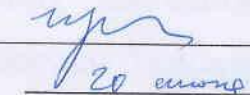


МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК  
Кафедра программного обеспечения

ДОПУЩЕНО К ЗАЩИТЕ В ГЭК  
И ПРОВЕРЕНО НА ОБЪЕМ  
ЗАИМСТВОВАНИЯ  
Заведующий кафедрой  
д.п.н., профессор  
И.Г. Захарова  
  
2016 г.

**МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ**

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ  
АВТОМАТИЗАЦИИ ДЕЯТЕЛЬНОСТИ АДМИНИСТРАТОРА БАЗ  
ДАННЫХ ДЛЯ ФИЛИАЛА ООО «УГМК-СТАЛЬ» «МЗ «ЭЛЕКТРОСТАЛЬ  
ТЮМЕНИ»

02.04.03. Математическое обеспечение и администрирование информационных  
систем

Магистерская программа «Высокопроизводительные вычислительные  
системы»

Выполнил работу  
Студент 2 курса  
очной формы обучения



Дубровин  
Михаил  
Григорьевич

Научный руководитель  
к.пед.н., доцент



Плотоненко  
Юрий  
Анатольевич

Рецензент  
Зав. лабораторией разработки программных комплексов  
контроля добычи нефти УА НАЦРН им. Шпильмана



Галкин  
Вячеслав  
Юрьевич

Тюмень 2016

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ .....	7
§ 1.1 Металлургический завод "Электросталь Тюмени" .....	7
§ 1.2 Информационная структура предприятия .....	11
§ 1.3 Роль администратора баз данных в структуре предприятия .....	17
§ 1.4 Сравнительный анализ систем мониторинга баз данных .....	27
§ 1.5 Функциональные требования к программному обеспечению .....	33
§ 1.6 Нефункциональные требования к программному обеспечению .....	36
ГЛАВА 2. ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ .....	39
§ 2.1 Построение диаграммы описания бизнес – процессов по стандарту IDEF0 .....	40
§ 2.2 Построение диаграммы потоков данных (DFD) .....	48
§ 2.3 Проектирование базы данных .....	50
ГЛАВА 3. РЕАЛИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ .....	55
§ 3.1 Описание инструментальных средств .....	55
§ 3.2 Описание алгоритма работы приложения .....	60
§ 3.3 Описание компонентов программного обеспечения .....	62
§ 3.4 Описание пользовательского интерфейса .....	67
ЗАКЛЮЧЕНИЕ .....	83
СПИСОК СОКРАЩЕНИЙ .....	85
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ .....	86
СПИСОК ТАБЛИЦ И ИЛЛЮСТРАЦИЙ .....	90
ПРИЛОЖЕНИЯ .....	92
Приложение 1. Алгоритм работы приложения .....	92

Приложение 2. Создание нового соединения .....	94
Приложение 3. Взаимодействие с интерфейсом WMI .....	96
Приложение 4. Взаимодействие с Oracle .....	101
Приложение 5. SQL запрос к базе данных Oracle .....	104
Приложение 6. Сохранение отчета в формате Excel .....	105

## **ВВЕДЕНИЕ**

На предприятии «Электросталь Тюмени», работающем в режиме 24/7, бесперебойная и эффективная работа серверов баз данных является одним из наиболее приоритетных задач в обеспечении работоспособности информационной системы предприятия.

Поддержка работоспособности серверов баз данных организации является прямой обязанностью администратора базы данных, входящего в штат отдела автоматизации систем управления технологическими процессами и производством.

Для оперативной локализации проблем необходимо иметь механизм слежения, который бы предупреждал администратора о надвигающихся неисправностях. Такой превентивный подход к задаче обнаружения сводит к минимуму время незапланированных простоев баз данных, поскольку специалист получает возможность заранее спланировать мероприятия по ликвидации неисправностей.

Ежедневные задачи по мониторингу и обслуживанию баз данных занимают значительное рабочее время специалиста. С течением времени, в связи с появлением новых задач, информационная структура предприятия расширяется. Появляются новые сервера баз данных, а также увеличивается нагрузка на текущие БД. Увеличивается время, затраченное на ежедневные задачи мониторинга и администрирования.

Для обеспечения высокой доступности и высокой производительности систем баз данных предприятия, необходимо комплексное средство, позволяющее оценивать состояние работоспособности систем, снижать время отклика на критические ситуации в работе информационной системы, автоматизировать время ежедневного мониторинга баз данных.

Объект исследования работы - автоматизация деятельности администратора баз данных на предприятии.

Предметом исследования работы является разработка программного обеспечения для автоматизации деятельности администратора баз данных.

Целью выпускной квалификационной работы является разработка программного обеспечения для автоматизации работы администратора баз данных Oracle, находящихся под управлением серверных операционных систем Windows, предназначенного для филиала ООО «УГМК – Сталь» МЗ «Электросталь Тюмени».

Для достижения поставленной цели необходимо решить следующие задачи:

- проанализировать работу выбранного предприятия и его информационную структуру;
- описать назначение серверов баз данных и роль администратора баз данных в информационной структуре предприятия;
- проанализировать назначение программного обеспечения, для автоматизации деятельности администратора баз данных, а также существующие программные продукты;
- составить требования к программному обеспечению;
- проанализировать и отобразить бизнес – процессы, связанные с работой АБД;
- спроектировать базу данных программного обеспечения;

- выбрать инструментальные средства для реализации программного обеспечения;
- реализовать программное обеспечение; - ввести приложение в эксплуатацию.

Методы исследования диссертационной работы:

- методы анализа предметной области;
- методы проектирования программного обеспечения;
- методы разработки информационных приложений; - методы администрирования баз данных.

Выпускная квалификационная работа состоит из введения, основной части, включающей три главы, заключения и списка использованных источников.

Во введении обосновывается актуальность работы, описываются объект и предмет исследования, формируется цель и задачи выпускной работы, перечисляются используемые в работе методы исследования, показывается практическая значимость результатов проекта, приведен обзор работы по главам и использованных источников.

В первой главе выпускной квалификационной работы проанализирована структура работы выбранной организации, проанализирована информационная структура предприятия. В частности, рассмотрено назначение сервера базы данных предприятия, а также специалиста, отвечающего за работоспособность этих систем. Изучены задачи и обязанности администратора баз данных, ответственного за работоспособность данного сектора информационной структуры предприятия. Проанализирована необходимость в программном обеспечении для автоматизации деятельности администратора баз данных, а также рассмотрены программные продукты, существующие на рынке программного

обеспечения, созданные для обозначенных целей. В конце первой главы выявлены требования к программному обеспечению.

Во второй главе, на основе анализа предметной области, построены диаграммы IDEF0 и DFD, которые позволяют графически отобразить деятельность администратора баз данных предприятия. Далее, построена ER – диаграмма и физическая модель базы данных программного обеспечения.

В третьей главе описаны инструментальные средства, используемые при разработке приложения. Далее, описаны основные структурные модули приложения. В третьей главе описан алгоритм работы программного обеспечения. В конце третьей главы продемонстрирован пользовательский интерфейс приложения и описание работы с программой.

В заключении представлены результаты проделанной работы.

На основе детального анализа структуры выбранного предприятия, а также изучения источников [3,11,12,27,32] исследован производственный цикл металлургического завода. На основе источников [3,27,32,33,35] и анализа предметной области рассмотрена информационная структура предприятия.

После изучения источников [1,2,9,13,16,18,25,26] проанализированы задачи и обязанности администратора баз данных, а также методологии администрирования баз данных. На основе источников [1,13,16,17,18,21,22,23,33,36] детально изучена структура серверов баз данных Oracle а также особенности их администрирования. Тенденции в области администрирования баз данных Oracle рассмотрены с помощью источников [7,9,10,19,20,27]. Опыт создания автоматизированных систем мониторинга баз данных рассмотрен с помощью источников [7,8,15,19,24,28]. Возможности удаленного взаимодействия с серверными операционными системами Windows с помощью технологии WMI рассмотрены с использованием источников [4,29,31,33].

В процессе изучения источников [5,6,38] проанализированы методологии проектирования информационных систем.

Результаты выпускной квалификационной работы предназначены для внедрения в филиал ООО «УГМК-Сталь» «МЗ «Электросталь Тюмени» в отдел автоматизации систем управления технологическими процессами и производством.

## **ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

### **§ 1.1 Металлургический завод "Электросталь Тюмени"**

Предприятие ООО «Электросталь Тюмени» является мини – заводом, что означает, что оно работает по неполному металлургическому циклу (без коксохимического и доменного производства), используя в качестве исходного сырья лом чёрных металлов.

Строительство металлургического завода по производству проката — результат исполнения договора о сотрудничестве между Уральской горнометаллургической компанией и правительством Тюменской области. Поставку и монтаж основного технологического оборудования осуществляют итальянские фирмы DANIELI, SIC и SIAD. Генеральный подрядчик – ОАО ПО «Монтажник», г. Магнитогорск.[11]

Стоимость проекта завода — около 22 млрд. рублей. Около 1 млрд. в целом на реализацию проекта выделено из бюджетов Федерации и Тюменской области. При строительстве завода использованы новейшие технологии в области металлургии.

Принцип работы предприятия основан на расплаве лома черных металлов. Завод ежегодно утилизирует около 650 тыс. тонн металлолома, которого только в тюменском регионе накоплено более 100 млн. тонн. Треть

сырья составляет чугун из Серова, так как он требуется для варки качественной стали. Остальное сырье — лом из Тюменской области, ХантыМансийского и Ямало-Ненецкого автономных округов. На заводе организовано производство широкой номенклатуры углеродистых, легированных, конструкционных и арматурных марок стали. Продукция предназначена для стройиндустрии, предприятий нефтегазового сектора и машиностроения. В основном, металлопрокат направляется на внутренний рынок — в Тюменскую область, Югру и Ямал.[12]

Проектная мощность завода — 550 тысяч тонн продукции ежегодно. Это арматура и металлические конструкции для жилищного и дорожного строительства. В частности, предприятие занимается выпуском строительной арматуры от 10 до 40 мм и круглого проката диаметром от 10 до 42 мм. [11]

Реквизиты предприятия:

Юридический адрес: область Свердловская, город Верхняя Пышма, проспект Успенский, дом 1

ОГРН: 1147232030219

ИНН: 7203311519

Фактический адрес: г. Тюмень, Старый Тобольский тракт, 1-й км.

Тел. / факс: 8(3452) 532-888; 535-693

Сайт: <http://www.steel.ugmk.com/>

Email: [info@tyumensteel.ru](mailto:info@tyumensteel.ru)

Далее рассмотрим производственный цикл металлургического завода.

Для начала, введем несколько обозначений:



1. ЭСПЦ: Электросталеплавильный цех.
2. ПЦ: Прокатный цех.
3. ДСП: Дуговая сталеплавильная печь.
4. АПК: Агрегат печь – ковш.
5. МНЛЗ: Машина непрерывного литья заготовок.

Цикл производства стали начинается со сбора металлолома в Скрапоразделочном цеху и загрузке его в машины или в вагоны. После того, как металлолом загружен, его перевозят в Шихтовое отделение, где его переносят в специальные металлические бадьи.

Далее, металлолом в бадьях переносится в ЭСПЦ, где загружается в 75 - тонную ДСП. В ДСП металлолом нагревается до высокой температуры и переходит в состояние жидкого металла.

В процессе плавления на ДСП заготовка металла отправляется в лабораторию, где производится его анализ на добавку необходимых материалов для получения определенной марки стали, и данные по анализам отправляются на пульт управления АПК.

Итак, полученный из ДСП металл, переливается в Сталеразливочный ковш. С помощью телеги, Старезаливочный ковш передвигается к агрегату АПК. На этапе АПК производятся необходимые добавки материалов для получения стали, вдувания газов, а также повторный нагрев металла. Образующиеся шлаки сливаются в специальные емкости и металл, оставаясь в сталеразливочном ковше, переносится к МНЛЗ.

На этапе МНЛЗ металл сначала переливается в промежуточный ковш из сталеразливочного и после этого разливается по четырем ручьям. В результате

получаются заготовки стали. Заготовки делятся по заданной длине, отсекаются лишние части заготовки.

Следующим этапом, заготовка либо попадает в адьюстаж для охлаждения и подготовки к отгрузке, либо отправляется в Прокатный цех для подготовки к приданию определенной формы.

В ПЦ заготовка первым делом попадает в Нагревательную печь для придания определенной температуры.

Далее по конвейеру заготовка переходит к прокатному стану, где с помощью волкового оборудования стали придается определенная заданная форма, и происходит её растягивание по длине.

Следующим этапом заготовки охлаждаются и сортируются в определенном порядке.

Последним этапом ПЦ осуществляется резка заготовок, согласно заданной длине, а также их взвешивание. После этого, с помощью магнитных кранов, полученная готовая продукция попадает в зону отгрузки, откуда она отправляется с территории завода. (Рис. 1)

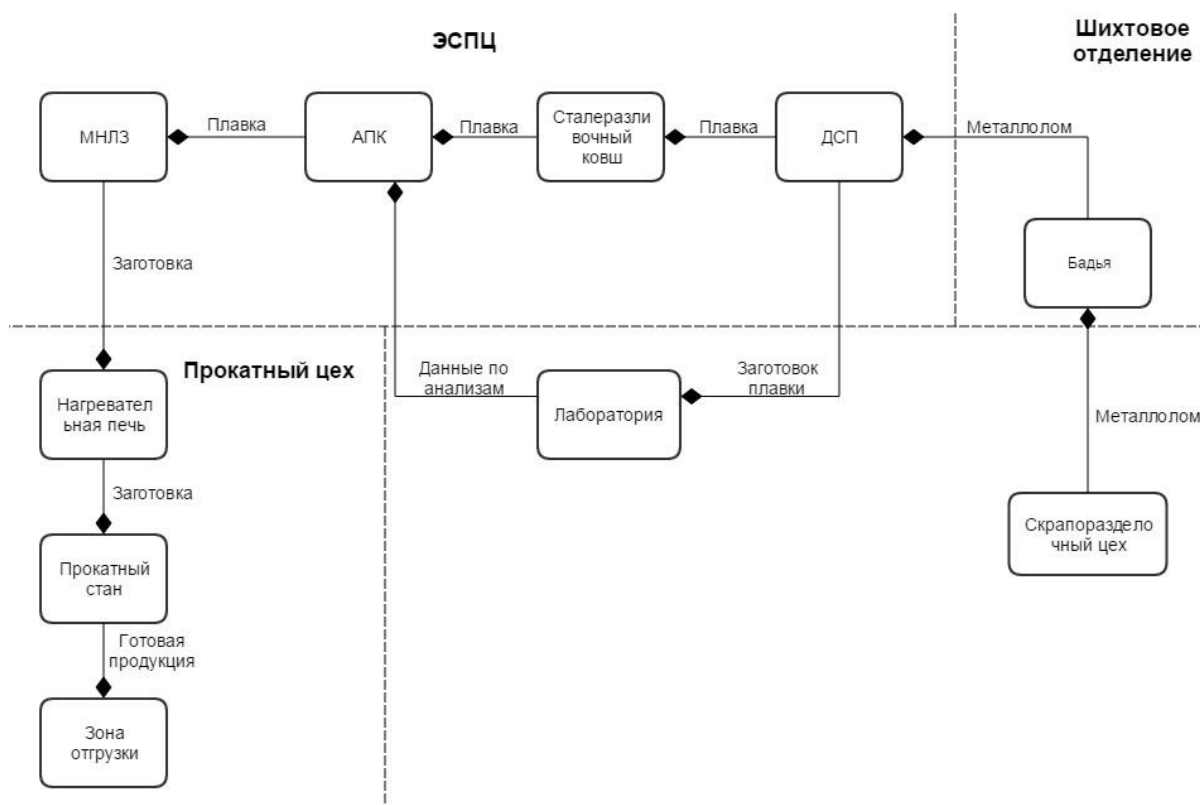


Рис. 1 Схема производственного цикла предприятия

## § 1.2 Информационная структура предприятия

Автоматизированная система управления или АСУ — комплекс аппаратных и программных средств, предназначенный для управления различными процессами в рамках технологического процесса, производства, предприятия.[33] АСУТП – это автоматизированная система управления технологическим процессом. Технологическим процессом металлургического завода является процесс выплавки определенной стали.

Специалисты по комплексной автоматизации предприятий придерживаются 5-уровневой информационной структуры предприятия.[33] (Рис. 2)

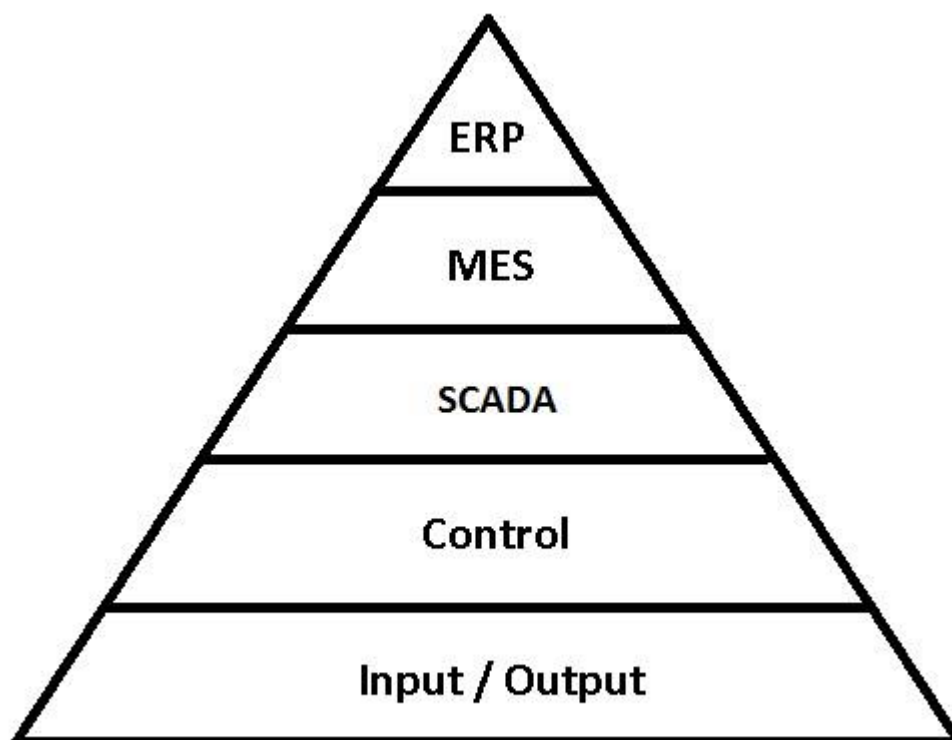


Рис. 2 Пятиуровневая информационная структура предприятия

В основе всей структуры является технологический объект. В качестве технологического объекта металлургического завода выступает технологический агрегат, с использованием которого производится процесс выплавки стали.

Далее представлены элементы информационной системы:

- ERP (Enterprise Resource Planning) – система планирования ресурсов производства;
- MES (Manufacture Execution System) – система исполнительного производства;
- SCADA (Supervisory Control and Data Acquisition) – система диспетчерского управления;
- Control – система локального управления;
- Input/Output – система ввода/вывода.

Уровень **Input/Output** – непосредственно взаимодействует с технологическим объектом. Его задачи:

1. Сбор информации с объекта. Измерение величин параметров объекта с помощью датчиков.
2. Результаты измерения этих величин передаются на следующий уровень Control.

Уровень **Control**. Основная задача – непосредственное управление технологическим объектом, т.е. изменение управляющих параметров для достижения желаемых выходных величин. [35]

Аппаратные средства АСУТП - программируемые логические контроллеры – ПЛК (PLC). ПЛК представляет собой специализированную ЭВМ или промышленный компьютер, способную надежно работать в цеховых условиях, т.е. при наличии помех от электрооборудования, нестабильных напряжениях в питающих сетях, изменении температур в широком диапазоне, вибрации, запыленности и т.п. Программное обеспечение этого уровня позволяет организовывать с помощью ПЛК регуляторы, каждый из которых воспринимает информацию с датчика физической величины, вычисляет управляющее воздействие и передаёт исполнительным механизмам соответствующий сигнал, который превращается в положение регулирующего органа. [35]

Уровень **SCADA** – уровень диспетчерского управления. Получает информацию о технологическом процессе с нижних уровней информационной структуры. Основная задача уровня SCADA – диспетчерское управление технологическим процессом.[35]

SCADA система передает информацию на более высокие уровни информационной структуры предприятия.

Уровень **MES** – уровень исполнения производства. Этот уровень получает информацию от SCADA системы. Уровень SCADA позволяет управлять отдельным технологическим аппаратом или отдельной технологической операцией. На уровне MES собирается информация с многих SCADA - систем. В этой информации не содержится сведений о локальных параметрах технологических процессов, а присутствует результат усреднения. На основе этой информации на уровне MES рассчитываются материальные и энергетические балансы, характеризующие всю технологическую схему. [35]

Наиболее высокий уровень – **ERP** – решение стратегических задач развития предприятий. Анализ результатов работы предприятия за относительно долгосрочный период (квартал, полугодие, год). Определяются критические точки технологии, а также технические и экономические решения, которые позволяют оптимизировать работу предприятия в целом.[35]

Целью информационной структуры предприятия является информационное обеспечение технологического процесса, прямое технологическое управление.

В качестве данных информационной технологии используют различные виды исходной информации. На уровне АСУТП такими данными являются характеристики работы технологического объекта.

Ответственным за состояние уровней Control, SCADA, MES и за использование информации, полученных с этих уровней является отдел автоматизации систем управления технологическими процессами и производством (АСУТПиП).

Целью деятельности отдела АСУТПиП является повышение уровня автоматизации производственных и технологических процессов, повышение

оперативности взаимодействия подразделений предприятия за счет внедрения современных информационных систем и систем связи, и, как следствие, улучшение технико - экономических показателей работы предприятия, повышение качества конкурентоспособности готовой продукции.

Для реализации этих целей отдел стремится достичь следующих показателей:

- своевременное сервисное обслуживание систем автоматизации и связи;
- своевременная техническая и информационная поддержка пользователей;
- обеспечение подразделений завода подключением к информационным ресурсам предприятия.

Далее продемонстрирована схема архитектуры информационных систем предприятия «Электросталь Тюмени». (Рис. 3)

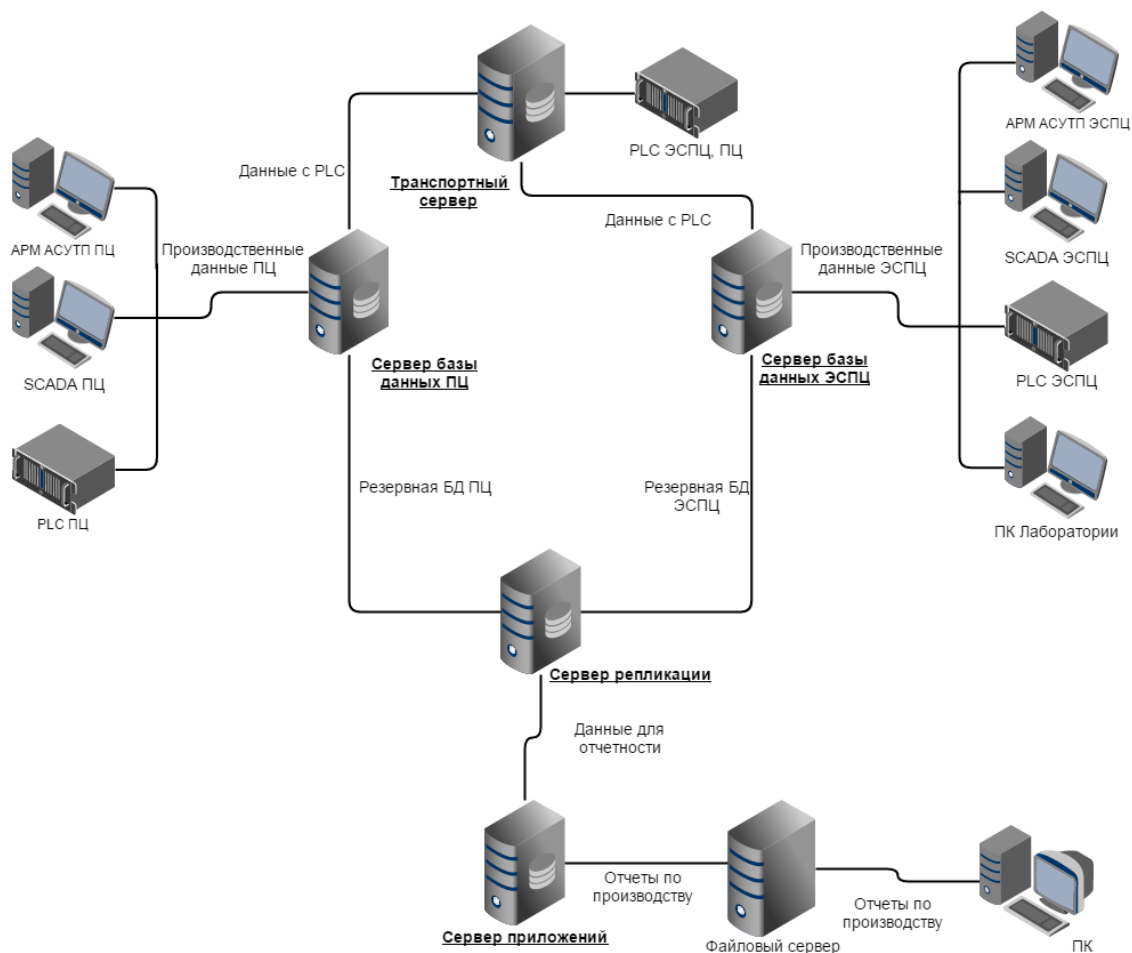


Рис. 3 Схема информационной системы предприятия

Производственный процесс предприятия подразумевает постоянный обмен информации с PLC агрегатов, SCADA систем и АРМов систем управления технологическими процессами с производственными базами данных соответствующих цехов. Поэтому, во время производственного цикла, важнейшим требованием к работе систем автоматизации является доступность баз данных, а также серверов баз данных в режиме 24/7.

Для получения дополнительной информации с PLC реализован «Транспортный сервер», который передает актуальную информацию на сервера баз данных цехов.



Данные с баз данных ЭСПЦ и ПЦ реплицируются в режиме реального времени на «Сервер репликации» для поддержки актуальной копии баз данных, участвующих в производственном процессе.

С помощью технологии репликации производственные данные, необходимые для формирования отчетности, отправляются на Сервер приложений.

С помощью программного обеспечения, отчеты по производству отправляются для хранения и последующего использования на файловый сервер. Конечные пользователи системы, руководители цехов, ведущие специалисты отделов, а также сотрудники отдела технологического контроля, получают информацию о производстве с помощью отчетов.

Работоспособность транспортного сервера, серверов баз данных ЭСПЦ и ПЦ, сервера репликации и сервера приложений, является зоной ответственности администратора баз данных организации.

### **§ 1.3 Роль администратора баз данных в структуре предприятия**

Согласно профессиональному стандарту по коду 06.011 от 17 сентября 2014 г., основной обязанностью, характеризующую работу администратора баз данных является поддержание эффективной работы баз данных, обеспечивающих функционирование информационных систем в организации. [2]

На предприятии «Электросталь Тюмени», работающем в режиме 24/7, бесперебойная работа серверов баз данных является одним из наиболее приоритетных задач в обеспечении работоспособности информационной системы предприятия.

Поддержка работоспособности серверов баз данных организации является прямой обязанностью администратора базы данных, входящего в штат отдела автоматизации систем управления технологическими процессами и производством.

Обязанности администратора баз данных индивидуальны для каждого предприятия и специфичны в зависимости от системы управления баз данных. В данной организации в качестве системы управления базой данных используется программный продукт компании Oracle – Oracle Database.

СУБД Oracle Database (далее просто Oracle) обеспечивает эффективные и действенные решения для основных средств баз данных. Oracle поддерживает самые большие базы данных — потенциальным размером до сотен гигабайт. Oracle поддерживает большое число пользователей, одновременно выполняющих разнообразные приложения, которые оперируют одними и теми же данными. Он минимизирует соперничество за данные и гарантирует согласованность данных. Oracle поддерживает все описанные выше возможности, при этом сохраняя высокую степень суммарной

производительности системы. Во многих случаях ПО Oracle должно работать 24 часа в сутки, не имея периодов разгрузки, ограничивающих пропускную способность базы данных. Нормальные системные операции — такие, как откат базы данных, а также частичные сбои компьютерной системы, — не прерывают работу с базой данных. Oracle может выборочно управлять доступностью данных, как на уровне базы данных, так и на более низких уровнях. Чтобы извлечь максимум преимуществ из имеющейся компьютерной системы или сети, Oracle позволяет разделять работу между сервером базы данных и прикладными программами клиентов. Программное обеспечение Oracle совместимо с промышленными стандартами включая большинство стандартных операционных систем — таких, как семейство Microsoft Windows NT, а также различные версии Linux. Приложения, разрабатываемые для Oracle, могут использоваться в любой операционной системе с минимумом модификаций или вообще без таковых. Oracle удовлетворяет промышленно принятым стандартам по языку доступа к данным, операционным системам, интерфейсам с пользователем и сетевым протоколам. Oracle является классической реляционной СУБД, использующей механизм клиент-сервер.

[21]

СУБД Oracle использует язык SQL — простой, мощный язык доступа к базе данных, который является стандартным для реляционных СУБД. SQL, реализованный корпорацией Oracle для Oracle, на 100% согласуется со стандартом ANSI/ISO языка SQL. Кроме того, в Oracle реализовано расширение стандартных функций SQL, называемое PL/SQL. Поддержка механизмов ODBC, BDE и JDBC позволяет осуществлять миграцию между различными программными и аппаратными платформами с минимальными затратами на модификацию программного обеспечения. Все вышперечисленное позволяет использовать СУБД Oracle для автоматизации практически любых процессов, где идет работа с большим объемом данных

— начиная от простейших АРМ небольшого предприятия и заканчивая системами управления ядерными реакторами и ускорителями частиц. [21]

Далее представлены обязанности администратора баз данных текущего предприятия, основываясь на должностной инструкции и поставленных задачах.

### **1. Установка и обновление программного обеспечения Oracle и прикладных средств.**

Обязанность заключается, в умении администратора устанавливать и обновлять программное обеспечение Oracle. При этом, администратор должен точно представлять, для чего служит тот или иной устанавливаемый продукт Oracle. Знать редакции и состав компонентов устанавливаемого программного обеспечения Oracle. Располагать информацией об особенностях установки программного обеспечения под различные платформы. Следить за обновлениями (patch) программного обеспечения Oracle и осуществлять их установку.

### **2. Распределение внешней памяти системы и планирование дальнейших потребностей в ней.**

В этой обязанности администратор Oracle должен представлять, как и для чего Oracle использует внешнюю (дисковую) память. Быть знакомым с файловыми системами платформ, на которых она расположена. Иметь понятие об устройствах ASM (Automatic Storage Management или Автоматическое управление хранением файлов БД) и RAW устройствах (сырые). Знать, что такое RAID и разбираться в его конфигурировании. Уметь оптимально, с точки зрения производительности, размещать файлы Oracle во внешней памяти. Владеть навыками настройки ASM. Просчитывать потребности Oracle во внешней памяти на ближайшее будущее. [9]

### **3. Обеспечение работоспособности операционной системы и прикладного программного обеспечения.**

Администратору баз данных необходимо отслеживать работоспособность операционной системы. Для этого нужно знать ограничения в потреблении памяти процессов Oracle для выбранной операционной системы, а также проверять системные журналы ОС на наличие системных ошибок. Что касается прикладного программного обеспечения, администратору необходимо следить за работоспособностью служб и процессов, взаимодействующих с базой данных и участвующих в производственном процессе предприятия.

### **4. Создание первичных структур хранения базы данных (табличных пространств).**

Администратор должен создавать и конфигурировать табличные пространства во внешней памяти. Первичные структуры хранения обычно планируются заранее и добавляются в процессе создания базы данных. Иногда могут создаваться так же и дополнительные структуры. Обычно это связано с серьёзными изменениями приложений или конфигурации оборудования. Особенностью обязанности является то, что надо правильно распределить файлы при создании табличного пространства. Так как табличные пространства являются контейнером для первичных объектов, правильное распределение файлов пространства во внешней памяти имеет одно из решающих значений для дальнейшей производительности системы.[9]

### **5. Создание первичных объектов (таблиц, представлений, индексов).**

Администратор должен создавать первичные объекты Oracle. Это одна из первых обязанностей администратора. Обычно, в рабочей базе все первичные объекты приложения создаются сразу. Но может потребоваться и

дополнительное создание первичных объектов по мере дополнения или изменения приложения.

## **6. Регистрация пользователей и обеспечение безопасности системы.**

Обязанность включает в себя работы по созданию и настройке учётных записей пользователей Oracle, а также охватывает круг действий по проведению в жизнь политики безопасности базы данных.

Администратор создаёт учётные записи пользователей. Выдаёт им пароли. В зависимости от устройства системы заносит данные о пользователях во внешние справочники, например список операторов (связь данных о пользователе и его учётной записи). Здесь же при необходимости пользователю задаётся профиль или он включается в определённую ресурсную группу. В эту обязанность так же включены действия, осуществляемые для обеспечения защиты базы данных в целом.

Физическая безопасность. Администратор должен проводить мероприятия по защите сервера базы данных от физического доступа посторонними лицами. То есть, к примеру, не допускать в серверную лишних людей, не оставлять открытой консоль сервера базы данных, запираться на ключ шкафы и устройства.

Программная безопасность. Администратор должен предпринять меры по защите системы от взлома через установленное программное обеспечение. Понятно, что лишний код, это ещё одна лишняя лазейка в систему, поэтому не стоит ставить на сервер лишние программные продукты и опции, которые никогда не будут использоваться. Для защиты от взлома Oracle постоянно выпускает так называемые патчи безопасности. Их установка так же входит в обязанность.

Безопасность экземпляра. Администратор должен настроить экземпляр базы данных для обеспечения максимальной его безопасности. Некорректные

значения инициализационных параметров экземпляра могут привести к его несанкционированному доступу или к его падению.

Сетевая безопасность. Администратор должен настроить сетевые службы Oracle для предотвращения несанкционированного доступа к ним. Неправильно задав параметры листнера можно открыть удалённый доступ к его управлению. [9]

### **7. Обеспечение соответствия лицензионному соглашению с корпорацией Oracle.**

Администратор должен следить за соответствием аппаратного и программного обеспечения сервера баз данных, лицензионному соглашению, заключённому с корпорацией Oracle.

### **8. Управление и мониторинг доступа пользователей к базе данных.**

Администратор должен разрешать или запрещать доступ пользователей к объектам базы данных, в соответствии с принятой политикой безопасности, а также осуществлять аудит несанкционированного доступа пользователей к этим объектам.

Кроме предоставления доступа к объектам базы данных, администратор должен уметь управлять и самими учётными записями пользователей: блокировать их, включать в различные профили и ресурсные группы, назначать табличные пространства и текущие схемы. В общем, уметь гибко настроить доступ пользователя к самой СУБД, а не к объектам базы данных. Естественно, кроме управления доступом пользователей, надо следить и за тем, что эти пользователи делают в базе данных. Для этого в Oracle имеются специальные встроенные средства аудита.

### **9. Мониторинг и оптимизация производительности базы данных.**

Администратор должен наблюдать за производительностью системы, и в случае её ухудшения принять меры для её настойки.

База данных – «живой организм». Меняются пользовательские и системные данные. Меняется статистика и планы запросов. Накапливаются ошибки. Изменяется используемая память и увеличивается в размерах сама база. Всё это, в конце концов, может привести к ухудшению производительности системы. Заметить это ухудшение как можно раньше, это одна из главных задач администратора. Для этого требуется постоянно производить мониторинг системы по основным показателям производительности. Если эти значения изменились в худшую сторону, администратор должен принять меры к углублённому анализу и на его основе приступить к настройке базы данных. Но это то, что касается базы данных в целом. [9]

#### **10. Планирование процедур создания резервных копий и восстановления базы данных.**

Администратор должен создать процедуры действий, по которым будут осуществляться резервное копирование и восстановление базы данных.

Эта обязанность просто сводится к составлению инструкций по резервному копированию и восстановлению. В инструкциях описывается последовательность выполняемых команд. Чем подробнее, тем лучше, особенно, что касается восстановления. Такая инструкция является самым низшим документом и предназначена в основном только для администратора.

#### **11. Резервное копирование и восстановление базы данных.**

Администратор должен создавать резервные копии базы данных в соответствии с политикой резервного копирования и восстановления, а также восстанавливать из резервных копий всю или часть базы данных в случаях её повреждения.



Обычно в документе политики должны быть чётко прописаны инструкции как проводить резервное копирование базы данных. Администратор должен просто следовать им в соответствии с расписанием. Что же касается восстановления, тут немного сложнее, так как не всегда возможно составить подробную инструкцию по восстановлению базы данных. В основном здесь роль играет опыт и знания администратора.

## **12. Связь с корпорацией Oracle для получения технической помощи.**

Администратор в случаях невозможности самому решить возникшую проблему, должен обратиться в службу технической поддержки Oracle. [9]

Чем оперативнее администратор локализует проблему, тем скорее он ее решит. Лучший для этого способ - иметь соответствующий механизм слежения, который бы предупреждал администратора о надвигающихся неисправностях. Большинство неполадок легче исправить до того, как они приобретут глобальный характер. Такой превентивный подход к задаче обнаружения сводит к минимуму время незапланированных простоев баз данных, поскольку специалист получает возможность заранее спланировать мероприятия по ликвидации неисправностей.

С течением времени, в связи с появлением новых задач, информационная структура предприятия расширяется. Появляются новые сервера баз данных, а также увеличивается нагрузка на текущие БД. Время, затраченное на ежедневные задачи мониторинга и администрирования, увеличивается пропорционально увеличению количества БД. В связи с этим, необходимо автоматизированное средство, которое позволит сократить время на решение повседневных задач АБД, вне зависимости от роста нагрузки на текущие БД и кол-ва баз данных, находящихся в зоне ответственности АБД.

Эмпирическим путем, на основе изучения источников литературы и предметной области, получено, что работоспособность базы данных характеризуют следующие показатели сервера Oracle:

- % попадания в буферный кэш;
- % сортировок из памяти;
- % мягких разборов sql;
- % попадания в словарный кэш;
- % попадания в библиотечный кэш;
- % свободного места в разделяемом пуле;
- % используемых процессов;
- % используемых сессий;
- количество переключений журналов redo log;
- размер памяти, потребляемый процессом;
- размер разделяемой памяти, потребованный для выполнения sql запроса;
- кол-во обращений к буферу, потребованное для выполнения sql запроса;
- кол-во чтений с диска, потребованное для выполнения sql запроса;
- % заполненности табличного пространства;

Также на работоспособность сервера Oracle влияют следующие параметры операционной системы:

- % загрузки центрального процессора;
- % загрузки физической памяти;
  
- % загрузки виртуальной памяти;
  
- % заполненности логического диска;
  
- % нагрузки процесса ОС на центральный процессор;
  
- кол-во потребляемой памяти процесса ОС.

Итак, для обеспечения высокой доступности и высокой производительности систем баз данных предприятия, необходимо комплексное средство, позволяющее видеть и оценивать состояние работоспособности, а также снижать время отклика на критические ситуации в работе информационной системы, автоматизировать процесс мониторинга.

Далее, рассмотрены самые популярные и эффективные программные продукты, направленные на мониторинг систем баз данных Oracle.

#### **§ 1.4 Сравнительный анализ систем мониторинга баз данных**

После проведения анализа рынка программного обеспечения, направленного на автоматизацию деятельности администрирования баз данных Oracle, выявлены наиболее популярные и эффективные комплексные системы, позволяющие проводить комплексный мониторинг серверов баз данных.

Далее представлено описание этих систем:

##### **1. Oracle Enterprise Manager**

Разработчик: Oracle Corporation, США.

Сайт: <http://www.oracle.com/ru/products/enterprise-manager/>

Функциональность: OEM обладает огромным функционалом для управления и мониторинга баз данных, возможности Oracle Enterprise Manager охватывают наиболее полноценный список задач администрирования БД. Enterprise Manager также позволяет проводить администрирование нескольких баз данных из одной центральной консоли управления. В Enterprise Manager реализована возможность исполнения команд как экземпляра базы данных так и операционной системы.

Стоимость: Стандартная комплектация программного обеспечения бесплатна для администрирования Oracle. Но для полноценного мониторинга баз данных, предупреждения возможных проблем и обеспечения высокой производительности и доступности необходима покупка дополнительных модулей, таких как Diagnostics Pack, Tuning Pack, Database Lifecycle Management Pack и других. Стоимость каждого дополнительного модуля для одного сервера начинается от 3750 долларов и может достигать нескольких десятков тысяч долларов, в зависимости от количества процессоров и ядер сервера. [30]

## 2. Toad DBA Suite for Oracle

Разработчик: Dell Software, США.

Сайт: <http://software.dell.com/products/toad-for-oracle/>

Функциональность: Решение Toad DBA Suite for Oracle представляет собой большой набор средств администрирования баз данных Oracle, позволяющих повышать эффективность работы благодаря автоматизации

обслуживания, обеспечению оптимальной производительности и устранению рисков, связанных с изменениями в базе данных. TOAD for Oracle позволяет управлять большим количеством баз данных за меньшее время благодаря его возможностям работы со многими экземплярами баз данных.

Стоимость: Стоимость одного продукта Toad DBA Suite for Oracle составляет около 6000\$.

### 3. Spotlight on Oracle

Разработчик: Dell Software, США.

Сайт: <http://software.dell.com/products/spotlight-on-oracle/>

Функциональность: Программное обеспечение Spotlight on Oracle – это инструмент визуальной диагностики в режиме реального времени, которая позволяет быстро устранять узкие места в среде Oracle и кластерах Oracle RAC. Решение Spotlight on Oracle помогает быстрее находить и исправлять ошибки в производительности, уменьшать простои и снижать расходы на ИТинфраструктуру в организации.

Стоимость: Стоимость лицензии на использование Spotlight on Oracle начинается с 1200\$ за мониторинг одного экземпляра Oracle. Возможность мониторинга операционной системы начинается с 400\$ за процессор сервера.

### 4. Lab128

Разработчик: 128 Consulting LLC, США

Сайт: [http://www.lab128.com/lab128\\_product.html](http://www.lab128.com/lab128_product.html)

Функциональность: Lab128 - это передовой инструментарий абсолютно необходимый в ситуации, когда производительность баз данных Oracle требует неотложного внимания. Lab128 использовал и внедрил способ записи истории активных сессий (Active Session History - ASH) для Oracle версии 9i,

еще до того как была выпущена версия 10g с встроенной ASH. Lab128 использует методику анализа производительности путем частых выборок из v\$sqlstats/v\$sql, и эта методика просто незаменима, обеспечивая новые уникальные возможности в диагностике. Lab128 был первым, где использовалось множество других методик, которых пока нет в других продуктах. Разработка Lab128 была мотивированна проблемами в реальной практике и стремлением обеспечить максимально возможную производительность.

Стоимость: Стоимость продукта Lab128 начинается с 599\$ за поддержку взаимодействия с двумя экземплярами, до 999\$ с неограниченным количеством экземпляров баз данных. Цена указана за одно лицо, использующее программное обеспечение.

Для объективной оценки подобных систем, введем критерии, на основании которых будет производиться оценка данного типа программного обеспечения.

Далее описаны выбранные критерии:

Мониторинг базы данных подразумевает подробный анализ работоспособности компонентов экземпляра базы данных;

Мониторинг операционной системы является важным компонентом для администрирования сервера базы данных, включающим возможность мониторинга системных журналов ОС, а также статуса служб и процессов операционной системы Windows;

Мониторинг прикладного оборудования, такого как жестких дисков, процессора, памяти;

Возможность удаленного мониторинга позволяет администратору баз данных отслеживать состояние нескольких серверов из одного рабочего места;

Регистрация инцидентов позволяет сохранять информацию об отклонениях системы от состояния нормальной работы;

Возможность оповещений администратора базы данных о критическом состоянии системы;

Поддержка кроссплатформенности ОС и БД подразумевает поддержку СУБД Oracle версий 10g – 11g и ОС Windows Server 2003 – 2012, используемых в структуре предприятия;

Хранение истории исследуемых параметров позволяет диагностировать источник возможных проблем;

Сбор статистики в активном режиме позволяет собирать данные по состоянию работоспособности серверов базы данных без необходимости нахождения администратора базы данных на рабочем месте;

Отсутствие необходимости добавления дополнительных схем, таблиц или модулей для функционирования программного обеспечения; Простота установки и поддержки программного обеспечения;

Стоимость лицензии на использование программного обеспечения.

На основании выбранных критериев, составим таблицу для сравнительной оценки возможностей каждого из программных продуктов.

Таблица 1  
Сравните

льный анализ ПО для мониторинга БД

<b>Наименование ПО / Критерий</b>	Enterprise Manager	Toad for Oracle	Spotlight	Lab128
Мониторинг базы данных	Да	Да	Да	Да
Мониторинг операционной системы	Да(доп.)	Нет	Да	Нет
Мониторинг серверного оборудования	Да(доп.)	Нет	Нет	Нет
Удаленный мониторинг	Да	Да	Да	Да
Регистрация инцидентов	Да	Да	Да	Нет
Централизованный мониторинг	Нет	Нет	Нет	Нет
Оповещения об инцидентах	Да	Нет	Да	Да
Кроссплатформенность исследуемых объектов	Нет	Да	Да	Да
Хранение истории	Да	Да	Да	Да
Активный мониторинг	Да	Да	Да	Да
Отсутствие необходимости добавления доп. модулей	Нет	Да	Нет	Да
Простота использования	Да	Да	Да	Да
Стоимость	От 3750\$	От 6000\$	От 1600\$	От 600\$

Итак, на основе проведенного анализа можно сделать вывод, что данные программные продукты не отвечают всем предъявленным требованиям, поэтому принято решение о разработке собственного программного обеспечения, направленного на комплексный мониторинг серверов баз данных. Далее, представлены функциональные требования к программному обеспечению, выделенные на основе проведенного анализа.



## § 1.5 Функциональные требования к программному обеспечению

Программное обеспечение предназначено для комплексного мониторинга серверов баз данных предприятия ООО «Электросталь Тюмени».

Прежде чем, производить мониторинг сервера базы данных, необходим функционал для создания подключения с удаленным сервером, по которому существует необходимость в исследовании его работоспособности. Исходя из этого, у программного обеспечения должна быть возможность создания нового соединения с удаленным сервером, после чего соединение возможно перевести в режим активности, и проводить комплексный мониторинг сервера базы данных.

Для удобства проведения мониторинга, программное обеспечение должно быть разделено на три раздела.

Первая часть программного обеспечения предназначается для сбора параметров экземпляра и базы данных Oracle, таких как:

- информация по базе данных, экземпляру и параметрам базе данных;
- данные по наполненности табличных пространств и файлов данных;
- сведения по компонентам системной глобальной и процессной памяти;
- информация по сессиям;
- сведения по выполненным SQL запросам;
- самые длинные события ожидания и метрики производительности экземпляра;

- журнал переключений файлов повтора.

Вторая часть приложения служит для сбора параметров по операционной системе и аппаратному обеспечению, таких как:

- информация по загрузке центрального процессора и нагрузке на физическую и виртуальную память;
- сведения по наполненности логических дисков;
- данные об ошибках из журнала событий операционной системы;
- информация по запущенным процессам и их потреблению памяти, а также о статусах служб.

На первых двух этапах ПО должен быть реализован сбор показаний с исследуемых параметров с определенной периодичностью. Периодичность обращения к параметрам должна задаваться в разделе конфигурации программного обеспечения. У каждого значения параметра должно быть задано нормальное значение, отклоненное и критическое значение. При выходе значения параметра за пределы нормального значения, должна производиться обработка события – уведомление об отклонения параметра администратору базы данных с помощью отображения события на экране, а также сохранение информации о событии, для проведения дальнейшего анализа.

Третья часть программного обеспечения предназначена для обработки информации, полученной в ходе исследования значений параметров, характеризующих работоспособность сервера базы данных.

Последний подраздел предназначен для формирования отчетности по событиям за определенный период.

У программного обеспечения должен быть реализован раздел конфигураций, в котором возможно изменения настроек программного обеспечения.

Итак, функции программного обеспечения должны включать:

- создание и удаление подключения к удаленному серверу;
- подключение нескольких удаленных серверов в активном режиме;
- мониторинг удаленного сервера в режиме реального времени;
- просмотр визуальной информации о состоянии производительности базы данных сервера;
- просмотр визуальной информации о состоянии работоспособности операционной системы сервера;
- возможность просмотра критических событий в работе сервера;
- оповещение о критическом состоянии в работе сервера администратору баз данных;
- формирование отчетности по критическим событиям;
- конфигурирование настроек программного обеспечения.

Функционал программного обеспечения позволит улучшить значения следующих показателей:

- время простоя предприятия, связанного с некорректной работой производственных БД;

- время, затраченное на восстановление базы данных, в случае сбоя;
- время, затраченное на проведение ежедневного мониторинга БД предприятия;
- кол-во определенных проблем, потенциально опасных для работы системы.

### **§ 1.6 Нефункциональные требования к программному обеспечению**

В процессе анализа требований к программному обеспечению проанализированы и выделены нефункциональные требования согласно ГОСТ 34.602-89.

#### **Требования к структуре и функционированию системы.**

- программное обеспечение не должно требовать и производить установку дополнительных служб, клиентов, схем данных и таблиц для тех серверов, которые будут исследованы с помощью данного программного обеспечения;
- программное обеспечение не должно требовать установки и должно запускаться с помощью исполняемого файла.

#### **Требования к численности и квалификации персонала системы и режиму его работы.**

Система не требует выделения дополнительных должностных лиц, для её обслуживания и предназначена для следующих сотрудников организации:

- Администратор баз данных / Отдел по автоматизированным системам управления технологическим процессом и производством / Бюро систем управления производством;

- Ведущий инженер - программист / Отдел  
по

автоматизированным системам управления технологическим процессом и производством / Бюро систем управления производством.

Для управления программным обеспечением нет необходимости повышения квалификации; указанный персонал должен работать в соответствии с рабочим графиком, определенным организацией.

### **Требования к эргономике и технической эстетике.**

Взаимодействие пользователей с прикладным программным обеспечением, входящим в состав системы должно осуществляться посредством визуального графического интерфейса (GUI). Интерфейс системы должен быть понятным и удобным, не должен быть перегружен графическими элементами и должен обеспечивать быстрое отображение экранных форм. Навигационные элементы должны быть выполнены в удобной для пользователя форме.

Все надписи экранных форм, а также сообщения, выдаваемые пользователю (кроме системных сообщений) должны быть на русском языке.

Система должна обеспечивать корректную обработку аварийных ситуаций, вызванных неверными действиями пользователей, неверным форматом или недопустимыми значениями входных данных. В указанных случаях система должна выдавать пользователю соответствующие сообщения, после чего возвращаться в рабочее состояние.

### **Требования к защите информации от несанкционированного доступа.**

При осуществлении доступа приложению подразумевается строгая персонализация, отсутствие групповых учетных записей, а также не допускается использование учетных записей другого пользователя. Доступ

предоставляется только для заранее определенного списка пользователей, анонимный доступ не разрешается.

Данные, представляющие информационную ценность предприятия, должны быть представлены в зашифрованном виде.

### **Требования к информационному обеспечению системы.**

Для хранения данных, собираемых программным обеспечением, а также для сохранения конфигураций программного обеспечения необходимо создать отдельную базу данных.

### **Требования к программному обеспечению системы.**

В качестве операционных систем, для функционирования программного обеспечения могут быть использованы: - Windows XP SP1+;

- Windows 7.

Программное обеспечение должно поддерживать мониторинг следующих версий СУБД Oracle:

- Oracle 9.2.0.4 (и выше);

- Oracle 10g (R1 и R2) ;

- Oracle 11g (R1 и R2).

Программное обеспечение должно поддерживать мониторинг следующих версий серверных операционных систем:

- Windows 2003 Server;

- Windows 2008 Server; - Windows 2012 Server.

Требования, не описанные в данном разделе, к программному обеспечению не предъявляются.

## ГЛАВА 2. ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

На сегодняшний день в программной инженерии существуют два основных подхода к разработке ПО ИС, принципиальное различие между которыми обусловлено разными способами декомпозиции систем. Первый подход называют функционально-модульным или структурным. В его основу положен принцип функциональной декомпозиции. Второй, объектноориентированный подход, использует объектную декомпозицию. При этом структура системы описывается в терминах объектов и связей между ними, а поведение системы описывается в терминах обмена сообщениями между объектами. [38]

Сущность структурного подхода к разработке ПО ИС заключается в его декомпозиции (разбиении) на автоматизируемые функции: система разбивается на функциональные подсистемы, которые, в свою очередь, делятся на подфункции, те – на задачи и так далее до конкретных процедур.

Основными средствами, описывающими функциональную структуру системы и отношения между её элементами являются:

- DFD (Data Flow Diagrams) – диаграммы потоков данных;
- SADT (Structured Analysis and Design Technique) – метод структурного анализа и проектирования) – модели и соответствующие функциональные диаграммы;
- ERD (Entity-Relationship Diagrams) – диаграммы сущность – связь.

На стадии формирования требований к ПО SADT-модели и DFD отражают существующую и предлагаемую структуру бизнес-процессов

организации и взаимодействие между ними. С помощью ERD выполняется описание используемых в организации данных на концептуальном уровне, не зависящем от средств реализации базы данных (СУБД).

Перечисленные модели в совокупности дают полное описание ПО ИС независимо от того, является ли система существующей или вновь разрабатываемой. [38]

## **§ 2.1 Построение диаграммы описания бизнес – процессов по стандарту IDEF0**

В качестве субъекта исследования выступает деятельность администратора баз данных на предприятии.

Основной функцией отражающей систему в целом является обеспечение работоспособности базы данных.

При описании бизнес-процесса выбран уровень детализации, при котором подробно описывается функциональная деятельность администрирования баз данных. (Рис. 4) На вход функции подаются:

- Метрики работоспособности базы данных и операционной системы;
- Исходящие и входящие потоки данных;
- Заявки пользователей на модификацию систем баз данных.

В качестве управления на систему в целом действуют:

- Регламентированные требования к работе базы данных;
- Технологический процесс производства стали;
- Политика информационной безопасности предприятия;



- Должностные инструкции персонала, ответственного за обеспечение работоспособности баз данных.

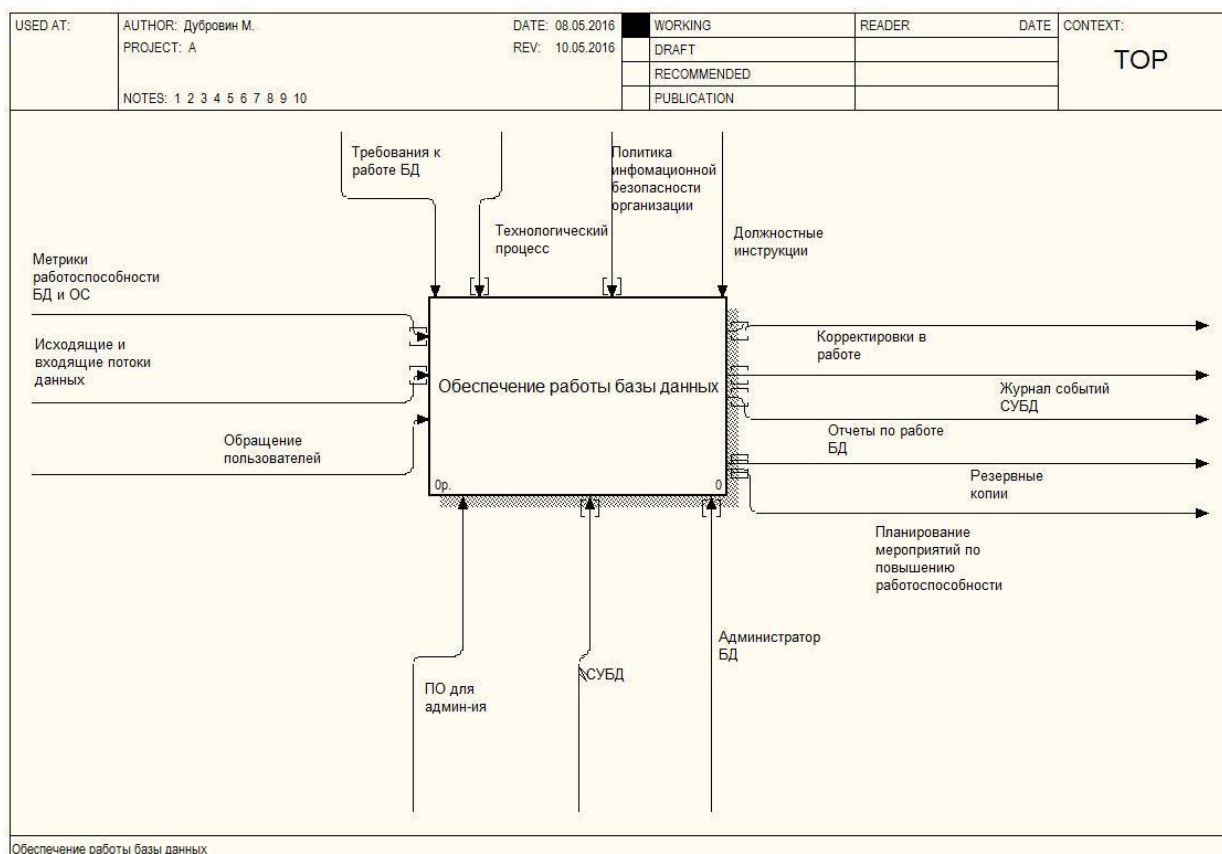
Механизмами обеспечения работоспособности являются:

- Система управления базой данных Oracle;
- Администратор баз данных;
- Программное обеспечение, способствующее администрированию баз данных.

После преобразования входной информации, при воздействии управления с помощью описанных выше механизмов получаем:

- Корректировки в работе СУБД и базы данных;
- Журнал событий СУБД;
- Отчетность по работоспособности базы данных;
- Резервные копии базы данных;
- Планирование мероприятий по повышению

работоспособности БД.



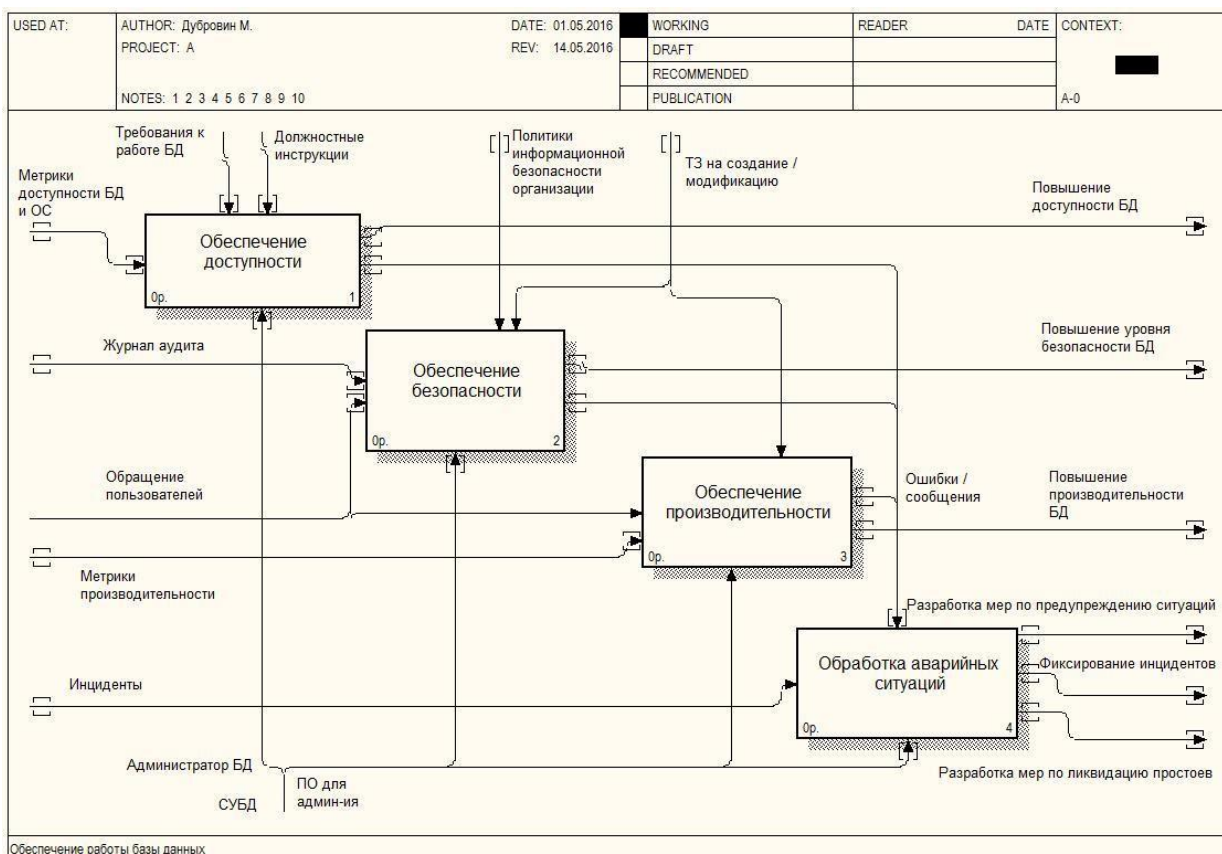
**Рис. 4 IDEF0. Обеспечение работы базы данных**

Далее выполняется детализация контекстной диаграммы на диаграммы декомпозиции. (Рис. 5)

Основная функция разбивается на следующие работы:

- Обеспечение доступности БД;
- Обеспечение безопасности БД;
- Обеспечение производительности БД; - Обработка аварийных ситуаций.

Входом работы «Обеспечение доступности» являются Метрики доступности БД и ОС, а выходом являются реализация повышения доступности БД, а также ошибки и сообщения, в случае отклонения системы от нормальной работы. В качестве управления выступают требования к



**Рис. 5 IDEF0. Декомпозиция функции "Обеспечение работы БД"**

работе БД, а также должностные инструкции администратора баз данных.

Входом для работы «Обеспечение безопасности» являются журнал аудита базы СУБД и заявки пользователей. На выходе работы получается повышение уровня безопасности БД и СУБД, а также ошибки и сообщения, в случае отклонения системы от нормальной работы. Механизмами работы являются политика информационной безопасности и технические задания на модификацию системы управления технологическими процессами.

Входом для работы «Обеспечение производительности» являются метрики производительности СУБД и заявки пользователей. На выходе работы получается повышение уровня производительности БД и СУБД, а также ошибки и сообщения, в случае отклонения системы от нормальной работы. Механизмами работы являются политика информационной

безопасности, а также технические задания на модификацию системы управления технологическими процессами.

Входом для работы «Обработка аварийных ситуаций» являются сообщения и ошибки о критических ситуациях в режиме работы СУБД и БД. Выходом являются разработка мер по предупреждению критических ситуаций, фиксирование инцидентов и разработка мер по ликвидации простоев базы данных.

В свою очередь каждый блок - работа диаграммы декомпозиции может быть так же декомпозирован.

Далее приведена диаграмма декомпозиции для работы «Обеспечение доступности». (Рис. 6)

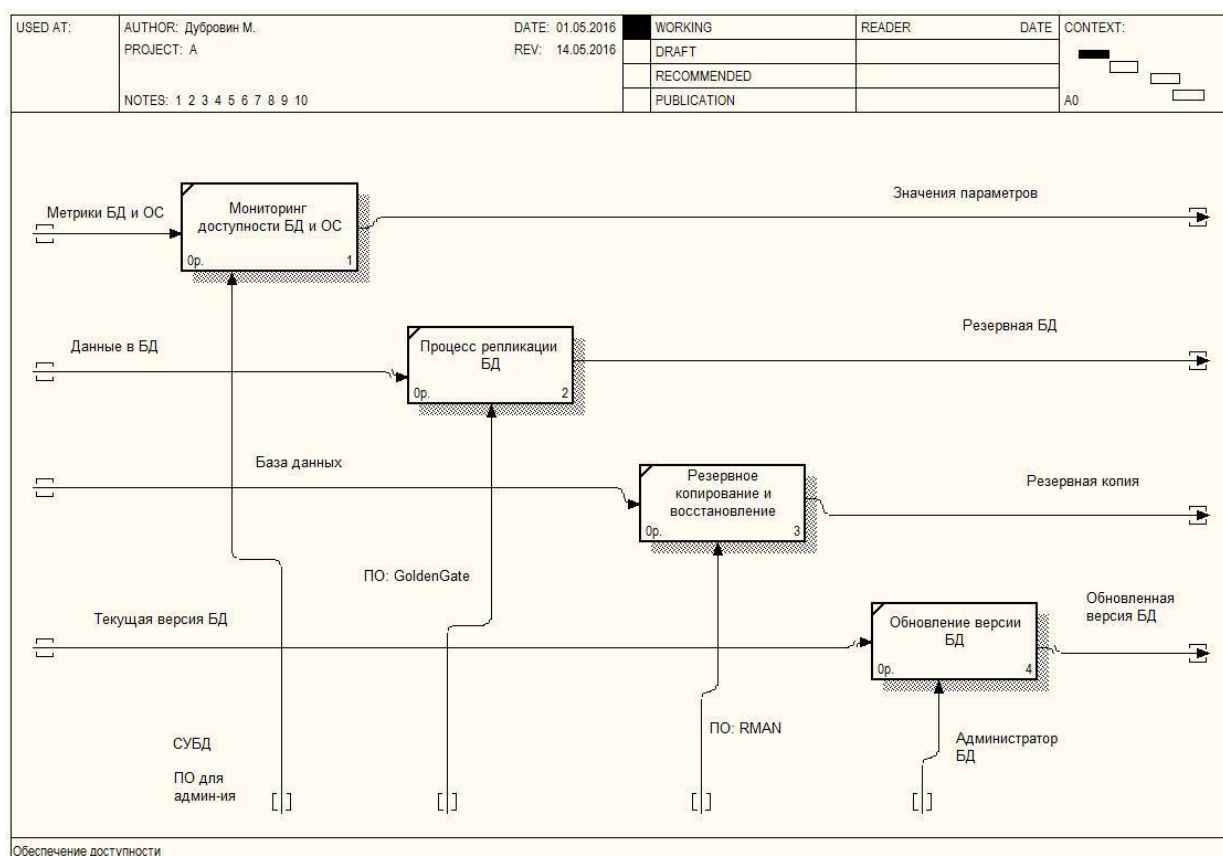


Рис. 6 IDEF0. Обеспечение доступности

Она состоит из следующих блоков:

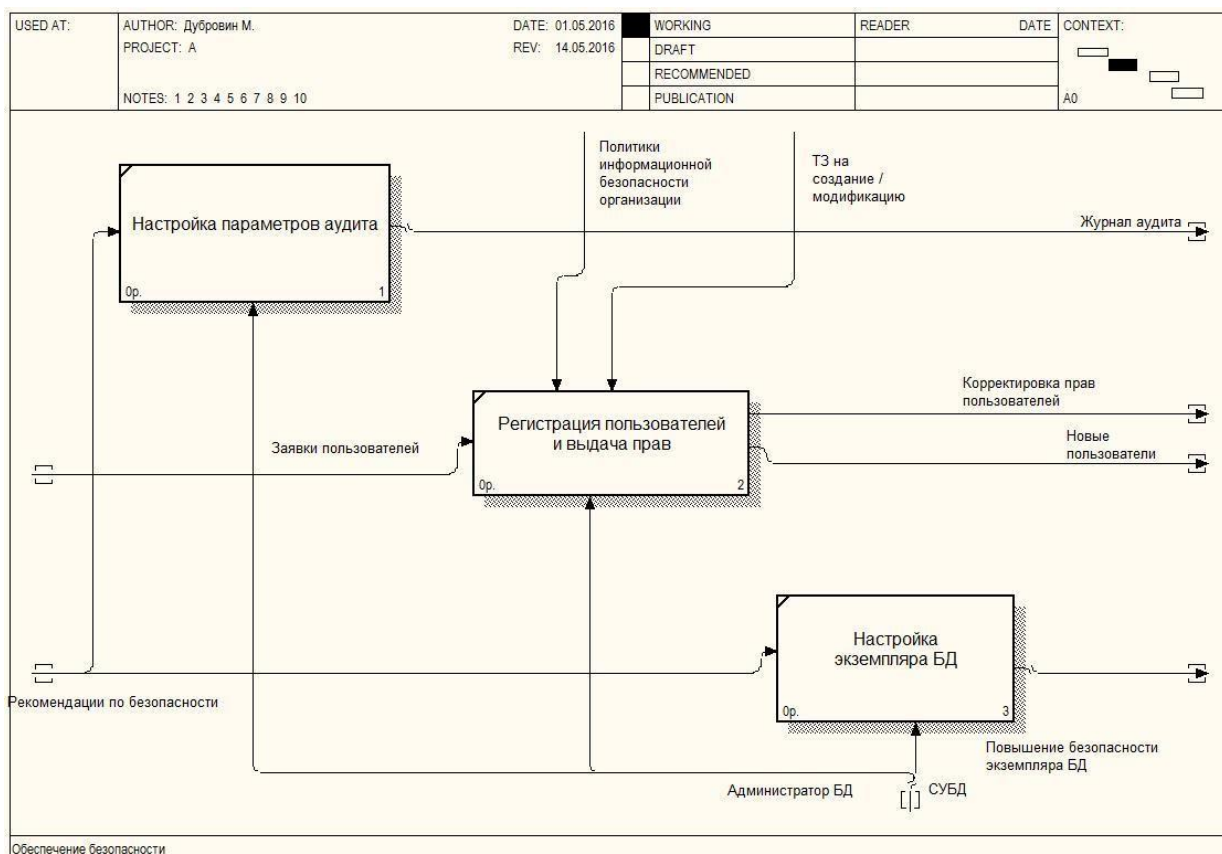
- Мониторинг доступности БД и ОС. Фиксирование параметров показателей, характеризующих работоспособность базы данных, операционной системы, а также определенных процессов ОС.

- Процесс репликации базы данных. Реализация репликации данных, для получения резервной копии исследуемой базы данных. Репликация данных обеспечивается с помощью программного обеспечения GoldenGate, разработанное компанией Oracle.

- Резервное копирование и восстановление. Периодическое выполнения резервного копирования базы данных с помощью ПО RMAN и восстановление базы данных из резервной копии при необходимости.

- Обновление версий СУБД. Выполнение обновления СУБД для получения более современной версии в соответствии с лицензионным соглашением, а также для устранения ошибок в СУБД.

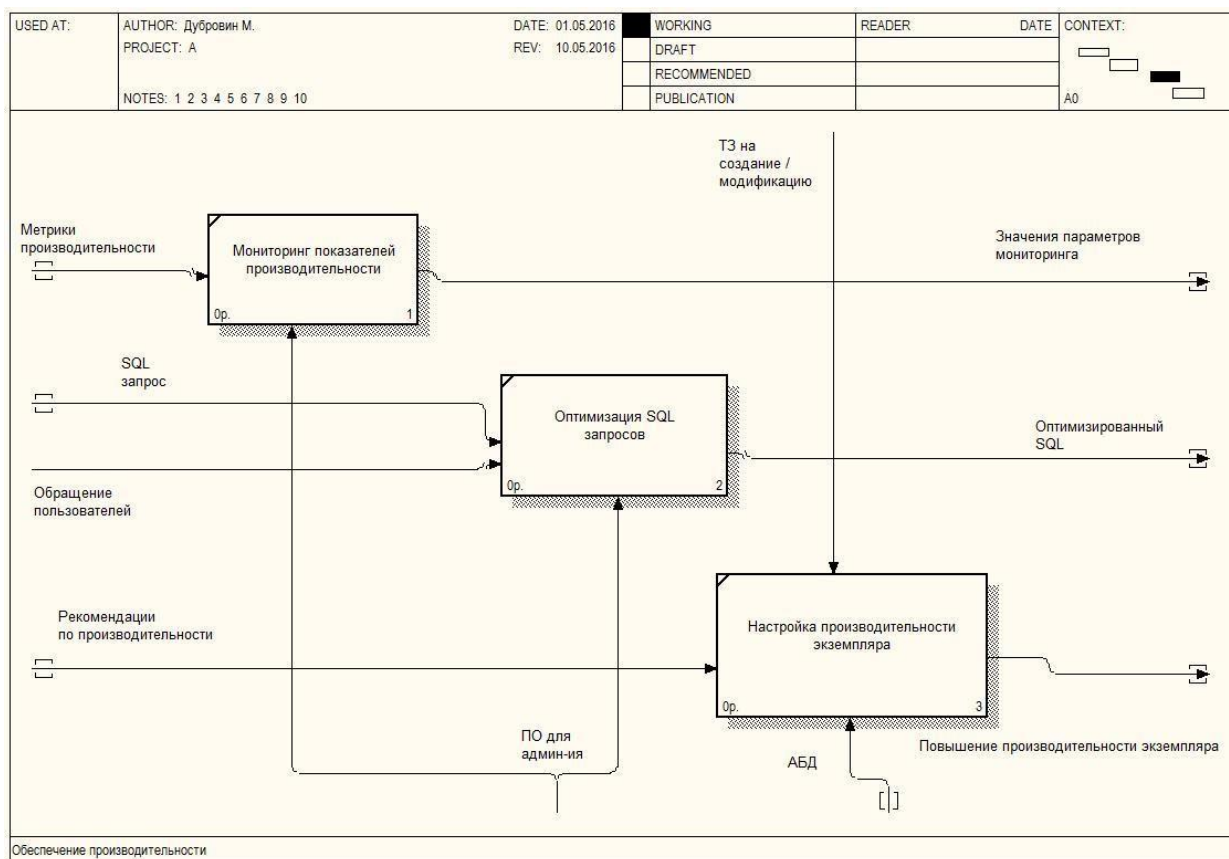
Далее приведена диаграмма декомпозиции для работы «Обеспечение безопасности». (Рис. 7) Она включает в себя следующие блоки:



**Рис. 7 IDEF0. Обеспечение безопасности БД**

- Настройка параметров аудита. Аудит операций с базой данных для предотвращения несанкционированного доступа к базе данных.
- Регистрация пользователей и выдача прав. Создание новых пользователей в случае расширения или модификации системы управления технологическими процессами, а также выдача им прав, необходимых для выполнения функциональных обязанностей.
- Настройка экземпляра БД. Настройка параметров экземпляра базы данных, для повышения информационной безопасности БД и предотвращения несанкционированного доступа.

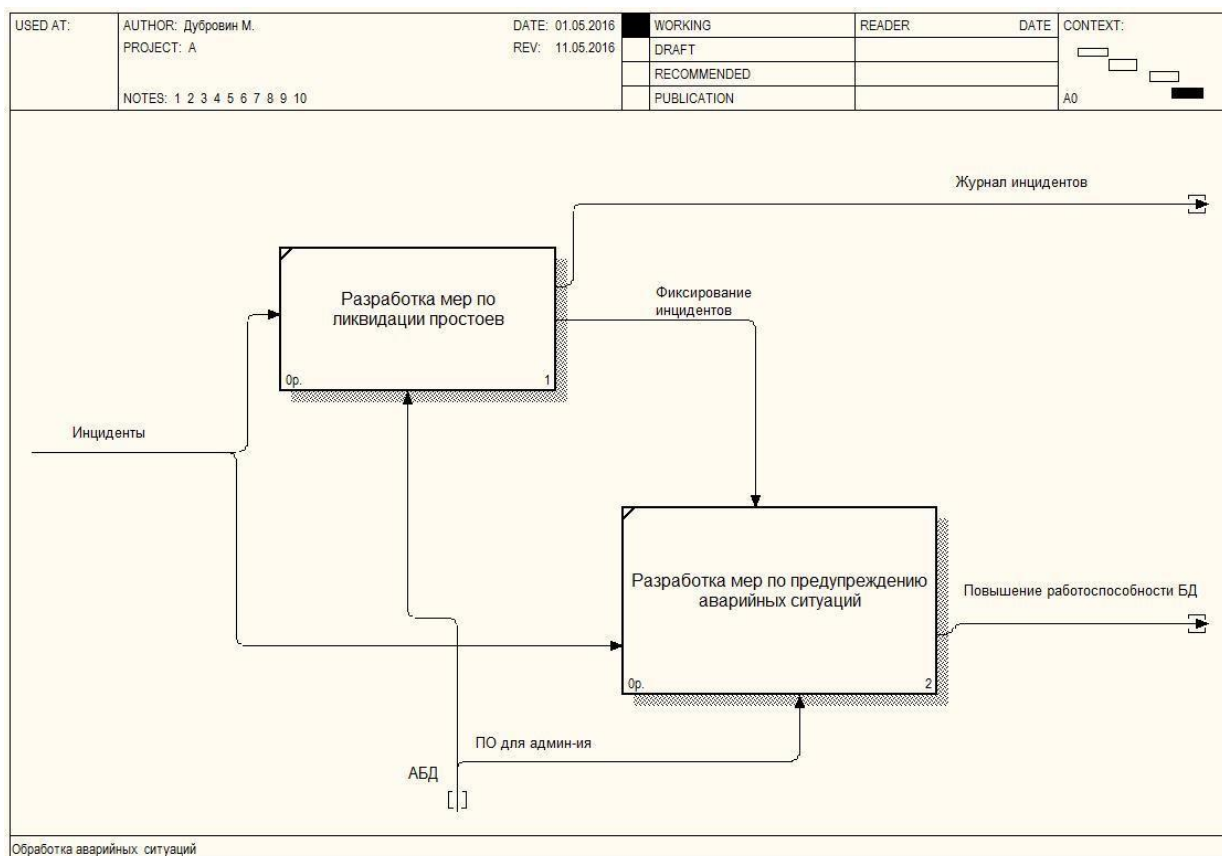
Далее приведена диаграмма декомпозиции для работы «Обеспечение производительности». (Рис. 8) Она включает в себя следующие блоки:



**Рис. 8 IDEF0. Обеспечение производительности БД**

- Мониторинг показателей производительности. Фиксирование параметров производительности экземпляра базы данных.
- Оптимизация SQL запросов. Оптимизация SQL запросов при обращении пользователей информационной системы, либо разработчиков.
- Настройка производительности экземпляра. Настройка параметров экземпляра базы данных, для обеспечения высокой производительности, в зависимости от требований, предъявляемых к серверу базы данных.

Далее приведена диаграмма декомпозиции для работы «Обеспечение производительности». (Рис. 9) Она включает в себя следующие блоки:



**Рис. 9 IDEF0. Обработка аварийных ситуаций**

- Разработка мер по ликвидации простоев. Устранения неполадок в работе сервера базы данных, на основе инцидентов в работоспособности и сообщений, в результате проведения мониторинга сервера БД.
- Разработка мер по предупреждению аварийных ситуаций. Разработка мер по предупреждению критических ситуаций в работе сервера базы данных, на основе инцидентов в работоспособности и сообщений, в результате проведения мониторинга сервера БД.

## § 2.2 Построение диаграммы потоков данных (DFD)

Основной целью данного этапа проектирования является выделение внешних сущностей, осуществляющих информационный обмен с предметной областью, определение маршрутов и содержания информационных потоков, выделение и описание существующих хранилищ данных. Дополнительной



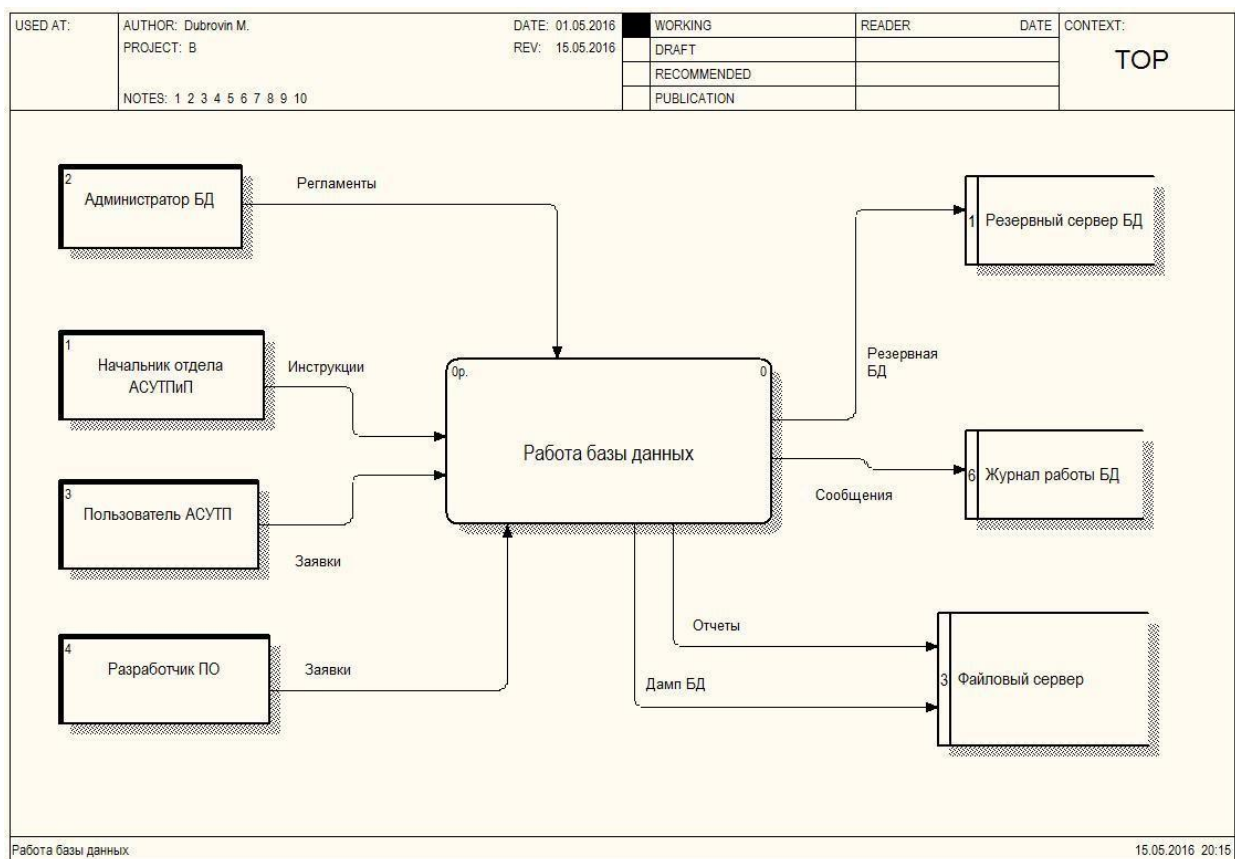
целью этапа является более глубокое и системное изучение предметной области. (Рис. 10)

На схеме информационных потоков выделены следующие внешние сущности:

- Администратор базы данных. Обеспечивает работу базы данных и составляет регламенты, описывающие требования к работе базы данных.
- Начальник отдела АСУТПиП. Выдает инструкции о работы базы данных и о политике информационной безопасности организации.
- Пользователь АСУТП. Заполняет заявки для работы с базой данных с указанием функциональных обязанностей и составляет обращение о некорректной работе базы данных.
- Разработчик ПО. Заполняет заявки на предоставление необходимых прав для работы с базой данных, а также на оптимизацию SQL запросов, используемых при разработке программного обеспечения.

На схеме информационных потоков выделены следующие накопители данных:

- Резервный сервер базы данных. Хранит актуальную резервную копию базы данных, направленную на восстановление работы, в случае критической ситуации.
- Журнал работы БД. Хранит данные о работе базы данных, с помощью которых можно отслеживать работоспособность базы данных и анализировать инциденты.
- Файловый сервер. Хранит информацию с отчетами о работе базы данных за определенный промежуток времени, а также последние дампы базы данных.



**Рис. 10 DFD Работа базы данных**

В связи с тем, что автоматизированная система мониторинга базы данных не направлена на сокращение документооборота организации, декомпозиция диаграммы потоков данных является избыточной.

### § 2.3 Проектирование базы данных

Для хранения конфигурационных настроек, а также для хранения данных, проводимых мониторингом необходимо создать информационную базу данных. Для представления данных в базе данных и отношений между ними реализована ER – диаграмма.

Далее представлены сущности, используемые в программном обеспечении:

1. Подключение. Характеризуется логином и паролем для подключения к базе данных; логином и паролем для подключения к

ОС; именем компьютера, к которому производится подключение; системным идентификатором БД; статусом активности.

2. Пользователь. Определяется логином и паролем для доступа к программному обеспечению.

3. Тип метрики. Характеризуется типом метрики и его описанием. 4. Значение метрики. Характеризуется типом, значением, описанием и датой фиксации метрики.

5. Граница метрики. Определяется верхней границей нормального состояния метрики и нижней границы критического состояния.

6. Проблема. Определяется типом и значением метрики, в которой возникла проблема; временем возникновения проблемы и описанием критической ситуации.

7. Статус проблемы. Характеризуется описанием статуса.

Основываясь на данных сущностях, построена ER- диаграмма для описания структуры базы данных приложения. (Рис. 11)

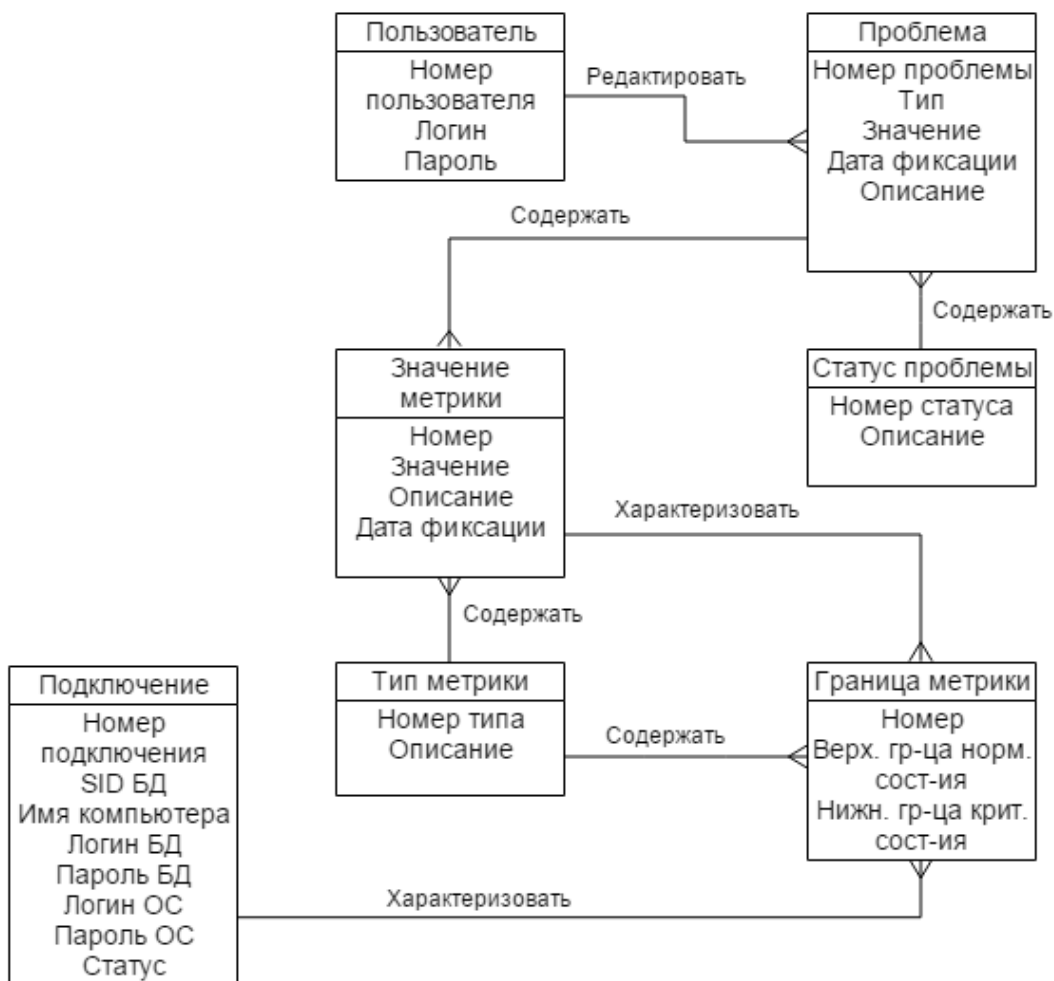


Рис. 11 ER диаграмма

На основе логической диаграммы построена физическая модель базы данных, содержащая 7 таблиц. Рассмотрим каждую из них.

1. Таблица «mon\_users». Содержит информацию о пользователях, у которых есть доступ к структуре программного обеспечения.

Таблица 2

Описание таблицы mon\_users

Имя	Тип	Ключи	Описание
ID	Integer	Primary	ID пользователя

Login	Text		Логин
Pas	Text		Пароль

2. Таблица «mon\_connect». Содержит информацию о подключениях к исследуемым серверам баз данных.

Таблица 3  
Описание

таблицы mon\_connect

Имя	Тип	Ключи	Описание
Id_con	Integer	Primary	ID подключения
server	Text		Наименование или IP сервера
oslogin	Text		Логин ОС для подключения
ospas	Text		Пароль ОС
SID	Text		SID БД
dblogin	Text		Логин к БД
dbpas	Text		Пароль к БД
status	Text		Статус подключения

3. Таблица «mon\_metric\_dic». Хранит словарь метрик, обращение к которым будет использовано для исследования работоспособности СУБД и ОС.

Таблица 4  
Описание

таблицы mon\_metric\_dic

Имя	Тип	Ключи	Описание
ID	Integer	Primary	ID метрики
value	Text		Описание метрики

4. Таблица «mon\_metric\_set». Содержит данные с верхней границей нормального состояния метрики и нижней границей критического состояния.

Таблица 5

Описание таблицы mon\_metric\_set

Имя	Тип	Ключи	Описание
Id_ms	Integer	Primary	ID
Id_con	Integer	Foreign	ID подключения
Id_dic	Integer	Foreign	ID типа метрики
lowervalue	Integer		Нижняя гр-ца крит-ого состояния метрики
upvalue	Integer		Верхняя гр-ца норм-ого состояния метрики

5. Таблица «mon\_metric\_his». Предназначена для хранения значений метрик, с указанием даты фиксации.

Таблица 6

Описание таблицы mon\_metric\_his

Имя	Тип	Ключи	Описание
Id_met	Integer	Primary	ID значений метрики
Id_ms	Integer	Foreign	ID границ метрики
ts	Datetime		Дата фиксации
value	Integer		Значение метрики
desc	Text		Описание комп-та метрики

6. Таблица «mon\_problem\_status». Содержит словарь статусов критических ситуаций.

Таблица 7

Описание таблицы mon\_problem\_status

Имя	Тип	Ключи	Описание
ID	Integer	Primary	ID статуса проблемы
status	Text		Описание статуса

7. Таблица «mon\_problem\_his». Предназначена для хранения

критических ситуаций в работе сервера БД.

Таблица 8

Описание таблицы mon\_problem\_his

Имя	Тип	Ключи	Описание
Id	Integer	Primary	ID проблемы
Id_us	Integer	Foreign	ID пользователя
Id_m	Integer	Foreign	ID значения метрики
Id_st	Integer	Foreign	ID статуса проблемы
Description	Text		Описание проблемы

Далее представлена физическая модель базы данных. (Рис. 12)

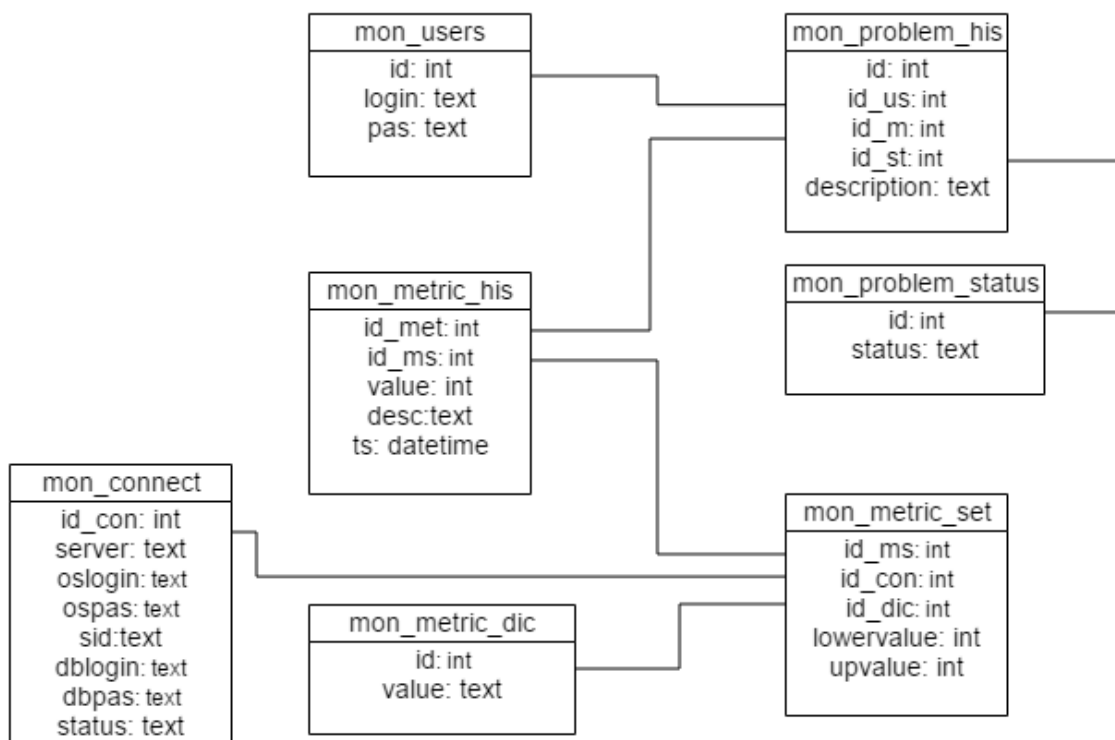


Рис. 12 Физическая модель базы данных

## ГЛАВА 3. РЕАЛИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

### § 3.1 Описание инструментальных средств

Базовым языком программирования, используемого при разработке программного обеспечения является Delphi. Разработка проводилась в интегрированной среде разработки Embarcadero RAD Studio 2010.

Выбор языка программирования основывается на следующих показателях:

- Предприятие имеет лицензионное соглашение на разработку программного обеспечения в среде Embarcadero RAD Studio 2010.
- Среда разработки обеспечивает реализацию всех функциональных и нефункциональных требований, предъявленных к программному обеспечению.
- Наличие стабильных библиотек и модулей, необходимых для разработки программного обеспечения;
- Все программное обеспечение для систем автоматизации предприятия, представленное компанией Danieli Automation, написано на выбранном языке программирования. В связи с этим, облегчается процесс поддержки программных продуктов предприятия.
- Быстрота разработки программного обеспечения связана с достаточным уровнем владения ЯП у разработчика приложения.

Для расширения функционала стандартной реализации IDE Embarcadero RAD Studio 2010 использованы следующие компоненты:

- ZeosLib. Проект с открытым исходным кодом, поддерживающий несколько систем управления базами данных для Delphi. Необходим для взаимодействия с БД SQLite.



- ODAC. Oracle Data Access Components представляет собой набор из нескольких невизуальных компонентов, которые предназначены для ЯП Delphi и служат для получения доступа к данным сервера базы данных Oracle. Основным преимуществом ODAC является прямой доступ к данным сервера Oracle. Не требует установки других уровней средств доступа к данным (таких как BDE и ODBC).

- TMS Component Pack. Набор TMS Component Pack поможет создавать многофункциональные и современные приложения для Windows с помощью более чем 360 компонентов. Используется для создания качественного интерфейса программного обеспечения.

- EhLib. Библиотека EhLib включает в себя компоненты и классы для Borland Delphi и предназначена для расширения функциональности клиентской части приложений, работающей с БД: вывод, печать и занесения данных конечным пользователем.

В качестве БД для хранения данных программного обеспечения выбрана SQLite.

SQLite — это постоянно совершенствуемая библиотека, осуществляющая работу автономного механизма баз данных SQL, который не нуждается в сервере.[14]

Выбор SQLite обоснован следующими признаками:

- Код, используемый в SQLite, является общественным достоянием, и потому он может использоваться в любых целях — коммерческих или частных.

- SQLite не нуждается в установке перед использованием. Нет никакой процедуры инсталляции. Отсутствуют какие-либо серверные процессы,

которые необходимо запускать, останавливать или конфигурировать. Нет никакой потребности в создании администратором нового образца базы данных или в назначении пользователям прав доступа. Эта библиотека не использует никакие файлы конфигурации.

- SQLite является внедренным механизмом базы данных SQL. В отличие от большинства других баз данных SQL SQLite не имеет отдельного серверного процесса. SQLite читает и пишет непосредственно обычным файлом на диске. Полная база данных SQL с многочисленными таблицами, индексами, триггерами содержится в единственном файле на жестком диске. Формат файла базы данных является межплатформенным. Можно свободно копировать базу данных между 32- и 64-разрядными системами.[14]

- SQLite библиотека позволяет читать, обновлять и искать данные на основе языка запросов SQL, ориентируясь на стандарт SQL 92.

- Возможность хранить данные в базе объемом до 2 терабайт.

Для удаленного обращения к объектам операционных систем Windows используется технология Windows Management Instrumentation.

WMI — это одна из базовых технологий для централизованного управления и слежения за работой различных частей компьютерной инфраструктуры под управлением платформы Windows.[31]

По своей сути WMI – это расширенная и адаптированная компанией Microsoft реализация стандарта WBEM (WebBased Enterprise Management компании DMTF Inc ). В основе WBEM лежит идея создания универсального интерфейса мониторинга и управления к различным системам и компонентам распределенной информационной среды предприятия с использованием

объектно-ориентированной идеологии и широко распространенных веб-технологий представления информации: протоколов XML и HTTP.

Таким образом, WMI – это открытая унифицированная библиотека (репозиторий) однотипных интерфейсов доступа к параметрам, настройке и свойствам различных систем и их компонентов.

Так как WMI построен по объектно-ориентированному принципу, то все данные операционной системы представлены в виде объектов и их свойств и методов.[31]

Все классы группируются в пространства имен, которые иерархически упорядочены и логически связаны друг с другом по определенной технологии или области управления. В WMI имеется одно корневое пространство имен Root, которое в свою очередь имеет 4 подпространства: CIMv2, Default, Security и WMI.

Классы имеют свойства и методы и находятся в иерархической зависимости друг от друга, то есть классы-потомки могут наследовать или переопределять свойства классов-родителей, а также добавлять свои свойства.

Свойства классов используются для однозначной идентификации экземпляра класса и для описания состояния используемого ресурса. Обычно все свойства классов доступны только для чтения, хотя некоторые из них можно модифицировать определенным методом. Методы классов позволяют выполнить действия над управляемым ресурсом.

Для обращения к объектам WMI используется специфический язык запросов WMI Query Language (WQL), который является одной из разновидностей SQL. Основное его отличие от ANSI SQL — это невозможность изменения данных, то есть с помощью WQL возможна лишь выборка данных с помощью команды SELECT. Помимо ограничений на работу с объектами, WQL не поддерживает такие операторы как DISTINCT,

JOIN, ORDER, GROUP, математические функции. [34]

Создание диаграмм IDEF0 и DFD производилось с помощью программного обеспечения AllFusion Process Modeller r7. Создание ER – диаграммы, физической модели базы данных и других схем, использованных в работе, производилось с помощью онлайн – ресурса <https://www.gliffy.com/>.

## § 3.2 Описание алгоритма работы приложения

Приложение работает согласно архитектуре «клиент – сервер», где в качестве клиента выступает реализованное программное обеспечение, а сервером является исследуемый сервер базы данных.

Алгоритм работы приложения реализован следующим образом:

Этап 1. Пройти аутентификацию для доступа к разделам программного обеспечения. Таблица пользователей `mon_users`, у которых есть доступ к приложению, хранится в локальной базе данных SQLite.

Этап 2. Создать строку подключения к Oracle и WMI. Данные для подключения вводятся с помощью интерфейса приложения, проверяется возможность подключения, и в случае успешного подключения, переход к этапу 3.

Этап 3. Сохранить строку соединения к Oracle и WMI в локальную базу данных SQLite. Пароли подключения к базе данных и интерфейсу Windows сохраняются в зашифрованном виде. Данные по подключению сохраняются в таблицу `mon_connect`.

Этап 4. Задать граничные значения метрик ОС и БД. Словарь метрик хранится в таблице `mon_metric_dic`. С помощью интерфейса задаются граничные значения метрик, характеризующих нормальное состояние метрики, отклоненное от нормального состояния и критическое значение, для созданного соединения. Указанные граничные значения сохраняются в таблицу `mon_metric_set`.

Этап 5. Создать сессию с Oracle и интерфейсом ОС Windows.

Этап 6. Отправить запросы SQL к Oracle, и WQL запросы к интерфейсу WMI. Запросы направлены на получение характеристик сервера Oracle и операционной системы.

Этап 7. Обработать данные, полученные в результате выполнения запросов. Значения метрик сравниваются со значениями заданными на этапе 3.

Этап 8. Сохранить полученные данные в соответствующие таблицы базы SQLite. Полученные значения на этапе 7 сохраняются в таблицу `mon_metric_his` для ведения истории. Значения метрик, которые свидетельствуют об отклоненном от нормального состояния и о критическом состоянии сохраняются в отдельную таблицу `mon_problems`.

Этап 9. Вывести данные о проблемах работоспособности за конкретный период из SQLite в интерфейс программы. В одном из разделов интерфейса приложения выводятся данные из таблицы `mon_problems`.

Этап 10. Сформировать отчет о работоспособности сервера за конкретный временной период. Открывается заранее подготовленный шаблон из директории `Templates`, в него заносятся данные и сохраняются в формате `Excel`.

Алгоритм работы программного обеспечения, представленный в виде блок - схемы согласно ГОСТ 19.701-90 описан в Приложении 1.

### **§ 3.3 Описание компонентов программного обеспечения**

Для корректного функционирования программного обеспечения директория программы должна содержать следующие компоненты:

- Исполняемый файл, для запуска программного обеспечения.

- Папка «Data», содержащая файл базы данных SQLite и библиотеку sqlite3.dll, необходимую для взаимодействия с базой данных.
- Папка «Templates», включающая шаблоны отчетов в формате Microsoft Excel.

Далее представлены модули исходного кода программного обеспечения, используемые в формировании исполняемого файла:

AboutUnit. Информационный модуль, предоставляющий информацию о разработчике и организации, а также версию программного обеспечения.

AuthUnit. Модуль, предназначенный для авторизации пользователя для получения доступа к компонентам приложения.

ConDMUnit. Невизуальный модуль, предназначенный для хранения компонентов доступа к данным. Модуль ориентирован на хранение запросов для взаимодействия с базой данных SQLite, используемых при создании, редактировании, либо удалении данных о соединениях с серверами Oracle и ОС Windows.

ConnectUnit. Модуль, содержащий форму создания нового соединения с сервером базы данных.

DBDMUnit. Невизуальный модуль, предназначенный для хранения компонентов доступа к данным. Модуль ориентирован на хранение запросов, которые отправляются к серверу Oracle, для получения данных о работоспособности базы данных и экземпляра сервера Oracle Database. Модуль так же хранит компоненты, предназначенные для визуального отображения данных, собираемых запросами.

DMUnit. Невизуальный модуль, предназначенный для хранения коллекций изображений, используемых в процессе эксплуатации ПО, а также компонентов, предназначенных для открытия соединения с базами данных Oracle и SQLite.

HelpUnit. Информационный модуль, предназначен для получения справочной информации по использованию программного обеспечения.

MainUnit. Основной модуль, содержащий все визуальные компоненты основного окна программного обеспечения. Содержит блоки: «Навигация» с возможностью перехода между разделами; «Подключения» с информацией по текущим подключениям; Раздел конфигураций программного обеспечения; Верхнее меню для управления программным обеспечением; Стартовая страница ПО.

MsgUnit. Модуль, предназначенный для отображения формы сообщения. Поддерживает три типа сообщений: предупреждающий, сообщение об ошибке и уведомление об удачном выполнении операции.

Схема взаимодействия модулей программного обеспечения представлена далее. (Рис. 13)



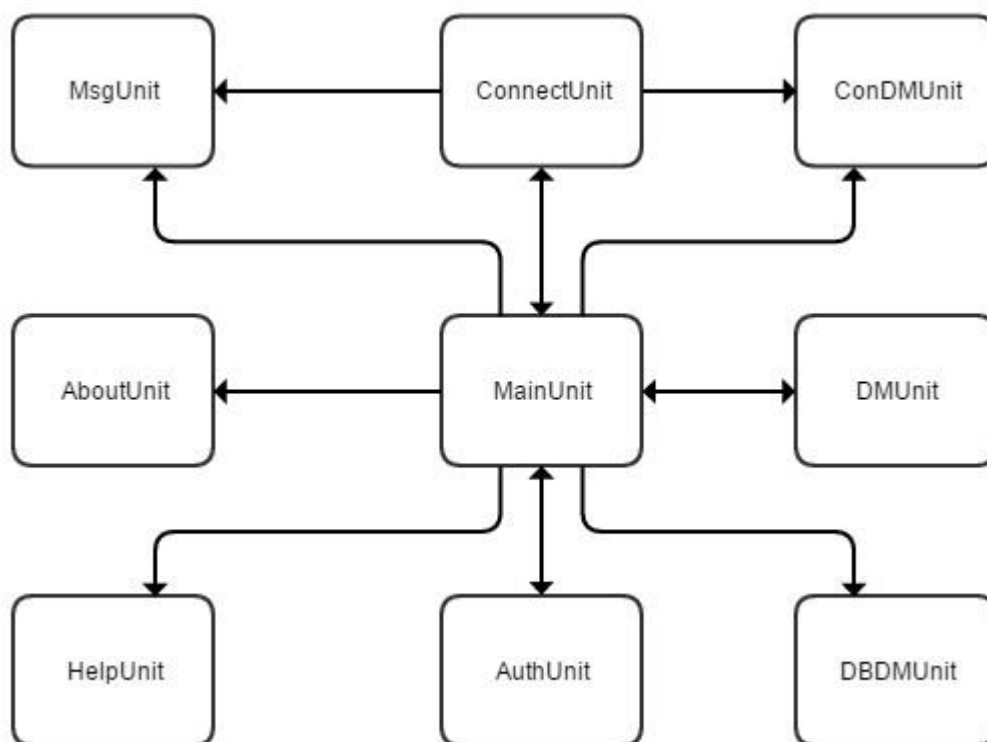


Рис. 13 Схема взаимодействия модулей ПО

В качестве методологии программирования использован структурный подход к программированию. В качестве подпрограмм выступают процедуры и функции.

Для реализации алгоритмов программного обеспечения созданы следующие процедуры и функции:

Procedure TimerChecked(checkindex:integer). Конфигурация времени таймера в зависимости от выбранного параметра.

Procedure ActivePanel(navindex:integer). Первоначальное отображение данных блока, в зависимости от выбранного блока.

Procedure OraConnect(servname:string; dbname:string; dblogin:string; dbpass:string). Выполнение соединения с сервером Oracle на основе данных, необходимых для установления соединения.

Procedure WMICConnect(servname:string; oslogin:string; ospas:string).  
Выполнение соединения с интерфейсом WMI на основе данных, необходимых для установления соединения.

Function CryptString(varstr:widestring):widestring. Шифрование строки.

Function ShowMsg(text:string; MSGType:byte):TModalResult. Вывод модального окна с сообщением в зависимости от типа сообщения. Поддерживаемые типы сообщений: предупреждение, ошибка, успешное выполнение.

Остальные процедуры предназначены для обработки действий пользователя, а также для обработки изменения данных.

Общее количество процедур составляет 94.

Листинг исходного кода, отображающий событие, которое вызывается при нажатии кнопки создания нового соединения в модуле ConnectUnit, отображен в Приложении 2.

Для получения информация об интерфейсе WMI использовано 7 WQL запросов. Процедура, исходный код которой представлен в Приложении 3 отображает событие, которое вызывается при открытии вкладки «Службы / процессы» в блоке «ОС» и демонстрирует взаимодействие с интерфейсом WMI OS Windows.

В Приложении 4 отображена процедура, которая показывает взаимодействие программного обеспечения с сервером Oracle, на примере вкладки «Файлы» блока «База данных».

Для получения информации о сервере Oracle используется 16 SQL запросов. Пример SQL запроса к базе данных Oracle отображен в

Приложении 5. Запрос направлен на получение информации о файлах данных базы данных Oracle.

Процедура, отображающая процесс выгрузки отчета в формате Microsoft Excel продемонстрирована в Приложении 6.

### § 3.4 Описание пользовательского интерфейса

При запуске приложения пользователь попадает на главную страницу ПО. На главной странице есть возможность нажать на кнопку «Запуск», чтобы произвести инициализацию ПО. Так же возможно просмотреть справку по использованию приложения, перейти к настройкам программы, либо нажать кнопку «Первый запуск», после нажатия которой происходит инициализация приложения и переход к созданию первого соединения. (Рис.

14)

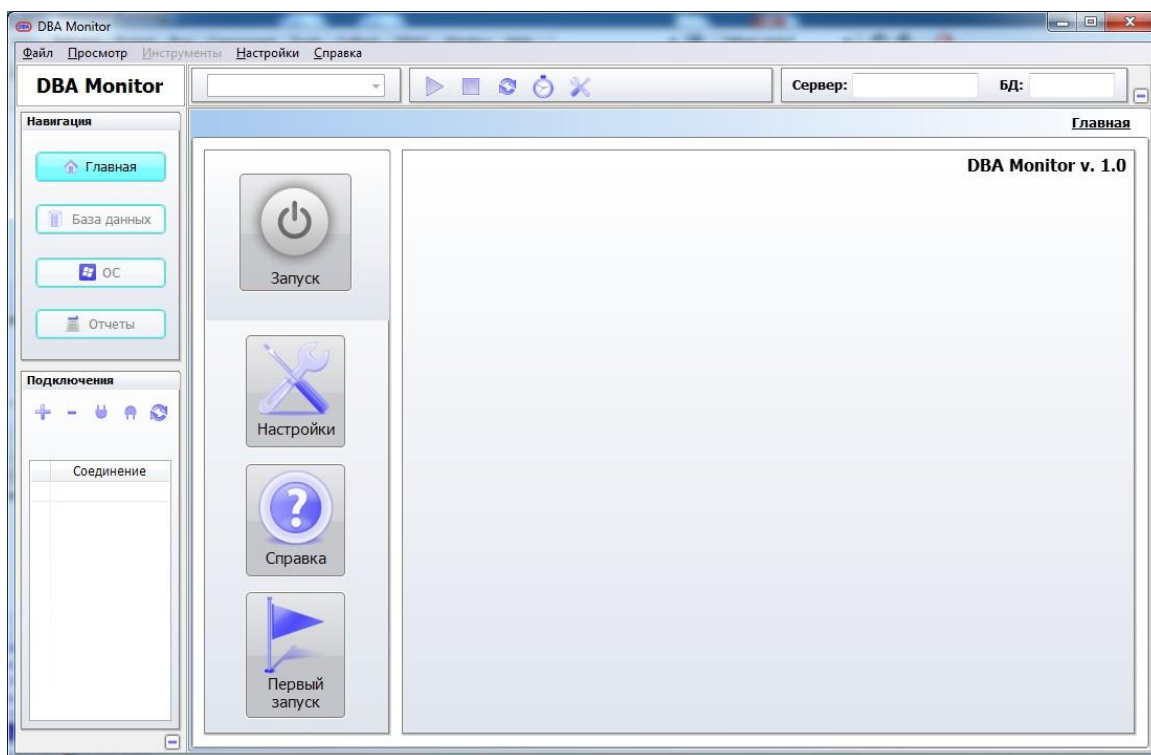


Рис. 14 Главная страница приложения

При нажатии кнопки «Запуск», пользователю приложения необходимо пройти аутентификацию для использования функций программного обеспечения. Для аутентификации необходимо ввести Логин и Пароль. (Рис.

15)

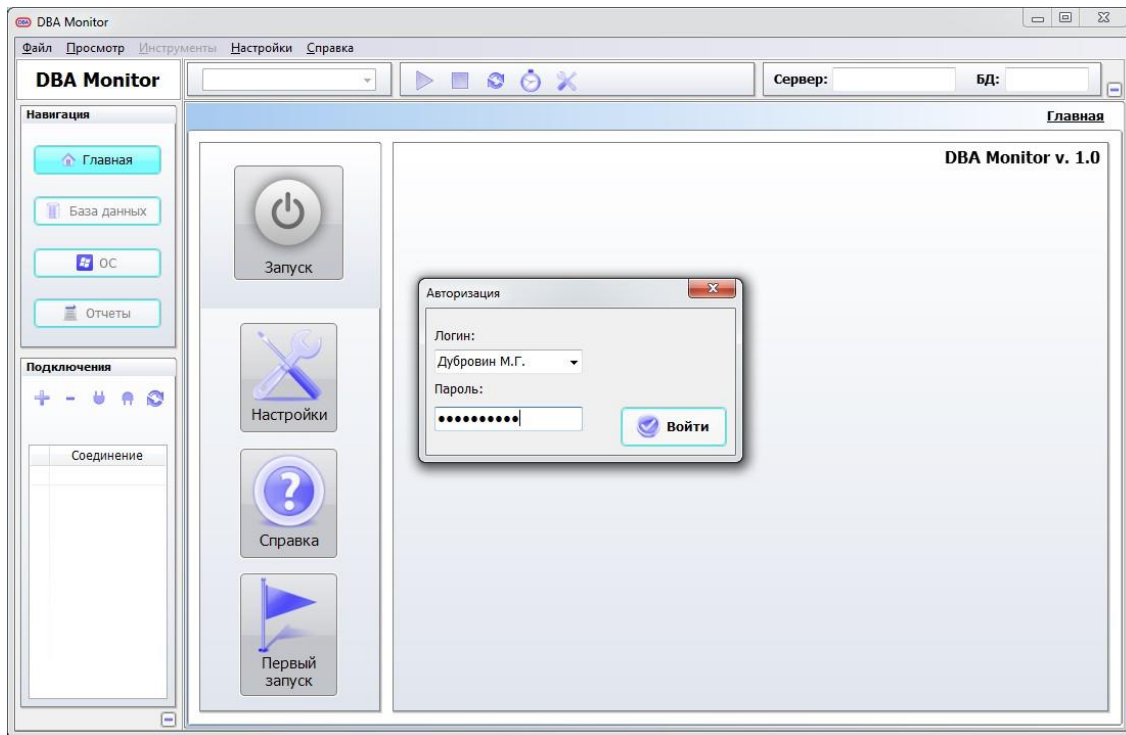


Рис. 15 Аутентификация пользователя

В случае удачной аутентификации состояние кнопки переходит в статус «Запущено», появляется доступ ко всем функциям программного обеспечения. Появляется список созданных подключений, существует возможность перехода к категориям приложения, а также появляется доступ к функциям верхнего меню приложения. (Рис. 16)

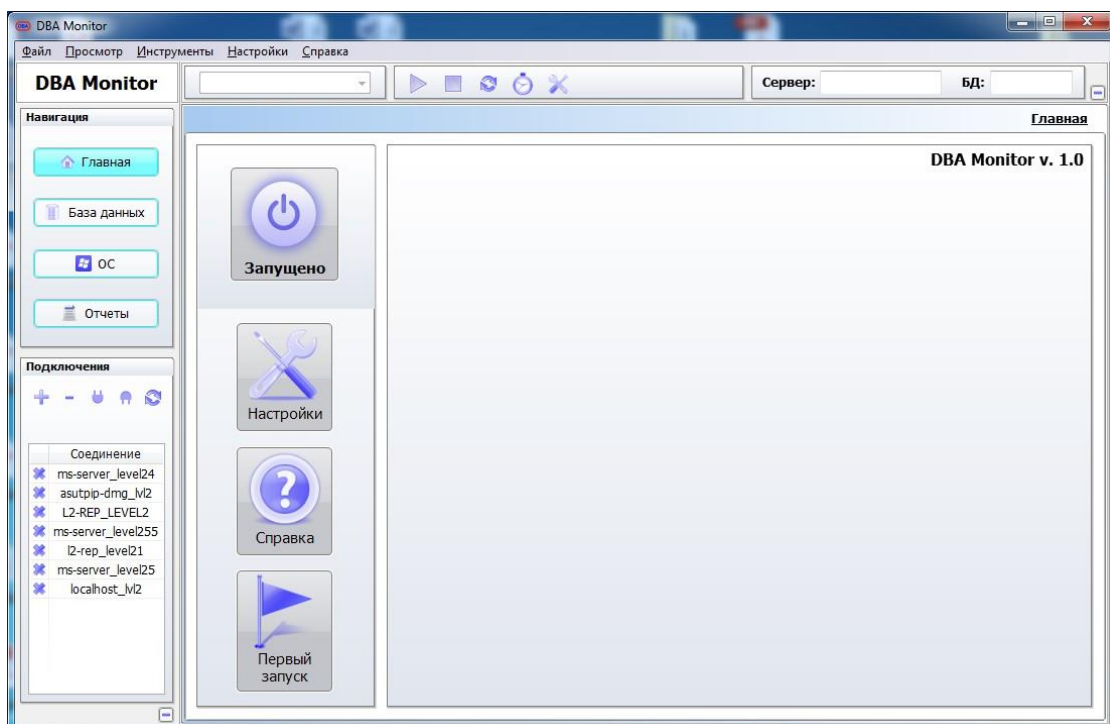


Рис. 16 Главная страница. Нажатие кнопки "Запуск"

При нажатии кнопки создания нового подключения, визуально отображающейся в виде плюса, располагающейся в блоке «Подключения», открывается форма создания нового соединения с сервером базы данных, которая разделена на три шага. В данном окне, для перехода к следующему этапу, нужно ввести данные, необходимые для подключения к базе данных.

(Рис. 17)

Создать соединение

Шаг 1 Шаг 2 Шаг 3 Соединение

**Шаг 1. Аутентификация БД**

Проверка соединения с исследуемой базой данных

Имя Сервера(IP): pc-01

Имя БД: xe

Логин: system

Пароль: ●●●●●●

Далее →

Отмена

Рис. 17 Создание соединения. Шаг 1

При нажатии кнопки «Далее» осуществляется попытка установления соединения с базой данных. При неудачном соединении отображается диалоговое окно с текстом ошибки. (Рис. 18)

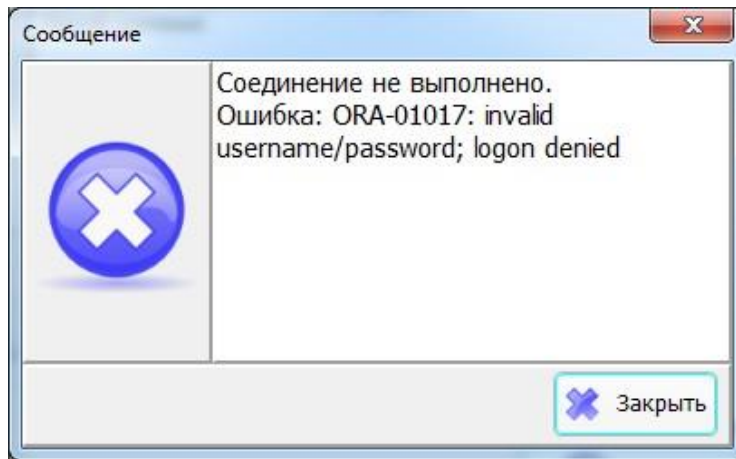


Рис. 18 Сообщение об ошибке

При удачном соединении отображается диалоговое окно с информацией об успешном выполнении операции. (Рис. 19)

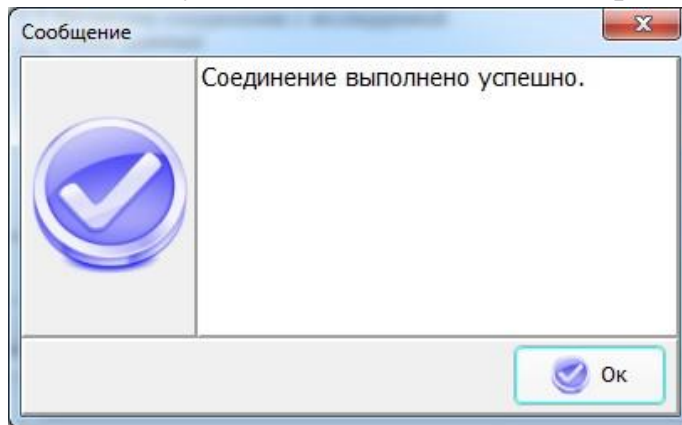


Рис. 19 Сообщение о выполнении операции

На втором шаге необходимо ввести данные, необходимые для установления соединения с интерфейсом WMI. (Рис. 20)

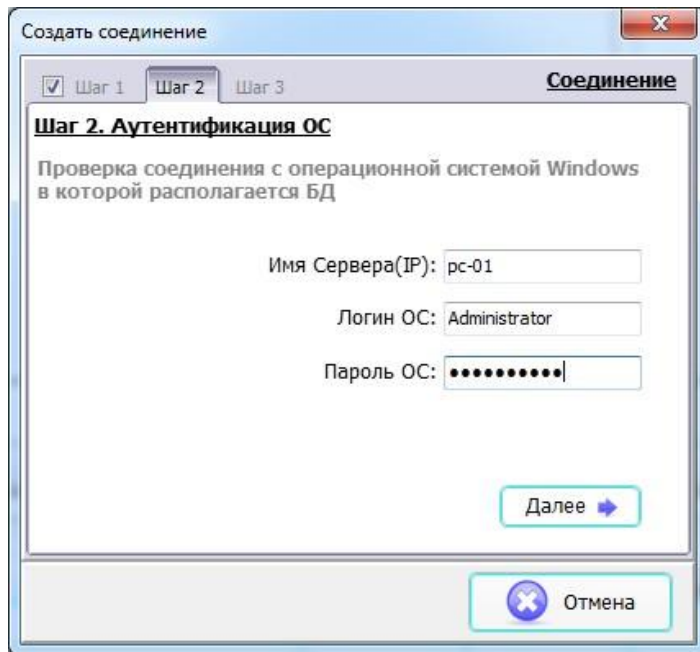


Рис. 20 Создание соединения. Шаг 2

При удачном соединении на втором шаге на заключительном шаге необходимо ввести уникальное имя соединения. В итоге создается запись в базе данных с новым соединением, и новое соединение отображается в блоке «Подключения». (Рис. 21)

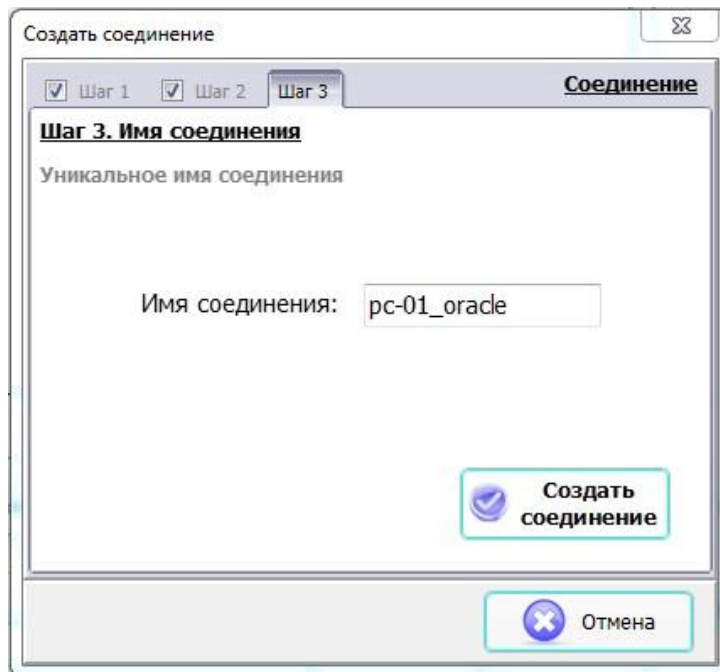




Рис. 21 Создание соединения. Шаг 3

Для того чтобы определить граничные значения метрик созданного подключения необходимо перейти на вкладку конфигураций, нажав на соответствующую кнопку верхнего меню. (Рис. 22)

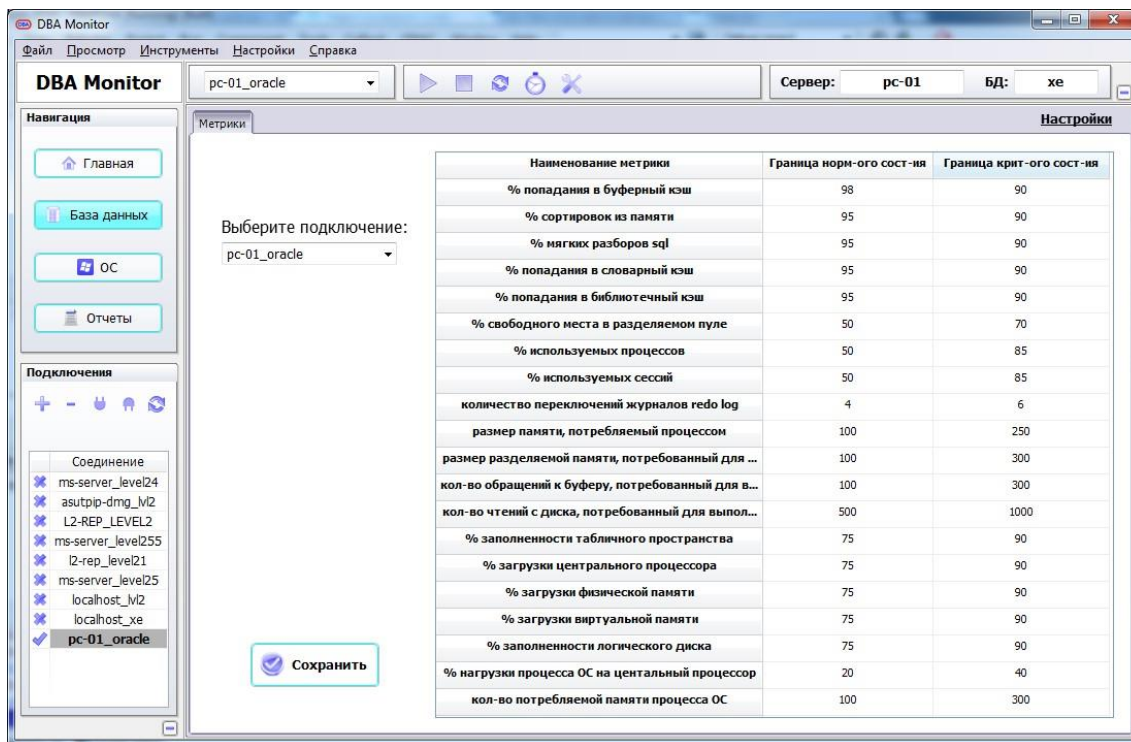


Рис. 22 Конфигурация метрик

Для осуществления мониторинга разделов «База данных» и «Операционная система» необходимо перевести одно или несколько соединений в активный режим. Для этого необходимо выбрать исследуемый сервер базы данных и нажать третью кнопку в блоке «Подключения». При установлении удачного соединения производится переход к подразделу «Общая информация» раздела «База данных». В данном разделе возможно просмотреть информацию о базе данных, экземпляре базы данных и параметрах экземпляра. (Рис. 23)

В левой части верхнего блока приложения отображается наименование текущего активного подключения. Для выбора другого подключения, его нужно выбрать в данной строке выбора. В центральной части верхнего блока приложения отображаются управляющие кнопки, которые позволяют

запускать и останавливать режим периодического мониторинга, при включении которого, с течением заданного периода производится обновление визуализируемых данных. Третья кнопка блока позволяет обновить данные в активном разделе. Четвертая кнопка позволяет выбрать время обновления таймера из заданных значений. Пятая кнопка предназначена для перехода к форме конфигурации приложения. Правая часть верхнего меню предназначена для отображения информации о текущем активном подключении.

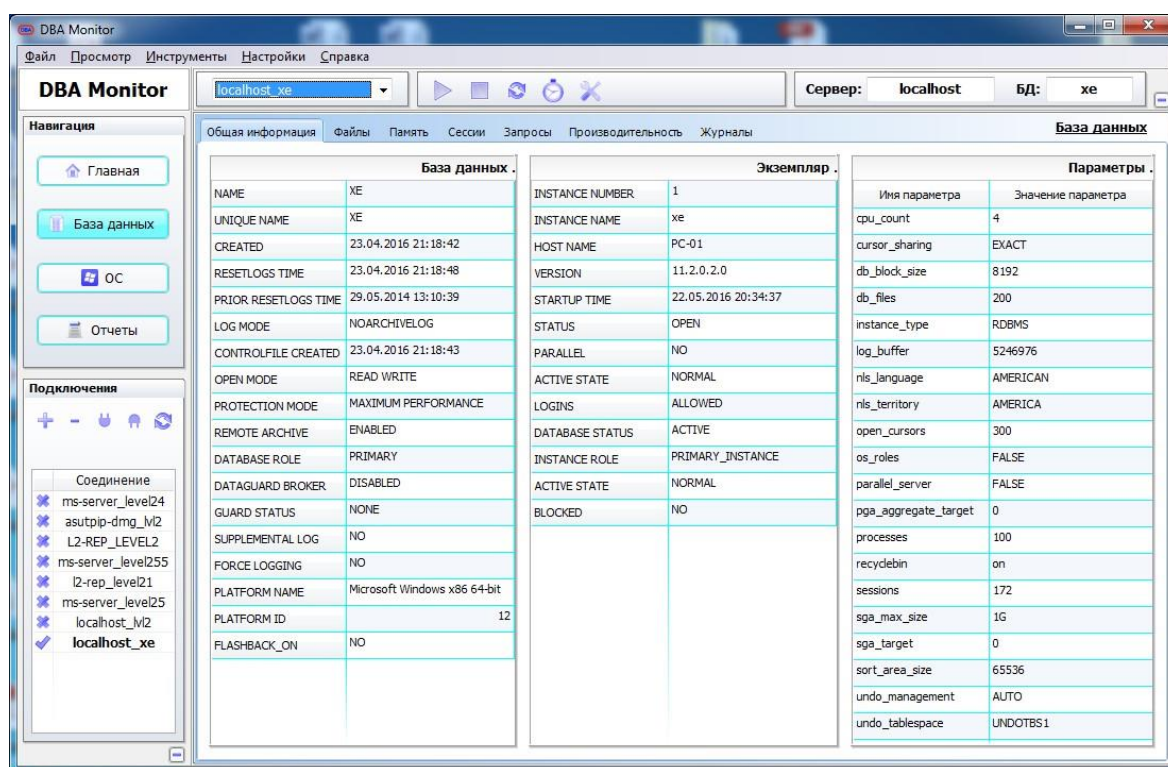


Рис. 23 Раздел "База данных". Подраздел 1

Во втором подразделе раздела «База данных», пользователь приложения может оценить состояние табличных пространств базы данных и файлов данных на основе визуальной информации. (Рис. 24)

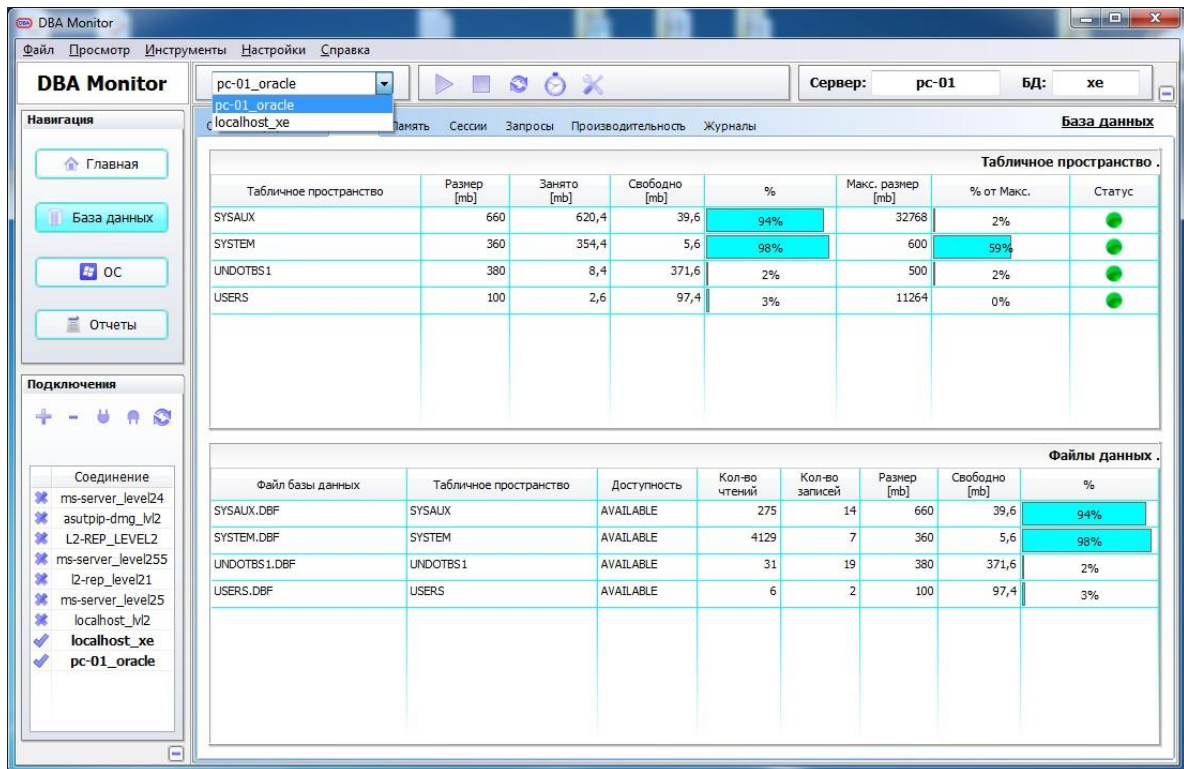


Рис. 24 Раздел "База данных". Подраздел 2

В подразделе «Память» пользователь приложения может оценить состояние структур памяти экземпляра базы данных Oracle. (Рис. 25). Для удобства мониторинга возможно скрыть левый и верхние управляющие панели, нажав соответствующие кнопки «минус» на них.

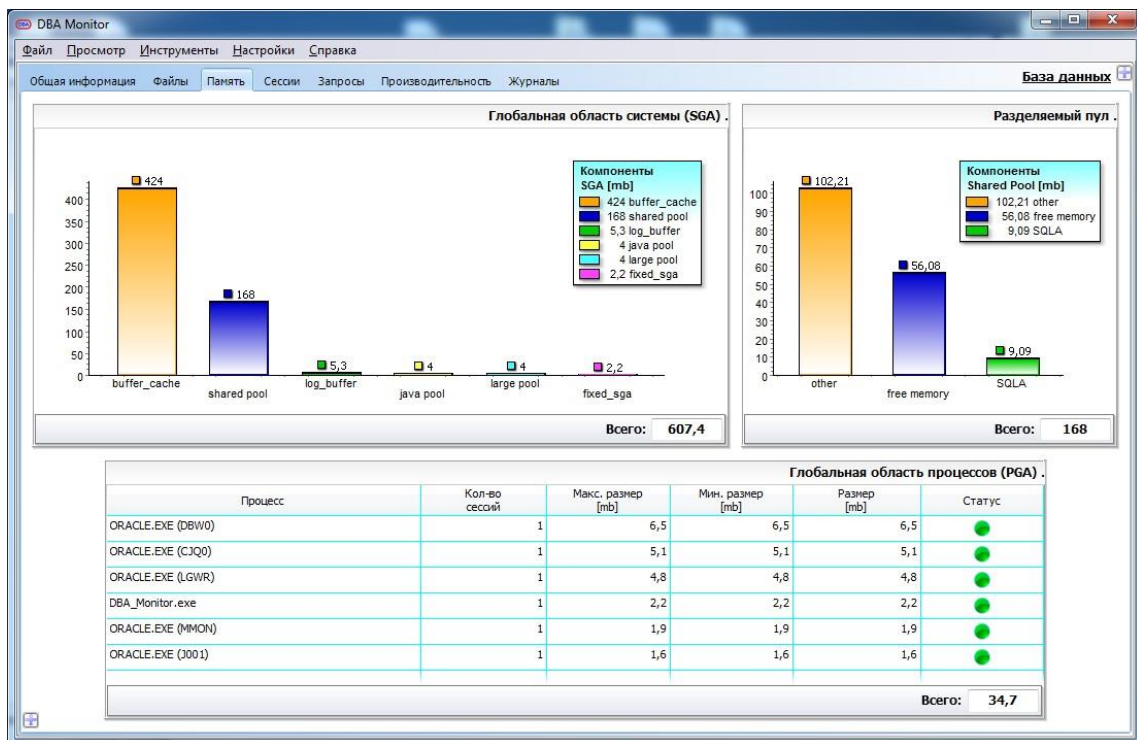


Рис. 25 Раздел "База данных". Подраздел 3

В подразделе «Сессии» пользователь приложения может увидеть активные и уже использованные сессии, а также подробную информацию по ним. (Рис. 26)

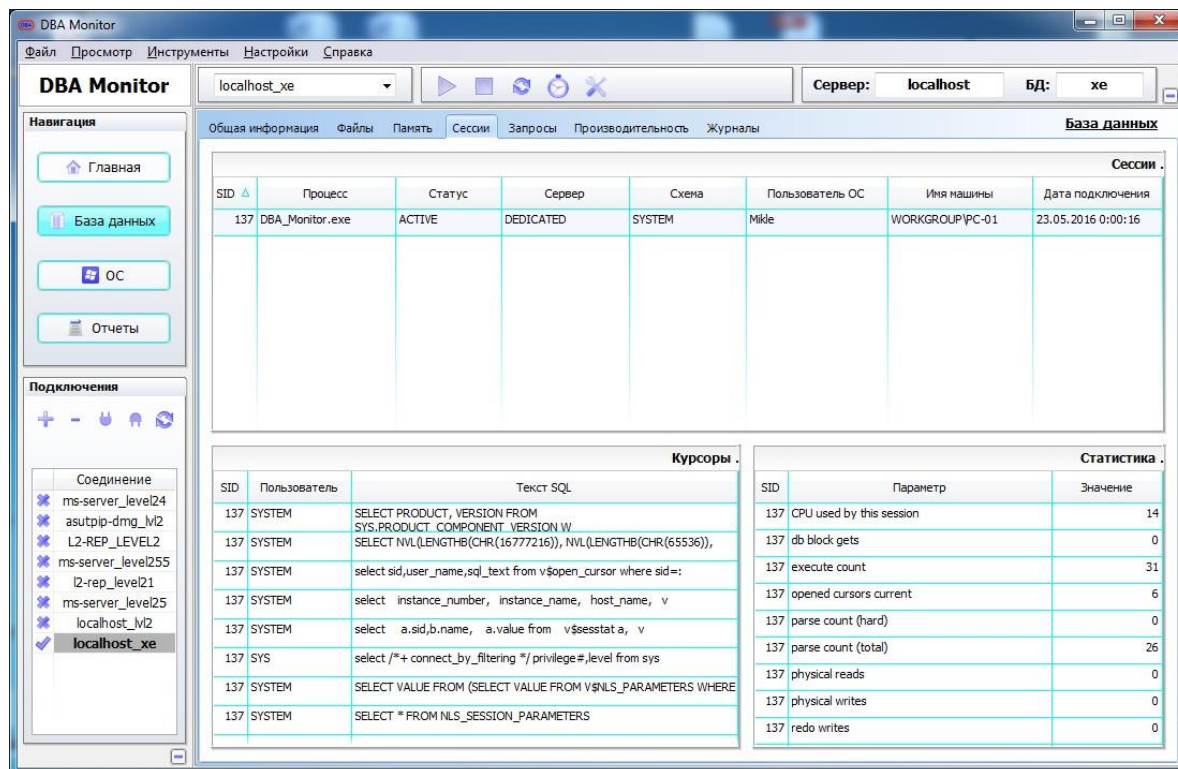


Рис. 26 Раздел "База данных". Подраздел 4

В следующем разделе, который называется «Запросы» возможно просмотреть последние использованные запросы к исследуемой базе данных и подробную информацию по их использованию. (Рис. 27)

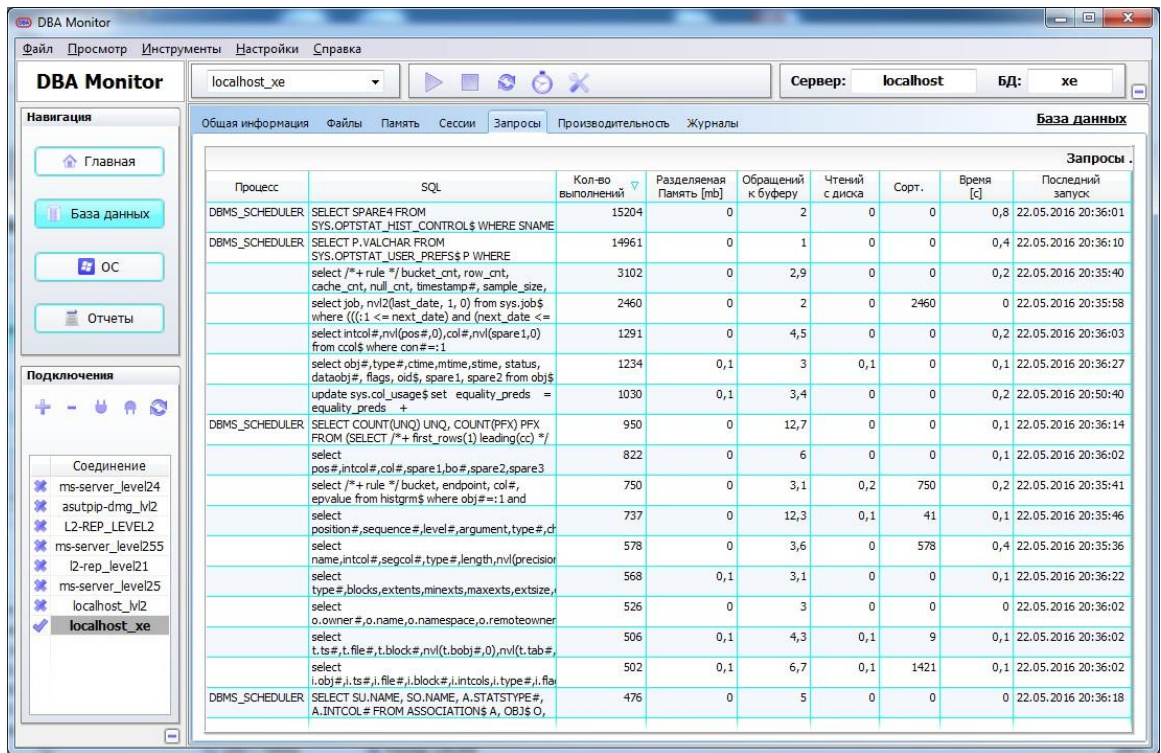


Рис. 27 Раздел "База данных". Подраздел 5

В следующем разделе «Производительность» возможно просмотреть ожидания экземпляра БД, которые негативно влияют на производительность, а также основные метрики производительности экземпляра. (Рис. 28)

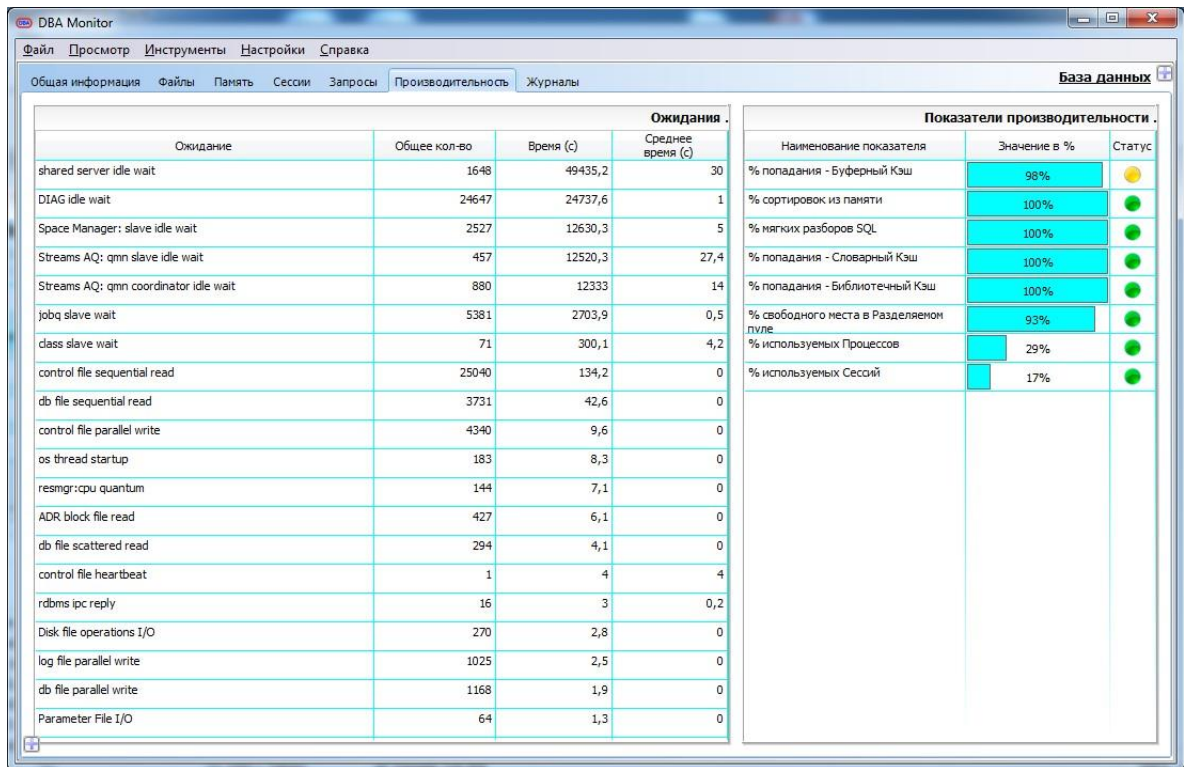


Рис. 28 Раздел "База данных". Подраздел 6

Последний подраздел «Журналы» блока «Базы данных» отображает график переключений Журналов повтора, что соответствует графику активности базы данных. (Рис. 29)

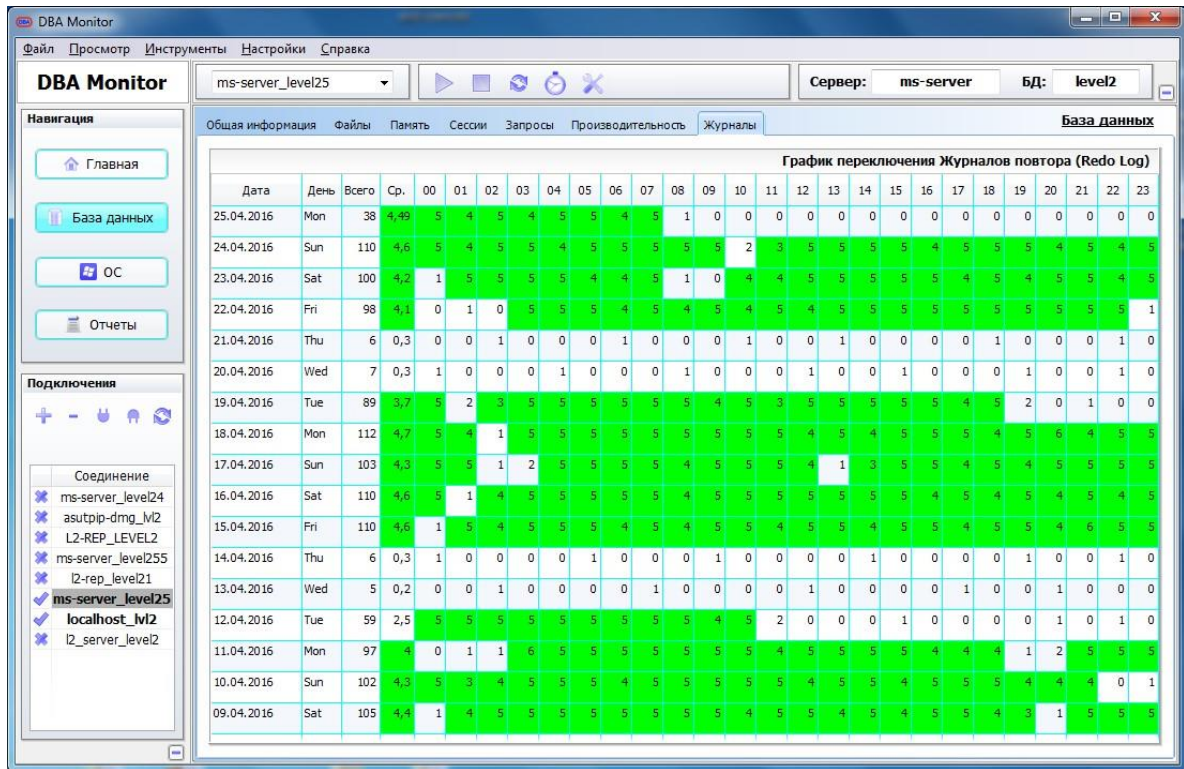


Рис. 29 Раздел "База данных". Подраздел 7

В разделе «ОС» в первом подразделе пользователю доступна информация о загрузке центрального процессора, физической и виртуальной памяти. Так же в данном разделе отображается информация о наполненности логических дисков сервера. (Рис. 30)

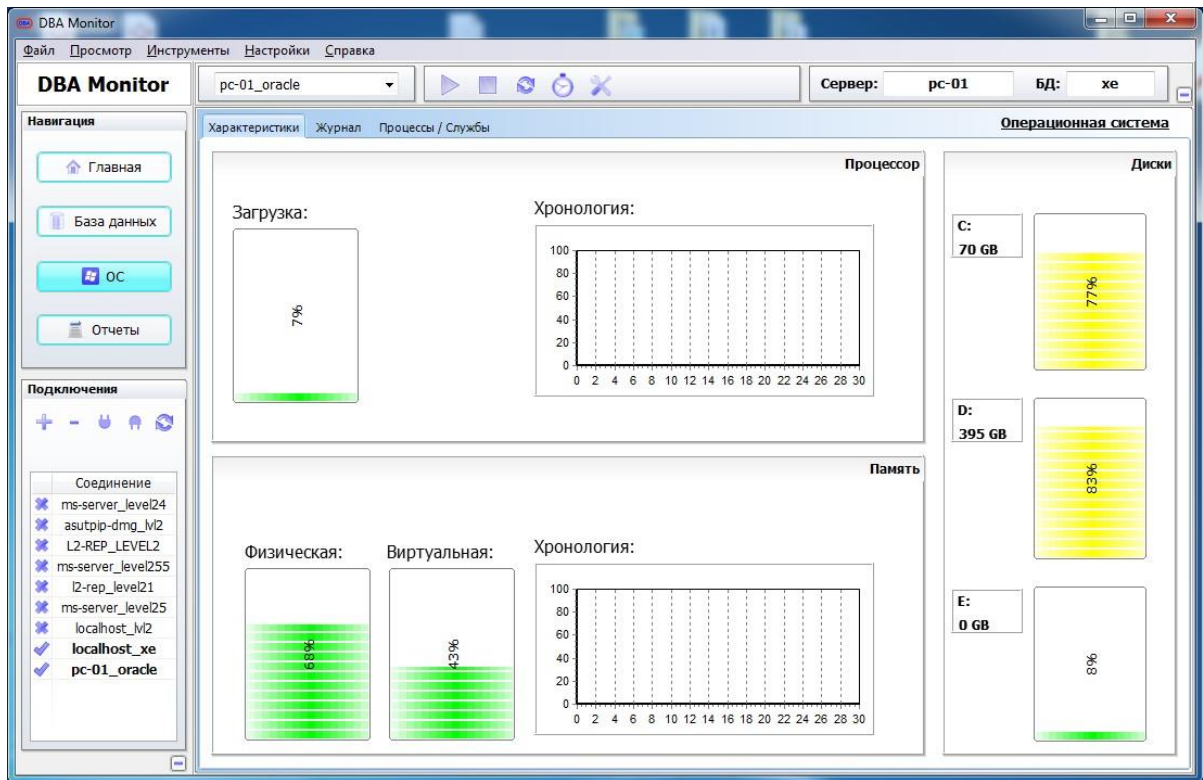


Рис. 30 Раздел ОС. Подраздел 1

Для того, чтобы периодически обновлять информацию на форме, необходимо выбрать время периодического обновления. (Рис. 31)

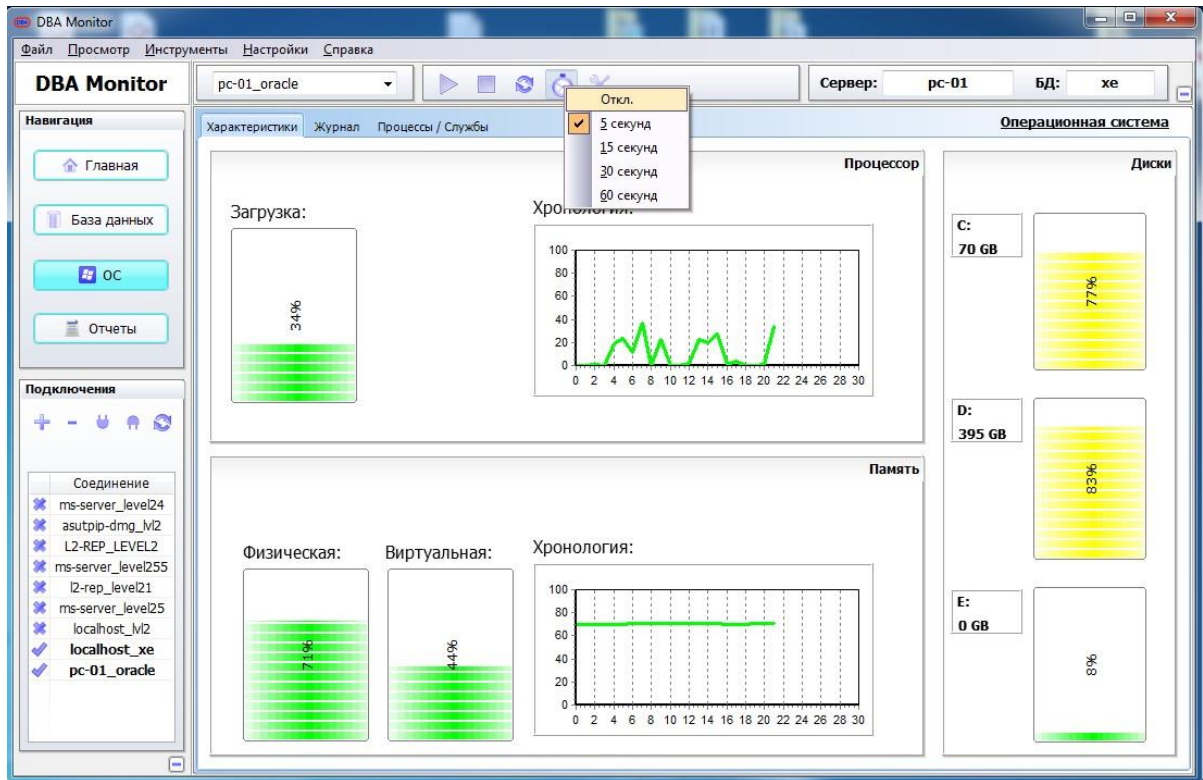


Рис. 31 Периодическое обновление формы

В подразделе «Журнал» отображаются последние сообщения из журнала сообщений Windows типа «Ошибка» либо «Предупреждение». (Рис. 32)

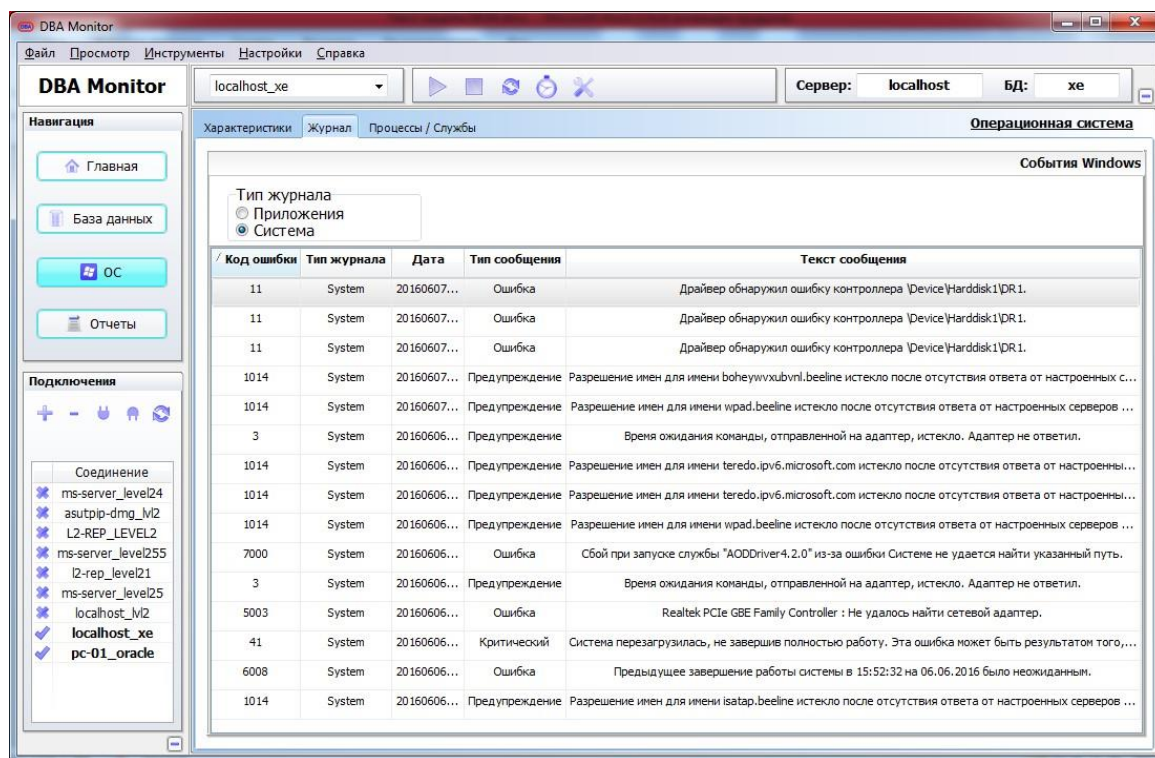


Рис. 32 Раздел ОС. Подраздел 2

Последний подраздел «Процессы/Службы» блока «ОС» отображает данные о запущенных процессах и службах. (Рис. 33)



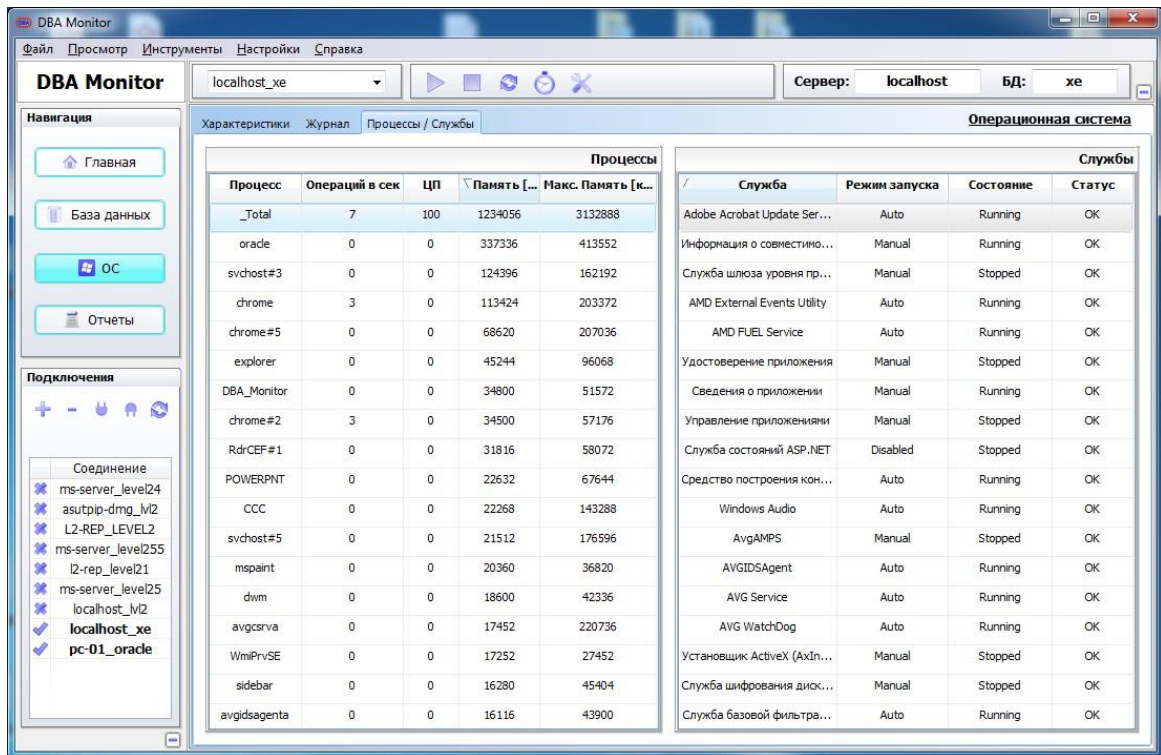
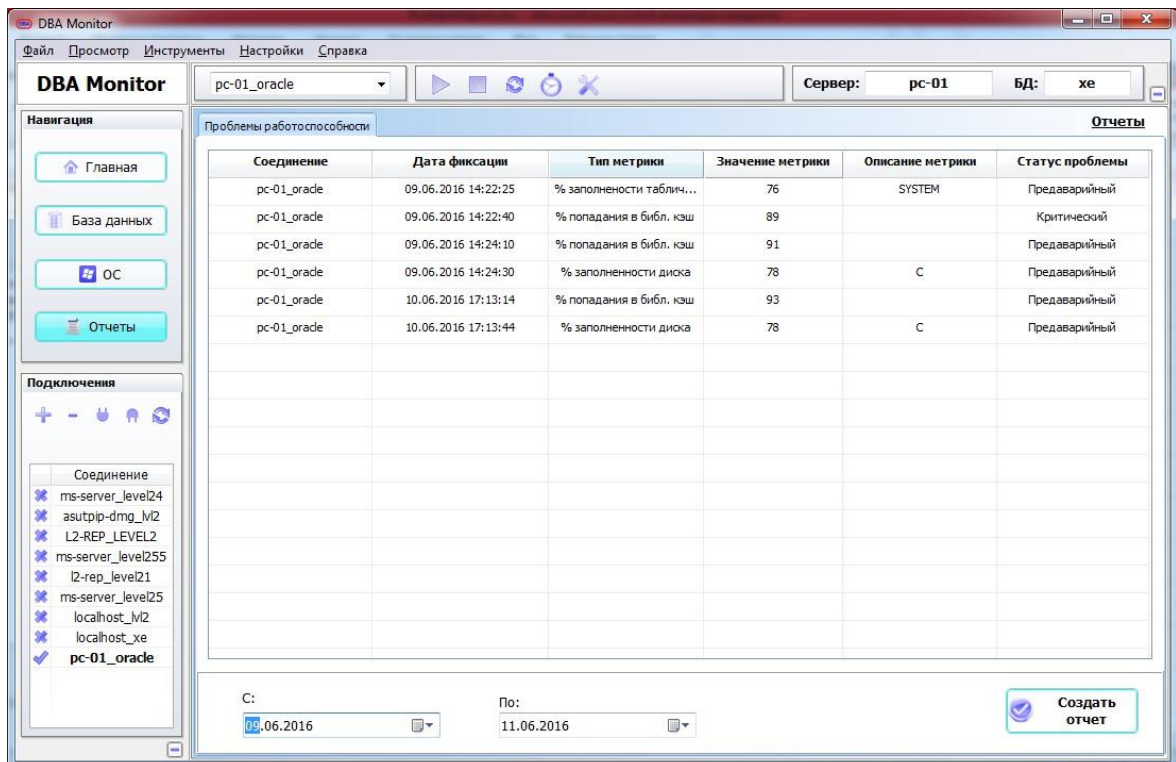


Рис. 33 Раздел ОС. Подраздел 3

В разделе «Отчеты» возможно просмотреть последние проблемы в работоспособности выбранного сервера базы данных, а также сформировать отчет в формате Excel. (Рис. 34)



## ЗАКЛЮЧЕНИЕ

Первым этапом выполнения работы было изучение структуры работы выбранной организации, также произведен анализ информационной структуры предприятия. В частности, рассмотрено назначение сервера базы данных предприятия и специалиста, отвечающего за работоспособность этих систем. Далее, были изучены задачи и обязанности администратора баз данных, ответственного за работоспособность данного сектора информационной структуры предприятия. После этого, произведен анализ необходимости в программном обеспечении для автоматизации деятельности администратора баз данных и рассмотрены программные продукты, существующие на рынке программного обеспечения, созданные для обозначенных целей. В завершении проведения анализа предметной области, выявлены функциональные требования и нефункциональные требования к программному обеспечению.

Следующим этапом выполнения работы было построение диаграмм IDEF0 и DFD, которые позволяют графически отобразить деятельность администратора баз данных предприятия. На основе проведенного анализа, для проектирования базы данных приложения, была построена ER – диаграмма и физическая модель базы данных программного обеспечения.

В последующей части работы, проанализированы и описаны инструментальные средства, используемые при разработке приложения. Далее описан алгоритм работы приложения и основные структурные модули. В конце работы, продемонстрирован пользовательский интерфейс приложения и описание работы с ним.

В процессе выполнения работы была выявлена недостаточная изученность выбранной области исследования. Для более детального освоения данной сферы были рассмотрены последующие источники.

На основе детального анализа структуры выбранного предприятия, а также изучения источников [3,11,12,27,32] исследован производственный цикл металлургического завода. На основе источников [3,27,32,33] и анализа предметной области рассмотрена информационная структура предприятия.

После изучения источников [1,2,9,13,16,18,25,26] проанализированы задачи и обязанности администратора баз данных, а также методология администрирования баз данных. На основе источников

[1,13,16,17,18,21,22,23,33,35] детально изучена структура серверов баз данных Oracle и особенности их администрирования. Тенденции в области администрирования баз данных Oracle рассмотрены с помощью источников [7,9,10,19,20,27]. Опыт создания автоматизированных систем мониторинга баз данных рассмотрен с помощью источников [7,8,15,19,24,28].

Возможности удаленного взаимодействия с серверными операционными системами Windows с помощью технологии WMI рассмотрены с использованием источников [4,29,31,33].

В процессе изучения источников [5,6,37] проанализированы методологии проектирования информационных систем.

Программное обеспечение находится в опытной эксплуатации отдела АСУТПиП предприятия. Цель выпускной работы достигнута в полном объеме. Все задачи, поставленные перед работой, выполнены.

Разработанное программное обеспечение не является абсолютно законченной версией. В перспективе, возможно добавление новых функций и модулей, что позволит автоматизировать иные функциональные обязанности специалиста.

## СПИСОК СОКРАЩЕНИЙ

АБД	Администратор баз данных
АРМ	Автоматизированное рабочее место
АПК	Агрегат Печь - Ковш
АСУ	Автоматизированная система управления
АСУТП	Автоматизированная система управления технологическим процессом
АСУТПиП	Автоматизированная система управления технологическим процессом и производством
БД	База данных
ДСП	Дуговая Сталеплавильная печь
МНЛЗ	Машина непрерывного литья заготовок
ОС	Операционная система
ПЦ	Прокатный цех
ПО	Программное обеспечение
СУБД	Система управления базой данных
ЭСПЦ	Электросталеплавильный цех
ЯП	Язык программирования

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Алапати С. Oracle Database 11g: руководство администратора баз данных – М.: Вильямс, 2010. – 1440 с.
2. Ассоциация предприятий компьютерных и информационных технологий. Профессиональные стандарты в области ИТ.  
[Электронный ресурс]:  
<http://www.apkit.ru/committees/education/meetings/standarts.php>
3. Бойко В.И., Смоляк В.А. Автоматизированные системы управления технологическими процессами в черной металлургии: учебное пособие. - Днепропетровск: ДГТУ, 1997. - 576 с.
4. Бочкарев В. Администрирование с помощью WMI. [Электронный ресурс]:  
<http://www.sysengineering.ru/administration/administrationusingwmi.aspx>
5. Гвоздева Т.В. Проектирование информационных систем. – М.:Высшее образование, 2009. – 512 с.
6. Димов Э.М., Диязитдинова А.Р., Качков Д.А. Проектирование информационных систем: учебное пособие. – Самара: ПГАТИ, 2004 – 78 с.
7. Дмитриенко П. В. Методика оценки эффективности систем мониторинга вычислительных ресурсов. // Компьютерные исследования и моделирование., - 2012, № 3 - С. 661–668.
8. ДьюМоолен Р. Секреты упреждающего мониторинга баз данных.  
[Электронный ресурс]: <http://www.oracloid.ru/index.php?t=243>
9. Жилин С. Обязанности администратора Oracle. [Электронный ресурс]:  
<http://alldba.ru/index.php/subd/oracle-database/administrirovanieoracle/obyazannosti>
10. Интернет – журнал «FORS» [Электронный ресурс]:  
<http://www.fors.ru/pressroom/journal>
11. Иван Литкевич. «vsluh.ru». Металлургический завод «Электросталь

- Тюмени». [Электронный ресурс]:  
<http://tumen.bezformata.ru/listnews/zavod-elektrostal-tyumeni/13037806/>
12. ИИС «Металлоснабжение и сбыт». Электросталь Тюмени  
[Электронный ресурс]: <http://www.metalinfo.ru/ru/news/77097>
  13. Кайт Т. Oracle для профессионалов. Архитектура, методики программирования и особенности версий 9i, 10g и 11g – М.: Вильямс. – 848 с.
  14. Каримов А.Ф. Двигатель базы данных "SQLite". [Электронный ресурс]:  
<http://www.codenet.ru/db/other/sqlite/>
  15. Клайн К. Мониторинг SQL Server. [Электронный ресурс]: <http://www.osp.ru/win2000/2006/07/3821302/>
  16. Луни К., Брила Б. Oracle Database 10g Настольная книга администратора баз данных – М: Лори, 2005. – 729 с.
  17. Маллинс К. С. Администрирование баз данных. Полное справочное руководство по методам и процедурам. М.: КУДИЦ - ОБРАЗ, 2003.
  18. Мальков А. Средства мониторинга работы СУБД и приложений.  
[Электронный ресурс]: <http://www.ntrlab.ru/?p=195>
  19. Марносов Е. С., Несбытнов А. Е., Саньков С. Г. Разработка автоматизированной системы мониторинга аппаратного и программного обеспечения серверов с целью сокращения времени реагирования на произошедшие критические ситуации. [Электронный ресурс]:  
[http://www.volpi.ru/files/publications/E\\_\\_S\\_\\_Marnosov\\_\\_A\\_\\_E\\_\\_Nesbytnov\\_\\_S\\_\\_G\\_\\_Sankov\\_Razrabotka\\_avtomatizirovannoj\\_sistemy\\_monito.pdf](http://www.volpi.ru/files/publications/E__S__Marnosov__A__E__Nesbytnov__S__G__Sankov_Razrabotka_avtomatizirovannoj_sistemy_monito.pdf)
  20. Миллсап К., Хольт Д. Oracle. Оптимизация производительности. – СПбЖ Символ-Плюс, 2006. – 464 с.
  21. Мовчан А. Поговорим про Oracle Database. [Электронный ресурс]:

- <http://www.nestor.minsk.by/kg/2007/24/kg72412.html>
22. Нанда А., Фейерштейн С. Oracle PL/SQL для администраторов баз данных –СПб: Символ-Плюс, 2008. - 496 с.
  23. Открытые системы. СУБД [Электронный ресурс]: <http://www.osp.ru/dbms/>
  24. Охтилев М.Ю., Соколов Б.В., Юсупов. Р.М. Теоретические и технологические основы концепции проактивного мониторинга и управления сложными объектами. // Известия ЮФУ. Технические науки., - 2015, №1.
  25. Персональный блог Алесандра Рындина. Ведущий консультант Oracle по хранилищам данных. [Электронный ресурс]: <http://www.oraclegis.com/blog/>
  26. Персональный блог Марка Ривкина. Начальник отдела технического консалтинга по серверным технологиям Oracle CIS. [Электронный ресурс]: <http://mrivkin.narod.ru/>
  27. Пирожников В.Е., Каблуковский А.Ф. Автоматизация контроля и управления электросталеплавильными установками. – М.: Металлургия, 1974. – 205 с.
  28. Полторацкий В. Е. Методы администрирования информационных ресурсов автоматизированных систем. // Известия ЮФУ. Технические науки., - 2003, №4.
  29. Попов А., Шикин Е. Администрирование Windows с помощью WMI и WMIС. – СПб.: БХВ-Петербург, 2004. – 752 с.
  30. Ривкин М. Oracle Enterprise manager и его пакеты. [Электронный ресурс]: [http://mrivkin.narod.ru/Publ/OEM\\_PACK.htm](http://mrivkin.narod.ru/Publ/OEM_PACK.htm)
  31. Свободная энциклопедия «Википедия». WMI. [Электронный ресурс]: <https://ru.wikipedia.org/wiki/WMI>

32. Смирнов А.Н., Сафонов В.М., Дорохова Л.В., Цупрун А.Ю.  
Металлургические мини - заводы - Донецк: Норд-Пресс, 2005. – 469 с.
33. Соснин О.М. Основы автоматизации технологических процессов и производств. – М.:Издательский центр «Академия»,2007. – 240 с.
34. Терехов А. А вы знакомы с WMI? [Электронный ресурс]: [http://teran.karelia.pro/articles/item\\_4413.html](http://teran.karelia.pro/articles/item_4413.html)
35. Фафурин В. А., Терюшов И. Н. Автоматизация технологических процессов и производств – Казань: Издательство Казанского государственного технологического университета, 2008. – 552 с.
36. Oracle Database Online Documentation 10g Release 2 (10.2).  
Administration. [Электронный ресурс]:  
[https://docs.oracle.com/cd/B19306\\_01/nav/portal\\_3.htm](https://docs.oracle.com/cd/B19306_01/nav/portal_3.htm)
37. Oracle Database Online Documentation 11g Release 2 (11.2). Database Administration. [Электронный ресурс]:  
[https://docs.oracle.com/cd/E11882\\_01/nav/portal\\_4.htm](https://docs.oracle.com/cd/E11882_01/nav/portal_4.htm)
38. Market Journal. Структурное проектирование. [Электронный ресурс]:  
<http://www.market-journal.com/metodyjekonomiki/38.html>

## СПИСОК ТАБЛИЦ И ИЛЛЮСТРАЦИЙ

Таблицы:

Таблица 1 .....	31
Таблица 2 .....	51
Таблица 3 .....	52
Таблица 4 .....	52



Таблица 5 .....		
53		
Таблица 6 .....		
53		
Таблица 7 .....		
53	Таблица	8
.....		54

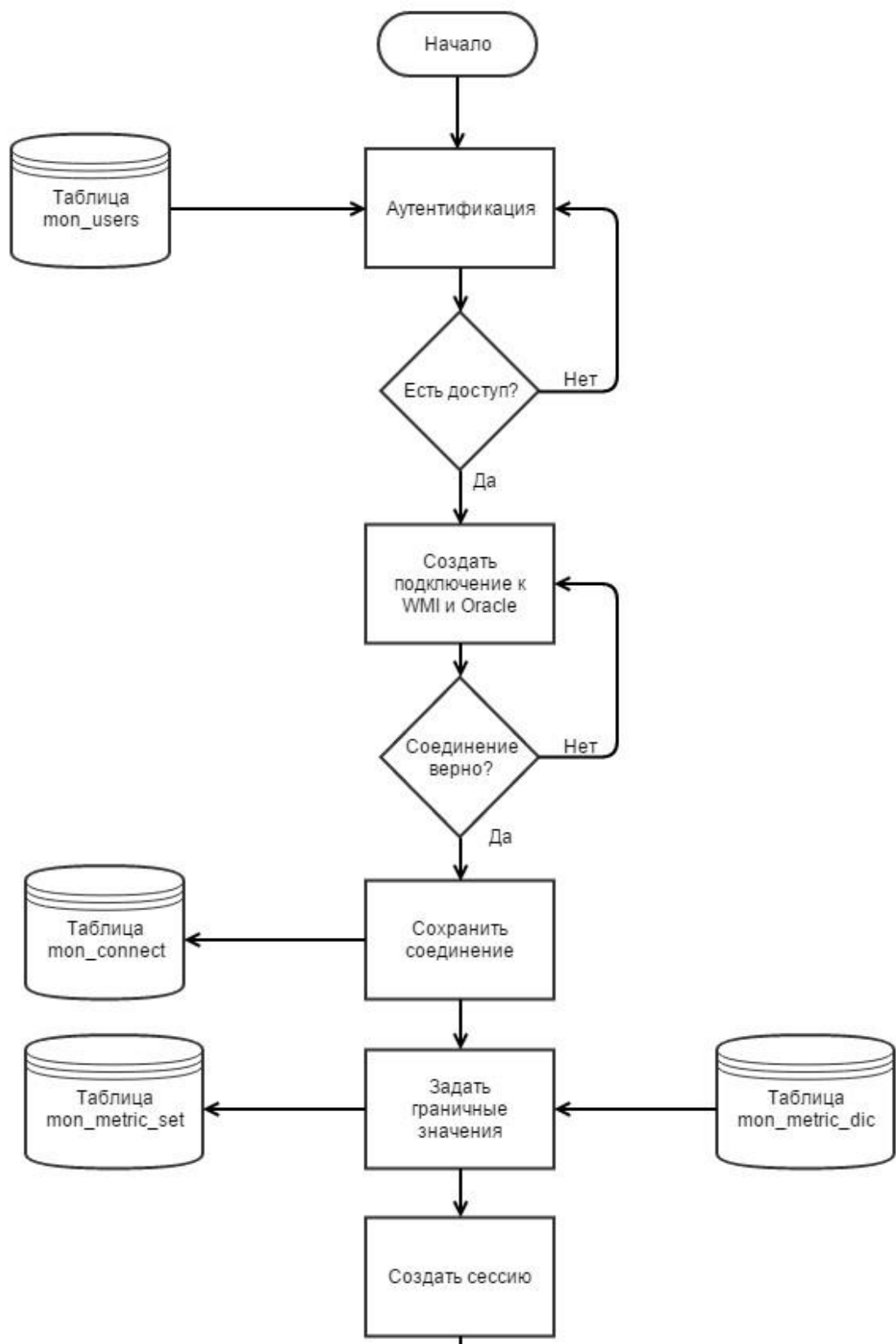
Иллюстрации:

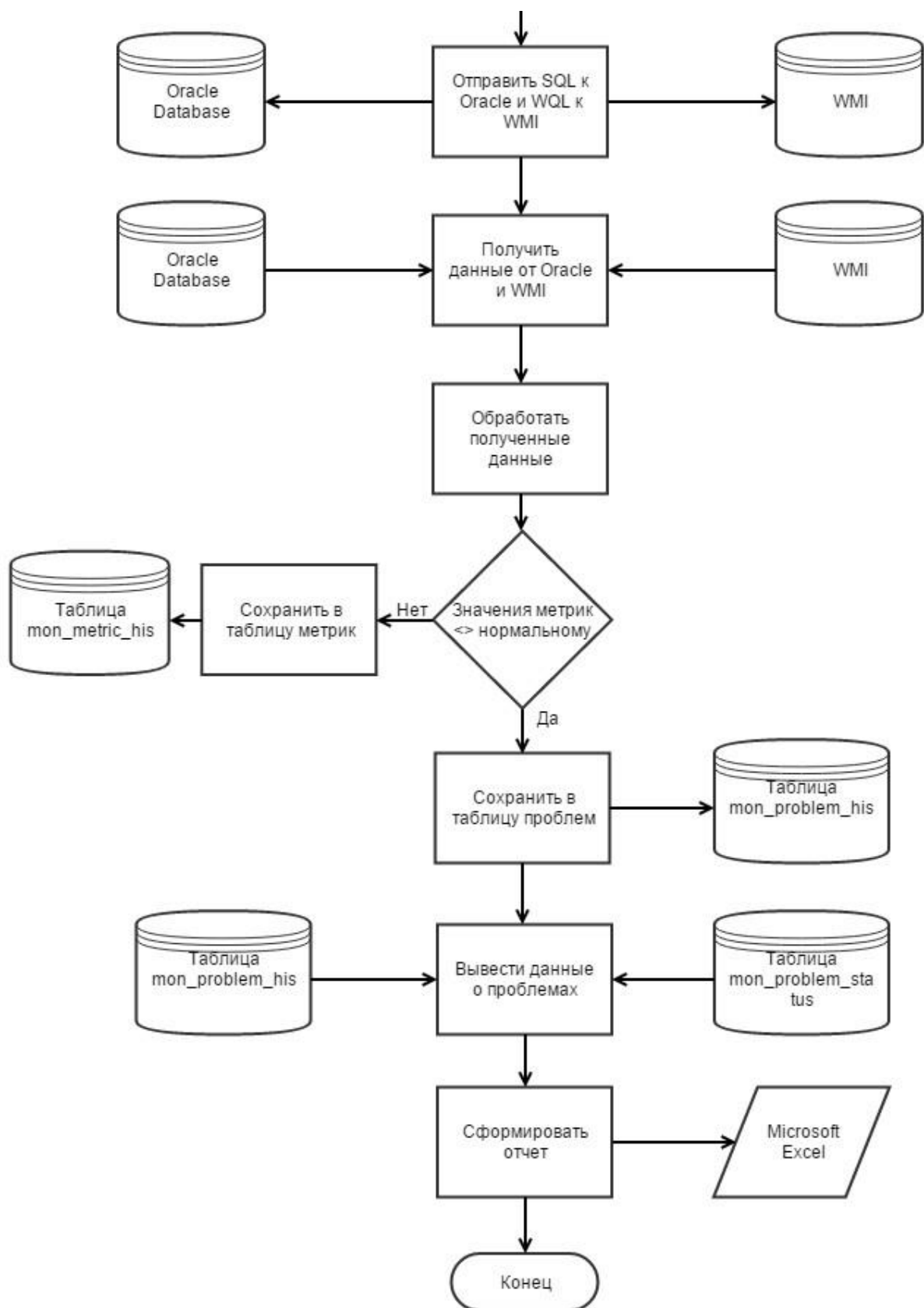
Рис. 1 Схема производственного цикла предприятия.....	11
Рис. 2 Пятиуровневая информационная структура предприятия.....	12
Рис. 3 Схема информационной системы предприятия .....	16
Рис. 4 IDEF0. Обеспечение работы базы данных .....	42
Рис. 5 IDEF0. Декомпозиция функции "Обеспечение работы БД" .....	43
Рис. 6 IDEF0. Обеспечение доступности.....	44
Рис. 7 IDEF0. Обеспечение безопасности БД .....	46
Рис. 8 IDEF0. Обеспечение производительности БД.....	47
Рис. 9 IDEF0. Обработка аварийных ситуаций.....	48
Рис. 10 DFD Работа базы данных .....	50
Рис. 11 ER диаграмма .....	52
Рис. 12 Физическая модель базы данных .....	55
Рис. 13 Схема взаимодействия модулей ПО .....	65
Рис. 14 Главная страница приложения .....	68
Рис. 15 Аутентификация пользователя.....	69
Рис. 16 Главная страница. Нажатие кнопки "Запуск" .....	70
Рис. 17 Создание соединения. Шаг 1 .....	70
Рис. 18 Сообщение об ошибке.....	71
Рис. 19 Сообщение о выполнении операции.....	71
Рис. 20 Создание соединения. Шаг 2 .....	72

Рис. 21 Создание соединения. Шаг 3 .....	73
Рис. 22 Конфигурация метрик .....	73
Рис. 23 Раздел "База данных". Подраздел 1 .....	74
Рис. 24 Раздел "База данных". Подраздел 2 .....	75
Рис. 25 Раздел "База данных". Подраздел 3 .....	76
Рис. 26 Раздел "База данных". Подраздел 4 .....	76
Рис. 27 Раздел "База данных". Подраздел 5 .....	77
Рис. 28 Раздел "База данных". Подраздел 6 .....	78
Рис. 29 Раздел "База данных". Подраздел 7 .....	78
Рис. 30 Раздел ОС. Подраздел 1 .....	79
Рис. 31 Периодическое обновление формы .....	80
Рис. 32 Раздел ОС. Подраздел 2 .....	80
Рис. 33 Раздел ОС. Подраздел 3 .....	81
Рис. 34 Раздел Отчеты. Подраздел 2 .....	82

## **ПРИЛОЖЕНИЯ**

### **Приложение 1. Алгоритм работы приложения**





## Приложение 2. Создание нового соединения

```

procedure TConnectForm.ConCreateButtonClick(Sender: TObject);
var  conname:string; begin
    screen.Cursor:=crhourglass;
    conname:=connameedit.Text;

    // проверяем чтобы имя соединения было не пустым
if (conname<>") then  begin
    servname:=servnameedit.Text;
    dbname:=dbnameedit.Text;
    dblogin:=dbloginedit.Text;
    dbpass:=dbpassedit.Text;
    oslogin:=osloginedit.text;   ospas:=ospasedit.text;

    // шифруем пароль от БД и от ОС
    dbpass:=dm1.CryptString(dbpass);   ospas:=dm1.CryptString(ospas);

    // передаем параметры подключения в запрос на вставку записи
    condm.zq_ins_connect.ParamByName('conname').AsString:=conname;
    condm.zq_ins_connect.ParamByName('servname').AsString:=servname;
    condm.zq_ins_connect.ParamByName('dbname').AsString:=dbname;
    condm.zq_ins_connect.ParamByName('dblogin').AsString:=dblogin;
    condm.zq_ins_connect.ParamByName('dbpass').AsString:=dbpass;
    condm.zq_ins_connect.ParamByName('oslogin').AsString:=oslogin;
    condm.zq_ins_connect.ParamByName('ospas').AsString:=ospas;   try
        condm.zq_ins_connect.ExecSQL;
    msgform.ShowMsg('Выполнено успешно.',1);   connectform.Close;

```

```

        condm.zq_sel_name_con.refresh; //обновляем данные в связи с новой
записью  except  on e:exception do

        msgform.ShowMsg('Соединение не создано.'+#10+'Ошибка:
'+e.Message,0);

end; end else

        msgform.ShowMsg('Имя соединения не может быть пустым.',0);

screen.Cursor:=crdefault; end;

```

### **Приложение 3. Взаимодействие с интерфейсом WMI**

```

procedure TMainForm.OSPage3Show (Sender: TObject); var

        objectSet  :  ISWbemObjectSet;

obj : ISWbemObject;

        i,q,j:integer;

query:string;

        properties:      ISWbemPropertySet;

begin

        // Получение данных по процессам Windows

        // текст запроса  query:='select Name,IODataOperationsPerSec,
PercentProcessorTime,'+

        'WorkingSetPrivate,WorkingSetPeak from
Win32_PerfFormattedData_PerfProc_Process';

```

```

Try
    // выполнение запроса
objectSet:=dm1.services.ExecQuery(query,'WQL',wbemFlagReturnImmediately
and wbemFlagForwardOnly,nil);

    for i := 0 to objectSet.count - 1 do
begin

    obj          :=          objectSet.ItemIndex(i);
properties:=obj.Properties_;

    // заполнение таблицы данными

    ProcessGrid.Cells[0,i+1] := obj.Properties_.Item('Name',0).get_value;

    ProcessGrid.Cells[1,i+1] :=
obj.Properties_.Item('IODataOperationsPerSec',0).get_value;

    ProcessGrid.Cells[2,i+1] :=
obj.Properties_.Item('PercentProcessorTime',0).get_value;

    ProcessGrid.Cells[3,i+1] :=
floattostr(obj.Properties_.Item('WorkingSetPrivate',0).get_value/1024);

    ProcessGrid.Cells[4,i+1] :=
floattostr(obj.Properties_.Item('WorkingSetPeak',0).get_value/1024);

    ProcessGrid.RowCount:=i+1;

    //сохранение в таблице значений метрик % от процессора
dm1.zq_ins_metric.ParamByName('disc').asString:=obj.Properties_.Item('Name',0)
.get_value;

```

```

dm1.zq_ins_metric.ParamByName('met_id').asinteger:=20;
dm1.zq_ins_metric.ParamByName('ts').AsDateTime:=now;

dm1.zq_ins_metric.ParamByName('value').AsInteger:=obj.Properties_.Item('PercentProcessorTime',0).get_value;    dm1.zq_ins_metric.execsqli;
dm1.zq_sel_max_mid.open;

    m_id:=dm1.zq_sel_max_mid.fields[0].asinteger; //id значения метрики

    //запрос к границам метрики
dm1.zq_sel_m_border.parambyname('met_id').asinteger:=20;
dm1.zq_sel_m_border.parambyname('con_id').asinteger:=connect_id;
dm1.zq_sel_m_border.execsqli;    low:=dm1.zq_sel_m_border.fields[0].asinteger;
up:=dm1.zq_sel_m_border.fields[1].asinteger;

    if obj.Properties_.Item('PercentProcessorTime',0).get_value>=low    // >
границы крит. состояния
    then
begin
    dm1.zq_ins_problem.parambyname('stat_id').asinteger:=2;
dm1.zq_ins_problem.ParamByName('met_id').AsInteger:=m_id;
dm1.zq_ins_problem.ExecSQL;    end

    else if obj.Properties_.Item('PercentProcessorTime',0).get_value>=up    // >
нормального состояния
    then
begin

```



```

    dm1.zq_ins_problem.parambyname('stat_id').asinteger:=1;
dm1.zq_ins_problem.ParamByName('met_id').AsInteger:=m_id;
dm1.zq_ins_problem.ExecSQL;    end;

//      метрика      Потребляемой      памяти      процессом
dm1.zq_ins_metric.ParamByName('disc').asString:=obj.Properties_.Item('Name',0)
.get_value;

    dm1.zq_ins_metric.ParamByName('met_id').asinteger:=21;
dm1.zq_ins_metric.ParamByName('ts').AsDateTime:=now;
dm1.zq_ins_metric.ParamByName('value').AsInteger:=obj.Properties_.Item('Work
ingSetPrivate',0).get_value /1024;    dm1.zq_ins_metric.execsqli;
dm1.zq_sel_max_mid.open;

    m_id:=dm1.zq_sel_max_mid.fields[0].asinteger; //id значения метрики
dm1.zq_sel_m_border.parambyname('met_id').asinteger:=21;
dm1.zq_sel_m_border.parambyname('con_id').asinteger:=connect_id;
dm1.zq_sel_m_border.execsqli;

    low:=dm1.zq_sel_m_border.fields[0].asinteger;
up:=dm1.zq_sel_m_border.fields[1].asinteger;

    if obj.Properties_.Item('WorkingSetPrivate',0).get_value>=low    // > границы
крит. состояния
    then
begin
    dm1.zq_ins_problem.parambyname('stat_id').asinteger:=2;
dm1.zq_ins_problem.ParamByName('met_id').AsInteger:=m_id;
dm1.zq_ins_problem.ExecSQL;    end

```

```

        else if obj.Properties_.Item('WorkingSetPrivate',0).get_value >= up      // >
нормального состояния
        then
begin
        dm1.zq_ins_problem.parambyname('stat_id').asinteger:=1;
dm1.zq_ins_problem.ParamByName('met_id').AsInteger:=m_id;
dm1.zq_ins_problem.ExecSQL;
        end; end;
except      on
e:exception do
        msgform.ShowMsg('Не выполнено.'+#10+'Ошибка: '+e.Message,0);
end;

// Получение данных по службам Windows try
query:='select Caption, StartMode, State, Status from Win32_Service';
objectSet:=dm1.services.ExecQuery(query,'WQL',wbemFlagReturnImmediately
and wbemFlagForwardOnly,nil);
        for i := 0 to objectSet.count - 1 do
begin
        obj := objectSet.ItemIndex(i);
        ServiceGrid.Cells[0,i+1] := obj.Properties_.Item('Caption',0).get_value;
        ServiceGrid.Cells[1,i+1] := obj.Properties_.Item('StartMode',0).get_value;
        ServiceGrid.Cells[2,i+1] := obj.Properties_.Item('State',0).get_value;
ServiceGrid.Cells[3,i+1] := obj.Properties_.Item('Status',0).get_value; end;
        ServiceGrid.RowCount:=i+1;
Except
        on e:exception do

```

```
msgform.ShowMsg('Не выполнено.'+#10+'Ошибка: '+e.Message,0);  
end; end;
```

#### Приложение 4. Взаимодействие с Oracle

```
procedure TMainForm.DBPage3Show(Sender: TObject); begin  
    if dm1.OraSession1.Connected=true then  
begin    try  
        // запрос к информации по tablespace и datafile    if  
dbdm.oq_files_tblspc.Active=true then dbdm.oq_files_tblspc.refresh  
else dbdm.oq_files_tblspc.open;  
        if dbdm.OQ_files_datafile.Active=true then dbdm.OQ_files_datafile.refresh  
else dbdm.OQ_files_datafile.Open;  
  
        dbdm.OQ_files_tblspc.First;  
        while not dmdm.oq_files_tblspc.Eof do // перебор tablespace  
begin  
            //сохранение в таблице значений метрик % заполненности tablespace  
dm1.zq_ins_metric.ParamByName('met_id').asinteger:=14;  
dm1.zq_ins_metric.ParamByName('ts').AsDateTime:=now;  
  
dm1.zq_ins_metric.ParamByName('disc').Asstring:=dbdm.oq_files_tblspc.Fields[0  
].asstring;  
  
dm1.zq_ins_metric.ParamByName('value').AsInteger:=dbdm.oq_files_tblspc.Field  
s[6].asinteger;  
  
        dm1.zq_ins_metric.execsql;  
  
        dm1.zq_sel_max_mid.open;
```

```

m_id:=dm1.zq_sel_max_mid.fields[0].asinteger; //id значения метрики

//запрос к границам метрики
dm1.zq_sel_m_border.parambyname('met_id').asinteger:=14;
dm1.zq_sel_m_border.parambyname('con_id').asinteger:=connect_id;
dm1.zq_sel_m_border.execsql;

low:=dm1.zq_sel_m_border.fields[0].asinteger;
up:=dm1.zq_sel_m_border.fields[1].asinteger;

if dbdm.oq_files_tblspc.Fields[0].asinteger>=low // > границы крит.
состояния
then
begin
dm1.zq_ins_problem.parambyname('stat_id').asinteger:=2;
dm1.zq_ins_problem.ParamByName('met_id').AsInteger:=m_id;
dm1.zq_ins_problem.ExecSQL;
dbtablespacegrid.fields[7].asinteter:=2; end

else if dbdm.oq_files_tblspc.Fields[0].asinteger>=up // > нормального
состояния then begin
dm1.zq_ins_problem.parambyname('stat_id').asinteger:=1;
dm1.zq_ins_problem.ParamByName('met_id').AsInteger:=m_id;
dm1.zq_ins_problem.ExecSQL; dbtablespacegrid.fields[7].asinteter:=1;
end;

dmdm.oq_files_tblspc.next;
end;

```

```

        except          on
e:exception do
    msgform.ShowMsg('Не выполнено.'+#10+'Ошибка: '+e.Message,0);
end;    end;    end;
end;

```

### **Приложение 5. SQL запрос к базе данных Oracle**

```

select
    regexp_substr(dat.file_name,'[^\]+$') as file_name, -- имя файла
    dat.tablespace_name, -- наименование табличного пространства
    dat.status, -- статус доступности  stat.phyrds, -- кол-во чтений
    stat.phywrts, -- кол-во записей
    dat.bytes/1024/1024 as total_size, -- размер(мб)
    round(sum(free.bytes)/1024/1024,1) as free_size, -- свободное пространство
    round((dat.bytes-sum(free.bytes))/dat.bytes*100,1) as proc -- % своб. пр-ва from
    dba_free_space free,
    dba_data_files dat,  v$filestat
    stat
where dat.file_id=free.file_id(+) and dat.file_id=stat.file#(+)
group by  free.file_id, dat.tablespace_name,
    dat.file_name,  dat.bytes,  dat.status,  dat.file_id,
    stat.PHYRDS,
    stat.PHYWRTS
order by dat.tablespace_name, dat.file_id

```

### **Приложение 6. Сохранение отчета в формате Excel**

```

procedure TMainForm.CreateReportButtonClick(Sender: TObject);
var s:string; i,j:integer;
    TemplateDir:string;
    FileName:string; begin
    try
        TemplateDir:=extractfilepath(paramstr(0));
    dm1.ExcelApplication1.Connect;

        //Подключение к шаблону

    dm1.ExcelWorkbook1.ConnectTo(dm1.ExcelApplication1.Workbooks.Add(TemplateDir+'\Templates\ProblemReport.xlsx',0));

        dm1.ExcelWorksheet1.ConnectTo(dm1.ExcelWorkbook1.ActiveSheet as
        _worksheet);

        j:=0;

        // выгрузка данных в ячейки Excel
    while not dm1.zq_sel_problem.Eof do
    begin
        for i := 0 to dm1.zq_sel_problem.Fields.Count-1 do

    dm1.ExcelWorksheet1.Cells.Item[j+1,i+1].value:=dm1.zq_sel_problem.Fields[i].AsVariant;

        dm1.zq_sel_problem.Next;
    inc(j); end;

        // Сохранение отчета

```

```
FileName:='Отчет по работоспособности сервера.xlsx';  
dm1.ExcelApplication1.DisplayAlerts[LOCALE_USER_DEFAULT]:=false;  
dm1.ExcelWorkbook1.SaveAs(FileName,          xlAddIn,  EmptyParam,  
EmptyParam,  EmptyParam,EmptyParam,  xlNoChange,  EmptyParam,  
EmptyParam, EmptyParam, EmptyParam, 0);  
dm1.ExcelApplication1.DisplayAlerts[LOCALE_USER_DEFAULT]:=true;  
dm1.ExcelApplication1.Disconnect; except  showmessage('Ошибка  
выполнения. Сохранение в Excel не выполнено.');
```