2011. – №3(11). – С. 86–93.
3. Рощина Я. Оптимизация процесса оценки кредитоспособности заёмщиков при ипотечном кредитовании в РФ. / Я. Рощина // Российское предпринимательство. – 2010. – №1(1). – С.113–117.
4. Ермакова К.О. Автоматизация процесса кредитования / К.О. Ермакова // Проблемы экономики и менеджмента. – 2015. – №7(47). – С. 32–36.
5. Deductor:Loans - комплексное скоринговое решение потребительского кредитования [Электронный ресурс]. – URL: <http://www.olap.ru/home.asp?artId=157> (дата обращения: 17.10.2019).
6. Diasoft. Брошюра «Кредитный конвейер Flextera» [Электронный ресурс]. – URL: https://www.diasoft.ru/banks/solutions/retail/kreditny_konveyer_fiz_litsa/ (дата обращения: 17.10.2019).
7. Полянский Ю. Организация бизнес-процессов принятия кредитных решений с учетом рисков конкретного вида бизнеса. / Ю. Полянский // Форум «Интеграция корпоративных прикладных систем (ICAS-2011)». – 2011. – С.3–7
8. SAS Real-Time Decision Manager (SAS RTDM) [Электронный ресурс]. – URL: [http://www.tadviser.ru/index.php/%D0%9F%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82:SAS_Real-Time_Decision_Manager_\(SAS_RTDM\)](http://www.tadviser.ru/index.php/%D0%9F%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82:SAS_Real-Time_Decision_Manager_(SAS_RTDM)) (дата обращения: 18.10.2019).

9. SAS(R) Intelligent Decisioning [Электронный ресурс]. – URL: https://www.sas.com/ru_ru/software/intelligent-decisioning.html (дата обращения: 18.10.2019).
10. Метод анализа иерархий: процедура применения [Электронный ресурс]. – URL: <http://vamocenka.ru/metod-analiza-ierarxij-procedura-primeneniya/> (дата обращения: 18.10.2019).
11. Чуричев А. Кредитный конвейер на базе SAS Real-Time Decision Manager [Электронный ресурс]. – URL: https://www.sas.com/content/dam/SAS/ru_ru/doc/Events/Presentation/bishkek-event/alexandr-churichev-sas-based-credit-conveyor-bishkek.pdf (дата обращения: 18.10.2019).
12. Тэйлор Д. SAS RTDM – обзор [Электронный ресурс]. – URL: <http://jtonedm.com/2013/05/29/first-look-sas-real-time-decision-manager-2/> (дата обращения: 18.10.2019).
13. SAS(R) IT Resource Management 3.21: Administrator's Guide, Second Edition [Электронный ресурс]. – URL: <http://support.sas.com/documentation/cdl/en/itrmxag/64938/HTML/default/viewer.htm#titlepage.htm> (дата обращения: 18.10.2019).
14. SAS(R) 9.2 Intelligence Platform: Overview, Second Edition [Электронный ресурс]. – URL: <https://support.sas.com/documentation/cdl/en/biov/63145/PDF/default/biov.pdf> (дата обращения: 18.10.2019).
15. Ephraim Schwartz. SAS launches Real-Time Decision Manager [Электронный ресурс]. – URL: <http://www.infoworld.com/t/data-management/sas-launches-real-time-decision-manager-282> (дата обращения: 04.11.2019).
16. Шувалов С., Путятинский С. Построение кредитного конвейера на базе Neoflex. Конференция IBM. – Impact 2012. М. 2012. – URL: <https://www.ibm.com/ru/events/presentations/impact2012/neoflex.pdf> (дата обращения: 06.11.2019).
17. Неофлекс представил концепцию построения кредитных конвейеров, позволяющую минимизировать риски банков при массовом

- кредитовании [Электронный ресурс]. – URL: https://www.cnews.ru/news/line/neofleks_predstavil_kontseptsuyu_postroeniya (дата обращения: 06.11.2019).
18. Андиева Е., Семенова И. Поддержка принятия решений в системе кредитования / Е. Андиева, И. Семенова. – Омск: СибАДИ. – 2010. – С.168.
19. Red Hat. Особенности JBoss Enterprise Application Platform [Электронный ресурс]. – URL: <http://www.redhat.com/resourcelibrary/articles/jboss-enterprise-application-platform-features> (дата обращения: 09.11.2019).
20. Command line tool and library for transferring data with URLs [Электронный ресурс]. – URL: <https://curl.haxx.se/> (дата обращения: 09.11.2019).
21. VPN туннелирование – особенности VPN соединения [Электронный ресурс]. – URL: <http://www.freevpnservice.ru/vpn-tunneling-article/> (дата обращения: 14.11.2019).
22. Ильиных Т. Е., Шустова Л. И. Проектирование реляционных баз данных в нотациях IDEF1X / Т.Е. Ильиных, Л.И. Шустова // М.: МИФИ. – 2000. – С. 136.
23. Этапы проектирования баз данных [Электронный ресурс]. – URL: http://www.mstu.edu.ru/study/materials/zelenkov/ch_5_1.html (дата обращения: 03.11.2019).
24. Уфук Явуз. Расчет пропускной способности сети для Oracle Data Guard в системах SAP [Электронный ресурс]. – URL: <http://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/6006ac18-7900-2e10-a793-ee2d6b39063f?QuickLink=index&overridelayout=true&49873160777130> (дата обращения: 18.11.2019).
25. Шевченко Д.Н., Крачевня И.Н. Имитационное моделирование на GPSS / Д.Н Шевченко, И.Н. Крачевня // Гомель: БелГУТ. – 2007. – С. 97.

26. Краткое описание языка GPSS [Электронный ресурс]. – URL: http://bigor.bmstu.ru/?cnt/?doc=110_Simul/3017.mod (дата обращения: 02.12.2019).
27. NetApp Best Practice Guidelines for Oracle Database [Электронный ресурс]. – 2015. – URL: http://www.netwell.ru/download/documents/techlibrary/NetApp_Best_Practice_Guidelines_for_Oracle_Data_base.pdf (дата обращения: 09.11.2019)
28. Oracle Active Data Guard [Электронный ресурс]. – URL: <https://www.oracle.com/technetwork/ru/database/activedataguard-ru-434799-ru.pdf> (дата обращения: 09.11.2019).
29. Oracle Active Data Guard Real-Time Data Protection and Availability [Электронный ресурс]. – URL: <https://www.oracle.com/technetwork/database/availability/active-data-guard-ds-12c-1898869.pdf> (дата обращения: 09.11.2019).
30. Oracle Data Guard Concepts and Administration 10g Release 2 [Электронный ресурс]. – URL: https://docs.oracle.com/cd/B19306_01/server.102/b14239/title.htm (дата обращения: 09.11.2019).
31. Обзор Oracle Data Guard [Электронный ресурс]. – URL: <http://www.oracle.com/technetwork/articles/database-performance/dataguardoverview-098960.html> (дата обращения: 10.11.2019).
32. Red Hat JBoss Operations Network. Gain greater efficiency with a single point of control [Электронный ресурс]. – URL: <https://www.redhat.com/en/technologies/jboss-middleware/operations-network> (дата обращения: 20.11.2019).
33. Red Hat JBoss Middleware [Электронный ресурс]. – URL: <https://redhat.axoft.ru/products/JBoss-Middleware> (дата обращения: 20.11.2019).
34. VPN. Кратко о частных соединениях [Электронный ресурс]. – URL: <http://www.freevpnservice.ru/vpn-info/> (дата обращения: 14.11.2019).

35. Типы RAID-массивов [Электронный ресурс]. – URL: <https://integrus.ru/blog/typy-raid-massivov.html> (дата обращения: 03.12.2019).
36. Преимущества СУБД Oracle [Электронный ресурс]. – URL: <http://www.interface.ru/home.asp?artId=32990> (дата обращения: 17.10.2019).
37. Янис Вреде. HardwareLUXx: тест жёстких дисков ёмкостью 1 ТБ [Электронный ресурс]. – 2012. – URL: <http://www.hardwareluxx.ru/index.php/artikel/hardware/storage/23266-roundup-hdd-1-tb.html?start=13> (дата обращения: 09.11.2019).
38. Документация Oracle. Метрика Redo Generated (per second) [Электронный ресурс]. – URL: http://docs.oracle.com/cd/B16240_01/doc/doc.102/e16282/oracle_database_help/oracle_database_instance_throughput_redosize_ps.html (дата обращения: 20.10.2019).
39. Документация Oracle. Архитектура памяти экземпляра БД Oracle [Электронный ресурс]. – URL: http://docs.oracle.com/cd/B19306_01/server.102/b14220/memory.htm (дата обращения: 06.11.2019).
40. The Transaction Processing Performance Council [Электронный ресурс]. – URL: <http://www.tpc.org/> (дата обращения: 06.12.2019).
41. BaseGroupLabs – технологии анализа данных [Электронный ресурс]. – URL: <http://www.basegroup.ru/deductor/> (дата обращения: 09.12.2019).

ПРИЛОЖЕНИЕ А. Описание используемых технологий

А.1 СУБД Oracle 10g

Проведем сравнение СУБД Oracle и СУБД Microsoft SQL Server:

- MS SQL Server не поддерживает индексы типа Bitmap и Reverse, в отличие от Oracle;
- в MS SQL Server не может быть изменен размер блока базы данных. В Oracle – может;
- СУБД Oracle имеет большую производительность и поддерживаемое количество пользователей, чем MS SQL Server;
- MS SQL Server имеет меньшую стоимость лицензии и обслуживания, чем Oracle [36].

СУБД Oracle обладает следующими возможностями, которые полезны для использования в системе:

- позволяет организовывать кластеры серверов СУБД (технология Real Application Cluster), что позволяет увеличить отказоустойчивость и производительность СУБД без замены оборудования [27];
- возможность воспроизведения нагрузки на тестовой базе данных в точном соответствии с нагрузкой на рабочем сервере при помощи технологии Real Application Testing, что позволяет наиболее адекватно проводить нагрузочное тестирование системы;
- обеспечение отказоустойчивости, репликации и балансировки нагрузки с помощью технологии Active Data Guard [28];
- высокий уровень производительности обработки запросов.

А.2 Утилита cURL

cURL – свободная кроссплатформенная служебная программа командной строки, позволяющая взаимодействовать с множеством различных серверов по множеству различных протоколов с синтаксисом URL.

Программа поддерживает протоколы: FTP, FTPS, HTTP, HTTPS, TFTP, SCP, SFTP, Telnet, DICT, File, LDAP, а также POP3, IMAP и SMTP. Также

cURL поддерживает сертификаты HTTPS, методы HTTP POST, HTTP PUT, загрузку на FTP, загрузку через формы HTTP [20].

В СРП данная утилита используется для вызова внешних веб-сервисов.

A.3 Red Hat JBOSS Enterprise Application Platform

Red Hat JBoss Enterprise Application Platform (EAP) является открытой платформой для JAVA-приложений и является основой для промежуточного слоя RedHat JBoss. EAP позволяет разворачивать приложения на физической, виртуальной и облачной средах; поддерживает широкий спектр фреймворков, базирующихся на виртуальной машине Java (JVM); включает в себя сервер приложений JBoss Application Server [19], имеющий возможности кластеризации, обмена сообщениями, распределенного кэширования и других технологий создания устойчивой и масштабируемой платформы.

JBoss Enterprise Application Platform оптимизирован на работу и на нескольких ядрах и в виртуальной среде. Это позволяет снизить потребность в физических ресурсах и тем самым увеличить производительность [32].

JBoss Enterprise Application Platform сервер приложений, который может использоваться для работы с приложениями обработки больших объемов транзакций в условиях повышенных требований к надежности. Также обеспечивает: отказоустойчивость, кэширование, интеллектуальную балансировку нагрузки и распределенное развертывание [33].

A.4 VPN

VPN (virtual private network — виртуальные частные сети) — метод, позволяющий шифровать информационное соединение между компьютером пользователя и другим компьютером сети. Другой компьютер может принадлежать организации пользователя, доверенному лицу или коммерческому VPN сервису [34].

Использование VPN соединения или других разновидностей туннелирования с целью шифрования информации является способом обеспечения её закрытости для всех, кроме тех людей, которым эта

информация предназначена. У этого способа есть еще один дополнительный эффект, который заключается в том, что различные виды трафика выглядят для «подслушивающего» одинаково.

Система VPN подразумевает шифрование всех данных между компьютерами сети.

Системы VPN передают все данные через зашифрованную сеть.

Существует несколько различных стандартов установки VPN сетей, включая IPSec, SSL/TLS и PPTP, которые различаются по своей сложности, по обеспечиваемому уровню безопасности и по совместимости с отдельными операционными системами.

VPN системы, созданные на основе SSL/TLS, довольно просты в настройке и обеспечивают высокий уровень безопасности [21, 34].

IPSec работает на уровне сети Интернет, который ответственен за передачу пакетов данных в Интернет архитектуре, в то время как другие стандарты используют уровень приложений. Этот факт говорит о том, что IPSec является более гибким инструментом, так как он может быть использован для защиты всех остальных протоколов, находящихся на более высоких уровнях. Кроме того, приложения не должны быть разработаны специально для использования IPSec, в то время как функции SSL/TLS или других протоколов высокого уровня должны быть встроены в приложение.

A.5 Oracle DataGuard

Технология DataGuard является центральным компонентом интегрированного набора решений Oracle Database High Availability (HA - высокая готовность Oracle Database), который гарантирует непрерывность ведения бизнеса, сводя к минимуму время различных видов запланированных и незапланированных простоев, которые могут затронуть предприятие [28].

Active DataGuard – представляет собой решение, которое хранит актуальную копию производственной БД для перенаправления интенсивных

запросов, отчетов и резервных копий с производственной базы данных на резервную [29].

Oracle Active DataGuard позволяет открывать физические резервные базы (physical stand by database) данных на чтение – для создания отчетов, простых или сложных запросов, сортировки, доступа через веб-интерфейс и т.д., в то время, как изменения из производственной базы данных применяются на резервной[30]. Все запросы из резервной физической базы данных выполняются в реальном времени и возвращают текущие результаты. Любая операция, которая требует быстрый доступ на чтение, может быть выполнена в резервной физической базе данных, что увеличивает производительность и разгружает основную базу данных.

Active DataGuard отслеживает изменения через RMAN (Oracle Recovery Manager – менеджер восстановления Oracle), что позволяет выгружать быстрые инкрементальные резервные копии с основной на резервную базу данных. Скорость инкрементальных резервных копий может в несколько раз превышать скорость бэкапов резервных баз данных в ранних версиях продукта [31].

A.6 RAID10

RAID (redundant array of independent disks — избыточный массив независимых дисков) — массив из нескольких дисков (запоминающих устройств), управляемых контроллером, связанных между собой скоростными каналами передачи данных и воспринимаемых внешней системой как единое целое. В зависимости от типа используемого массива может обеспечивать различные степени отказоустойчивости и быстродействия. Служит для повышения надёжности хранения данных и/или для повышения скорости чтения/записи.

RAID 10 — зеркалированный массив, данные в котором записываются последовательно на несколько дисков, как в RAID 0. Эта архитектура представляет собой массив типа RAID 0, сегментами которого вместо

отдельных дисков являются массивы RAID 1. Соответственно, массив этого уровня должен содержать как минимум 4 диска (и всегда чётное количество). Общая схема RAID 10 представлена на рисунке А.1. RAID 10 объединяет в себе высокую отказоустойчивость и производительность [35].

Несмотря на то, что для данного уровня RAID возможно сохранение целостности данных при выходе из строя половины дисков, необратимое разрушение массива происходит при выходе из строя уже двух дисков, если они находятся в одной зеркальной паре.

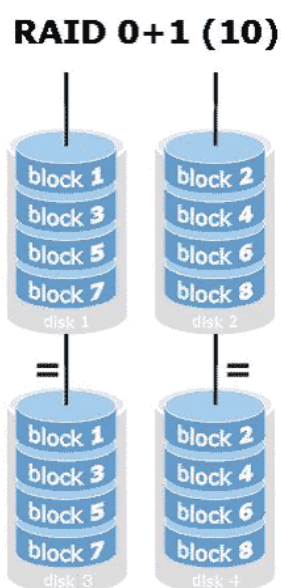


Рисунок А.1 - Схема RAID-10

ПРИЛОЖЕНИЕ Б. Оптимизация схемы MAIN

Пусть U – универсальная схема отношения (множество всех атрибутов предметной области) и F – множество ФЗ.

Перед выполнением алгоритма можно привести все ФЗ к одному атрибуту в правой части (по свойству декомпозиции) и удалить лишние ФЗ. Но это не обязательно.

Алгоритм:

1. Построение УНП:

$$\text{УНП}^+ = (A_1 \rightarrow A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_{17} A_{18} A_{19} A_{20} A_{21} A_{22} A_{23} A_{24} A_{25} A_{26} A_{27} A_{28} A_{29} A_{30} A_{31} A_{32} A_{33} A_{34} A_{35} A_{36} A_{37} A_{38} A_{39} B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} B_{13} B_{14} B_{15} B_{16} B_{17} B_{18} B_{19} B_{20} B_{21} B_{22} B_{23} B_{24} B_{25} B_{26} B_{27} B_{28} B_{29} B_{30} B_{31} M_1 M_2 M_3 M_4 M_5 M_6 M_7 N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8 N_9 N_{10} N_{11} N_{12} N_{13} N_{14} N_{15} N_{16};$$
$$A_2 \rightarrow A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_{17} A_{18} A_{19} A_{20} A_{21} A_{22} A_{23} A_{24} A_{25} A_{26} A_{27} A_{28} A_{29} A_{30} A_{31} A_{32} A_{33} A_{34} A_{35} A_{36} A_{37} A_{38} A_{39} M_1 M_2 M_3 M_4 M_5 M_6 M_7 B_6 B_7;$$
$$A_7 \rightarrow B_6;$$
$$A_{13} \rightarrow B_7;$$
$$A_{10} A_{11} A_{12} A_{14} \rightarrow A_3 A_4 A_5 A_6 A_{39};$$
$$A_1 F_1 \rightarrow A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_{17} A_{18} A_{19} A_{20} A_{21} A_{22} A_{23} A_{24} A_{25} A_{26} A_{27} A_{28} A_{29} A_{30} A_{31} A_{32} A_{33} A_{34} A_{35} A_{36} A_{37} A_{38} A_{39} B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} B_{13} B_{14} B_{15} B_{16} B_{17} B_{18} B_{19} B_{20} B_{21} B_{22} B_{23} B_{24} B_{25} B_{26} B_{27} B_{28} B_{29} B_{30} B_{31} F_2 F_3 F_4 N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8 N_9 N_{10} N_{11} N_{12} N_{13} N_{14} N_{15} N_{16} M_1 M_2 M_3 M_4 M_5 M_6 M_7;$$
$$A_1 C_1 C_5 \rightarrow A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_{17} A_{18} A_{19} A_{20} A_{21} A_{22} A_{23} A_{24} A_{25} A_{26} A_{27} A_{28} A_{29} A_{30} A_{31} A_{32} A_{33} A_{34} A_{35} A_{36} A_{37} A_{38} A_{39} B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} B_{13} B_{14} B_{15} B_{16} B_{17} B_{18} B_{19} B_{20} B_{21} B_{22} B_{23} B_{24} B_{25} B_{26} B_{27} B_{28} B_{29} B_{30} B_{31} C_2 C_3 C_4 C_6 N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8 N_9 N_{10} N_{11} N_{12} N_{13} N_{14} N_{15} N_{16} M_1 M_2 M_3 M_4 M_5 M_6 M_7;$$
$$A_8 A_9 B_8 C_1 \rightarrow C_2 C_3 M_1 M_2 M_3 M_4 M_5 M_6 M_7;$$
$$B_8 B_9 \rightarrow B_{10};$$
$$B_1 B_2 B_3 B_4 \rightarrow B_5;$$
$$A_8 \rightarrow M_1 M_2 M_3 M_4 M_5 M_6 M_7;$$
$$N_1 \rightarrow N_2 N_3 N_4 N_5 N_6 N_7 N_8 N_9 N_{10} N_{11} N_{12} N_{13} N_{14} N_{15} N_{16}).$$

2. Так как нет ФЗ, включающей все атрибуты из U , добавляем в УНП тривиальную ФЗ:

$$A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_{17} A_{18} A_{19} A_{20} A_{21} A_{22} A_{23} A_{24} A_{25} A_{26} \\ A_{27} A_{28} A_{29} A_{30} A_{31} A_{32} A_{33} A_{34} A_{35} A_{36} A_{37} A_{38} A_{39} B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} B_{13} B_{14} B_{15} \\ B_{16} B_{17} B_{18} B_{19} B_{20} B_{21} B_{22} B_{23} B_{24} B_{25} B_{26} B_{27} B_{28} B_{29} B_{30} B_{31} C_1 C_2 C_3 C_4 C_5 C_6 F_1 F_2 F_3 F_4 M_1 M_2 M_3 M_4 \\ M_5 M_6 M_7 N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8 N_9 N_{10} N_{11} N_{12} N_{13} N_{14} N_{15} N_{16} \rightarrow \emptyset$$

3. Привести все нетривиальные ФЗ из УНП к неприводимому виду:

$$A_{10} A_{11} A_{12} A_{14} \rightarrow A_3 A_4 A_5 A_6 A_{39};$$

$A_{10}^+ \rightarrow A_{10};$	$A_3 A_4 A_5 A_6 A_{39} \notin A_{10}^+$
$A_{11}^+ \rightarrow A_{11};$	$A_3 A_4 A_5 A_6 A_{39} \notin A_{11}^+$
$A_{12}^+ \rightarrow A_{12};$	$A_3 A_4 A_5 A_6 A_{39} \notin A_{12}^+$
$A_{14}^+ \rightarrow A_{14};$	$A_3 A_4 A_5 A_6 A_{39} \notin A_{14}^+$
$(A_{10} A_{11})^+ \rightarrow A_{10} A_{11};$	$A_3 A_4 A_5 A_6 A_{39} \notin (A_{10} A_{11})^+$
$(A_{10} A_{12})^+ \rightarrow A_{10} A_{12};$	$A_3 A_4 A_5 A_6 A_{39} \notin (A_{10} A_{12})^+$
$(A_{10} A_{14})^+ \rightarrow A_{10} A_{14};$	$A_3 A_4 A_5 A_6 A_{39} \notin (A_{10} A_{14})^+$
$(A_{11} A_{12})^+ \rightarrow A_{11} A_{12};$	$A_3 A_4 A_5 A_6 A_{39} \notin (A_{11} A_{12})^+$
$(A_{11} A_{14})^+ \rightarrow A_{11} A_{14};$	$A_3 A_4 A_5 A_6 A_{39} \notin (A_{11} A_{14})^+$
$(A_{12} A_{14})^+ \rightarrow A_{12} A_{14};$	$A_3 A_4 A_5 A_6 A_{39} \notin (A_{12} A_{14})^+$
$(A_{10} A_{11} A_{12})^+ \rightarrow A_{10} A_{11} A_{12};$	$A_3 A_4 A_5 A_6 A_{39} \notin (A_{10} A_{11} A_{12})^+$
$(A_{10} A_{11} A_{14})^+ \rightarrow A_{10} A_{11} A_{14};$	$A_3 A_4 A_5 A_6 A_{39} \notin (A_{10} A_{11} A_{14})^+$
$(A_{10} A_{12} A_{14})^+ \rightarrow A_{10} A_{12} A_{14};$	$A_3 A_4 A_5 A_6 A_{39} \notin (A_{10} A_{12} A_{14})^+$
$(A_{11} A_{12} A_{14})^+ \rightarrow A_{11} A_{12} A_{14};$	$A_3 A_4 A_5 A_6 A_{39} \notin (A_{11} A_{12} A_{14})^+$

Из рассматриваемого ФЗ нельзя удалить атрибуты в левой части.

$$A_1 F_1 \rightarrow A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_{17} A_{18} A_{19} A_{20} A_{21} A_{22} A_{23} A_{24} A_{25} \\ A_{26} A_{27} A_{28} A_{29} A_{30} A_{31} A_{32} A_{33} A_{34} A_{35} A_{36} A_{37} A_{38} A_{39} B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} B_{13} B_{14} \\ B_{15} B_{16} B_{17} B_{18} B_{19} B_{20} B_{21} B_{22} B_{23} B_{24} B_{25} B_{26} B_{27} B_{28} B_{29} B_{30} B_{31} F_2 F_3 F_4 N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8 N_9 \\ N_{10} N_{11} N_{12} N_{13} N_{14} N_{15} N_{16} M_1 M_2 M_3 M_4 M_5 M_6 M_7;$$

$$A_1^+ \rightarrow A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_{17} A_{18} A_{19} A_{20} A_{21} A_{22} A_{23} A_{24} A_{25} \\ A_{26} A_{27} A_{28} A_{29} A_{30} A_{31} A_{32} A_{33} A_{34} A_{35} A_{36} A_{37} A_{38} A_{39} B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} B_{13} B_{14} \\ B_{15} B_{16} B_{17} B_{18} B_{19} B_{20} B_{21} B_{22} B_{23} B_{24} B_{25} B_{26} B_{27} B_{28} B_{29} B_{30} B_{31} M_1 M_2 M_3 M_4 M_5 M_6 M_7 N_1 N_2 N_3 \\ N_4 N_5 N_6 N_7 N_8 N_9 N_{10} N_{11} N_{12} N_{13} N_{14} N_{15} N_{16};$$

$$A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_{17} A_{18} A_{19} A_{20} A_{21} A_{22} A_{23} A_{24} A_{25} A_{26} \\ A_{27} A_{28} A_{29} A_{30} A_{31} A_{32} A_{33} A_{34} A_{35} A_{36} A_{37} A_{38} A_{39} B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} B_{13} B_{14} B_{15} \\ B_{16} B_{17} B_{18} B_{19} B_{20} B_{21} B_{22} B_{23} B_{24} B_{25} B_{26} B_{27} B_{28} B_{29} B_{30} B_{31} F_2 F_3 F_4 N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8 N_9 N_{10} \\ N_{11} N_{12} N_{13} N_{14} N_{15} N_{16} M_1 M_2 M_3 M_4 M_5 M_6 M_7 \notin A_1^+$$

$$B_{16}B_{17}B_{18}B_{19}B_{20}B_{21}B_{22}B_{23}B_{24}B_{25}B_{26}B_{27}B_{28}B_{29}B_{30}B_{31}C_2C_3C_4C_6N_1N_2N_3N_4N_5N_6N_7N_8N_9 \\ N_{10}N_{11}N_{12}N_{13}N_{14}N_{15}N_{16}M_1M_2M_3M_4M_5M_6M_7 \notin (A_1C_1)^+$$

$$(A_1C_5)^+ \rightarrow A_1A_2A_3A_4A_5A_6A_7A_8A_9A_{10}A_{11}A_{12}A_{13}A_{14}A_{15}A_{16}A_{17}A_{18}A_{19}A_{20}A_{21}A_{22}A_{23} \\ A_{24}A_{25}A_{26}A_{27}A_{28}A_{29}A_{30}A_{31}A_{32}A_{33}A_{34}A_{35}A_{36}A_{37}A_{38}A_{39}B_1B_2B_3B_4B_5B_6B_7B_8B_9B_{10}B_{11}B_{12} \\ B_{13}B_{14}B_{15}B_{16}B_{17}B_{18}B_{19}B_{20}B_{21}B_{22}B_{23}B_{24}B_{25}B_{26}B_{27}B_{28}B_{29}B_{30}B_{31}C_5N_1N_2N_3N_4N_5N_6N_7N_8 \\ N_9N_{10}N_{11}N_{12}N_{13}N_{14}N_{15}N_{16}M_1M_2M_3M_4M_5M_6M_7;$$

$$A_2A_3A_4A_5A_6A_7A_8A_9A_{10}A_{11}A_{12}A_{13}A_{14}A_{15}A_{16}A_{17}A_{18}A_{19}A_{20}A_{21}A_{22}A_{23}A_{24}A_{25}A_{26}A_{27} \\ A_{28}A_{29}A_{30}A_{31}A_{32}A_{33}A_{34}A_{35}A_{36}A_{37}A_{38}A_{39}B_1B_2B_3B_4B_5B_6B_7B_8B_9B_{10}B_{11}B_{12}B_{13}B_{14}B_{15}B_{16} \\ B_{17}B_{18}B_{19}B_{20}B_{21}B_{22}B_{23}B_{24}B_{25}B_{26}B_{27}B_{28}B_{29}B_{30}B_{31}C_2C_3C_4C_6N_1N_2N_3N_4N_5N_6N_7N_8N_9N_{10} \\ N_{11}N_{12}N_{13}N_{14}N_{15}N_{16}M_1M_2M_3M_4M_5M_6M_7 \notin (A_1C_5)^+ \\ (C_1C_5)^+ \rightarrow C_1C_5;$$

$$A_2A_3A_4A_5A_6A_7A_8A_9A_{10}A_{11}A_{12}A_{13}A_{14}A_{15}A_{16}A_{17}A_{18}A_{19}A_{20}A_{21}A_{22}A_{23}A_{24}A_{25}A_{26}A_{27} \\ A_{28}A_{29}A_{30}A_{31}A_{32}A_{33}A_{34}A_{35}A_{36}A_{37}A_{38}A_{39}B_1B_2B_3B_4B_5B_6B_7B_8B_9B_{10}B_{11}B_{12}B_{13}B_{14}B_{15}B_{16} \\ B_{17}B_{18}B_{19}B_{20}B_{21}B_{22}B_{23}B_{24}B_{25}B_{26}B_{27}B_{28}B_{29}B_{30}B_{31}C_2C_3C_4C_6N_1N_2N_3N_4N_5N_6N_7N_8N_9N_{10} \\ N_{11}N_{12}N_{13}N_{14}N_{15}N_{16}M_1M_2M_3M_4M_5M_6M_7 \notin (C_1C_5)^+$$

Из рассматриваемого ФЗ нельзя удалить атрибуты в левой части.

$$A_8A_9B_8C_1 \rightarrow C_2C_3;$$

$A_8^+ \rightarrow A_8;$	$C_2C_3 \notin A_8^+$
$A_9^+ \rightarrow A_9;$	$C_2C_3 \notin A_9^+$
$B_8^+ \rightarrow B_8;$	$C_2C_3 \notin B_8^+$
$C_1^+ \rightarrow C_1;$	$C_2C_3 \notin C_1^+$
$(A_8A_9)^+ \rightarrow A_8A_9;$	$C_2C_3 \notin (A_8A_9)^+$
$(A_8B_8)^+ \rightarrow A_8B_8;$	$C_2C_3 \notin (A_8B_8)^+$
$(A_8C_1)^+ \rightarrow A_8C_1;$	$C_2C_3 \notin (A_8C_1)^+$
$(A_9B_8)^+ \rightarrow A_9B_8;$	$C_2C_3 \notin (A_9B_8)^+$
$(A_9C_1)^+ \rightarrow A_9C_1;$	$C_2C_3 \notin (A_9C_1)^+$
$(B_8C_1)^+ \rightarrow B_8C_1;$	$C_2C_3 \notin (B_8C_1)^+$
$(A_8A_9B_8)^+ \rightarrow A_8A_9B_8;$	$C_2C_3 \notin (A_8A_9B_8)^+$
$(A_8A_9C_1)^+ \rightarrow A_8A_9C_1;$	$C_2C_3 \notin (A_8A_9C_1)^+$
$(A_8B_8C_1)^+ \rightarrow A_8B_8C_1;$	$C_2C_3 \notin (A_8B_8C_1)^+$
$(A_9B_8C_1)^+ \rightarrow A_9B_8C_1;$	$C_2C_3 \notin (A_9B_8C_1)^+$

Из рассматриваемого ФЗ нельзя удалить атрибуты в левой части.

$$B_8B_9 \rightarrow B_{10};$$

$$\begin{array}{ll}
B_1^+ \rightarrow B_1; & B_8 B_9 \notin B_1^+ \\
B_9^+ \rightarrow B_9; & B_8 B_9 \notin B_9^+
\end{array}$$

Из рассматриваемого ФЗ нельзя удалить атрибуты в левой части.

$$B_1 B_2 B_3 B_4 \rightarrow B_5;$$

$$\begin{array}{ll}
B_1^+ \rightarrow B_1; & B_5 \notin B_1^+ \\
B_2^+ \rightarrow B_2; & B_5 \notin B_2^+ \\
B_3^+ \rightarrow B_3; & B_5 \notin B_3^+ \\
B_4^+ \rightarrow B_4; & B_5 \notin B_4^+ \\
(B_1 B_2)^+ \rightarrow B_1 B_2; & B_5 \notin (B_1 B_2)^+ \\
(B_1 B_3)^+ \rightarrow B_1 B_3; & B_5 \notin (B_1 B_3)^+ \\
(B_1 B_4)^+ \rightarrow B_1 B_4; & B_5 \notin (B_1 B_4)^+ \\
(B_2 B_3)^+ \rightarrow B_2 B_3; & B_5 \notin (B_2 B_3)^+ \\
(B_2 B_4)^+ \rightarrow B_2 B_4; & B_5 \notin (B_2 B_4)^+ \\
(B_3 B_4)^+ \rightarrow B_3 B_4; & B_5 \notin (B_3 B_4)^+ \\
(B_1 B_2 B_3)^+ \rightarrow B_1 B_2 B_3; & B_5 \notin (B_1 B_2 B_3)^+ \\
(B_1 B_2 B_4)^+ \rightarrow B_1 B_2 B_4; & B_5 \notin (B_1 B_2 B_4)^+ \\
(B_1 B_3 B_4)^+ \rightarrow B_1 B_3 B_4; & B_5 \notin (B_1 B_3 B_4)^+ \\
(B_2 B_3 B_4)^+ \rightarrow B_2 B_3 B_4; & B_5 \notin (B_2 B_3 B_4)^+
\end{array}$$

Из рассматриваемого ФЗ нельзя удалить атрибуты в левой части.

4. Разбить ФЗ из УНП на классы эквивалентности:

$$\begin{aligned}
Kl_1 = & A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_{17} A_{18} A_{19} A_{20} A_{21} A_{22} A_{23} A_{24} \\
& A_{25} A_{26} A_{27} A_{28} A_{29} A_{30} A_{31} A_{32} A_{33} A_{34} A_{35} A_{36} A_{37} A_{38} A_{39} B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} B_{13} \\
& B_{14} B_{15} B_{16} B_{17} B_{18} B_{19} B_{20} B_{21} B_{22} B_{23} B_{24} B_{25} B_{26} B_{27} B_{28} B_{29} B_{30} B_{31} M_1 M_2 M_3 M_4 M_5 M_6 M_7 N_1 N_2 \\
& N_3 N_4 N_5 N_6 N_7 N_8 N_9 N_{10} N_{11} N_{12} N_{13} N_{14} N_{15} N_{16};
\end{aligned}$$

$$\begin{aligned}
Kl_2 = & A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_{17} A_{18} A_{19} A_{20} A_{21} A_{22} A_{23} A_{24} A_{25} \\
& A_{26} A_{27} A_{28} A_{29} A_{30} A_{31} A_{32} A_{33} A_{34} A_{35} A_{36} A_{37} A_{38} A_{39} M_1 M_2 M_3 M_4 M_5 M_6 M_7 B_6 B_7;
\end{aligned}$$

$$Kl_3 = A_{10} A_{11} A_{12} A_{14} A_3 A_4 A_5 A_6 A_{39};$$

$$\begin{aligned}
Kl_4 = & A_1 F_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_{17} A_{18} A_{19} A_{20} A_{21} A_{22} A_{23} A_{24} \\
& A_{25} A_{26} A_{27} A_{28} A_{29} A_{30} A_{31} A_{32} A_{33} A_{34} A_{35} A_{36} A_{37} A_{38} A_{39} B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} B_{13} \\
& B_{14} B_{15} B_{16} B_{17} B_{18} B_{19} B_{20} B_{21} B_{22} B_{23} B_{24} B_{25} B_{26} B_{27} B_{28} B_{29} B_{30} B_{31} F_2 F_3 F_4 N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8 \\
& N_9 N_{10} N_{11} N_{12} N_{13} N_{14} N_{15} N_{16};
\end{aligned}$$

$$Kl_5 = A_1 C_1 C_5 A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_{17} A_{18} A_{19} A_{20} A_{21} A_{22} A_{23}$$

$$A_{24}A_{25}A_{26}A_{27}A_{28}A_{29}A_{30}A_{31}A_{32}A_{33}A_{34}A_{35}A_{36}A_{37}A_{38}A_{39}B_1B_2B_3B_4B_5B_6B_7B_8B_9B_{10}B_{11}B_{12} \\ B_{13}B_{14}B_{15}B_{16}B_{17}B_{18}B_{19}B_{20}B_{21}B_{22}B_{23}B_{24}B_{25}B_{26}B_{27}B_{28}B_{29}B_{30}B_{31}C_2C_3C_4C_6N_1N_2N_3N_4N_5N_6 \\ N_7N_8N_9N_{10}N_{11}N_{12}N_{13}N_{14}N_{15}N_{16};$$

$$Kl_6 = A_8A_9B_8C_1C_2C_3M_1M_2M_3M_4M_5M_6M_7;$$

$$Kl_7 = B_8B_9B_{10};$$

$$Kl_8 = B_1B_2B_3B_4B_5;$$

$$Kl_9 = A_7B_6;$$

$$Kl_{10} = A_{13}B_7;$$

$$Kl_{11} = A_8M_1M_2M_3M_4M_5M_6M_7;$$

$$Kl_{12} = N_1N_2N_3N_4N_5N_6N_7N_8N_9N_{10}N_{11}N_{12}N_{13}N_{14}N_{15}N_{16};$$

$$Kl_{13} = A_1A_2A_3A_4A_5A_6A_7A_8A_9A_{10}A_{11}A_{12}A_{13}A_{14}A_{15}A_{16}A_{17}A_{18}A_{19}A_{20}A_{21}A_{22}A_{23}A_{24} \\ A_{25}A_{26}A_{27}A_{28}A_{29}A_{30}A_{31}A_{32}A_{33}A_{34}A_{35}A_{36}A_{37}A_{38}A_{39}B_1B_2B_3B_4B_5B_6B_7B_8B_9B_{10}B_{11}B_{12}B_{13} \\ B_{14}B_{15}B_{16}B_{17}B_{18}B_{19}B_{20}B_{21}B_{22}B_{23}B_{24}B_{25}B_{26}B_{27}B_{28}B_{29}B_{30}B_{31}C_1C_2C_3C_4C_5C_6F_1F_2F_3F_4M_1M_2 \\ M_3M_4M_5M_6M_7N_1N_2N_3N_4N_5N_6N_7N_8N_9N_{10}N_{11}N_{12}N_{13}N_{14}N_{15}N_{16}$$

5. Построить граф иерархии классов эквивалентности (Рисунок Б.1);
6. В каждом классе эквивалентности оставить одну ФЗ;
7. Редуцировать атрибуты справа в оставшихся ФЗ;
8. Исключить из рассмотрения ФЗ, у которых пустая правая часть (кроме $U \rightarrow \emptyset$);
9. На основе атрибутов оставшихся классов эквивалентности получить схему отношения:

$$\rho = (R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9, R_{10}, R_{11}, R_{12}, R_{13})$$

$$R_1 = A_1A_2B_1B_2B_3B_4B_8B_9B_{11}B_{12}B_{13}B_{14}B_{15}B_{16}B_{17}B_{18}B_{19}B_{20}B_{21}B_{22}B_{23}B_{24}B_{25}B_{26} \\ B_{27}B_{28}B_{29}B_{30}B_{31}N_1;$$

$$R_2 = A_2A_7A_8A_9A_{10}A_{11}A_{12}A_{13}A_{14}A_{15}A_{16}A_{17}A_{18}A_{19}A_{20}A_{21}A_{22}A_{23}A_{24}A_{25}A_{26}A_{27} \\ A_{28}A_{29}A_{30}A_{31}A_{32}A_{33}A_{34}A_{35}A_{36}A_{37}A_{38};$$

$$R_3 = A_{10}A_{11}A_{12}A_{14}A_3A_4A_5A_6A_{39};$$

$$R_4 = A_1F_1F_2F_3F_4; \quad R_9 = A_1C_1C_5F_1$$

$$R_5 = A_1C_1C_5C_4C_6; \quad R_{10} = A_7B_6$$

$$R_6 = A_8A_9B_8C_1C_2C_3; \quad R_{11} = A_{13}B_7$$

$$R_7 = B_8B_9B_{10}; \quad R_{12} = A_8M_1M_2M_3M_4M_5M_6M_7$$

$$R_8 = B_1B_2B_3B_4B_5; \quad R_{13} = N_1N_2N_3N_4N_5N_6N_7N_8N_9N_{10}N_{11}N_{12}N_{13}N_{14}N_{15}N_{16}$$

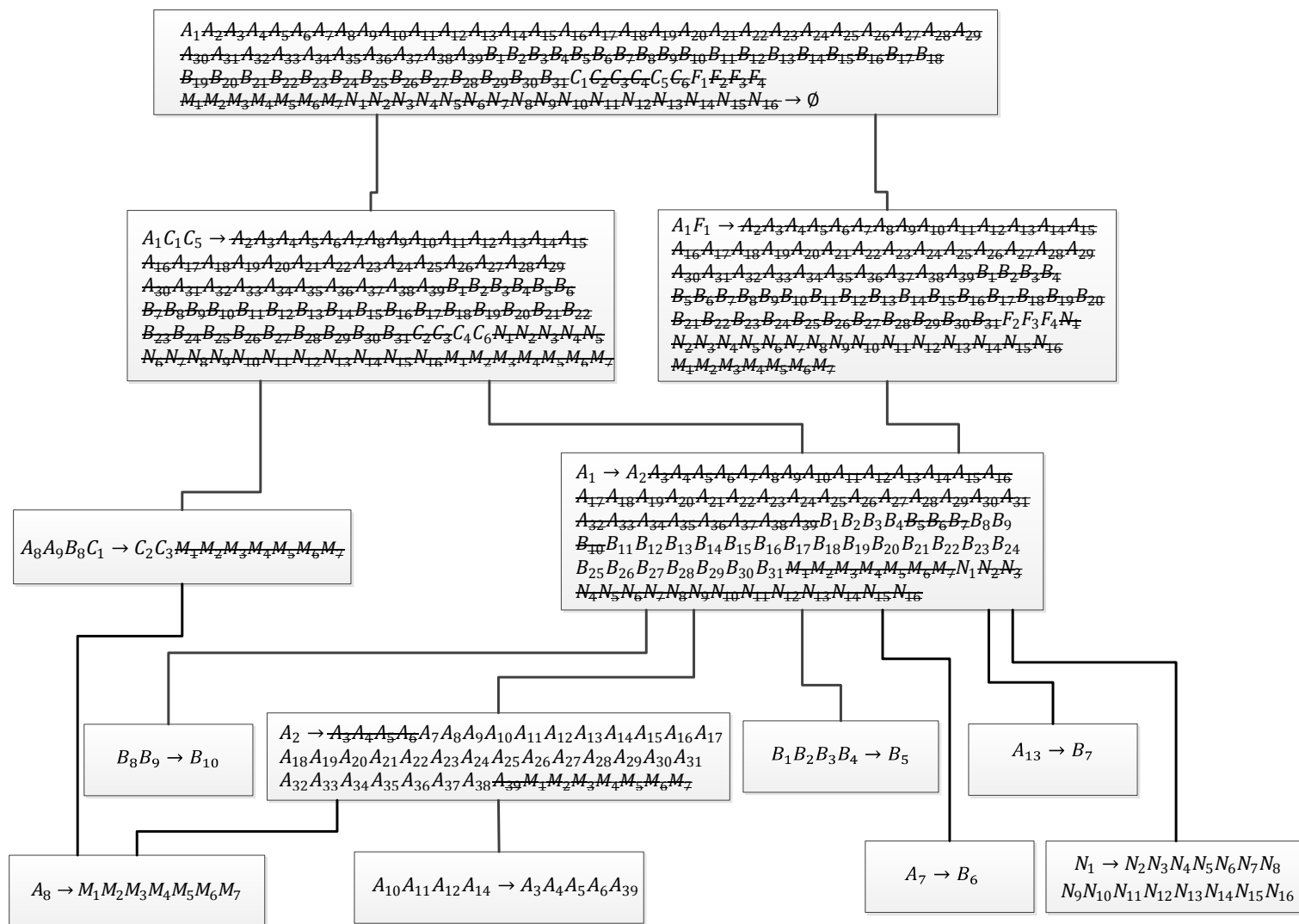


Рисунок Б.1 – Граф иерархии классов эквивалентности

10. Проверка на свойство сохранения ФЗ:

$$H = \emptyset$$

$$G^+ = (A_1 \rightarrow A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_{17} A_{18} A_{19} A_{20} A_{21} A_{22} A_{23} A_{24} A_{25} A_{26} A_{27} A_{28} A_{29} A_{30} A_{31} A_{32} A_{33} A_{34} A_{35} A_{36} A_{37} A_{38} A_{39} B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} B_{13} B_{14} B_{15} B_{16} B_{17} B_{18} B_{19} B_{20} B_{21} B_{22} B_{23} B_{24} B_{25} B_{26} B_{27} B_{28} B_{29} B_{30} B_{31} M_1 M_2 M_3 M_4 M_5 M_6 M_7 N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8 N_9 N_{10} N_{11} N_{12} N_{13} N_{14} N_{15} N_{16};$$

$$A_2 \rightarrow A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_{17} A_{18} A_{19} A_{20} A_{21} A_{22} A_{23} A_{24} A_{25} A_{26} A_{27} A_{28} A_{29} A_{30} A_{31} A_{32} A_{33} A_{34} A_{35} A_{36} A_{37} A_{38} A_{39} M_1 M_2 M_3 M_4 M_5 M_6 M_7 B_6 B_7;$$

$$A_7 \rightarrow B_6;$$

$$A_{13} \rightarrow B_7;$$

$$A_{10} A_{11} A_{12} A_{14} \rightarrow A_3 A_4 A_5 A_6 A_{39};$$

$$A_1 F_1 \rightarrow A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_{17} A_{18} A_{19} A_{20} A_{21} A_{22} A_{23} A_{24} A_{25} A_{26} A_{27} A_{28} A_{29} A_{30} A_{31} A_{32} A_{33} A_{34} A_{35} A_{36} A_{37} A_{38} A_{39} B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} B_{13} B_{14} B_{15} B_{16} B_{17} B_{18} B_{19} B_{20} B_{21} B_{22} B_{23} B_{24} B_{25} B_{26} B_{27} B_{28} B_{29} B_{30} B_{31} F_2 F_3 F_4 N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8 N_9 N_{10} N_{11} N_{12} N_{13} N_{14} N_{15} N_{16} M_1 M_2 M_3 M_4 M_5 M_6 M_7;$$

$$A_1 C_1 C_5 \rightarrow A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_{17} A_{18} A_{19} A_{20} A_{21} A_{22} A_{23} A_{24} A_{25} A_{26} A_{27} A_{28} A_{29} A_{30} A_{31} A_{32} A_{33} A_{34} A_{35} A_{36} A_{37} A_{38} A_{39} B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} B_{13} B_{14} B_{15} B_{16} B_{17} B_{18} B_{19} B_{20} B_{21} B_{22} B_{23} B_{24} B_{25} B_{26} B_{27} B_{28} B_{29} B_{30} B_{31} C_2 C_3 C_4 C_6 N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8 N_9 N_{10} N_{11} N_{12} N_{13} N_{14} N_{15} N_{16} M_1 M_2 M_3 M_4 M_5 M_6 M_7;$$

$$A_8 A_9 B_8 C_1 \rightarrow C_2 C_3 M_1 M_2 M_3 M_4 M_5 M_6 M_7;$$

$$B_8 B_9 \rightarrow B_{10};$$

$$B_1 B_2 B_3 B_4 \rightarrow B_5;$$

$$A_8 \rightarrow M_1 M_2 M_3 M_4 M_5 M_6 M_7;$$

$$N_1 \rightarrow N_2 N_3 N_4 N_5 N_6 N_7 N_8 N_9 N_{10} N_{11} N_{12} N_{13} N_{14} N_{15} N_{16}).$$

$$H = (A_1 \rightarrow A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_{17} A_{18} A_{19} A_{20} A_{21} A_{22} A_{23} A_{24} A_{25} A_{26} A_{27} A_{28} A_{29} A_{30} A_{31} A_{32} A_{33} A_{34} A_{35} A_{36} A_{37} A_{38} A_{39} B_5 B_6 B_7 B_{10} M_1 M_2 M_3 M_4 M_5 M_6 M_7 N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8 N_9 N_{10} N_{11} N_{12} N_{13} N_{14} N_{15} N_{16};$$

$$A_2 \rightarrow A_3 A_4 A_5 A_6 A_{39} M_1 M_2 M_3 M_4 M_5 M_6 M_7 B_6 B_7;$$

$$A_1 F_1 \rightarrow A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_{17} A_{18} A_{19} A_{20} A_{21} A_{22} A_{23} A_{24} A_{25} A_{26} A_{27} A_{28} A_{29} A_{30} A_{31} A_{32} A_{33} A_{34} A_{35} A_{36} A_{37} A_{38} A_{39} B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} B_{13} B_{14} B_{15} B_{16} B_{17} B_{18} B_{19} B_{20} B_{21} B_{22} B_{23} B_{24} B_{25} B_{26} B_{27} B_{28} B_{29} B_{30} B_{31} N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8 N_9 N_{10} N_{11} N_{12} N_{13} N_{14} N_{15} N_{16} M_1 M_2 M_3 M_4 M_5 M_6 M_7;$$

$$A_1 C_1 C_5 \rightarrow A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_{17} A_{18} A_{19} A_{20} A_{21} A_{22} A_{23} A_{24}$$

$$A_{25}A_{26}A_{27}A_{28}A_{29}A_{30}A_{31}A_{32}A_{33}A_{34}A_{35}A_{36}A_{37}A_{38}A_{39}B_1B_2B_3B_4B_5B_6B_7B_8B_9B_{10}B_{11}B_{12}B_{13} \\ B_{14}B_{15}B_{16}B_{17}B_{18}B_{19}B_{20}B_{21}B_{22}B_{23}B_{24}B_{25}B_{26}B_{27}B_{28}B_{29}B_{30}B_{31}C_2C_3N_1N_2N_3N_4N_5N_6N_7N_8N_9 \\ N_{10}N_{11}N_{12}N_{13}N_{14}N_{15}N_{16}M_1M_2M_3M_4M_5M_6M_7;$$

$$A_8A_9B_8C_1 \rightarrow M_1M_2M_3M_4M_5M_6M_7)$$

$$(G - H) = (A_1 \rightarrow A_2B_1B_2B_3B_4B_8B_9B_{11}B_{12}B_{13}B_{14}B_{15}B_{16}B_{17}B_{18}B_{19}B_{20}B_{21}B_{22}B_{23}B_{24} \\ B_{25}B_{26}B_{27}B_{28}B_{29}B_{30}B_{31}N_1;$$

$$A_2 \rightarrow A_7A_8A_9A_{10}A_{11}A_{12}A_{13}A_{14}A_{15}A_{16}A_{17}A_{18}A_{19}A_{20}A_{21}A_{22}A_{23}A_{24}A_{25}A_{26}A_{27}A_{28}A_{29} \\ A_{30}A_{31}A_{32}A_{33}A_{34}A_{35}A_{36}A_{37}A_{38};$$

$$A_7 \rightarrow B_6;$$

$$A_{13} \rightarrow B_7;$$

$$A_{10}A_{11}A_{12}A_{14} \rightarrow A_3A_4A_5A_6A_{39};$$

$$A_1F_1 \rightarrow F_2F_3F_4;$$

$$A_1C_1C_5 \rightarrow C_4C_6;$$

$$A_8A_9B_8C_1 \rightarrow C_2C_3;$$

$$B_8B_9 \rightarrow B_{10};$$

$$B_1B_2B_3B_4 \rightarrow B_5;$$

$$A_8 \rightarrow M_1M_2M_3M_4M_5M_6M_7;$$

$$N_1 \rightarrow N_2N_3N_4N_5N_6N_7N_8N_9N_{10}N_{11}N_{12}N_{13}N_{14}N_{15}N_{16})$$

Проверим, содержатся ли ФЗ из Н в замыкании $(G-H)^+$:

$$A_1^+ = A_1A_2A_3A_4A_5A_6A_7A_8A_9A_{10}A_{11}A_{12}A_{13}A_{14}A_{15}A_{16}A_{17}A_{18}A_{19}A_{20}A_{21}A_{22}A_{23}A_{24} \\ A_{25}A_{26}A_{27}A_{28}A_{29}A_{30}A_{31}A_{32}A_{33}A_{34}A_{35}A_{36}A_{37}A_{38}A_{39}B_1B_2B_3B_4B_5B_6B_7B_8B_9B_{10}B_{11}B_{12}B_{13} \\ B_{14}B_{15}B_{16}B_{17}B_{18}B_{19}B_{20}B_{21}B_{22}B_{23}B_{24}B_{25}B_{26}B_{27}B_{28}B_{29}B_{30}B_{31}N_1N_2N_3N_4N_5N_6N_7N_8N_9N_{10} \\ N_{11}N_{12}N_{13}N_{14}N_{15}N_{16}M_1M_2M_3M_4M_5M_6M_7$$

$$A_1 \rightarrow A_3A_4A_5A_6A_7A_8A_9A_{10}A_{11}A_{12}A_{13}A_{14}A_{15}A_{16}A_{17}A_{18}A_{19}A_{20}A_{21}A_{22}A_{23}A_{24}A_{25} \\ A_{26}A_{27}A_{28}A_{29}A_{30}A_{31}A_{32}A_{33}A_{34}A_{35}A_{36}A_{37}A_{38}A_{39}B_5B_6B_7B_{10}M_1M_2M_3M_4M_5M_6M_7N_1N_2N_3 \\ N_4N_5N_6N_7N_8N_9N_{10}N_{11}N_{12}N_{13}N_{14}N_{15}N_{16} \in (G - H)^+$$

$$A_2^+ = A_2A_3A_4A_5A_6A_7A_8A_9A_{10}A_{11}A_{12}A_{13}A_{14}A_{15}A_{16}A_{17}A_{18}A_{19}A_{20}A_{21}A_{22}A_{23}A_{24} \\ A_{25}A_{26}A_{27}A_{28}A_{29}A_{30}A_{31}A_{32}A_{33}A_{34}A_{35}A_{36}A_{37}A_{38}A_{39}M_1M_2M_3M_4M_5M_6M_7B_6B_7$$

$$A_2 \rightarrow A_3A_4A_5A_6A_{39}M_1M_2M_3M_4M_5M_6M_7B_6B_7 \in (G - H)^+$$

$$A_1F_1^+ = A_1A_2A_3A_4A_5A_6A_7A_8A_9A_{10}A_{11}A_{12}A_{13}A_{14}A_{15}A_{16}A_{17}A_{18}A_{19}A_{20}A_{21}A_{22}A_{23} \\ A_{24}A_{25}A_{26}A_{27}A_{28}A_{29}A_{30}A_{31}A_{32}A_{33}A_{34}A_{35}A_{36}A_{37}A_{38}A_{39}B_1B_2B_3B_4B_5B_6B_7B_8B_9B_{10}B_{11}B_{12} \\ B_{13}B_{14}B_{15}B_{16}B_{17}B_{18}B_{19}B_{20}B_{21}B_{22}B_{23}B_{24}B_{25}B_{26}B_{27}B_{28}B_{29}B_{30}B_{31}N_1N_2N_3N_4N_5N_6N_7N_8$$

$$N_9 N_{10} N_{11} N_{12} N_{13} N_{14} N_{15} N_{16} M_1 M_2 M_3 M_4 M_5 M_6 M_7 F_1 F_2 F_3 F_4$$

$$\begin{aligned} A_1 F_1 &\rightarrow A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_{17} A_{18} A_{19} A_{20} A_{21} A_{22} A_{23} A_{24} \\ A_{25} A_{26} A_{27} A_{28} A_{29} A_{30} A_{31} A_{32} A_{33} A_{34} A_{35} A_{36} A_{37} A_{38} A_{39} B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} B_{13} \\ B_{14} B_{15} B_{16} B_{17} B_{18} B_{19} B_{20} B_{21} B_{22} B_{23} B_{24} B_{25} B_{26} B_{27} B_{28} B_{29} B_{30} B_{31} N_1 N_2 N_3 N_4 N_5 N_6 N_7 N_8 N_9 N_{10} \\ N_{11} N_{12} N_{13} N_{14} N_{15} N_{16} M_1 M_2 M_3 M_4 M_5 M_6 M_7 &\in (G - H)^+ \end{aligned}$$

$$\begin{aligned} (A_1 C_1 C_5)^+ &\rightarrow A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_{17} A_{18} A_{19} A_{20} A_{21} A_{22} A_{23} \\ A_{24} A_{25} A_{26} A_{27} A_{28} A_{29} A_{30} A_{31} A_{32} A_{33} A_{34} A_{35} A_{36} A_{37} A_{38} A_{39} B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} \\ B_{13} B_{14} B_{15} B_{16} B_{17} B_{18} B_{19} B_{20} B_{21} B_{22} B_{23} B_{24} B_{25} B_{26} B_{27} B_{28} B_{29} B_{30} B_{31} C_1 C_2 C_3 C_4 C_5 C_6 N_1 N_2 N_3 N_4 \\ N_5 N_6 N_7 N_8 N_9 N_{10} N_{11} N_{12} N_{13} N_{14} N_{15} N_{16} M_1 M_2 M_3 M_4 M_5 M_6 M_7 \end{aligned}$$

$$\begin{aligned} A_1 C_1 C_5 &\rightarrow A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9 A_{10} A_{11} A_{12} A_{13} A_{14} A_{15} A_{16} A_{17} A_{18} A_{19} A_{20} A_{21} A_{22} A_{23} \\ A_{24} A_{25} A_{26} A_{27} A_{28} A_{29} A_{30} A_{31} A_{32} A_{33} A_{34} A_{35} A_{36} A_{37} A_{38} A_{39} B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 B_9 B_{10} B_{11} B_{12} \\ B_{13} B_{14} B_{15} B_{16} B_{17} B_{18} B_{19} B_{20} B_{21} B_{22} B_{23} B_{24} B_{25} B_{26} B_{27} B_{28} B_{29} B_{30} B_{31} C_2 C_3 N_1 N_2 N_3 N_4 N_5 N_6 N_7 \\ N_8 N_9 N_{10} N_{11} N_{12} N_{13} N_{14} N_{15} N_{16} M_1 M_2 M_3 M_4 M_5 M_6 M_7 &\in (G - H)^+ \end{aligned}$$

$$(A_8 A_9 B_8 C_1)^+ \rightarrow A_8 A_9 B_8 C_1 M_1 M_2 M_3 M_4 M_5 M_6 M_7$$

$$A_8 A_9 B_8 C_1 \rightarrow M_1 M_2 M_3 M_4 M_5 M_6 M_7 \in (G - H)^+$$

Таким образом, все ФЗ из Н содержатся в $(G-H)^+$. Следовательно схема обладает свойством сохранения ФЗ.

11. Проверка на свойство соединения без потерь: проверка представлена в таблице Б.1.

Таким образом, схема обладает свойством соединения без потерь.

Таблица Б.1 – Проверка свойства соединения без потерь

	A 1	A 2	A 3	A 4	A 5	A 6	A 7	A 8	A 9	A 10	A 11	A 12	A 13	A 14	A 15	A 16	A 17	A 18	A 19	A 20	A 21	A 22	A 23	A 24	A 25	A 26	A 27	A 28	A 29	A 30	A 31	A 32	A 33	A 34	
R ₁	a	a	a ₅	a ₅	a ₅	a ₅	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂		
R ₂		a	a ₅	a ₅	a ₅	a ₅	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	
R ₃			a	a	a	a				a	a	a		a																					
R ₄	a	a ₁	a ₅	a ₅	a ₅	a ₅	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂		
R ₅	a	a ₁	a ₅	a ₅	a ₅	a ₅	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂		
R ₆								a	a																										
R ₇																																			
R ₈																																			
R ₉	a	a ₁	a ₅	a ₅	a ₅	a ₅	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂	a ₂		
R ₁₀																																			
R ₁₁																																			
R ₁₂								a																											
R ₁₃																																			

Продолжение таблицы Б.1

	A 35	A 36	A 37	A 38	A 39	B 1	B 2	B 3	B 4	B 5	B 6	B 7	B 8	B 9	B 10	B 11	B 12	B 13	B 14	B 15	B 16	B 17	B 18	B 19	B 20	B 21	B 22	B 23	B 24	B 25	B 26	B 27	B 28	B 29
R ₁	a ₂	a ₂	a ₂	a ₂	a ₅	a	a	a	a	a ₉	a ₃	a ₄	a	a	a ₈	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	
R ₂	a	a	a	a	a ₅						a ₃	a ₄																						
R ₃					a																													
R ₄	a ₂	a ₂	a ₂	a ₂	a ₅	a ₁	a ₁	a ₁	a ₁	a ₉	a ₃	a ₄	a ₁	a ₁	a ₈	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁		
R ₅	a ₂	a ₂	a ₂	a ₂	a ₅	a ₁	a ₁	a ₁	a ₁	a ₉	a ₃	a ₄	a ₁	a ₁	a ₈	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁		
R ₆													a																					
R ₇													a	a	a																			
R ₈						a	a	a	a	a																								
R ₉	a ₂	a ₂	a ₂	a ₂	a ₅	a ₁	a ₁	a ₁	a ₁	a ₉	a ₃	a ₄	a ₁	a ₁	a ₈	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁	a ₁		
R ₁₀											a																							
R ₁₁												a																						
R ₁₂																																		
R ₁₃																																		

Продолжение таблицы Б.1

	B 30	B 31	C 1	C 2	C 3	C 4	C 5	C 6	F 1	F 2	F 3	F 4	M 1	M 2	M 3	M 4	M 5	M 6	M 7	N 1	N 2	N 3	N 4	N 5	N 6	N 7	N 8	N 9	N 10	N 11	N 12	N 13	N 14	N 15	N 16
R ₁	a	a											a ₁₀	a ₁₀	a ₁₀	a ₁₀	a ₁₀	a ₁₀	a ₁₀	a	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	
R ₂																																			
R ₃																																			
R ₄	a ₁	a ₁							a	a	a	a	a ₁₀	a ₁₀	a ₁₀	a ₁₀	a ₁₀	a ₁₀	a ₁₀	a ₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	
R ₅	a ₁	a ₁	a	a ₇	a ₇	a ₇	a	a					a ₁₀	a ₁₀	a ₁₀	a ₁₀	a ₁₀	a ₁₀	a ₁₀	a ₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	
R ₆			a	a	a								a ₁₀	a ₁₀	a ₁₀	a ₁₀	a ₁₀	a ₁₀	a ₁₀																
R ₇																																			
R ₈																																			
R ₉	a ₁	a ₁	a	a ₇	a ₇	a ₇	a	a ₇	a	a ₆	a ₆	a ₆	a ₁₀	a ₁₀	a ₁₀	a ₁₀	a ₁₀	a ₁₀	a ₁₀	a ₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	a ₁₁	
R ₁₀																																			
R ₁₁																																			
R ₁₂													a	a	a	a	a	a	a																
R ₁₃																				a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	

ПРИЛОЖЕНИЕ В. Расчет минимальной пропускной способности

Если первичные и вторичные БД расположены на разных узлах сети, то процесс репликации ведётся путём обмена журналами изменений. Таким образом для постоянной поддержки актуальности записей вторичной БД требуется определенная величина пропускной способности сети.

Для расчёта пропускной способности сети используется формула (В.1).

$$Band = \left(\frac{RGPS}{0,7} \right) * 8 / 1000000 \quad (В.1)$$

Band – величина требуемой пропускной способности, в Мб/с.

RGPS – метрика, представляющая собой объем изменений генерируемых в первичной БД в секунду, в байтах.

Журнал операций представляет собой буфер в SGA (разделяемая область памяти, хранящая данные и управляющую информацию отдельного экземпляра БД Oracle), хранящий информацию об изменениях, сделанных в БД. Данный журнал содержит информацию, необходимую для реконструкции и отмены изменений, сделанных в БД (операции insert, update, delete, create, alter, drop) [37,40].

Изменения, производящиеся в ходе обработки одной заявки, указаны в таблице В.1.

Таблица В.1 – Изменения, производимые в БД

Таблица	Число записей	Средний объем одной записи, байт
APPLICATION	1	910
APP_RESULT	1	432
BKI_RESULT	1	400
RESPONSE	1	1840
RULE_CHECK	130	91

Таким образом по одной заявке добавляется:

$$910 + 432 + 400 + 1840 + 91 * 130 = 15412 \text{ байт.}$$

При параллельной обработке 30 заявок и минимальном времени обработки заявки 2 секунды получаем оценку $RGPS = 15412 * \frac{30}{2} = 231180$ байт/с

Таким образом требуется среднее значение пропускной способности сети:

$$Band = \frac{231180}{0.7} * \frac{8}{1000000} = 2,642 \text{ Мб/с.}$$

ПРИЛОЖЕНИЕ Г. Листинг модуля проверки взаимодействия с БД

```
%macro mCheckTable(mvTableName) ;
    %DWL_INIT;

    %local mvLib mvTable;
    %let mvLib=%sysfunc(lowercase(%scan(&mvTableName, 1, .)));
    %let mvTable=%sysfunc(lowercase(%scan(&mvTableName, 2, .)));

    %local mvPath mvTestFile;
    %let mvPath=%sysfunc(pathname(&mvLib));
    %let mvTestFile=&mvPath/&mvTable..Z;

    %if %sysfunc(fileexist(&mvTestFile)) %then %do;
        data _null_;
            _error_=system("rm &mvTestFile");
        run;
        %IF %DWL_NOT_OK %THEN %GOTO EXIT;
    %end;

%EXIT:
    %DWL_TERM;
%mend;

%macro mCheckSelectTable(mvTable) ;
    %DWL_INIT;

    /*TODO: добавить проверку, что libname смотрит на одну папку
и что что SAS-BASE*/
    /*Библиотека и название таблицы*/
    %local mvLib mvPureTable;
    %if %length(%scan(&mvTable, 2, .))=0 %then %do;
        %let mvLib=WORK;
        %let mvPureTable=&mvTable;
    %end;
    %else %do;
        %let mvLib=%scan(&mvTable, 1, .);
        %let mvPureTable=%scan(&mvTable, 2, .);
    %end;
    %let mvPureTable=%sysfunc(lowercase(&mvPureTable));

    /*Путь к библиотеке*/
    %local mvLibPath; %let mvLibPath=%sysfunc(pathname(&mvLib));

    %local mvFilePath mvTestdFilePath;
    %let mvFilePath=&mvLibPath./&mvPureTable..sas7bdat;
    %let mvTestdFilePath=&mvLibPath./&mvPureTable..Z;

    %put mvFilePath=&mvFilePath;
    %put mvTestdFilePath=&mvTestdFilePath;

    /*Проверка существования таблицы*/
```

```

%if ^%sysfunc(exist(&mvTable)) %then %do;
    %DWL_RAISE_ERROR("Table &mvTable does not exist");
    %if %DWL_NOT_OK %then %goto EXIT;
%end;

/*Проверка индексов для таблицы*/
proc datasets lib=&mvLib nolist;
    modify &mvPureTable; index delete _all_;
quit;
%if %DWL_NOT_OK %then %goto EXIT;

data _null_;
    _error_ = system("compress -c &mvFilePath >
&mvTestdFilePath");
    if _error_ ^=0 then abort;
run;
%if %DWL_NOT_OK %then %goto EXIT;

/*Проверка существования таблицы*/
%if ^%sysfunc(fileexist(&mvTestdFilePath)) %then %do;
    %DWL_RAISE_ERROR("&mvTable was not compressed");
    %if %DWL_NOT_OK %then %goto EXIT;
%end;

%EXIT:
    %DWL_TERM;
%mend;

%macro mCheckModifyTable(mvTable);
    %DWL_INIT;

    /*TODO: добавить проверку, что libname смотрит на одну папку
и что что SAS-BASE*/
    /*Библиотека и название таблицы*/
    %local mvLib mvPureTable;
    %if %length(%scan(&mvTable, 2, .))=0 %then %do;
        %let mvLib=WORK;
        %let mvPureTable=&mvTable;
    %end;
    %else %do;
        %let mvLib=%scan(&mvTable, 1, .);
        %let mvPureTable=%scan(&mvTable, 2, .);
    %end;
    %let mvPureTable=%sysfunc(lowercase(&mvPureTable));

    /*Путь к библиотеке*/
    %local mvLibPath; %let mvLibPath=%sysfunc(pathname(&mvLib));

    %local mvFilePath mvTestdFilePath;
    %let mvFilePath=&mvLibPath./&mvPureTable..sas7bdat;
    %let mvTestdFilePath=&mvLibPath./&mvPureTable..Z;

```

```

%put mvFilePath=&mvFilePath;
%put mvTestdFilePath=&mvTestdFilePath;

/*Проверка существования файла*/
%if ^%sysfunc(fileexist(&mvTestdFilePath)) %then %do;
    %DWL_RAISE_ERROR("file &mvTestdFilePath does not
exist");
    %if %DWL_NOT_OK %then %goto EXIT;
%end;

/*Разархивация*/
data _null_;
    _error_ = system("uncompress -c &mvTestdFilePath >
&mvFilePath");
    if _error_ ^=0 then abort;
run;
%if %DWL_NOT_OK %then %goto EXIT;

/*Проверка существования таблицы*/
%if ^%sysfunc(exist(&mvTable)) %then %do;
    %DWL_RAISE_ERROR("&mvTable was not uncompressed");
    %if %DWL_NOT_OK %then %goto EXIT;
%end;

%EXIT:
    %DWL_TERM;
%mend;

```

ПРИЛОЖЕНИЕ Д. Листинг модуля проверки правил и стоп-листов

```
%macro mLookupCase (mvDimension, mvKeyField, mvDataFields,
mvHashObjName=, mvAbortOnFailure=1, mvMultiData=0, mvCheckOnly=0);

    /*Если название объекта не задано*/
    %if %length(&mvHashObjName)=0 %then %do;
        /*Если узел содержит опции*/
        %local mvEndTableIndex;
        %let mvEndTableIndex=%index(&mvDimension,%STR( ) );
        %if &mvEndTableIndex=0 %then %let
mvHashObjName=__h_&mvDimension;
        %else %let mvHashObjName=__h_%substr(&mvDimension, 1,
&mvEndTableIndex-1);
    %end;

    %local mvDefinedHashFlag ;
    %let mvDefinedHashFlag = __defined_&mvHashObjName ;

    retain &mvDefinedHashFlag 0; drop &mvDefinedHashFlag ;
    if &mvDefinedHashFlag ^= 1 then do;
        &mvDefinedHashFlag = 1;

        %if %length (&mvDataFields)>0 %then %do;
            call
missing(%sysfunc(translate(%sysfunc(compbl(&mvDataFields)), ", ", "
")));
        %end;

        declare hash &mvHashObjName(dataset: "&mvDimension" %if
&mvMultiData=1 %then %do; %STR(, multidata:'y') %end;);
        __rc = &mvHashObjName..defineKey(
            %local mvField i ;
            %let i = 1;
            %let mvField = %scan(&mvKeyField, &i, %str( ));
            %do %while (&mvField ^= %str());
                %if &i ^= 1 %then %do; , %end;
                "&mvField"

                %let i = %eval(&i + 1);
                %let mvField = %scan(&mvKeyField, &i, %str( ));
            %end;
        );

        %let i = 1;
        %let mvField = %scan(&mvDataFields, &i, %str( ));
        %do %while (&mvField ^= %str());
            __rc = &mvHashObjName..defineData("&mvField");
            %let i = %eval(&i + 1);
            %let mvField = %scan(&mvDataFields, &i, %str( ));
        %end;

        __rc = &mvHashObjName..defineDone();
    end;

    %if &mvCheckOnly %then %do;
        __rc = &mvHashObjName..check();
    %end;
```

```

%end;
%else %do;
    __rc = &mvHashObjName..find();
%end;

%if &mvAbortOnFailure %then %do;
    if __rc then do;
        putlog "ERROR: lookup into &mvDimension failed for key
value " &mvKeyField;
        abort;
    end;
%end;
drop __rc;
%mend mLookupCase;

%macro mLookupTestNode ;
    option mprint mlogic symbolgen;
    data test_fact;
        key1 = 1; key2 = 1; output;
        key1 = 2; key2 = 4; output;
        key1 = 2; key2 = 2; output;
        key1 = 1; key2 = 1; output;
    run;

    data test_dim;
        retain data0 "data0";
        key1 = 1; key2 = 1; data1 = 1; data2 = "XXX"; output;
        key1 = 2; key2 = 1; data1 = 4; data2 = "YYY"; output;
        key1 = 2; key2 = 2; data1 = 4; data2 = "???"; output;
        key1 = 3; key2 = 1; data1 = 9; data2 = "ZZZ"; output;
    run;

    data _null_;
        attrib
            data0 length=$5 label="data0"
            data1 length=$8 label="data1"
            data2 length=$3 label="data2"
        ;
        set test_fact;

        if _n_ >= 2 then do;
            %mLookupHash(mvDimension = test_dim,
                        mvKeyField = key1 key2,
                        mvDataFields = data0 data1,
                        mvHashObjName = h1);
            %mLookupHash(mvDimension = test_dim,
                        mvKeyField = key1 key2,
                        mvDataFields = data2,
                        mvHashObjName = h2);
        end;
        put _all_;
    run;
%mend mLookupTestNode;

```

```

/*Выдает атрибуты для всех данных измерения (без исторических и без
sas_code)*/
%macro mDimAttrib(mvBA, mvDimAlias, mvAttrHist, mvField, mvExtCond);

    %local mvAttrib;

    %local mvCond;
    %IF (%LENGTH(&mvDimAlias)>0) %THEN %let
mvCond=%STR(OWNER_ALIAS="&mvDimAlias");

    %IF %LENGTH(&mvField)>0 %THEN %DO;
        %if %LENGTH(&mvCond)>0 %then %let mvCond=&mvCond AND;
        %let mvField="%sysfunc(Compbl(&mvField))";
        %let mvField=%sysfunc(TRANWRD(&mvField,%str( ),%str(", ")));
        %let mvCond=&mvCond %STR(%UPCASE(NAME) IN
(%UPCASE(&mvField)));
    %end;

    %IF %LENGTH(&mvExtCond)>0 %THEN %DO;
        %if %LENGTH(&mvCond)>0 %then %let mvCond=&mvCond AND;
        %let mvCond=&mvCond %STR((&mvExtCond));
    %end;

    %IF %LENGTH(&mvCond)>0 %THEN %LET mvCond=%STR((WHERE=(&mvCond)));
    %PUT mvCond=&mvCond;

    %LET d_id = %SYSFUNC(OPEN(META.&mvBA. &mvCond, i));

    %IF (&d_id=0) %THEN %DO;
        %put %sysfunc(sysmsg());
        %DWL_RAISE_ERROR("error opening table META.&mvBA");
    %END;

    %LOCAL vType vHist vName vLen vFmt vLab;
    %LET vType = %SYSFUNC(VarNum(&d_id,TYPE));
    %LET vHist = %SYSFUNC(VarNum(&d_id,HIST));
    %LET vName = %SYSFUNC(VarNum(&d_id,NAME));
    %LET vLen = %SYSFUNC(VarNum(&d_id,LEN));
    %LET vFmt = %SYSFUNC(VarNum(&d_id,FMT));
    %LET vLab = %SYSFUNC(VarNum(&d_id,LAB));

    %LOCAL mvType mvHist mvName mvLen mvFmt mvLab mvHFormat;

    %LET rc=%SYSFUNC(FETCH(&d_id));
    %DO %WHILE (&rc =0);

        %LET mvType=%SYSFUNC(GETVARC(&d_id, &vType));
        %LET mvHist=%SYSFUNC(GETVARC(&d_id, &vHist));
        %LET mvName=%SYSFUNC(GETVARC(&d_id, &vName));
        %LET mvLen=%SYSFUNC(GETVARC(&d_id, &vLen));
        %LET mvFmt=%SYSFUNC(GETVARC(&d_id, &vFmt));
        %LET mvLab=%SYSFUNC(GETVARC(&d_id, &vLab));

        %LET mvAttrib=&mvAttrib
                                %STR(&mvName    LENGTH=&mvLen    FORMAT=&mvFmt
LABEL="&mvLab");

```

```

        %IF (&mvHist=1 & &mvAttrHist=1) %THEN %DO;
            %LET mvHFormat=%SYSFUNC (COMPRESS (%SUBSTR (&mvFmt,
1,%EVAL (%LENGTH (&mvFmt)-2)) %STR (H.)));
            %LET mvAttrib=&mvAttrib
                %STR (&mvName._H LENGTH=&mvLen
FORMAT=&mvHFormat LABEL="&mvLab");
            %END;

        %LET rc=%SYSFUNC (FETCH (&d_id));
        %END;

    %LET rc = %SYSFUNC (CLOSE (&d_id));
    ATTRIB &mvAttrib;

%mend;

/*Выдает атрибуты в формате SQL */
%macro mDimAttribSQL(mvBA, mvDimAlias, mvAttrHist, mvField,
mvExtCond);

    %local mvAttribSQL;

    %local mvCond;
    %IF (%LENGTH (&mvDimAlias)>0) %THEN %let
mvCond=%STR (OWNER_ALIAS="&mvDimAlias");

    %IF %LENGTH (&mvField)>0 %THEN %DO;
        %if %LENGTH (&mvCond)>0 %then %let mvCond=&mvCond AND;
        %let mvField="%sysfunc (COMPBL (&mvField))";
        %let mvField=%sysfunc (TRANWRD (&mvField,%str ( ),%str ("","")));
        %let mvCond=&mvCond %STR (%UPCASE (NAME) IN
(%UPCASE (&mvField)));
        %end;

    %IF %LENGTH (&mvExtCond)>0 %THEN %DO;
        %if %LENGTH (&mvCond)>0 %then %let mvCond=&mvCond AND;
        %let mvCond=&mvCond %STR ((&mvExtCond));
        %end;

    %IF %LENGTH (&mvCond)>0 %THEN %LET mvCond=%STR ((WHERE= (&mvCond)));
    %PUT mvCond=&mvCond;

    %LET d_id = %SYSFUNC (OPEN (META.&mvBA. &mvCond, i));

    %IF (&d_id=0) %THEN %DO;
        %put %sysfunc (sysmsg());
        %DWL_RAISE_ERROR ("error opening table META.&mvBA");
        %END;

    %LOCAL vType vHist vName vLen vFmt vLab;
    %LET vType = %SYSFUNC (VarNum (&d_id,TYPE));
    %LET vHist = %SYSFUNC (VarNum (&d_id,HIST));
    %LET vName = %SYSFUNC (VarNum (&d_id,NAME));
    %LET vNameRef = %SYSFUNC (VarNum (&d_id,NAMEREF));
    %LET vLen = %SYSFUNC (VarNum (&d_id,LEN));
    %LET vFmt = %SYSFUNC (VarNum (&d_id,FMT));

```



```

%LET vLab = %SYSFUNC (VarNum(&d_id,LAB));
%LET vOwn = %SYSFUNC (VarNum(&d_id,OWNER_ALIAS));

%LOCAL mvType mvHist mvName mvLen mvFmt mvLab mvHFormat;

%LET rc=%SYSFUNC (FETCH(&d_id));
%DO %WHILE (&rc =0);

    %LET mvType=%SYSFUNC (GETVARC (&d_id, &vType));
    %LET mvHist=%SYSFUNC (GETVARC (&d_id, &vHist));
    %LET mvName=%SYSFUNC (GETVARC (&d_id, &vName));
    %LET mvNameRef=%SYSFUNC (GETVARC (&d_id, &vNameRef));
    %LET mvLen=%SYSFUNC (compress(%SYSFUNC (GETVARC (&d_id,
&vLen)), '$'));
    %LET mvFmt=%SYSFUNC (GETVARC (&d_id, &vFmt));
    %LET mvLab=%SYSFUNC (GETVARC (&d_id, &vLab));
    %LET mvOwn=%SYSFUNC (GETVARC (&d_id, &vOwn));

    %IF %LENGTH(&mvAttribSQL) %THEN %LET
mvAttribSQL=&mvAttribSQL%str(,);

    %if %length(&mvNameRef) %then %let mvNameExt
=&mvOwn..&mvNameRef as &mvName ;
                                %else %let mvNameExt
=&mvOwn..&mvName ;

    %LET mvAttribSQL=&mvAttribSQL
                                %STR(&mvNameExt LENGTH=&mvLen
FORMAT=&mvFmt LABEL="&mvLab");

    %IF (&mvHist=1 & &mvAttrHist=1) %THEN %DO;
        %LET mvHFormat=%SYSFUNC (COMPRESS(%SUBSTR(&mvFmt,
1,%EVAL(%LENGTH(&mvFmt)-2))%STR(H.)));
        %LET mvAttrib=&mvAttrib
                                %STR(&mvOwn..&mvName._H LENGTH=&mvLen
FORMAT=&mvHFormat LABEL="&mvLab");
        %END;

    %LET rc=%SYSFUNC (FETCH(&d_id));
    %END;

%LET rc = %SYSFUNC (CLOSE(&d_id));

&mvAttribSQL

%mend;

%macro mDimGetKeep(mvBA, mvDimAlias, mvKeepHist, mvField, mvExtCond,
mvVarTo, mvRefValues=1);
    %local mvKeep;

    /*TODO: убрать дублирование кода с mDimAttrib*/
    %local mvCond;
    %IF (%LENGTH(&mvDimAlias)>0) %THEN %let
mvCond=%STR(OWNER_ALIAS="&mvDimAlias");

```

```

%IF %LENGTH(&mvField)>0 %THEN %DO;
    %if %LENGTH(&mvCond)>0 %then %let mvCond=&mvCond AND;
    %let mvField="%sysfunc (COMPBL(&mvField))";
    %let mvField=%sysfunc (TRANWRD (&mvField,%str( ),%str(", ")));
    %let mvCond=&mvCond %STR(%UPCASE (NAME) IN
(%UPCASE (&mvField)));
%end;

%IF %LENGTH(&mvExtCond)>0 %THEN %DO;
    %if %LENGTH(&mvCond)>0 %then %let mvCond=&mvCond AND;
    %let mvCond=&mvCond %STR((&mvExtCond));
%end;

%IF %LENGTH(&mvCond)>0 %THEN %LET mvCond=%STR( (WHERE= (&mvCond)) );

%LET d_id = %SYSFUNC (OPEN (META.&mvBA. &mvCond, i));

%IF (&d_id=0) %THEN %DO;
    %DWL_RAISE_ERROR("error opening table META.&mvBA");
%END;

%LOCAL vType vHist vName vNameRef;
%LET vType = %SYSFUNC (VarNum (&d_id,TYPE));
%LET vHist = %SYSFUNC (VarNum (&d_id,HIST));
%LET vName = %SYSFUNC (VarNum (&d_id,NAME));
%LET vNameRef = %SYSFUNC (VarNum (&d_id,NAMEREF));

%LOCAL mvType mvHist mvName mvNameRef;

%LET rc=%SYSFUNC (FETCH (&d_id));
%DO %WHILE (&rc =0);

    %LET mvType=%SYSFUNC (GETVARC (&d_id, &vType));
    %LET mvHist=%SYSFUNC (GETVARC (&d_id, &vHist));
    %LET mvName=%SYSFUNC (GETVARC (&d_id, &vName));
    %LET mvNameRef=%SYSFUNC (GETVARC (&d_id, &vNameRef));

    %IF (%LENGTH (&mvNameRef)>0 and ("&mvType"!="REF") and
"&mvRefValues"="1") %THEN
        %LET mvCur=&mvNameRef;
    %ELSE
        %LET mvCur=&mvName;

    %LET mvKeep=&mvKeep %STR (&mvCur);

    %IF ("&mvHist"="1" & "&mvKeepHist"="1") %THEN %DO;
        %LET mvKeep=&mvKeep %STR (&mvCur._H);
    %END;

    %LET rc=%SYSFUNC (FETCH (&d_id));
%END;

%LET rc = %SYSFUNC (CLOSE (&d_id));

%let &&mvVarTo=&mvKeep;

```

```

%mend;

%macro mDimBy(mvBA, mvDimAlias);

    %local mvKeep;

    %LET d_id =
%SYSFUNC (OPEN(META.&mvBA. (WHERE=(OWNER_ALIAS="&mvDimAlias" AND
TRIM(TYPE) NE "REF")), i));

    %IF (&d_id=0) %THEN %DO;
        %DWL_RAISE_ERROR("error opening table META.&mvBA");
    %END;

    %LOCAL vType vHist vName vNameRef;
    %LET vType = %SYSFUNC(VarNum(&d_id,TYPE));
    %LET vHist = %SYSFUNC(VarNum(&d_id,HIST));
    %LET vName = %SYSFUNC(VarNum(&d_id,NAME));
    %LET vNameRef = %SYSFUNC(VarNum(&d_id,NAMEREF));

    %LOCAL mvType mvHist mvName mvNameRef;

    %LET rc=%SYSFUNC(FETCH(&d_id));
    %DO %WHILE (&rc =0);

        %LET mvType=%SYSFUNC(GETVARC(&d_id, &vType));
        %LET mvHist=%SYSFUNC(GETVARC(&d_id, &vHist));
        %LET mvName=%SYSFUNC(GETVARC(&d_id, &vName));
        %LET mvNameRef=%SYSFUNC(GETVARC(&d_id, &vNameRef));

        %IF (%LENGTH(&mvNameRef)>0 & ("&mvType"!="REF")) %THEN
            %LET mvCur=&mvNameRef;
        %ELSE
            %LET mvCur=&mvName;

        %LET mvKeep=&mvKeep &mvCur;
        %LET mvByLast=&mvCur;

        %IF (&mvHist=1) %THEN %DO;
            %LET mvKeep=&mvKeep &mvCur._H;
            %LET mvByLast=&mvCur._H;
        %END;

        %LET rc=%SYSFUNC(FETCH(&d_id));
    %END;

    %LET rc = %SYSFUNC(CLOSE(&d_id));

    BY &mvKeep;

%mend;

%macro mDimFillHistorical(mvBA, mvDimAlias, mvField);

```

```

%local mvCond;
%if %length(&mvDimAlias)>0 %then %do;
    %let mvCond=%str(&mvCond OWNER_ALIAS="&mvDimAlias" AND);
%end;
%if %length(&mvField)>0 %then %do;
    %let mvCond=%str(&mvCond name=%UPCASE("&mvField") AND );
%end;

%LOCAL mvAttrib mvHEncoding;
%LET d_id = %SYSFUNC (OPEN (META.&mvBA. (WHERE= (&mvCond
COMPRESS (HIST)="1")), i));

%IF (&d_id=0) %THEN %DO;
    %DWL_RAISE_ERROR("error opening table META.&mvBA");
%END;

%LOCAL vType vHist vCube vDetl vName vLen vFmt vLab;
%LET vType = %SYSFUNC (VarNum (&d_id,TYPE));
%LET vHist = %SYSFUNC (VarNum (&d_id,HIST));
%LET vCube = %SYSFUNC (VarNum (&d_id,CUBE));
%LET vDetl = %SYSFUNC (VarNum (&d_id,DETL));
%LET vName = %SYSFUNC (VarNum (&d_id,NAME));
%LET vLen = %SYSFUNC (VarNum (&d_id,LEN));
%LET vFmt = %SYSFUNC (VarNum (&d_id,FMT));
%LET vLab = %SYSFUNC (VarNum (&d_id,LAB));

%LOCAL mvType mvHist mvCube mvDetl mvName mvLen mvFmt mvLab
mvHFormat;

Attrib temp_hist length=$300; drop temp_hist;

%LET rc=%SYSFUNC (FETCH (&d_id));

%DO %WHILE (&rc =0);

    %LET mvType=%SYSFUNC (GETVARC (&d_id, &vType));
    %LET mvHist=%SYSFUNC (GETVARC (&d_id, &vHist));
    %LET mvCube=%SYSFUNC (GETVARC (&d_id, &vCube));
    %LET mvDetl=%SYSFUNC (GETVARC (&d_id, &vDetl));
    %LET mvName=%SYSFUNC (GETVARC (&d_id, &vName));
    %LET mvLen=%SYSFUNC (GETVARC (&d_id, &vLen));
    %LET mvFmt=%SYSFUNC (GETVARC (&d_id, &vFmt));
    %LET mvLab=%SYSFUNC (GETVARC (&d_id, &vLab));

    %LET mvHFormat=%SYSFUNC (COMPRESS (%SUBSTR (&mvFmt,
1,%EVAL (%LENGTH (&mvFmt)-2)) %STR (H.)));

    %LET mvHEncoding=&mvHEncoding %STR( temp_hist=put (&mvName,
&mvFmt) );

    %LET mvHEncoding=&mvHEncoding %STR(
&mvName._H=input (temp_hist,&mvHFormat) );
    %LET mvHEncoding=&mvHEncoding %STR( if (&mvName._H=.) then
do;

if &mvName ne . then

```

```

put "IORC ERROR
FOR &mvName=" &mvName=12.3 &mvName=&mvFmt;
                                _error_=0;
                                end; );

%LET rc=%SYSFUNC(FETCH(&d_id));
%END;

%LET rc = %SYSFUNC(CLOSE(&d_id));

%STR(&mvHEncoding);

%mend;

/*Обновляет измерение новыми данными (с учетом исторических форматов),
формирует урезанное измерение для join,
mvAllODS=1 если не нужно ориентироваться на поле ODS из метаданных при
построении измерения, а брать все поля*/
%macro mDimUpdate(mvBA, mvDimAlias, mvInTable, mvCutDimOut,
mvAllODS=0, mvHistExternal=0);
%DWL_INIT;

/*Имя таблицы измерения*/
%local mvLibEtl mvLibIDP mvTableEtl;
proc sql;
    select table, lib_etl, lib_idp into
        :mvTableEtl,
        :mvLibEtl,
        :mvLibIDP
    from meta.&mvBA._alias
    where alias="&mvDimAlias";
quit;
%IF %DWL_NOT_OK %THEN %GOTO EXIT;
%local mvDimTable mvDimTableIDP;
%LET mvDimTable=%sysfunc(compress(&mvLibETL..&mvTableEtl));
%let mvDimTableIDP=%sysfunc(compress(&mvLibIDP..&mvTableEtl));

/*Образующие поля*/
%LOCAL mvOdsFields;
%LET d_id =
%SYSFUNC(OPEN(META.&mvBA.(WHERE=(TRIM(OWNER_ALIAS)="&mvDimAlias" AND
TRIM(TYPE) ne "REF")), i));

%IF (&d_id=0) %THEN %DO;
    %DWL_RAISE_ERROR("error opening table META.&mvBA");
    %GOTO EXIT;
%END;

%LOCAL vName vHist vOds;
%LET vName = %SYSFUNC(VarNum(&d_id,NAME));
%LET vHist = %SYSFUNC(VarNum(&d_id,HIST));
%LET vOds = %SYSFUNC(VarNum(&d_id,ODS));

%LOCAL mvName mvHist mvOds;
%LET rc=%SYSFUNC(FETCH(&d_id));
%DO %WHILE (&rc =0);

```

```

%LET mvName=%SYSFUNC(GETVARC(&d_id, &vName));
%LET mvHist=%SYSFUNC(GETVARC(&d_id, &vHist));
%LET mvOds=%SYSFUNC(GETVARC(&d_id, &vOds));

%if (&mvOds=1) or (&mvAllODS=1) %THEN %LET
mvOdsFields=&mvOdsFields &mvName;
%if (&mvHistExternal=1 and &mvHist=1) %then %let
mvOdsFields=&mvOdsFields &mvName._h;
%LET rc=%SYSFUNC(FETCH(&d_id));
%END;
%LET rc = %SYSFUNC(CLOSE(&d_id));

%IF %LENGTH(%SCAN(&mvOdsFields, 2))>0 %THEN %LET
mvIndex=%STR(pkey=(&mvOdsFields));
%ELSE %LET mvIndex=&mvOdsFields;

/*Первичное создание измерения*/
%IF ^(%SYSFUNC(EXIST(&mvDimTable))) %THEN %DO;

/*Если это миграция из IDP*/
%if "&DWL_FIRSTMIGRATION"="1" %then %do;
%if %sysfunc(exist(&mvDimTableIDP)) %then %do;
proc copy in =&mvLibIDP out =&mvLibETL;
select &mvTableEtl;
run;
%IF %DWL_NOT_OK %THEN %GOTO EXIT;
%end;
%else %do;
%DWL_RAISE_ERROR("No dimension table &mvDimTableIDP
found");
%IF %DWL_NOT_OK %THEN %GOTO EXIT;
%end;
%end;
%else %if "&DWL_FIRST_LOAD"!="1" %then %do;
%DWL_RAISE_ERROR("No dimension table &mvDimTable");
%IF %DWL_NOT_OK %THEN %GOTO EXIT;
%end;
%else %do;
DATA &mvDimTable (index=(sas_code/unique &mvIndex));
%DimAttrib(mvBA=&mvBA, mvDimAlias=&mvDimAlias,
mvAttrHist=1);
regid=&DWL_REGID;
sas_code=&DWL_REGID;
RUN;
%IF %DWL_NOT_OK %THEN %GOTO EXIT;
%end;
%END;

%LOCAL mvDistTable; %LET mvDistTable=&mvDimAlias._DIST;
PROC SORT DATA=&mvInTable OUT=&mvDistTable NODUPKEY;
BY &mvOdsFields;
RUN;
%IF %DWL_NOT_OK %THEN %GOTO EXIT;

DATA &mvDistTable._H;
%DimAttrib(&mvBA, &mvDimAlias, 1);
SET &mvDistTable;

```

```

        %if ("&mvHistExternal" ^= "1") %then %do;
        %mDimFillHistorical(&mvBA, &mvDimAlias);
        %end;
        %mDimKeep(&mvBA, &mvDimAlias, 1);
RUN;
%IF %DWL_NOT_OK %THEN %GOTO EXIT;

PROC SORT DATA=&mvDistTable._H NODUPKEY;
        %mDimBy(&mvBA, &mvDimAlias);
RUN;
%IF %DWL_NOT_OK %THEN %GOTO EXIT;

/*Обновляем измерение*/
%local mvMaxDimId;%let mvMaxDimId=0;
proc sql;
        select max(sas_code)/1000 format=12. into: mvMaxDimId from
&mvDimTable;
quit;
%IF %DWL_NOT_OK %THEN %GOTO EXIT;

%LOCAL mvByLast;

/*Отрезаем данные с sas_code<0 (мигрированные из старого БОПОС)*/
DATA &mvDimTable._LZ;
        SET &mvDimTable (where=(sas_code<=0));
RUN;
%IF %DWL_NOT_OK %THEN %GOTO EXIT;

/*Сортируем исходное измерение, стирая индексы,
на случай изменения генеральной линии партии*/
PROC SORT DATA=&mvDimTable FORCE;
        %mDimBy(&mvBA, &mvDimAlias);
RUN;
%IF %DWL_NOT_OK %THEN %GOTO EXIT;

/*Обновление данных*/
DATA &mvDimTable._NEW &mvCutDimOut (index=(&mvIndex/unique));

        UPDATE &mvDistTable._H(in=innew)
&mvDimTable(where=(sas_code>0)) ;
        %mDimBy(&mvBA, &mvDimAlias);

        retain _mmm &mvMaxDimId; drop _mmm;
        if sas_code=. then do;
                _mmm+1;
                sas_code=_mmm*1000+&DWL_REGID;
        end;

        drop innew;

/*Если сразу две записи отвечают новой, то вылетает :( */
if (innew=1 and LAST.&mvByLast) then output &mvCutDimOut;
output &mvDimTable._NEW;

RUN;
%IF %DWL_NOT_OK %THEN %GOTO EXIT;

```

```

/*Добавляем данные с sas_code<0 и восстанавливаем сортировку и
индекс*/
PROC APPEND BASE=&mvDimTable._NEW DATA=&mvDimTable._LZ; RUN;
%IF %DWL_NOT_OK %THEN %GOTO EXIT;

PROC SQL; drop table &mvDimTable._LZ; QUIT;
%IF %DWL_NOT_OK %THEN %GOTO EXIT;

PROC SORT DATA=&mvDimTable._NEW;
    %mDimBy(&mvBA, &mvDimAlias);
RUN;
%IF %DWL_NOT_OK %THEN %GOTO EXIT;

PROC SQL;
    create unique index sas_code on &mvDimTable._NEW;
QUIT;
%IF %DWL_NOT_OK %THEN %GOTO EXIT;

%DWL_UNX_RENAME(&mvDimTable._NEW, &mvDimTable.);
%IF %DWL_NOT_OK %THEN %GOTO EXIT;

%EXIT:
    %DWL_TERM;
%mend;

```