Оглавление

[Введение 2](#_Toc133230014)

[1. Проектирование и реализация модели данных 5](#_Toc133230015)

[**1.1 Системный анализ предметной области и анализ требований** 5](#_Toc133230016)

[**1.1.1** **Анализ деятельности организации** 5](#_Toc133230017)

[**1.1.2** **Категории пользователей** 6](#_Toc133230018)

[**1.1.3** **Анализ требований** 7](#_Toc133230019)

[**1.1.4** **Выбор и обоснование проектных решений по видам обеспечения** 9](#_Toc133230020)

[**1.1.5** **Входная и выходная информация** 10](#_Toc133230021)

[**1.2** **Проектирование базы данных** 10](#_Toc133230022)

[**1.2.1** **Существующие в организации бизнес-процессы и процессы обработки данных исследуемого объекта** 10](#_Toc133230023)

[**1.2.2** **Инфологическое моделирование** 11](#_Toc133230024)

[**1.2.3** **Сущности** 11](#_Toc133230025)

[**1.2.4** **Взаимосвязи сущностей** 14](#_Toc133230026)

[**1.2.5** **Разработка инфологической модели** 15](#_Toc133230027)

[**1.2.6** **Даталогическое проектирование** 15](#_Toc133230028)

[**1.2.7** **Отношение и атрибуты** 16](#_Toc133230029)

[**1.2.8** **Ключевые поля и индексы** 17](#_Toc133230030)

[**1.2.9** **Даталогическая модель** 17](#_Toc133230031)

[**1.3** **Реализация** 18](#_Toc133230032)

[**1.3.1** **Особенности реализации** 18](#_Toc133230033)

[**1.3.2** **Разработка алгоритмов и кода хранимых процедур и триггеров** 19](#_Toc133230034)

[**1.3.3** **Тестирование и отладка хранимых процедур** 20](#_Toc133230035)

[**1.3.4** **Тестирование и отладка триггеров** 25](#_Toc133230036)

[**1.3.5** **Разработка прототипа пользовательского интерфейса** 29](#_Toc133230037)

[Выводы по первой главе 30](#_Toc133230038)

[Приложение к 1 главе 34](#_Toc133230039)

# Введение

Актуальность работы. Данная курсовая работа посвящена документообороту между заказчиком и испытательной лабораторией по анализу воды.

В современном мире состояние окружающего нас мира вызывает беспокойство, наверное, у всех жителей планеты. Существуют законы по охране окружающей среды, поднимаются темы огромного количества отходов от человеческой деятельности – особенно от заводов и фабрик, число которых с каждым годом увеличивается. Вместе с тем ученые говорят о том, что запасов питьевой воды, не загрязненной токсичными отходами, постоянно уменьшается. Поэтому сейчас у испытательных лабораторий, которые проводят анализы воды, очень много работы.

Именно из-за большой загрузки лабораторий данное исследование актуально. Гораздо удобнее и надежнее хранить документацию не только в бумажном виде, но и в структурированной безопасной системе, надежно защищенной от сбоев и потерь важных данных. Более быстрая обработка данных, их структурированность и легкость в поддержании порядка в больших объемах информации в документах поможет лабораториям повысить производительность и увеличить количество исследований, которые могут проводиться в лаборатории одновременно.

Новизна исследования заключается в разработке нового метода ведения документации между заказчиком и лабораторией с помощью базы данных, разработанной специально для этой цели с учетом особенностей испытательных лабораторий.

**Целью курсовой** **работы** является проектирование, разработка и защита базы данных для хранения и обработки информации о документообороте между заказчиком и лабораторией для самой лаборатории.

**Задачи** работы:

1. Изучить тонкости документооборота между заказчиком и лабораторией со стороны лаборатории;
2. Найти способ хранить информацию о заказчиках;
3. Найти способ хранить информацию непосредственно о документах, которые составляются в ходе работы лаборатории;
4. Создать возможность изменения, внесения или удаления данных;
5. Сделать систему надежной и удобной для пользователей.

**Объектом данного исследования** является информационная система документообороте между заказчиком и лабораторией.

**Предметом исследования** выступает процесс проектирования и реализации защищенной базы данных информационной системы документооборота между заказчиком и лабораторией.

**Методы исследования**: ознакомление с документацией и государственными стандартами для испытательных лабораторий (в частности по анализу воды), опрос сотрудников, работающих в лаборатории непосредственно с заказчиками и документацией, связанной с проведением анализов.

**Практическая значимость** данной работы в том, что спроектированную и разработанную базу данных можно внедрить в деятельность какой-либо испытательной лаборатории или разработать систему немного другого вида на основе уже созданной в данной работе системы.

Для достижения поставленной цели использовались методы анализа и синтеза. Информационной базой исследования являются публикации в сети Интернет, посвященные вопросу изучения разработки и защиты базы данных ХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХ.

В структуру курсовой работы входят две главы.

В первой главе анализируется предметная область, категории пользователей, для которых создается система, а также определяются их возможности в изменении состояния базы данных. Также в этой части выделяются функциональные и нефункциональные требования, бизнес-требования, выбирается наиболее подходящее программное обеспечение, анализируются входные и выходные данные системы.

Описываются происходящие в предприятии бизнес-процессы, также создается концептуальная (инфологическая) модель предметной области, и даталогическая модель. Также подробно описываются входящие в эти модель элементы: сущности, отношения, их атрибуты, связи между ними, ключи и индексы.

Описывается реализация базы данных: особенности реализации, разработка алгоритмов и кода хранимых процедур и триггеров, их тестирование и отладка. Также в этой части будет разработан прототип пользовательского интерфейса для созданной базы данных.

В конце главы формулируются выводы.

Вторая глава посвящена разработке системы защиты базы данных от несанкционированного доступа путём создания регистрационных имён, имён пользователей, ролей, применения разрешений и шифрования данных. В конце главы формулируются выводы.

Работа в ХХХХХХХХХХХХХХХХХХХ связана с ХХХХХХХХХХХХХХХХХ. Создание базы данных документооборота между заказчиком и лабораторией позволит автоматизировать и максимально устранить случайные ошибки и недостатки деятельности компании, возникающие без этой системы, а также создать её защиту путём использования регистрационных имен, применения разрешений и выполнения шифрования. Осуществление данной работы также позволит реализовать:

– уменьшение времени выполнения каждой функции системы обработки информации;

– автоматическое создание документации по предложениям и заявкам;

– простой и быстрый поиск информации о ХХХХХХХХХХХХХХХ;

– доступность информации по каждому сотруднику компании, лаборатории;

– обеспечение защиты базы данных от несанкционированного доступа.

# Проектирование и реализация модели данных

## **1.1 Системный анализ предметной области и анализ требований**

В главе 1 будет предоставлен системный анализ предметной области и анализ требований. Будут рассмотрены анализ деятельности организации, категории пользователей, для которых будет создана база данных, анализ требований, выбор и обоснование решений по видам обеспечения, а также входные и выходные данные.

### **Анализ деятельности организации**

Необходимо создать базу данных для документооборота в лаборатории по анализу воды, для чего нужно детально изучить предметную область и соответствующие бизнес-процессы. Изучение происходит с помощью опроса сотрудников, знакомства с документацией и государственными стандартами, регулирующими работу организации.

Лаборатория занимается отбором и определением количественных химических характеристик (далее по тексту КХА) проб воды по определенным показателям, внесенным в область аккредитации испытательной лаборатории. Работы производятся по договорам, заключенным между заказчиком и лабораторией.

Договор предусматривает отбор определенного количества проб воды с составлением актов отбора проб, их анализ по показателям, оговоренным в договоре, с выдачей протокола испытаний.

Отбор проб сопровождается оформлением акта отбора проб с фиксацией необходимых данных. Далее проба поступает в лабораторию и регистрируется с присвоением индивидуального (уникального) номера, которая сопровождает пробу весь срок нахождения в лаборатории до момента утилизации. Зарегистрированная проба передается исполнителям для проведения испытаний. По проведенным испытаниям, после необходимых расчетов, оформляется протокол КХА проб воды.

Бизнес-процессами лаборатории являются: составление договора с заказчиком, заполнение акта отбора проб, анализ отобранных проб и заполнение протокола испытаний.

В процессе работы лаборатории с заказчиками происходит хранение и обработка данных о заказчиках (наименование предприятия, адрес, банковские реквизиты); и о документации лаборатории, в которую входят:

- договоры;

- акты отбора проб (содержат номер акта отбора, дату отбора, адрес отбора);

- протоколы испытаний (содержат номер протокола (дублирует номер акта отбора), дату составления протокола, результаты соответствующих показателей). [6-9]

В результате проведенного исследования выдвигаются следующие требования к системе:

1) Хранение информации обо всех заказчиках.

2) Хранение информации о показателях, внесенных в область аккредитации испытательной лаборатории.

3) Хранение информации о методиках, с которыми работает лаборатория.

4) Хранение информации о заключенных договорах.

5) Хранение информации о составленных актах отбора проб.

6) Хранение информации о протоколах испытаний.

Система должна отвечать на следующие вопросы:

1) Работала ли лаборатория с этим заказчиком раньше?

2) По каким показателям лаборатория может проводить анализы? Какова цена каждого показателя?

3) Какие методики используются в этой лаборатории для анализов?

4) Когда и где происходил отбор проб для определенного анализа?

5) Каковы результаты произведенных испытаний по определенному акту отбора?

Ограничения предметной области:

В данной модели отражен только документооборот между лабораторией и заказчиком. В модель не включены учет расходных материалов, данные о сотрудниках лаборатории, данные о приборах (названия, количество, гарантия, срок использования). Также в эту модель не включена работа бухгалтерии, которая заключается в выписывании счетов после составления договора, контролем за их оплатой, ведением учета доходов и расходов.

Эта модель предполагает, что договор составляется на одно обращение заказчика, т.е. если заказчику понадобится снова обратиться в лабораторию для повторного проведения тех же анализов, ему придется составлять новый договор с лабораторией.

### **Категории пользователей**

Данная база данных предусмотрена для пользователей, работающих в лаборатории, таких, как:

- начальник лаборатории – проводит переговоры с заказчиками и в процессе составляет с ними договор, контролирует составление акта отбора и протокола исследований, также может непосредственно составлять акт и протокол самостоятельно;

- сотрудники лаборатории – они выполняют отбор проб и их анализ, в процессе чего оформляют отчетные документы (акты отбора проб и протокол исследования соответственно).

Заказчики не имеют никакого доступа к базе данных лаборатории, т.к. в базе данных содержится конфиденциальная информация, которой могут пользоваться только сотрудники лаборатории. Данные о заказчиках вносятся в базу данных начальником лаборатории при оформлении первого договора и остаются в базе на случай повторного обращения заказчика.

Начальник лаборатории является главой всего предприятия, поэтому именно он осуществляет администрирование базы данных.

Сотрудники лаборатории не имеют доступа ко всей базе данных. Они могут просматривать и вносить данные, необходимые для работы самой лаборатории как части предприятия, т.е. проведения анализов. Обновлением данных отдела лаборатории при необходимости занимается непосредственно начальник лаборатории.

Доступ к удалению данных из базы данных имеет только начальник лаборатории из соображений безопасности (во избежание удаления необходимой информации).

При этом не весь персонал лаборатории имеет доступ к этой базе данных. Например, сотрудник отдела кадров, в обязанности которого не входит работа с данными из этой базы данных, не будет иметь никакого доступа к ней. Также не будет доступа и у лаборантов, в обязанности которых не входит работа с документацией любого рода (лаборанты готовят рабочее место и необходимое оборудование для проведения анализов другими сотрудниками лаборатории, химиками, но сами анализы не проводят).

### **Анализ требований**

В курсовой работе будет рассмотрена автоматизация ХХХХХХХХХХХХХХХ. Разрабатываемая база данных будет хранить данные ХХХХХХХХХ ХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХ.

ХХХХХХХ компания представляет собой полноценное и рабочее предприятие, которое занимается ХХХХХХХХХХХХХХ ХХХХХХХХХ ХХХХХХХХХХ ХХХХХХХХХХХХХ и многое другое.

В результате исследования предметной области была выявлена необходимость в создании базы данных ХХХХХХХХХХХХ компании в связи с её отсутствием. При анализе необходимой системы были сформулированы функциональные и нефункциональные требованияк базе данных учета поставок фармацевтической компании.

К функциональным требованиям к базе данных ХХХХХХХХХХХХХХХ. относятся:

1) Возможность хранения и обновления данных о ХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХХ.

2) Высокое быстродействие (малое время отклика на запрос – не более 1,5 с).

3) Независимость данных.

4) Стандартизация построения и эксплуатации базы данных (СУБД).

5) Адекватность отображения данных предметной области, связанной с обработкой данных в системе ХХХХХХХХХХХХХХХХХХХХХХХХХХ.

6) Совместное использование данных многими пользователями (многопользовательский доступ к данным).

7) Дружелюбный и интуитивно понятный интерфейс пользователя.

Независимость данных – возможность изменения логической и физической структуры базы данных без изменения представлений о сотрудниках. Независимость данных предполагает инвариантность к характеру хранения данных, программному обеспечению и техническим средствам.

Стандартизация обеспечивает преемственность поколений СУБД, упрощает взаимодействие базы данных одного поколения СУБД с одинаковыми и различными моделями данных.

К нефункциональным требованиям к базе данных учета поставок фармацевтической компании относятся:

1) Создание рассматриваемой базы данных на базе программы MS SQL Server 2019 Developer и операционной системы Windows 10.

2) Требуемый объем оперативной памяти не менее 2 Гб.

3) Скорость процессора, требуемый объем на жестком диске.

К требованиям по защите информации относятся:

1) Реализация аутентификации и авторизации.

2) Разграничение прав доступа к данным.

3) Защита базы данных путём шифрования данных в ячейках и прозрачного шифрования.

4) Безопасность данных – защита данных от преднамеренного или непреднамеренного нарушения секретности, искажения или разрушения.

Безопасность данных включает их целостность и защиту. Целостность данных – устойчивость хранимых данных к разрушению и уничтожению, связанных с неисправностями технических средств, системными ошибками и ошибочными действиями пользователей.

Целостность обеспечивается триггерами целостности – специальными приложениями-программами, работающими при определенных условиях. Защита данных от несанкционированного доступа предполагает ограничение доступа к конфиденциальным данным и может достигаться:

– введением системы паролей;

– получением разрешений от администратора базы данных (АБД);

– запретом от АБД на доступ к данным;

– формированием видов – таблиц, производных от исходных и предназначенных конкретным пользователям.

### **Выбор и обоснование проектных решений по видам обеспечения**

Для выбора подходящей СУБД, где будет создана база данных, необходимо провести анализ существующих систем управления и выбрать наиболее удобную и выгодную предприятию систему.

MS SQL Server – это система управления реляционными базами данных, разработанная компанией Microsoft. Она обладает высокой производительностью и отказоустойчивостью, тесно интегрирована с ОС Windows. Этот сервер поддерживает удаленные подключения, работает с многими популярными типами данных, дает возможность создавать триггеры и хранимые процедуры, имеет практичные и удобные утилиты для настройки.

В качестве основного языка запросов в MS SQL Server используется Transact-SQL (совместная разработка Microsoft и Sybase). При этом Transact-SQL — это реализация стандарта ANSI/ISO по SQL (структурированному языку запросов), но имеющая некоторые расширения. Клиент (например, внешняя программа) отправляет запрос на языке SQL посредством специального API (например, MS SQL Server Management Studio). СУБД должным образом интерпретирует и выполняет запрос, а затем посылает клиенту результат выполнения.

Сегодня СУБД MS SQL широко применяется при работе с базами данных (БД) самых разных размеров, начиная от персональных, заканчивая крупными БД масштаба предприятия.

MS SQL Server долгое время был доступен исключительно для Windows, однако начиная с версии 16 эта система доступна и на Linux. Кроме того, MS SQL Server позволяет установить разные версии на одном компьютере.

SQL Server характеризуется следующими особенностями:

* Высокая производительность – MS SQL Server работает достаточно быстро.
* Надежность и безопасность – MS SQL Server предоставляет шифрование данных.
* С MS SQL Server относительно легко работать и вести администрирование.

Проанализировав возможности и особенности разных СУБД, было решено, что база данных будет создана в среде MS SQL Server, т.к. именно эта СУБД в данном случае отвечает всем требованиям и имеет меньше всего недостатков.

При проектировании базы данных будет использована онлайн-платформа **draw.io,** т.к. она имеет все необходимые для составления диаграмм функции и при этом не требует установки – можно работать в режиме онлайн.

Также для создания диаграмм, в которых описываются бизнес-процессы организации, может быть использована программа BPwin (AllFusion Process Modeler).

BPwin – это инструмент для визуального моделирования, анализа, документирования и оптимизации бизнес-процессов. BPwin предлагает средство для сбора всей необходимой информации о работе предприятия и графического изображения этой информации в виде целостной и непротиворечивой модели. Включает три стандартные методологии: IDEF0 (функциональное моделирование), DFD (моделирование потоков данных) и IDEF3 (моделирование потоков работ).

### **Входная и выходная информация**

Входными данными для этой базы данных будут данные о заказчиках, а также их пожелания по показателям, значения которых необходимо определить, которые также определены в договоре (на основе этой информации высчитывается необходимое количество проб).

Данные о новых заказчиках вводятся примерно 1 раз в 3 месяца (однако могут вводиться чаще или реже), обновляются не чаще 1 раза в год (чаще всего при повторном обращении заказчика в случае изменения каких-либо параметров).

Новые договоры составляются несколько раз в месяц, а соответственно и новые данные о показателях, результаты которых заказчикам необходимо получить, вводятся столько же раз (в среднем до 20 раз в месяц, возможно большее или меньшее количество).

Выходными данными будет протокол исследования – дата его составления и результаты показателей, определенных в договоре. Ими будут пользоваться как сотрудники лаборатории и начальник лаборатории для ведения отчетности работы, так и заказчик, т.к. протокол исследования содержит количественные результаты исследований, за которыми заказчик и обращается в лабораторию.

## **Проектирование базы данных**

В главе 2 будет предоставлено проектирование базы данных перед ее реализацией. Будут рассмотрены существующие в организации бизнес-процессы и процессы обработки данных исследуемого объекта, а также инфологическая и даталогическая модели и их составляющие (сущности и взаимосвязи между ними, отношения и атрибуты, ключевые поля и индексы).

### **Существующие в организации бизнес-процессы и процессы обработки данных исследуемого объекта**

Основными бизнес-процессами лаборатории являются: составление договора с заказчиком, заполнение акта отбора проб, анализ отобранных проб и заполнение протокола испытаний.

Контекстная диаграмма, отображающая основную функцию системы, а также входные и выходные данные и управляющую и регламентирующую функции, представлена в приложении на рис. 3. Модель построена с использованием BPwin (AllFusion Process Modeler).

Входными данными для этой базы данных будут данные о заказчиках (включая их пожелания по показателям, значения которых необходимо определить).

Управляющими данными являются государственные стандарты (ГОСТы) на составление договора, акта отбора проб, описание методики проведения анализа, а также на составление протокола исследования.

Механизмом исполнения являются начальник лаборатории и сотрудники лаборатории, а также приборы и реактивы, необходимые во время проведения собственно анализа.

Выходными данными будет протокол исследования, включающий результаты показателей, определенных в договоре.

Выходными данными будут пользоваться как сотрудники лаборатории и начальник лаборатории для ведения отчетности работы, так и заказчик, т.к. протокол исследования содержит количественные результаты исследований, за которыми заказчик и обращается в лабораторию.

В приложении на рис. 4 показана диаграмма декомпозиции первого уровня, которая является детализацией контекстной диаграммы. Контекстная диаграмма является родительской по отношению к диаграмме декомпозиции, которая, в свою очередь, является дочерней для контекстной диаграммы. Дочерняя диаграмма более подробно описывает функции системы, которые реализуются в рамках основной функции, отображенной в родительской диаграмме.

В приложении на рис. 5 показана диаграмма декомпозиции второго уровня, где подробно описан бизнес-процесс «Составление договора».

### **Инфологическое моделирование**

В этом разделе будет описано создание инфологической (концептуальной) модели. Она нужна для представления предметной области в удобной для восприятия человеком и независимой от СУБД форме. Для создания инфологической модели необходимо выделить сущности и взаимосвязи между ними. Концептуальная модель представляется в графическом виде.

### **Сущности**

Сущность – класс однотипных объектов, информация о которых должна быть учтена в модели в наименьшем единственном числе. Объектом в данном случае может быть предмет, явление, событие и др.

Каждая сущность имеет атрибуты сущности – именованные характеристики, которые являются свойствами сущности.

В данной предметной области выделяются следующие сущности и их атрибуты:

1. Сущность «Заказчик».

Заказчиком является юридическое или физическое лицо. Заказчик заключает договор, чтобы получить услуги, предоставляемые лабораторией.

Атрибуты сущности «Заказчик»:

* ID заказчика – первичный ключ; идентификационный номер, уникальный для каждого заказчика;
* Наименование – название организации или имя физического или юридического лица;
* Адрес – юридический адрес заказчика;
* Банковские реквизиты – банковские реквизиты заказчика.

1. Сущность «Договор».

Договор заключается между заказчиком и лабораторией, он определяет, какие именно показатели будут определяться в ходе исследования.

Атрибуты сущности «Договор»:

* Номер договора – первичный ключ;
* Дата договора – дата заключения договора;
* Предмет договора – то, на что направлено взаимодействие сторон; предметом договора чаще всего является природная, сточная или питьевая вода, в редких случаях в роли предмета договора могут выступать минеральная и др. воды;
* Количество проб – количество проб, которое нужно отобрать для успешного проведения исследований по всем необходимым показателям.

1. Сущность «Акт отбора и протокол».

Сущностью «Акт отбора и протокол» является объединение двух документов – акта отбора и протокола. Актом отбора является документ, который фиксирует момент взятия пробы; протоколом является документ, в котором отражены данные по проведенным анализам. Эти документы в объединены одну сущность, так как они имеют одинаковый номер, и на каждый акт отбора составляется только один протокол.

Атрибуты сущности «Акт отбора и протокол»:

* Номер пробы – первичный ключ; во время отбора пробы ей присваивается номер, он и является затем номером акта и протокола;
* Адрес отбора – фактический адрес места, откуда была взята проба;
* Дата отбора – дата отбора пробы и составления акта отбора пробы соответственно;
* Дата протокола – дата составления протокола, т.е. дата проведения исследований и определения необходимых показателей в отобранных пробах.

1. Сущность «Показатели».

Все показатели, внесенные в область аккредитации испытательной лаборатории.

Атрибуты сущности «Показатели»:

* Название показателя – первичный ключ;
* Цена – цена каждого показателя;
* Единицы измерения;
* Класс опасности – присвоен ли показателю класс опасности и какой он в случае, если присвоен;
* Норматив макс – верхняя граница нормального значения показателя для питьевой воды;
* Норматив мин – нижняя граница нормального значения показателя для питьевой воды.

1. Сущность «Методика».

Методика описывает процесс проведения анализа.

Атрибуты сущности «Методики»:

* Номер методики – первичный ключ; каждая методика описана в государственных стандартах и имеет определенный номер ГОСТ;
* Название – название методики;
* Приборы – нужны ли приборы для проведения исследования и если да, то какие;
* Посуда – нужна ли посуда для проведения исследования и если нужна, то какая.

1. Сущность «Показатель в договоре».

Промежуточная сущность, связывает договоры с соответствующими показателями и методиками их определения, так как в договоре может быть определено несколько показателей, при этом для определения некоторых показателей могут быть использованы разные методики.

Атрибуты сущности «Показатель в договоре»:

* Номер договора;
* Номер методики;
* Название показателя;
* Норматив в договоре макс – верхняя граница нормального значения для предмета договора;
* Норматив в договоре мин – нижняя граница нормального значения для предмета договора.

1. Сущность «Показатель в протоколе».

Промежуточная сущность, связывает протоколы с соответствующими показателями и содержит результат исследования.

Атрибуты сущности «Показатель в протоколе»:

* Название показателя;
* Номер пробы;
* Результат показателя – количественный результат проведенного анализа по данному показателю.

### **Взаимосвязи сущностей**

Между сущностями имеется определенная связь, которую можно охарактеризовать определенным глаголом, описывающим имеющуюся зависимость, с заданной в имеющейся взаимной зависимости модальностью («должен» и «может»). В данной предметной области среди выделенных сущностей существуют следующие связи:

Связь «Заказчик» – «Договор» (1:М).

Каждый заказчик может заключить несколько договоров; каждый договор должен быть заключен только одним заказчиком.

Связь «Договор» – «Показатель в договоре» (1:М).

Каждый договор должен содержать несколько показателей в договоре; каждый показатель в договоре должен содержаться только в одном договоре.

Связь «Показатели» – «Показатель в договоре» (1:М).

Каждый показатель в договоре должен быть только одним показателем; каждый показатель может несколько раз быть показателем в договоре.

Связь «Показатели» – «Показатель в протоколе» (1:М).

Каждый показатель в протоколе должен быть только одним показателем; каждый показатель может несколько раз быть показателем в протоколе.

Связь «Договор» – «Акт отбора и протокол» (1:М).

На каждый договор должно быть составлено несколько актов отбора и протоколов; по каждому акту отбора и протоколу должен быть заключен только один договор.

Связь «Акт отбора и протокол» – «Показатель в протоколе» (1:М).

Каждый акт отбора и протокол должны содержать несколько показателей в протоколе; каждый показатель в протоколе должен содержаться только в одном акте отбора и протоколе.

Связь «Методики» – «Показатель в договоре» (1:М).

Каждая методика может определять несколько показателей в договоре; каждый показатель в договоре должен быть определен с помощью только одной методики.

### **Разработка инфологической модели**

В предыдущих разделах был проведен анализ предметной области, были выделены сущности и связи, существующие между ними. Собственно инфологическая модель, созданная на основе исследования необходимых сущностей и связей между ними, представлена на рис. 1.

Инфологическая модель была выполнена с помощью онлайн-системы draw.io.

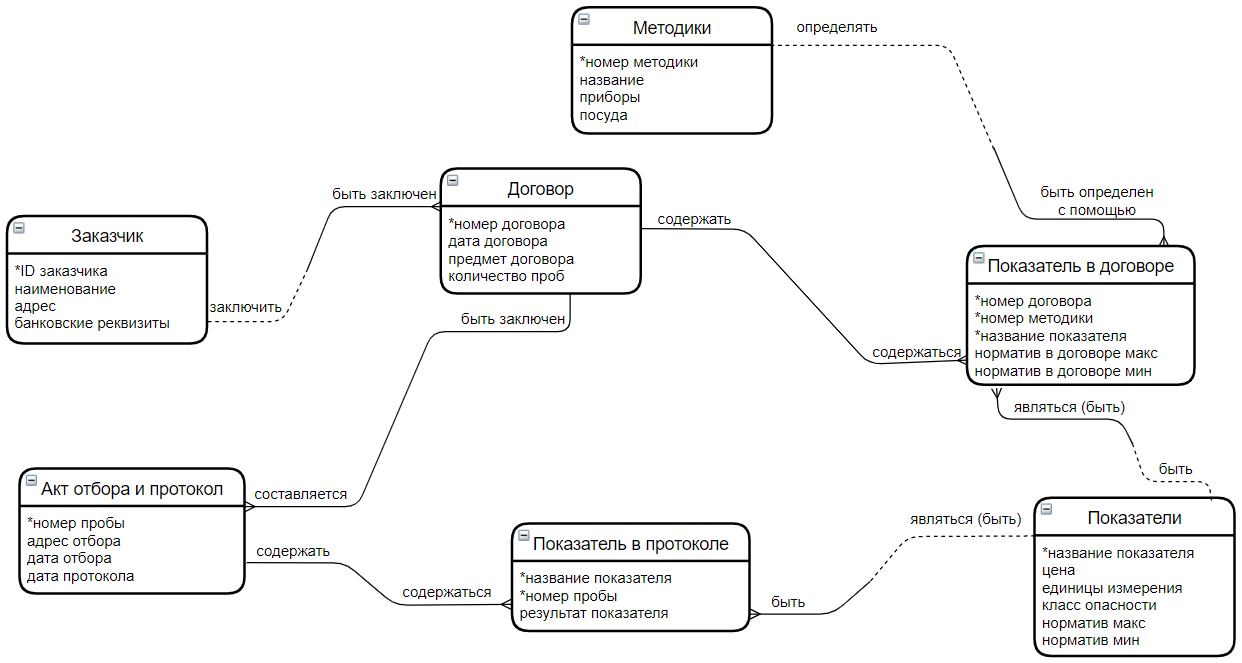


Рис. 1. Инфологическая модель ([www.draw.io](http://www.draw.io)).

### **Даталогическое проектирование**

В этом разделе будет создана даталогическая модель системы. Даталогическое проектирование – это проектирование логической структуры базы данных, при этом на выбор проектных решений оказывают влияние возможности физической организации данных, представленные конкретной СУБД. Логическая структура базы данных, отраженная в даталогической модель, и сама заполненная данными база данных являются отображением реальной предметной области, поэтому на выбор предметных решений непосредственное влияние оказывает специфика отображаемой предметной области, отраженной в инфологической модели.

Даталогическая модель описывает логическую структуру БД на языке описания данных. Она включает в себя определение всех информационных единиц и связей между ними, а также задание их имен, типов данных и количественных и качественных характеристик. Даталогическая модель представляется в графическом виде.

### **Отношение и атрибуты**

В этом подпункте будут описаны отношения и их атрибуты, которые войдут в модель. Свойства атрибутов отношений представлены в приложении в таблице 1.

1. Отношение «Customer».

Отношение «Customer» соответствует сущности «Заказчик» и атрибуты этого отношения соответствуют атрибутам сущности:

* сust\_id – ID заказчика, первичный ключ; идентификационный номер, уникальный для каждого заказчика;
* cust\_name – наименование;
* cust\_address – адрес;
* cust\_bank – банковские реквизиты.

1. Отношение «Contract».

Отношение «Contract» соответствует сущности «Договор» и атрибуты этого отношения соответствуют атрибутам сущности:

* contr\_num – номер договора;
* contr\_date – дата договора;
* contr\_subj – предмет договора;
* sample\_amount – количество проб.

1. Отношение «ActProtocol».

Отношение «ActProtocol» соответствует сущности «Акт отбора и протокол» и атрибуты этого отношения соответствуют атрибутам сущности:

* sample\_num – номер пробы;
* sample\_address – адрес отбора;
* sample\_date – дата отбора;
* prot\_date – дата протокола.

1. Отношение «Indicator».

Отношение «Indicator» соответствует сущности «Показатели» и атрибуты этого отношения соответствуют атрибутам сущности:

* ind\_name – название показателя;
* ind\_price – цена;
* ed\_izm – единицы измерения;
* danger\_class – класс опасности;
* standard\_max – норматив макс;
* standard\_min – норматив мин.

1. Отношение «Method».

Отношение «Method» соответствует сущности «Методики» и атрибуты этого отношения соответствуют атрибутам сущности:

* method\_num – номер методики;
* method\_name – название;
* devices – приборы;
* tools – посуда.

1. Отношение «Ind\_Contr».

Отношение «Ind\_Contr» соответствует сущности «Показатель в договоре» и атрибуты этого отношения соответствуют атрибутам сущности:

* contr\_max – норматив в договоре макс;
* contr\_min – норматив в договоре мин.

1. Отношение «Ind\_Prot».

Отношение «Ind\_Prot» соответствует сущности «Показатель в протоколе» и атрибуты этого отношения соответствуют атрибутам сущности:

* ind\_result – результат показателя.

### **Ключевые поля и индексы**

Ключ сущности – неизбыточный набор атрибутов, значения которых в совокупности являются уникальными для каждого экземпляра сущности. Сущность может иметь несколько различных ключей.

Первичный ключ – это атрибут или набор атрибутов, значения которых однозначно определяют каждый кортеж отношения и обеспечивают уникальность строк отношения.

Внешний ключ отношения – это набор атрибутов отношения, содержащий ссылки на ключи другого или того же самого отношения.

Системы баз данных обычно используют индексы для обеспечения быстрого доступа к реляционным данным. Индекс представляет собой отдельную физическую структуру данных, которая позволяет получать быстрый доступ к одной или нескольким строкам данных.

Все ключи и индексы отношений описаны в приложении в таблице 2.

### **Даталогическая модель**

После исследования всех отношений и связей между ними была составлена даталогическая модель.

Даталогическая модель представлена на рис. 2.

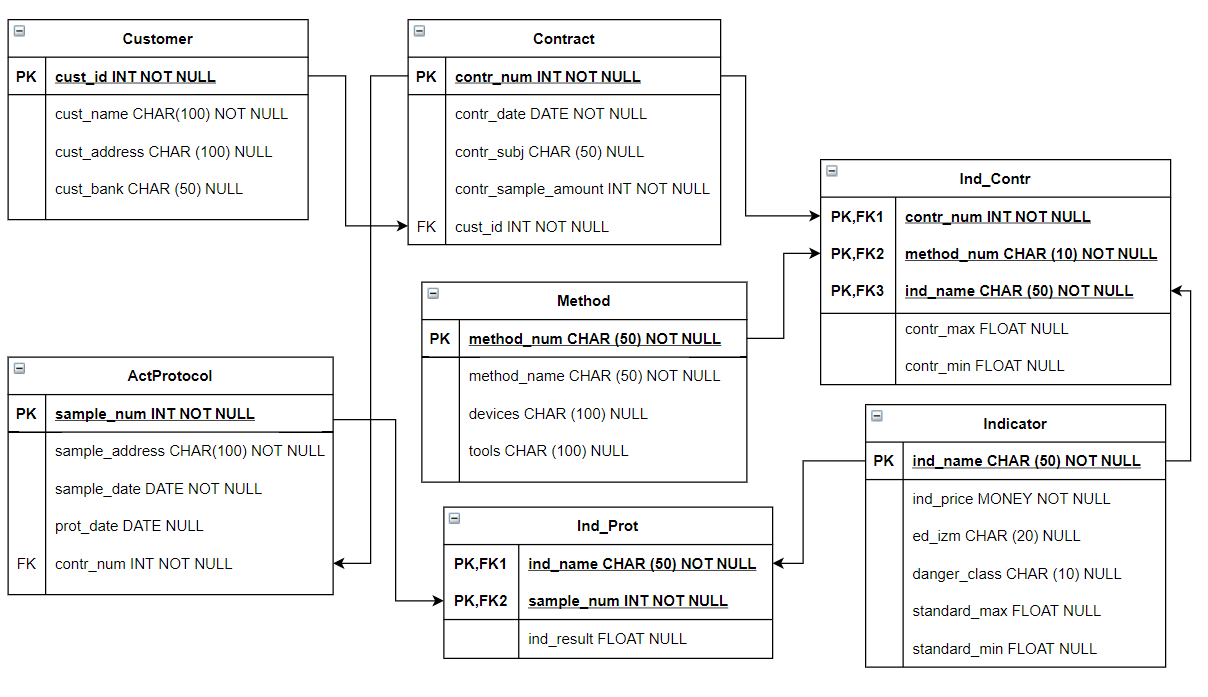


Рис. 2. Даталогическая модель ([www.draw.io](http://www.draw.io)).

## **Реализация**

В главе 3 будет представлена реализация спроектированной базы данных в выбранной среде управления, а также описаны особенности реализации. Также будут рассмотрены разработка хранимых процедур и триггеров, их тестирование и отладка, а также будет разработан прототип пользовательского интерфейса.

### **Особенности реализации**

База данных будет создана в выбранной ранее в ходе анализа СУБД Microsoft SQL Server. База данных будет назваться «Laboratory».

В базе данных «Laboratory» 7 таблиц, которые соответствуют рассмотренным ранее во второй части отношениям:

1. Customer;
2. Contract;
3. Indicator;
4. ActProtocol;
5. Method;
6. Ind\_Contr;
7. Ind\_Prot.

В таблицах указываются первичные и внешние ключи, создаются необходимые индексы, описанные ранее. Также создаются связи между отношениями (с помощью внешних ключей).

Примеры заполненных таблиц в приложении на рис. 6-12 (вместе с кодами создания каждой таблицы и связей между ними). На рис. 13-19 соответственно заполненные данными таблицы. Диаграмма базы данных на рис. 20.

При реализации поле contr\_subj (предмет договора в Договоре) отношения Contract следует сделать полем со списком, в который входят: «сточная вода», «природная вода», «питьевая вода». При этом ограничивать возможные значения только теми, что входят в список, нельзя, должна быть предусмотрена возможность написать значение вручную, т.к. им может быть другой тип (какой именно в таком случае указывается вручную).

Также следует сделать поле ed\_izm (единицы измерения) отношения Indicator полем со списком, в который входят: поле со списком, список: «ммоль/дм3», «ед. pH», «мг/л», «мг-экв/л». При этом ограничивать возможные значения только теми, что входят в список, нельзя, т.к. некоторые показатели имеют другие единицы измерения, которые необходимо зафиксировать вручную.

В этой базе данных сначала следует ввести данные в таблицы Indicator и Method. Данные в этих таблицах практически никогда не изменяются.

В Indicator (Показатель) вводятся данные о показателях, внесенных в область аккредитации испытательной лаборатории. Также в поля standard\_max и standard\_min вводятся максимальное и минимальное значения соответственно для питьевой воды. Далее на основе этих показателей вычисляются нормальные значения нужного показателя для природной или сточной воды. Для смеси эти значения вычисляются и вносятся в таблицу вручную.

В Method (Методика) вносятся данные о методиках, с помощью которых проводятся анализы в испытательной лаборатории. Методики описаны в ГОСТе, поэтому у каждой методики есть официальный ГОСТ номер, который состоит не только из цифр. Описание методики в таблицу не вносится. Также некоторые методики включают использование приборов, посуды и реактивов, но в этой базе данных не будут учитываться реактивы.

В данной базе данных в таблице Contract нет поля с общей суммой, которую должен заплатить заказчик за исследование. Общая сумма будет вычисляться по вызову хранимой процедуры.

### **Разработка алгоритмов и кода хранимых процедур и триггеров**

Хранимая процедура — это специальный тип пакета инструкций Trаnsаct-SQL, созданный, используя язык SQL и процедурные расширения. Хранимые процедуры сохраняются на стороне сервера, чтобы улучшить производительность и постоянство выполнения повторяемых задач.

Триггер – это механизм, который вызывается, когда в указанной таблице происходит определенное действие. Каждый триггер состоит из имени, действия и исполнения.

В этом разделе будет описано создание хранимых процедур и триггеров, улучшающих работу созданной базы данных.

### **Тестирование и отладка хранимых процедур**

Первая хранимая процедура: contract\_sum.

Эта хранимая процедура считает сумму договора с определенным номером и выводит ее на экран.

Процедура получает на вход параметр @contr\_num типа INT – это номер договора, для которого процедура должна посчитать сумму.

Процедура считает сумму цен показателей, названия которых совпадают с названиями показателей с определенным номером договора (тем, что мы получили в параметре). Затем эта сумма выводится в виде таблицы с названием столбца "Total contract sum".

Входные данные для процедуры contract\_sum: параметр @contr\_num типа INT, в параметр вводится номер договора, для которого определены показатели, значения которых будут определяться в ходе анализа проб воды.

Выходные данные процедуры contract\_sum: таблица, состоящая из одного столбца с названием "Total contract sum", в котором указывается сумма договора с номером, введенным в параметр.

Коды создания процедуры и ее вызова находятся в приложении. Создание процедуры изображено на рис. 21. Вызов процедуры изображен на рис. 22.

Вторая хранимая процедура: standard\_drinking\_water\_min.

Эта хранимая процедура считает минимальное нормальное значение для определенного показателя в договоре по предмету договора на основе данных из таблицы Indicator (нормального минимального значения показателя для питьевой воды).

Процедура получает на вход параметры @ind\_name\_p типа CHAR(50) – название показателя, для которого будет считаться минимальное значение, @contr\_num\_p типа INT – номер договора, для показателя из которого считается минимальное нормальное значение, @method\_num\_p типа CHAR(50) – название методики.

Процедура считает минимальное нормальное значение для определенного показателя в договоре. Сначала объявляется переменная @min\_drinking\_water типа FLOAT, затем она инициализируется значением standard\_min из таблицы Indicator из элемента отношения, ключом которого является @ind\_name\_p – один из входных параметров. Затем данные в таблице Ind\_Contr в столбце contr\_min обновляются на @min\_drinking\_water в той строке, ключ которой состоит из @ind\_name\_p, @contr\_num\_p и @method\_num\_p.

Входные данные для процедуры standard\_drinking\_water\_min: параметры @ind\_name\_p типа CHAR(50) – название показателя, для которого будет считаться минимальное значение, @contr\_num\_p типа INT – номер договора, для показателя из которого считается минимальное нормальное значение, @method\_num\_p типа CHAR(50) – название методики. В параметры передается все значение первичного ключа, чтобы однозначно идентифицировать поле, для которого процедура будет вычислять и обновлять значение.

Выходные данные процедуры standard\_drinking\_water\_min: значение в таблице Ind\_Contr в столбце contr\_min в поле, первичным ключом для которого является сочетание параметров @ind\_name\_p, @contr\_num\_p и @method\_num\_p, обновится значение.

Эту процедуру впоследствии также будет вызывать триггер trigger\_s\_drinking\_water\_min.

Коды создания процедуры и ее вызова находятся в приложении. Создание процедуры изображено на рис. 23. Вызов процедуры изображен на рис. 24.

Третья хранимая процедура standard\_drinking\_water\_max:

Эта хранимая процедура считает максимальное нормальное значение для определенного показателя в договоре по предмету договора на основе данных из таблицы Indicator (нормального максимального значения показателя для питьевой воды).

Процедура получает на вход параметры @ind\_name\_p типа CHAR(50) – название показателя, для которого будет считаться минимальное значение, @contr\_num\_p типа INT – номер договора, для показателя из которого считается минимальное нормальное значение, @method\_num\_p типа CHAR(50) – название методики.

Процедура считает максимальное нормальное значение для определенного показателя в договоре. Сначала объявляется переменная @max\_drinking\_water типа FLOAT, затем она инициализируется значением standard\_max из таблицы Indicator из элемента отношения, ключом которого является @ind\_name\_p – один из входных параметров. Затем данные в таблице Ind\_Contr в столбце contr\_max обновляются на @max\_drinking\_water в той строке, ключ которой состоит из @ind\_name\_p, @contr\_num\_p и @method\_num\_p.

Входные данные для процедуры standard\_drinking\_water\_max: параметры @ind\_name\_p типа CHAR(50) – название показателя, для которого будет считаться минимальное значение, @contr\_num\_p типа INT – номер договора, для показателя из которого считается минимальное нормальное значение, @method\_num\_p типа CHAR(50) – название методики. В параметры передается все значение первичного ключа, чтобы однозначно идентифицировать поле, для которого процедура будет вычислять и обновлять значение.

Выходные данные процедуры standard\_drinking\_water\_max: значение в таблице Ind\_Contr в столбце contr\_max в поле, первичным ключом для которого является сочетание параметров @ind\_name\_p, @contr\_num\_p и @method\_num\_p, обновится значение.

Эту процедуру впоследствии также будет вызывать триггер trigger\_s\_drinking\_water\_max.

Коды создания процедуры и ее вызова находятся в приложении. Создание процедуры изображено на рис. 25. Вызов процедуры изображен на рис. 26.

Четвертая хранимая процедура standard\_natural\_water\_min:

Эта хранимая процедура считает минимальное нормальное значение для определенного показателя в договоре по предмету договора на основе данных из таблицы Indicator (нормального минимального значения показателя для питьевой воды).

Процедура получает на вход параметры @ind\_name\_p типа CHAR(50) – название показателя, для которого будет считаться минимальное значение, @contr\_num\_p типа INT – номер договора, для показателя из которого считается минимальное нормальное значение, @method\_num\_p типа CHAR(50) – название методики.

Процедура считает минимальное нормальное значение для определенного показателя в договоре. Сначала объявляется переменная @min\_natural\_water типа FLOAT, затем она инициализируется значением standard\_min, умноженным на 0,2, из таблицы Indicator из элемента отношения, ключом которого является @ind\_name\_p – один из входных параметров. Затем данные в таблице Ind\_Contr в столбце contr\_min обновляются на @min\_natural\_water в той строке, ключ которой состоит из @ind\_name\_p, @contr\_num\_p и @method\_num\_p.

Входные данные для процедуры standard\_natural\_water\_min: параметры @ind\_name\_p типа CHAR(50) – название показателя, для которого будет считаться минимальное значение, @contr\_num\_p типа INT – номер договора, для показателя из которого считается минимальное нормальное значение, @method\_num\_p типа CHAR(50) – название методики. В параметры передается все значение первичного ключа, чтобы однозначно идентифицировать поле, для которого процедура будет вычислять и обновлять значение.

Выходные данные процедуры standard\_natural\_water\_min: значение в таблице Ind\_Contr в столбце contr\_min в поле, первичным ключом для которого является сочетание параметров @ind\_name\_p, @contr\_num\_p и @method\_num\_p, обновится значение.

Эту процедуру впоследствии также будет вызывать триггер trigger\_s\_natural\_water\_min.

Коды создания процедуры и ее вызова находятся в приложении. Создание процедуры изображено на рис. 27. Вызов процедуры изображен на рис. 28.

Пятая хранимая процедура standard\_natural\_water\_max:

Эта хранимая процедура считает максимальное нормальное значение для определенного показателя в договоре по предмету договора на основе данных из таблицы Indicator (нормального максимального значения показателя для питьевой воды).

Процедура получает на вход параметры @ind\_name\_p типа CHAR(50) – название показателя, для которого будет считаться максимальное значение, @contr\_num\_p типа INT – номер договора, для показателя из которого считается максимальное нормальное значение, @method\_num\_p типа CHAR(50) – название методики.

Процедура считает максимальное нормальное значение для определенного показателя в договоре. Сначала объявляется переменная @max\_natural\_water типа FLOAT, затем она инициализируется значением standard\_max, умноженным на 3, из таблицы Indicator из элемента отношения, ключом которого является @ind\_name\_p – один из входных параметров. Затем данные в таблице Ind\_Contr в столбце contr\_max обновляются на @max\_natural\_water в той строке, ключ которой состоит из @ind\_name\_p, @contr\_num\_p и @method\_num\_p.

Входные данные для процедуры standard\_natural\_water\_max: параметры @ind\_name\_p типа CHAR(50) – название показателя, для которого будет считаться максимальное значение, @contr\_num\_p типа INT – номер договора, для показателя из которого считается максимальное нормальное значение, @method\_num\_p типа CHAR(50) – название методики. В параметры передается все значение первичного ключа, чтобы однозначно идентифицировать поле, для которого процедура будет вычислять и обновлять значение.

Выходные данные процедуры standard\_natural\_water\_max: значение в таблице Ind\_Contr в столбце contr\_max в поле, первичным ключом для которого является сочетание параметров @ind\_name\_p, @contr\_num\_p и @method\_num\_p, обновится значение.

Эту процедуру впоследствии также будет вызывать триггер trigger\_s\_natural\_water\_max.

Коды создания процедуры и ее вызова находятся в приложении. Создание процедуры изображено на рис. 29. Вызов процедуры изображен на рис. 30.

Шестая хранимая процедура standard\_wastewater\_min:

Эта хранимая процедура считает минимальное нормальное значение для определенного показателя в договоре по предмету договора на основе данных из таблицы Indicator (нормального минимального значения показателя для питьевой воды).

Процедура получает на вход параметры @ind\_name\_p типа CHAR(50) – название показателя, для которого будет считаться минимальное значение, @contr\_num\_p типа INT – номер договора, для показателя из которого считается минимальное нормальное значение, @method\_num\_p типа CHAR(50) – название методики.

Процедура считает минимальное нормальное значение для определенного показателя в договоре. Сначала объявляется переменная @min\_wastewater типа FLOAT, затем она инициализируется значением standard\_min, умноженным на 0,05, из таблицы Indicator из элемента отношения, ключом которого является @ind\_name\_p – один из входных параметров. Затем данные в таблице Ind\_Contr в столбце contr\_min обновляются на @min\_wastewater в той строке, ключ которой состоит из @ind\_name\_p, @contr\_num\_p и @method\_num\_p.

Входные данные для процедуры standard\_wastewater\_min: параметры @ind\_name\_p типа CHAR(50) – название показателя, для которого будет считаться минимальное значение, @contr\_num\_p типа INT – номер договора, для показателя из которого считается минимальное нормальное значение, @method\_num\_p типа CHAR(50) – название методики. В параметры передается все значение первичного ключа, чтобы однозначно идентифицировать поле, для которого процедура будет вычислять и обновлять значение.

Выходные данные процедуры standard\_wastewater\_min: значение в таблице Ind\_Contr в столбце contr\_min в поле, первичным ключом для которого является сочетание параметров @ind\_name\_p, @contr\_num\_p и @method\_num\_p, обновится значение.

Эту процедуру впоследствии также будет вызывать триггер trigger\_s\_wastewater\_min.

Коды создания процедуры и ее вызова находятся в приложении. Создание процедуры изображено на рис. 31. Вызов процедуры изображен на рис. 32.

Седьмая хранимая процедура standard\_wastewater\_max:

Эта хранимая процедура считает минимальное нормальное значение для определенного показателя в договоре по предмету договора на основе данных из таблицы Indicator (нормального минимального значения показателя для питьевой воды).

Процедура получает на вход параметры @ind\_name\_p типа CHAR(50) – название показателя, для которого будет считаться минимальное значение, @contr\_num\_p типа INT – номер договора, для показателя из которого считается минимальное нормальное значение, @method\_num\_p типа CHAR(50) – название методики.

Процедура считает минимальное нормальное значение для определенного показателя в договоре. Сначала объявляется переменная @min\_wastewater типа FLOAT, затем она инициализируется значением standard\_min, умноженным на 0,05, из таблицы Indicator из элемента отношения, ключом которого является @ind\_name\_p – один из входных параметров. Затем данные в таблице Ind\_Contr в столбце contr\_min обновляются на @min\_wastewater в той строке, ключ которой состоит из @ind\_name\_p, @contr\_num\_p и @method\_num\_p.

Входные данные для процедуры standard\_wastewater\_min: параметры @ind\_name\_p типа CHAR(50) – название показателя, для которого будет считаться минимальное значение, @contr\_num\_p типа INT – номер договора, для показателя из которого считается минимальное нормальное значение, @method\_num\_p типа CHAR(50) – название методики. В параметры передается все значение первичного ключа, чтобы однозначно идентифицировать поле, для которого процедура будет вычислять и обновлять значение.

Выходные данные процедуры standard\_wastewater\_min: значение в таблице Ind\_Contr в столбце contr\_min в поле, первичным ключом для которого является сочетание параметров @ind\_name\_p, @contr\_num\_p и @method\_num\_p, обновится значение.

Эту процедуру впоследствии также будет вызывать триггер trigger\_s\_wastewater\_min.

Коды создания процедуры и ее вызова находятся в приложении. Создание процедуры изображено на рис. 33. Вызов процедуры изображен на рис. 34.

### **Тестирование и отладка триггеров**

Триггер workson\_sot проверяет наличие повторных значений в столбце Name в таблице Sotrudnic. Это означает, что если при каждом изменении столбца Name в таблице Sotrudnic обнаруживается с помощью функции IN, что вводимое имя уже присутствует в базе данных, то начинается блок BEGIN, где происходит откат транзакции с помощью инструкции ROLLBACK TRANSACTION, и система выводит, что в базе данных уже присутствует данное вводимое имя.

Первый триггер: trigger\_s\_drinking\_water\_min.

Триггер trigger\_s\_drinking\_water\_min при срабатывании вызывает хранимую процедуру standard\_drinking\_water\_min, созданную ранее.

В триггере создаются переменные @ind\_name, @method\_num, @contr\_num и @contr\_subj и инициализируются вставленными значениями в ind\_name, method\_num и contr\_num таблицы Ind\_Contr соответственно, а в переменную @contr\_subj записывается значение предмета договора из таблицы Contract. Далее это значение проверяется, и если оно равно 'вода питьевая', то происходит вызов хранимой процедуры standard\_drinking\_water\_min, на вход которой передаются параметры в форме ранее созданных и инициализированных переменных @ind\_name, @contr\_num, @method\_num.

Код создания триггера и код вызова его срабатывания указаны в приложении. Также в приложении создание триггера изображено на рис. 35, срабатывание триггера на рис. 36.

Второй триггер: trigger\_s\_drinking\_water\_max.

Триггер trigger\_s\_drinking\_water\_max при срабатывании вызывает хранимую процедуру standard\_drinking\_water\_max, созданную ранее.

В триггере создаются переменные @ind\_name, @method\_num, @contr\_num и @contr\_subj и инициализируются вставленными значениями в ind\_name, method\_num и contr\_num таблицы Ind\_Contr соответственно, а в переменную @contr\_subj записывается значение предмета договора из таблицы Contract. Далее это значение проверяется, и если оно равно 'вода питьевая', то происходит вызов хранимой процедуры standard\_drinking\_water\_max, на вход которой передаются параметры в форме ранее созданных и инициализированных переменных @ind\_name, @contr\_num, @method\_num.

Код создания триггера и код вызова его срабатывания указаны в приложении. Также в приложении создание триггера изображено на рис. 37, срабатывание триггера на рис. 38.

Третий триггер: trigger\_s\_natural\_water\_min.

Триггер trigger\_s\_natural\_water\_min при срабатывании вызывает хранимую процедуру standard\_natural\_water\_min, созданную ранее.

В триггере создаются переменные @ind\_name, @method\_num, @contr\_num и @contr\_subj и инициализируются вставленными значениями в ind\_name, method\_num и contr\_num таблицы Ind\_Contr соответственно, а в переменную @contr\_subj записывается значение предмета договора из таблицы Contract. Далее это значение проверяется, и если оно равно 'вода природная', то происходит вызов хранимой процедуры standard\_natural\_water\_min, на вход которой передаются параметры в форме ранее созданных и инициализированных переменных @ind\_name, @contr\_num, @method\_num.

Код создания триггера и код вызова его срабатывания указаны в приложении. Также в приложении создание триггера изображено на рис. 39, срабатывание триггера на рис. 40.

Четвертый триггер: trigger\_s\_natural\_water\_max.

Триггер trigger\_s\_natural\_water\_max при срабатывании вызывает хранимую процедуру standard\_natural\_water\_max, созданную ранее.

В триггере создаются переменные @ind\_name, @method\_num, @contr\_num и @contr\_subj и инициализируются вставленными значениями в ind\_name, method\_num и contr\_num таблицы Ind\_Contr соответственно, а в переменную @contr\_subj записывается значение предмета договора из таблицы Contract. Далее это значение проверяется, и если оно равно 'вода природная', то происходит вызов хранимой процедуры standard\_natural\_water\_max, на вход которой передаются параметры в форме ранее созданных и инициализированных переменных @ind\_name, @contr\_num, @method\_num.

Код создания триггера и код вызова его срабатывания указаны в приложении. Также в приложении создание триггера изображено на рис. 41, срабатывание триггера на рис. 42.

Пятый триггер: trigger\_s\_wastewater\_min.

Триггер trigger\_s\_wastewater\_min при срабатывании вызывает хранимую процедуру standard\_wastewater\_min, созданную ранее.

В триггере создаются переменные @ind\_name, @method\_num, @contr\_num и @contr\_subj и инициализируются вставленными значениями в ind\_name, method\_num и contr\_num таблицы Ind\_Contr соответственно, а в переменную @contr\_subj записывается значение предмета договора из таблицы Contract. Далее это значение проверяется, и если оно равно 'вода сточная', то происходит вызов хранимой процедуры standard\_wastewater\_min, на вход которой передаются параметры в форме ранее созданных и инициализированных переменных @ind\_name, @contr\_num, @method\_num.

Код создания триггера и код вызова его срабатывания указаны в приложении. Также в приложении создание триггера изображено на рис. 43, срабатывание триггера на рис. 44.

Шестой триггер: trigger\_s\_wastewater\_max.

Триггер trigger\_s\_wastewater\_max при срабатывании вызывает хранимую процедуру standard\_wastewater\_max, созданную ранее.

В триггере создаются переменные @ind\_name, @method\_num, @contr\_num и @contr\_subj и инициализируются вставленными значениями в ind\_name, method\_num и contr\_num таблицы Ind\_Contr соответственно, а в переменную @contr\_subj записывается значение предмета договора из таблицы Contract. Далее это значение проверяется, и если оно равно 'вода сточная', то происходит вызов хранимой процедуры standard\_wastewater\_max, на вход которой передаются параметры в форме ранее созданных и инициализированных переменных @ind\_name, @contr\_num, @method\_num.

Код создания триггера и код вызова его срабатывания указаны в приложении. Также в приложении создание триггера изображено на рис. 45, срабатывание триггера на рис. 46.

Седьмой триггер: contract\_date.

Триггер contract\_date запрещает вставлять в договор дату из будущего, т.е. позже сегодняшней.

В триггере создаются переменные @contr\_num и @contr\_date, далее в @contr\_num записывается самый большой на данный момент номер договора, а в @contr\_date записывается дата, вставленная в таблицу в поле, первичный ключ которого равен @contr\_num. Затем значение @contr\_date сравнивается с сегодняшней датой при помощи функции GETDATE() и, если оно оказывается больше, чем сегодняшняя дата, на экран выводится соответствующее сообщение, а значение даты в таблице меняется на сегодняшнее число.

Код создания триггера и код вызова его срабатывания указаны в приложении. Также в приложении создание триггера изображено на рис. 47, срабатывание триггера на рис. 48.

Восьмой триггер: prot\_act\_date.

Триггер prot\_act\_date запрещает вставлять в значение даты протокола дату более раннюю, чем дата акта отбора, т.е. поддерживает соблюдение бизнес-правил испытательной лаборатории.

В триггере создаются переменные @act\_date, @prot\_date и @sample\_num, далее в @sample\_num записывается наибольшее значение sample\_num из таблицы ActProtocol из той строки, куда была только что вставлена дата протокола prot\_date. Затем в @act\_date записывается значение даты акта отбора act\_date, которая соответствует пробе с номером @sample\_num. После этого в @prot\_date сохраняется значение только что вставленной даты, и значение даты протокола в переменной @prot\_date сравнивается со значением даты акта отбора в переменной @act\_date. Если значение @act\_date больше, то выводится соответствующее сообщение и значение даты протокола обновляется до сегодняшней даты.

Код создания триггера и код вызова его срабатывания указаны в приложении. Также в приложении создание триггера изображено на рис. 49, срабатывание триггера на рис. 50.

Девятый триггер: ind\_names.

Триггер ind\_names реализует целостность между таблицами Ind\_Contr и Ind\_Prot, запрещая вставлять в Ind\_Prot значение показателя, который не анализировался по соответствующему договору.

Триггер срабатывает на вставку значений в Ind\_Prot. Сначала объявляется переменная @sample\_num и ей присваивается только что вставленное в таблицу значение, затем объявляется переменная @contr\_num и ей присваивается значение contr\_num из таблицы ActProtocol, где номер пробы sample\_num равен переменной @sample\_num, определенной ранее. Далее проверяется условие: если вставляемое название показателя не соответствует хоть какому-нибудь из названий показателей в Ind\_Contr с таким же номером договора, то об этом выводится сообщение на экран и транзацкия откатывается.

Код создания триггера и код вызова его срабатывания указаны в приложении. Также в приложении создание триггера изображено на рис. 51, срабатывание триггера на рис. 52.

### **Разработка прототипа пользовательского интерфейса**

Пользовательский интерфейс упрощает пользователю взаимодействие с базой данных, делает его более удобным и интуитивно понятным, поэтому в этом разделе будет предложен прототип пользовательского интерфейса (он не будет функционален).

Для входа в систему у каждого пользователя будут свои логин и пароль. Таким образом система защищается от несанкционированного проникновение, также это позволяет ограничить некоторым пользователям возможности с работой в базе данных и оставить им доступными только те данные, которые нужны им непосредственно в их рабочей деятельности.

Вход в систему происходит в главном меню, и это является первым действием, которое необходимо выполнить пользователям прежде чем начать работу с базой данных. В приложении на рис. 53 изображен прототип первого окна – того, что видят пользователи, только зайдя в систему.

После авторизации появляется главное меню, причем у разных пользователей оно выглядит по-разному. На главном меню расположено шесть кнопок, пять из них при нажатии откроют окна с созданными ранее таблицами: Заказчик (Customer), Договор (Contract), Показатели (Indicator), Методики (Method) и Акт отбора и протокол (ActProtocol), а также кнопка выхода из системы.

Начальнику лаборатории доступны все кнопки и все действия с таблицами в базе данных; остальным сотрудникам разрешен доступ ко всем таблицам, кроме таблицы Заказчик (Customer) с данными заказчиков. На рис. 54 изображено главное меню для начальника лаборатории. Для остальных сотрудников оно выглядит также, за исключением того, что кнопка «Заказчик» неактивна – нажатие на нее сотрудниками, не являющимися начальником лаборатории, ни к чему не приведет. Главное меню сотрудников на рис. 55.

При нажатии начальником лаборатории на кнопку «Заказчик» открывается окно на рис. 56. На рисунке схематично изображена таблица и кнопки, находящиеся в открывшемся окне. Начальник лаборатории может просмотреть данные о заказчиках, вставить, удалить или изменить их при необходимости. Это окно больше никому не доступно.

При нажатии кнопка «Договоры» открывает окно с рис. 57. В нем можно увидеть таблицы – таблицу с договорами и вспомогательную таблицу Показатель в договоре. Также начальнику лаборатории доступны кнопки внести, удалить и изменить данные. Функция подсчета суммы договора по введенному номеру доступна всем, у кого есть доступ к таблице с договорами.

Окна Показатели и Методики (открываются при нажатии на кнопки с аналогичными названиями) похожи тем, что данные в них очень редко меняются, добавляются или удаляются. Также таблицы Показатели и Методики должны быть заполнены перед началом работы в системе. На рис. 58 и 59 показаны окна Показатели и Методики соответственно с точки зрения начальника лаборатории, т.к. все кнопки доступны.

Также из главного меню можно открыть окно Акты отбора и протоколы, в нем находится основная таблица Акты отбора и протоколы и вспомогательная – Показатель в протоколе. Здесь, как и в других окнах, кнопки для изменения, вставки и удаления данных доступны только начальнику лаборатории. Окно с этими таблицами на рис. 60.

В каждом окне есть кнопка «Вернуться назад», доступная всем пользователям всегда. Она закрывает открытое ранее окно и возвращает пользователя в главное меню.

# Выводы по первой главе

В результате работы получилось создать базу данных для обеспечения организованного документооборота между заказчиком и испытательной лабораторией для лаборатории.

Данная база данных содержит информацию о показателях, внесенных в область аккредитации испытательной лаборатории (название, цена, единицы измерения, класс опасности, минимальное и максимальное нормальные значения для питьевой воды); о методиках, с помощью которых проводят исследования (их ГОСТ номер, название, необходимые приборы и посуда); о заказчиках, когда-либо обращавшихся в лабораторию (их идентификационный номер, наименование предприятия/имя, адрес, банковские реквизиты); обо всех когда-либо заключенных договорах (их номер, дату, предмет договора, количество проб); также об актах отбора и протоколах (их номер (номер пробы), дата и адрес отбора, дата протокола). Также в базе данных есть информация о показателях в каждом договоре, о методике получения их количественного значения, а также о допустимых минимальном и максимальном значениях для каждого из них; также есть информация о результатах показателей, для определения которых проводили анализ проб.

База данных не хранит информацию о суммах договоров, но при необходимости она быстро вычисляема. Для этого была создана функция, которая при вызове вычисляет сумму договора по его номеру. Также система сама заполняет максимальное и минимальное значения для показателей в договоре после ввода названий показателей и номеров договора и методики в таблицу при условии, что предметом договора является или питьевая, или природная, или сточная вода (самые востребованные для анализов типы). Для этого тоже были созданы функции, они вызываются триггерами, то есть пользователю не надо вызывать их вручную, нужно просто заполнить необходимыми данными таблицу.

Также был создан триггер, запрещающий вносить дату договора будущим числом, и триггер, запрещающий вносить в дату протокола более раннюю дату, чем дата акта отбора с тем же номером (в таком случае в дату протокола заносится сегодняшнее число).

Целостность поддерживает также триггер, запрещающий вставление в результаты протокола значения показателя, которого нет в указанном договоре.

В работе также был создан прототип пользовательского интерфейса для разных категорий пользователей. В нем сотрудники могут в зависимости от своих прав просматривать, добавлять, изменять или удалять определенные данные. Для улучшения его работы следует добавить возможность поиска по базе данных, а также добавить возможность объединять несколько таблиц по какому-либо признаку.

Созданную систему можно улучшить, добавив новую категорию пользователей – бухгалтеров, а также немного изменив систему – для этого нужно добавить счет и скрыть от бухгалтеров информацию, касающуюся непосредственно количественных значений анализируемых показателей, а также скрыть от сотрудников лаборатории, не являющихся начальником лаборатории или бухгалтерами, информацию о выписанных счетах.

Также по ходу развития системы ее можно модернизировать, добавив другие «типы» договоров – договор на определенный временной промежуток (например, на календарный год) или на определенный объем работ (например, сопровождение при запуске сложного оборудования на производстве).

**Список литературы**

ГОСТ Р 52501-2005 (ИСО 3696:1987). НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ. ВОДА ДЛЯ ЛАБОРАТОРНОГО АНАЛИЗА. Технические условия. ВНЕСЕН Техническим комитетом по стандартизации ТК 131 "Химические реактивы и особо чистые химические вещества".

ГОСТ Р 8.613-2013. НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ. Государственная система обеспечения единства измерений. МЕТОДИКИ (МЕТОДЫ) ИЗМЕРЕНИЙ СОСТАВА И СВОЙСТВ ПРОБ ВОД. Общие требования к разработке. ВНЕСЕН Техническим комитетом по стандартизации ТК 53 "Основные нормы и правила по обеспечению единства измерений".

ГОСТ 3351-74\*. Вода питьевая. Методы определения вкуса, запаха, цветности и мутности.

ГОСТ Р 8.563-2009. НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ. Государственная система обеспечения единства измерений. МЕТОДИКИ (МЕТОДЫ) ИЗМЕРЕНИЙ. ВНЕСЕН Управлением метрологии Федерального агентства по техническому регулированию и метрологии.

Муравьев А.Г. Руководство по определению показателей качества воды полевыми ме-тодами. 3-е изд., доп. и перераб. – СПб.: «Крисмас+», 2004. – 248 с.

1. Кириллов, В.В. Введение в реляционные базы данных. - М.: БХВ-Петербург, **2016**. – **318** c.;
2. Фуфаев, Э.В. Базы данных / Э.В. Фуфаев, Д.Э. Фуфаев. - М.: Академия, **2016**. – 320 c.;
3. Полтавцева, М.А. Безопасность баз данных: учеб. пособие / М.А. Полтавцева, Д.П. Зегжда, А.Ф. Супрун. – СПб: СПбПУ, 2015. – 125 с.
4. Советов Б.Я. Базы данных: учебник для прикладного бакалавриата / Б.Я. Советов, В.В. Цехановский, В.Д. Чертовской. – М.: Издательство «Юрайт», 2015. – 463 с.
5. Голицына, О.Л. Базы данных / О.Л. Голицына, Н.В. Максимов, И.И. Попов. – М.: Форум, **2015**. – 400 c.;
6. Илюшечкин, В.М. Основы использования и проектирования баз данных: учебник / В.М. Илюшечкин. – М.: Юрайт, 2015. – 516 c.;
7. Скрыпников А.В. Безопасность систем баз данных. – М.: Воронеж: ВГУИТ, 2015. – 139 с.;
8. Петкович Д. Microsoft SQL Server 2012. Руководство для начинающих. – М.: БХВ-Петербург, 2013. – 816 с.;
9. Iteam.ru – школа русского менеджмента. AllFusion Proccess Modeler. Режим доступа: <https://blog.iteam.ru/allfusion-process-modeler-bpwin/>
10. rt.draw.io – сервис для создания блок-схем, графиков, диаграмм и других визуальных объектов. Режим доступа: <https://rt.draw.io/>

# Приложение к 1 главе

1. **Проектирование базы данных**

Рисунки к главе 2:

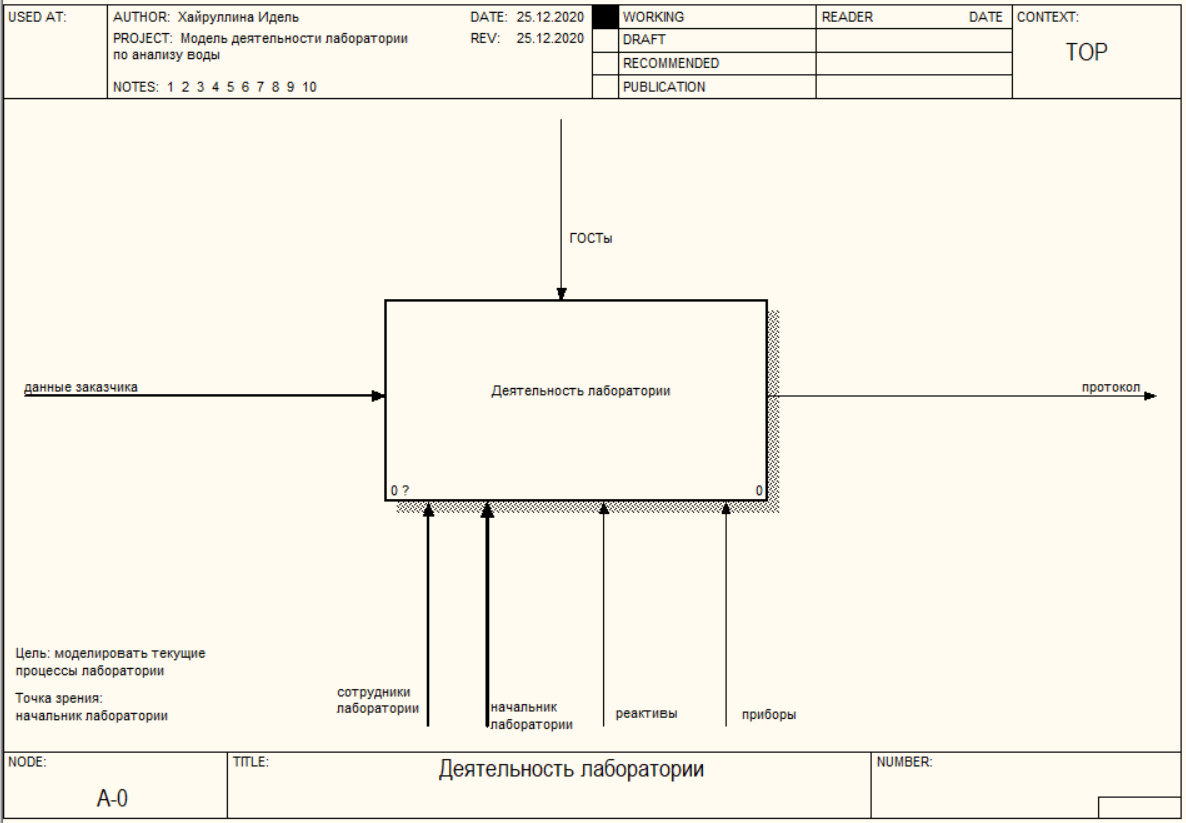


Рис. 3. Контекстная диаграмма.

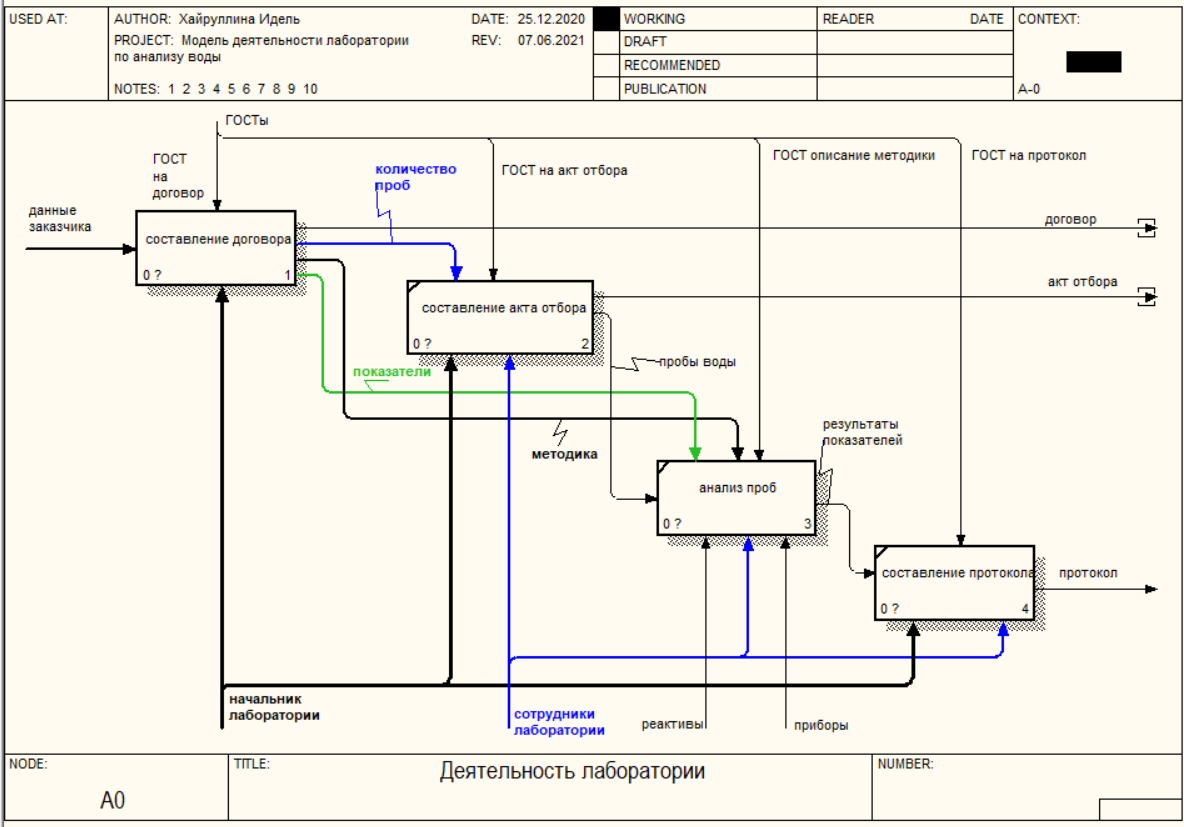


Рис. 4. Диаграмма декомпозиции первого уровня.

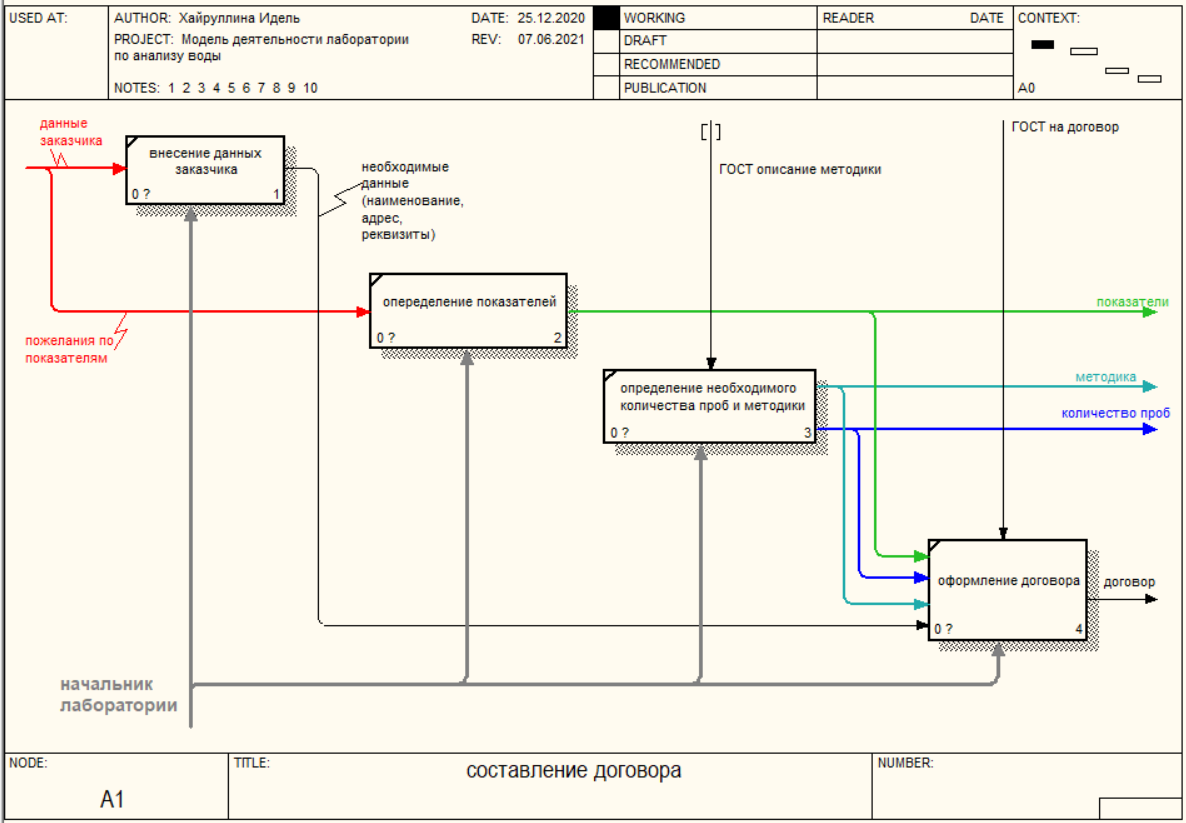


Рис. 5. Диаграмма декомпозиции второго уровня.

Таблицы к главе 2:

Таблица 1. Свойства атрибутов отношений.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Отношение | Атрибуты отношения | Тип атрибута | Свойства атрибута | Комментарий |
| Customer | cust\_id | INT | NOT NULL | Первичный ключ; счетчик. |
|  | cust\_name | CHAR(100) | NOT NULL |  |
|  | cust\_address | CHAR(100) | NULL |  |
|  | cust\_bank | CHAR(50) | NULL | Поле типа CHAR, но содержит цифры, из-за того, что с ним не нужно выполнять вычислений. |
| Contract | contr\_num | INT | NOT NULL | Первичный ключ; счетчик. |
|  | contr\_date | DATE | NOT NULL | Значение по умолчанию – сегодняшняя дата. |
|  | contr\_subj | CHAR(50) | NULL | Подстановка: поле со списком, список: «сточная», «природная», «питьевая». |
|  | sample\_amount | INT | NOT NULL |  |
| Indicator | ind\_name | CHAR(50) | NOT NULL | Первичный ключ; текстовое поле. |
|  | ind\_price | MONEY | NOT NULL |  |
|  | ed\_izm | CHAR(20) | NULL | Подстановка: поле со списком, список: «мг/дм3», «ед. pH», «˚Ж», «градусы», «баллы», «моль/дм3». |
|  | danger\_class | CHAR(10) | NULL |  |
|  | standard\_max | FLOAT | NULL |  |
|  | standard\_min | FLOAT | NULL |  |
| ActProtocol | sample\_num | INT | NOT NULL | Первичный ключ. |
|  | sample\_address | CHAR(100) | NOT NULL |  |
|  | sample\_date | DATE | NOT NULL | Значение по умолчанию – сегодняшняя дата. |
|  | prot\_date | DATE | NULL | Нет значения по умолчанию, т.к. результат может быть не готов и дата готовности неизвестна. |
| Method | method\_num | CHAR(50) | NOT NULL | Первичный ключ. |
|  | method\_name | CHAR(50) | NOT NULL |  |
|  | devices | CHAR(100) | NULL |  |
|  | tools | CHAR(100) | NULL |  |
| Ind\_Contr | contr\_max | FLOAT | NULL |  |
|  | contr\_min | FLOAT | NULL |  |
| Ind\_Prot | ind\_result | FLOAT | NULL |  |

Таблица 2. Ключи и индексы.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Отношение | Атрибут | Тип | Индекс | Комментарий |
| Customer | cust\_id | PK | Index Unique |  |
|  | cust\_name |  | Index |  |
| Contract | contr\_num | PK | Index Unique |  |
|  | contr\_date |  | Index |  |
|  | cust\_id | FK | Index | Служит для связи с отношением Customer |
| Indicator | ind\_name | PK | Index Unique |  |
|  | danger\_class |  | Index |  |
| ActProtocol | sample\_num | PK | Index Unique |  |
|  | sample\_address |  | Index |  |
|  | sample\_date |  | Index |  |
|  | contr\_num | FK | Index | Служит для связи с отношением Contract |
| Method | method\_num | PK | Index Unique |  |
|  | method\_name |  | Index |  |
| Ind\_Contr | contr\_num | FK | Index | Служит для связи с отношением Contract |
|  | method\_num | FK | Index | Служит для связи с отношением Method |
|  | ind\_name | FK | Index | Служит для связи с отношением Indicator |
| Ind\_Prot | ind\_name | FK | Index | Служит для связи с отношением Indicator |
|  | sample\_num | FK | Index | Служит для связи с отношением ActProtocol |

1. **Реализация**

Создание таблицы Customer:

Создание таблицы Customer с первичным ключом и индексом на рис. 6.

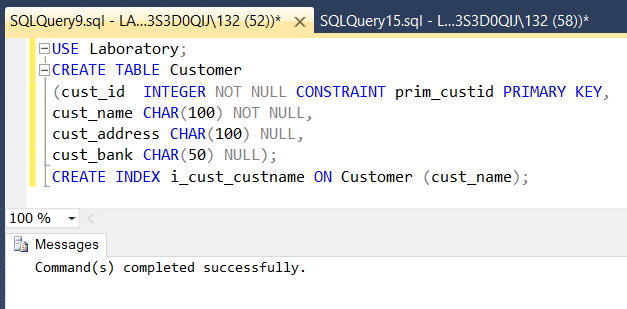


Рис. 6. Создание таблицы Customer.

Текст запроса с рис. 6:

USE Laboratory;

CREATE TABLE Customer

(сust\_id INTEGER NOT NULL CONSTRAINT prim\_custid PRIMARY KEY,

сust\_name CHAR(100) NOT NULL,

сust\_address CHAR(100) NULL,

сust\_bank CHAR(50) NULL);

CREATE INDEX i\_cust\_custname ON Customer (сust\_name);

Создание таблицы Contract:

Создание таблицы Contract с первичным и внешним ключами и необходимыми индексами на рис. 7.

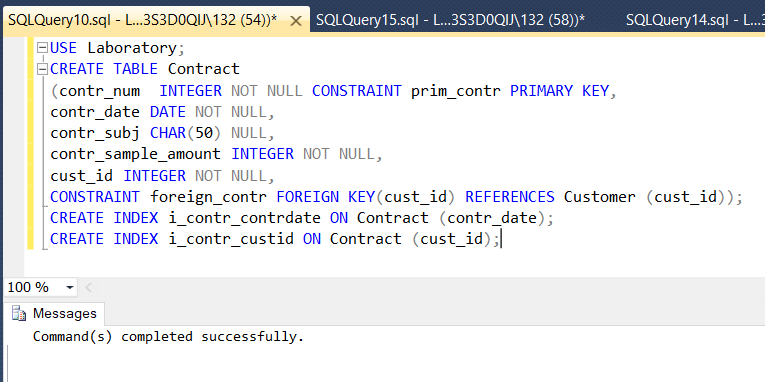


Рис. 7. Создание таблицы Contract.

Текст запроса с рис. 7:

USE Laboratory;

CREATE TABLE Cоntract

(contr\_num INTEGER NOT NULL CONSTRAINT prim\_contr PRIMARY KEY,

contr\_date DATE NOT NULL,

contr\_subj CHAR(50) NULL,

contr\_sample\_amount INTEGER NOT NULL,

сust\_id INTEGER NOT NULL,

CONSTRAINT foreign\_contr FOREIGN KEY(сust\_id) REFERENCES Customer (сust\_id));

CREATE INDEX i\_contr\_contrdate ON Cоntract (contr\_date);

CREATE INDEX i\_contr\_сustid ON Cоntract (сust\_id);

Создание таблицы Indicator:

Создание таблицы Indicator с первичным ключом и индексом на рис. 8.

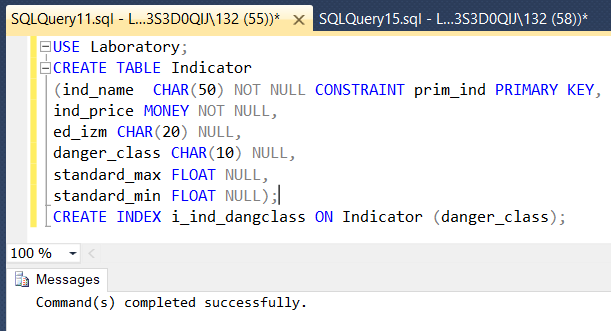


Рис. 8. Создание таблицы Indicator.

Текст запроса с рис. 8:

USE Laboratory;

CREATE TABLE Indicator

(ind\_name CHAR(50) NOT NULL CONSTRAINT prim\_ind PRIMARY KEY,

ind\_price MONEY NOT NULL,

ed\_izm CHAR(20) NULL,

danger\_class CHAR(10) NULL,

standard\_max FLOAT NULL,

standard\_min FLOAT NULL);

CREATE INDEX i\_ind\_dangclass ON Indicator (danger\_class);

Создание таблицы ActProtocol:

Создание таблицы ActProtocol с первичным и внешним ключами и необходимыми индексами на рис. 9.

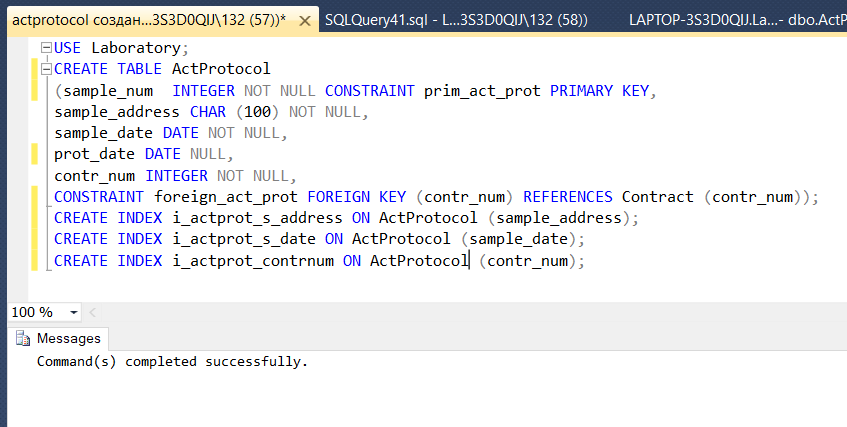


Рис. 9. Создание таблицы ActProtocol.

Текст запроса с рис. 9:

USE Laboratory;

CREATE TABLE ActProtocol

(sample\_num INTEGER NOT NULL CONSTRAINT prim\_act\_prot PRIMARY KEY,

sample\_address CHAR (100) NOT NULL,

sample\_date DATE NOT NULL,

prot\_date DATE NULL,

contr\_num INTEGER NOT NULL,

CONSTRAINT foreign\_act\_prot FOREIGN KEY (contr\_num) REFERENCES Cоntract (contr\_num));

CREATE INDEX i\_actprot\_s\_address ON ActProtocol (sample\_address);

CREATE INDEX i\_actprot\_s\_date ON ActProtocol (sample\_date);

CREATE INDEX i\_actprot\_contrnum ON ActProtocol (contr\_num);

Создание таблицы Method:

Создание таблицы Method с первичным ключом и индексом на рис. 10.

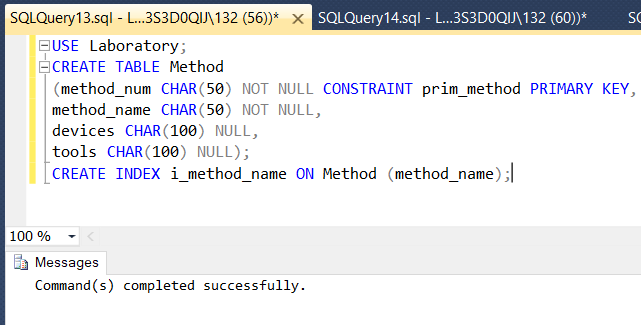


Рис. 10. Создание таблицы Method.

Текст запроса с рис. 10:

USE Laboratory;

CREATE TABLE Method

(method\_num CHAR(50) NOT NULL CONSTRAINT prim\_method PRIMARY KEY,

method\_name CHAR(50) NOT NULL,

devices CHAR(100) NULL,

tools CHAR(100) NULL);

CREATE INDEX i\_method\_name ON Method (method\_name);

Создание таблицы Ind\_Contr:

Создание таблицы Ind\_Contr с первичным и внешним ключами и необходимыми индексами на рис. 11.

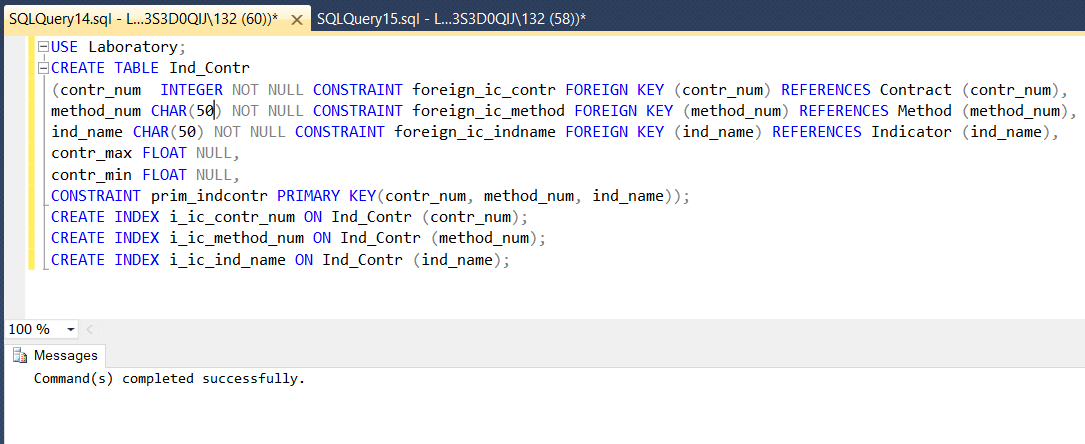


Рис. 11. Создание таблицы Ind\_Contr.

Текст запроса с рис. 11:

USE Laboratory;

CREATE TABLE Ind\_Contr

(contr\_num INTEGER NOT NULL CONSTRAINT foreign\_ic\_contr FOREIGN KEY (contr\_num) REFERENCES Cоntract (contr\_num),

method\_num CHAR(50) NOT NULL CONSTRAINT foreign\_ic\_method FOREIGN KEY (method\_num) REFERENCES Method (method\_num),

ind\_name CHAR(50) NOT NULL CONSTRAINT foreign\_ic\_indname FOREIGN KEY (ind\_name) REFERENCES Indicator (ind\_name),

contr\_max FLOAT NULL,

contr\_min FLOAT NULL,

CONSTRAINT prim\_indcontr PRIMARY KEY(contr\_num, method\_num, ind\_name));

CREATE INDEX i\_ic\_contr\_num ON Ind\_Contr (contr\_num);

CREATE INDEX i\_ic\_method\_num ON Ind\_Contr (method\_num);

CREATE INDEX i\_ic\_ind\_name ON Ind\_Contr (ind\_name);

Создание таблицы Ind\_Prot:

Создание таблицы Ind\_Prot с первичным и внешним ключами и необходимыми индексами на рис. 12.

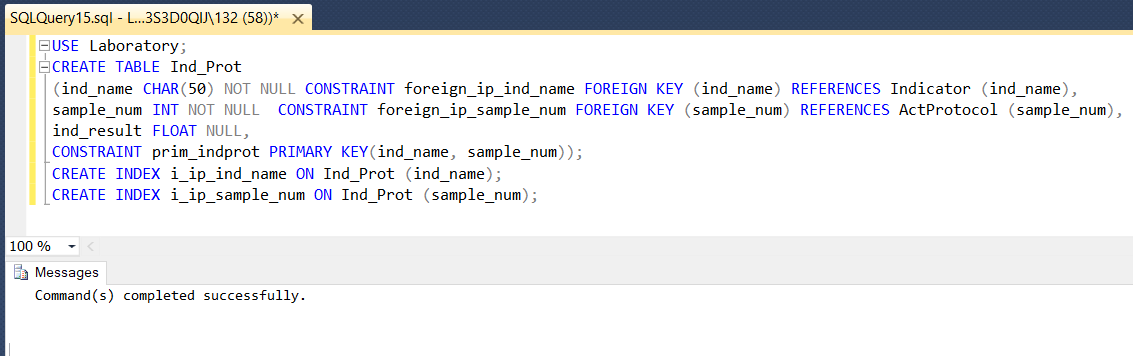


Рис. 12. Создание таблицы Ind\_Prot.

Текст запроса с рис. 12:

USE Laboratory;

CREATE TABLE Ind\_Prot

(ind\_name CHAR(50) NOT NULL CONSTRAINT foreign\_ip\_ind\_name FOREIGN KEY (ind\_name) REFERENCES Indicator (ind\_name),

sample\_num INT NOT NULL CONSTRAINT foreign\_ip\_sample\_num FOREIGN KEY (sample\_num) REFERENCES ActProtocol (sample\_num),

ind\_result FLOAT NULL,

CONSTRAINT prim\_indprot PRIMARY KEY(ind\_name, sample\_num));

CREATE INDEX i\_ip\_ind\_name ON Ind\_Prot (ind\_name);

CREATE INDEX i\_ip\_sample\_num ON Ind\_Prot (sample\_num);

На рис. 13-19 изображены заполненные данными таблицы Customer, Contract, Indicator, ActProtocol, Method, Ind\_Contr и Ind\_Prot соответственно.

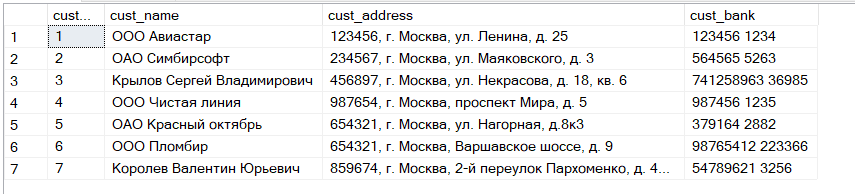


Рис. 13. Заполненная таблица Customer.

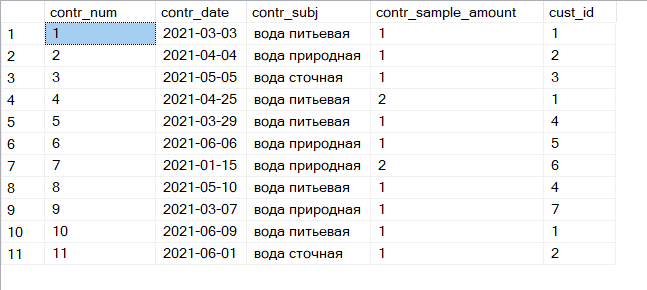


Рис. 14. Заполненная таблица Contract.

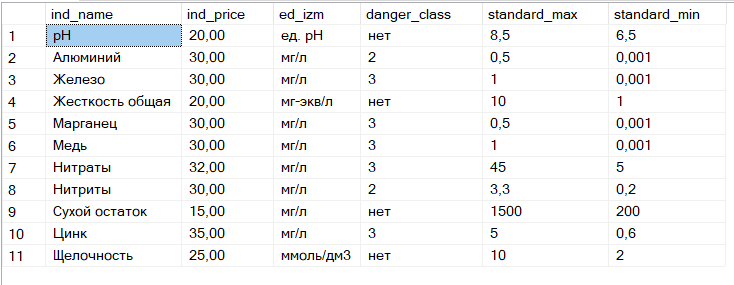


Рис. 15. Заполненная таблица Indicator.

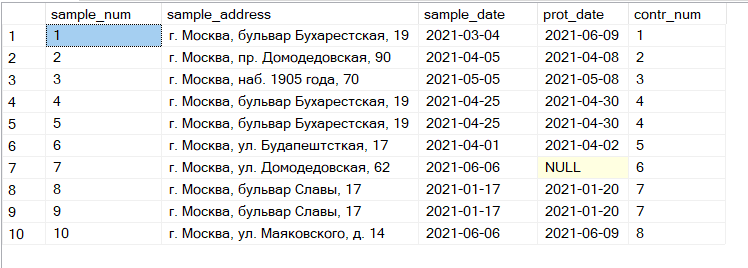


Рис. 16. Заполненная таблица ActProtocol.

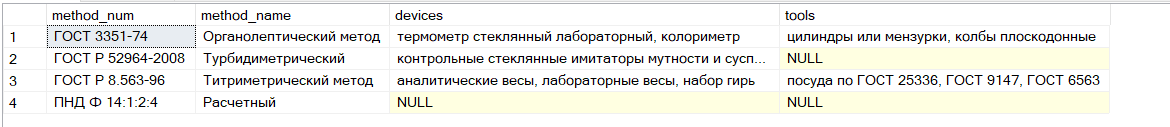


Рис. 17. Заполненная таблица Method.

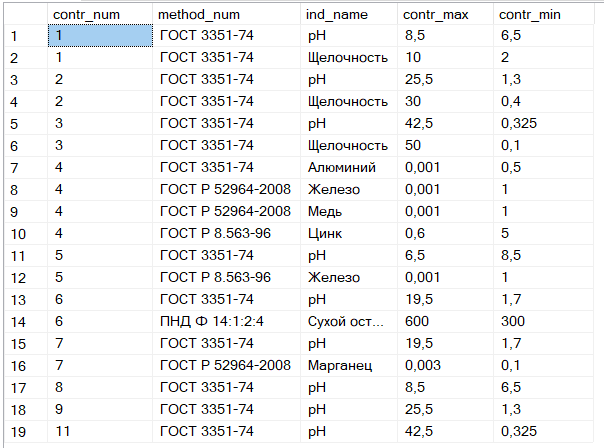


Рис. 18. Заполненная таблица Ind\_Contr.

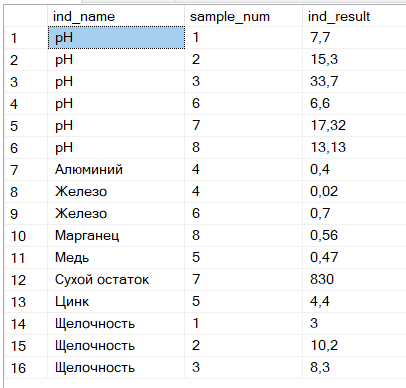


Рис. 19. Заполненная таблица Ind\_Prot.

На рис. 20 изображена диаграмма созданной базы данных Laboratory.

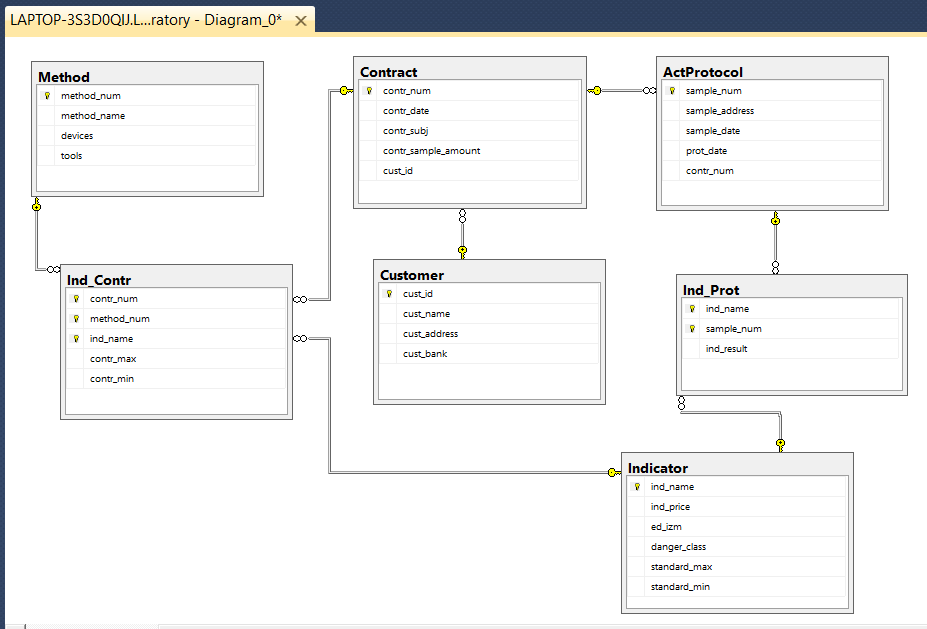


Рис. 20. Диаграмма базы данных Laboratory.

Первая хранимая процедура: contract\_sum:

Код создания процедуры:

GO

CREATE PROCEDURE contract\_sum (@contr\_num INT) AS

SELECT SUM(ind\_price) AS “Total contract sum”

FROM Indicator JOIN Ind\_Contr

ON Indicator.ind\_name=Ind\_Contr.ind\_name

WHERE Ind\_Contr.contr\_num=@contr\_num;

Создание процедуры изображено на рис. 21.

Код вызова процедуры:

EXECUTE contract\_sum 1;

EXECUTE contract\_sum 4;

EXECUTE contract\_sum 5;

Вызов процедуры изображен на рис. 22.

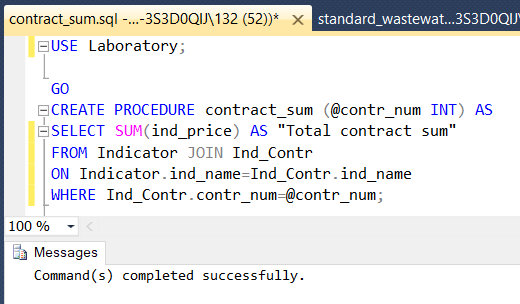


Рис. 21. Создание процедуры contract\_sum.

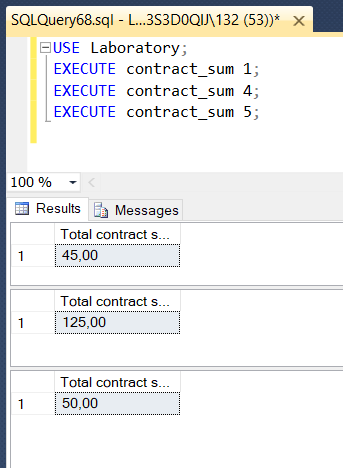


Рис. 22. Вызов процедуры contract\_sum.

Вторая хранимая процедура: standard\_drinking\_water\_min:

Код создания процедуры:

GO

CREATE PROCEDURE standard\_drinking\_water\_min

(@ind\_name\_p CHAR(50), @contr\_num\_p INT, @method\_num\_p CHAR(50)) AS

BEGIN

DECLARE @min\_drinking\_water AS FLOAT;

SELECT @min\_drinking\_water=standard\_min

FROM Indicator WHERE ind\_name=@ind\_name\_p

UPDATE Ind\_Contr SET contr\_min=@min\_drinking\_water

WHERE (ind\_name=@ind\_name\_p

AND contr\_num=@contr\_num\_p

AND method\_num=@method\_num\_p);

END;

Создание процедуры изображено на рис. 23.

Код вызова процедуры:

INSERT INTO Ind\_Contr (ind\_name, contr\_num, method\_num) VALUES ('pH', 8, 'ГОСТ 3351-74');

EXECUTE standard\_drinking\_water\_min 'pH', 8, 'ГОСТ 3351-74';

SELECT \* FROM Ind\_Contr WHERE ind\_name='pH' AND contr\_num=8 AND method\_num='ГОСТ 3351-74';

Вызов процедуры изображен на рис. 24.

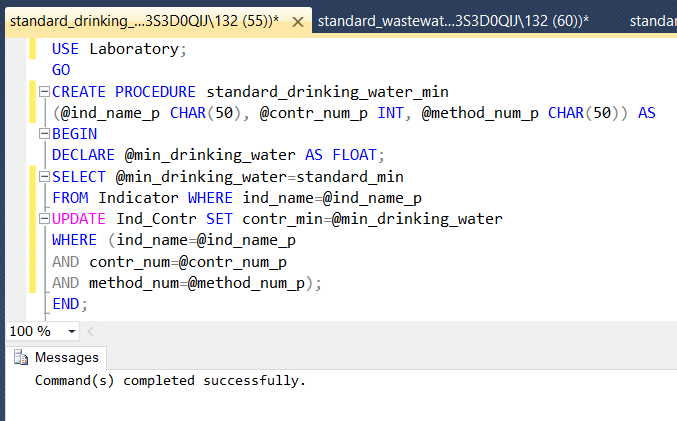


Рис. 23. Создание процедуры standard\_drinking\_water\_min.

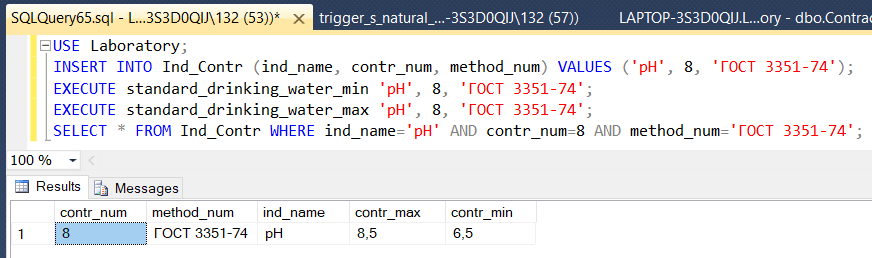


Рис. 24. Вызов процедуры standard\_drinking\_water\_min.

Третья хранимая процедура: standard\_drinking\_water\_max:

Код создания процедуры:

GO

CREATE PROCEDURE standard\_drinking\_water\_max

(@ind\_name\_p CHAR(50), @contr\_num\_p INT, @method\_num\_p CHAR(50)) AS

BEGIN

DECLARE @max\_drinking\_water AS FLOAT;

SELECT @max\_drinking\_water=standard\_max

FROM Indicator WHERE ind\_name=@ind\_name\_p

UPDATE Ind\_Contr SET contr\_max=@max\_drinking\_water

WHERE (ind\_name=@ind\_name\_p

AND contr\_num=@contr\_num\_p

AND method\_num=@method\_num\_p);

END;

Создание процедуры изображено на рис. 25.

Код вызова процедуры:

INSERT INTO Ind\_Contr (ind\_name, contr\_num, method\_num) VALUES ('pH', 8, 'ГОСТ 3351-74');

EXECUTE standard\_drinking\_water\_max 'pH', 8, 'ГОСТ 3351-74';

SELECT \* FROM Ind\_Contr WHERE ind\_name='pH' AND contr\_num=8 AND method\_num='ГОСТ 3351-74';

Вызов процедуры изображен на рис. 26.

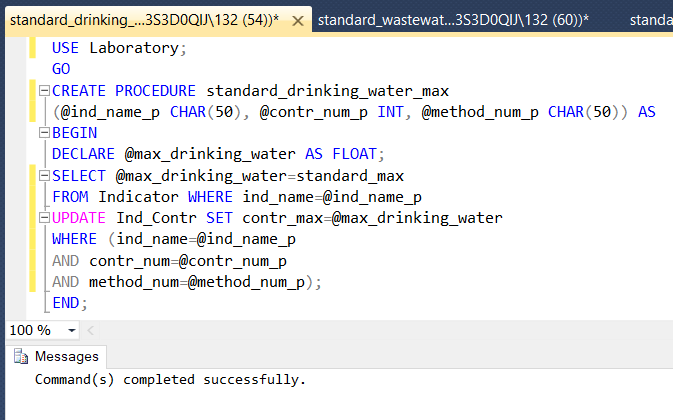


Рис. 25. Создание процедуры standard\_drinking\_water\_max.

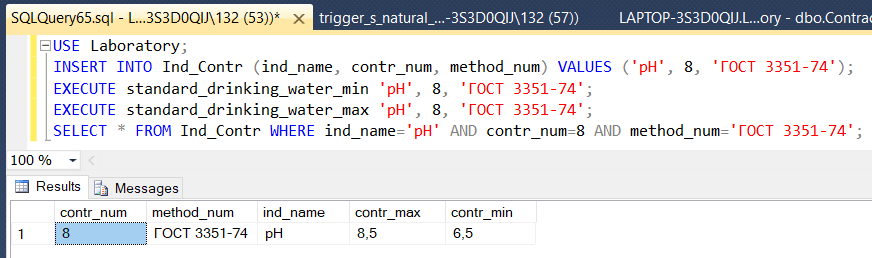


Рис. 26. Вызов процедуры standard\_drinking\_water\_max.

Четвертая хранимая процедура: standard\_natural\_water\_min:

Код создания процедуры:

GO

CREATE PROCEDURE standard\_natural\_water\_min

(@ind\_name\_p CHAR(50), @contr\_num\_p INT, @method\_num\_p CHAR(50)) AS

BEGIN

DECLARE @min\_natural\_water AS FLOAT;

SELECT @min\_natural\_water=(standard\_min\*0.2)

FROM Indicator WHERE ind\_name=@ind\_name\_p

UPDATE Ind\_Contr SET contr\_min=@min\_natural\_water

WHERE (ind\_name=@ind\_name\_p AND contr\_num=@contr\_num\_p AND method\_num=@method\_num\_p);

END;

Создание процедуры изображено на рис. 27.

Код вызова процедуры:

USE Laboratory;

INSERT INTO Ind\_Contr (ind\_name, contr\_num, method\_num) VALUES ('pH', 9, 'ГОСТ 3351-74');

EXECUTE standard\_natural\_water\_min 'pH', 9, 'ГОСТ 3351-74';

SELECT \* FROM Ind\_Contr WHERE ind\_name='pH' AND contr\_num=9 AND method\_num='ГОСТ 3351-74';

Вызов процедуры изображен на рис. 28.

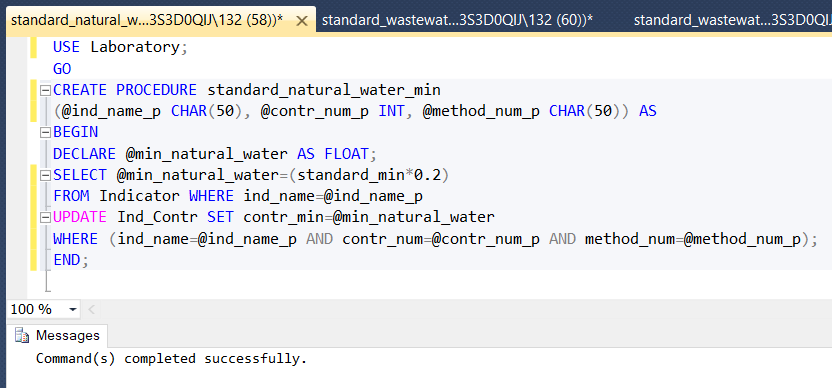


Рис. 27. Создание процедуры standard\_natural\_water\_min.

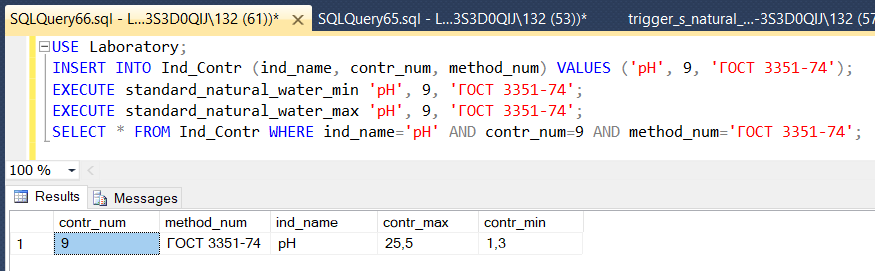


Рис. 28. Вызов процедуры standard\_natural\_water\_min.

Пятая хранимая процедура: standard\_natural\_water\_max:

Код создания процедуры:

GO

CREATE PROCEDURE standard\_natural\_water\_max

(@ind\_name\_p CHAR(50), @contr\_num\_p INT, @method\_num\_p CHAR(50)) AS

BEGIN

DECLARE @max\_natural\_water AS FLOAT;

SELECT @max\_natural\_water=(standard\_max\*3)

FROM Indicator WHERE ind\_name=@ind\_name\_p

UPDATE Ind\_Contr SET contr\_max=@max\_natural\_water

WHERE (ind\_name=@ind\_name\_p AND contr\_num=@contr\_num\_p AND method\_num=@method\_num\_p);

END

Создание процедуры изображено на рис. 29.

Код вызова процедуры:

USE Laboratory;

INSERT INTO Ind\_Contr (ind\_name, contr\_num, method\_num) VALUES ('pH', 9, 'ГОСТ 3351-74');

EXECUTE standard\_natural\_water\_max 'pH', 9, 'ГОСТ 3351-74';

SELECT \* FROM Ind\_Contr WHERE ind\_name='pH' AND contr\_num=9 AND method\_num='ГОСТ 3351-74';

Вызов процедуры изображен на рис. 30.

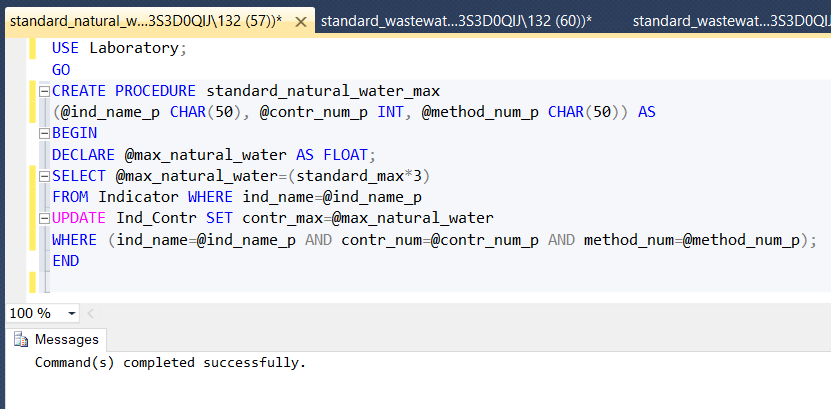


Рис. 29. Создание процедуры standard\_natural\_water\_max.

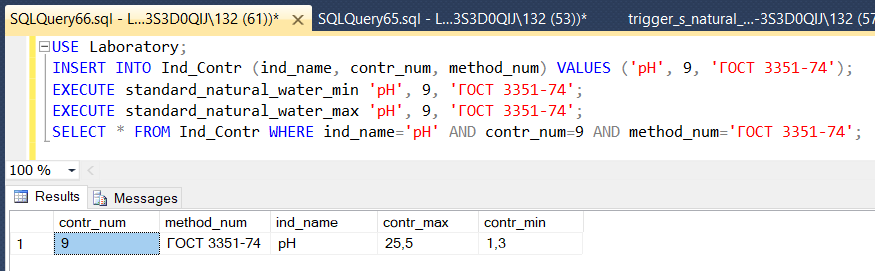


Рис. 30. Вызов процедуры standard\_natural\_water\_max.

Шестая хранимая процедура: standard\_wastewater\_min:

Код создания процедуры:

GO

CREATE PROCEDURE standard\_wastewater\_min

(@ind\_name\_p CHAR(50), @contr\_num\_p INT, @method\_num\_p CHAR(50)) AS

BEGIN

DECLARE @min\_wastewater AS FLOAT;

SELECT @min\_wastewater=(standard\_min\*0.05)

FROM Indicator WHERE ind\_name=@ind\_name\_p

UPDATE Ind\_Contr SET contr\_min=@min\_wastewater

WHERE (ind\_name=@ind\_name\_p

AND contr\_num=@contr\_num\_p

AND method\_num=@method\_num\_p);

END

Создание процедуры изображено на рис. 31.

Код вызова процедуры:

USE Laboratory;

INSERT INTO Ind\_Contr (ind\_name, contr\_num, method\_num) VALUES ('pH', 11, 'ГОСТ 3351-74');

EXECUTE standard\_wastewater\_min'pH', 11, 'ГОСТ 3351-74';

SELECT \* FROM Ind\_Contr WHERE ind\_name='pH' AND contr\_num=11 AND method\_num='ГОСТ 3351-74';

Вызов процедуры изображен на рис. 32.

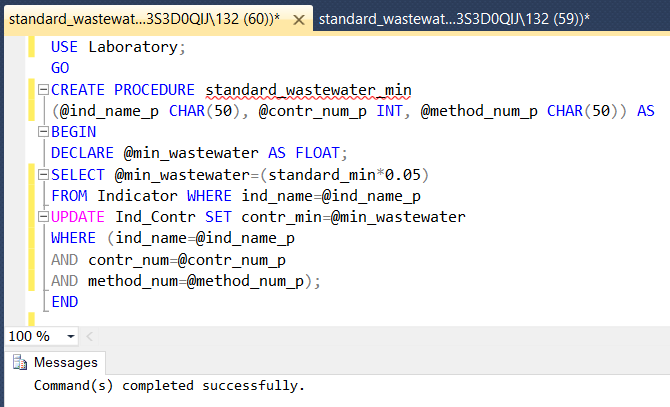


Рис. 31. Создание процедуры standard\_wastewater\_min.

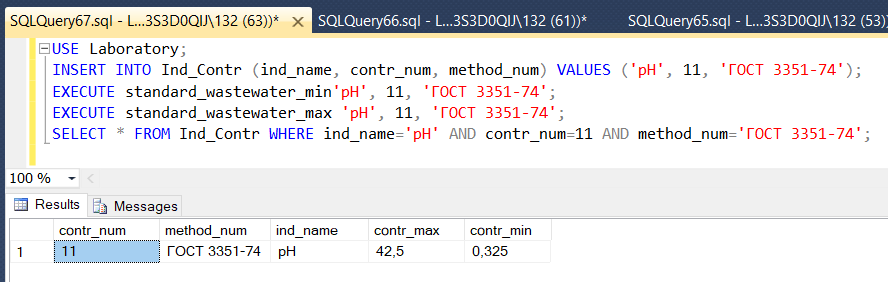


Рис. 32. Вызов процедуры standard\_wastewater\_min.

Седьмая хранимая процедура: standard\_wastewater\_max:

Код создания процедуры:

GO

CREATE PROCEDURE standard\_wastewater\_max

(@ind\_name\_p CHAR(50), @contr\_num\_p INT, @method\_num\_p CHAR(50)) AS

BEGIN

DECLARE @max\_natural\_water AS FLOAT;

SELECT @max\_natural\_water=(standard\_max\*3)

FROM Indicator WHERE ind\_name=@ind\_name\_p

UPDATE Ind\_Contr SET contr\_max=@max\_natural\_water

WHERE (ind\_name=@ind\_name\_p AND contr\_num=@contr\_num\_p AND method\_num=@method\_num\_p);

END

Создание процедуры изображено на рис. 33.

Код вызова процедуры:

USE Laboratory;

INSERT INTO Ind\_Contr (ind\_name, contr\_num, method\_num) VALUES ('pH', 11, 'ГОСТ 3351-74');

EXECUTE standard\_wastewater\_max 'pH', 11, 'ГОСТ 3351-74';

SELECT \* FROM Ind\_Contr WHERE ind\_name='pH' AND contr\_num=11 AND method\_num='ГОСТ 3351-74';

Вызов процедуры изображен на рис. 34.

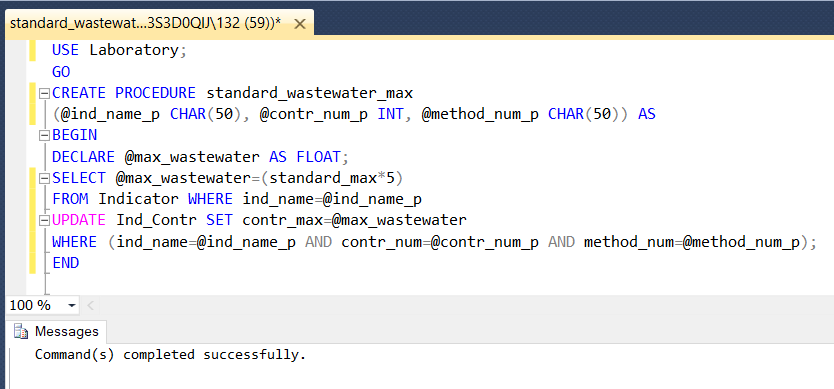


Рис. 33. Создание процедуры standard\_wastewater\_max.

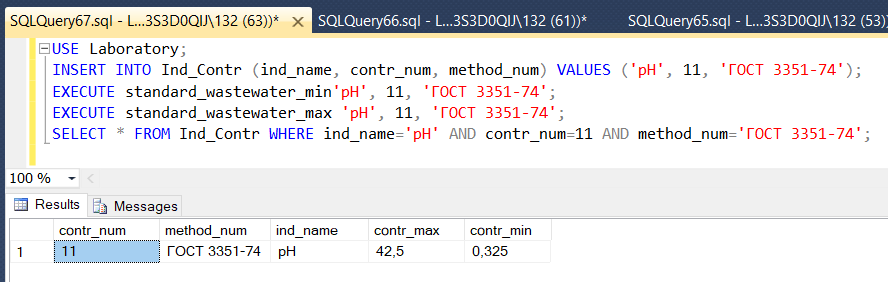


Рис. 34. Вызов процедуры standard\_wastewater\_max.

Первый триггер: trigger\_s\_drinking\_water\_min.

Код создания триггера:

GO

CREATE TRIGGER trigger\_s\_drinking\_water\_min ON Ind\_Contr

AFTER INSERT AS

BEGIN

DECLARE @ind\_name CHAR(50);

SELECT @ind\_name=(SELECT ind\_name FROM inserted);

DECLARE @method\_num CHAR(50);

SELECT @method\_num=(SELECT method\_num FROM inserted);

DECLARE @contr\_num INT;

SELECT @contr\_num=(SELECT contr\_num FROM inserted);

DECLARE @contr\_subj CHAR(50);

SELECT @contr\_subj = (SELECT Cоntract.contr\_subj

FROM Cоntract JOIN Ind\_Contr ON Cоntract.contr\_num=Ind\_Contr.contr\_num

WHERE Cоntract.contr\_num=@contr\_num AND Ind\_Contr.method\_num=@method\_num AND Ind\_Contr.ind\_name=@ind\_name);

IF @contr\_subj='вода питьевая'

BEGIN

EXECUTE standard\_drinking\_water\_min @ind\_name, @contr\_num, @method\_num;

END;

END;

Создание триггера изображено на рис. 35.

Код для вызова срабатывания триггера:

INSERT INTO Ind\_Contr (contr\_num, method\_num, ind\_name) VALUES (1, 'ГОСТ 3351-74', 'Щелочность');

INSERT INTO Ind\_Contr (contr\_num, method\_num, ind\_name) VALUES (1, 'ГОСТ 3351-74', 'pH');

SELECT \* FROM Ind\_Contr WHERE contr\_num=1;

Срабатывание триггера на рис. 36.

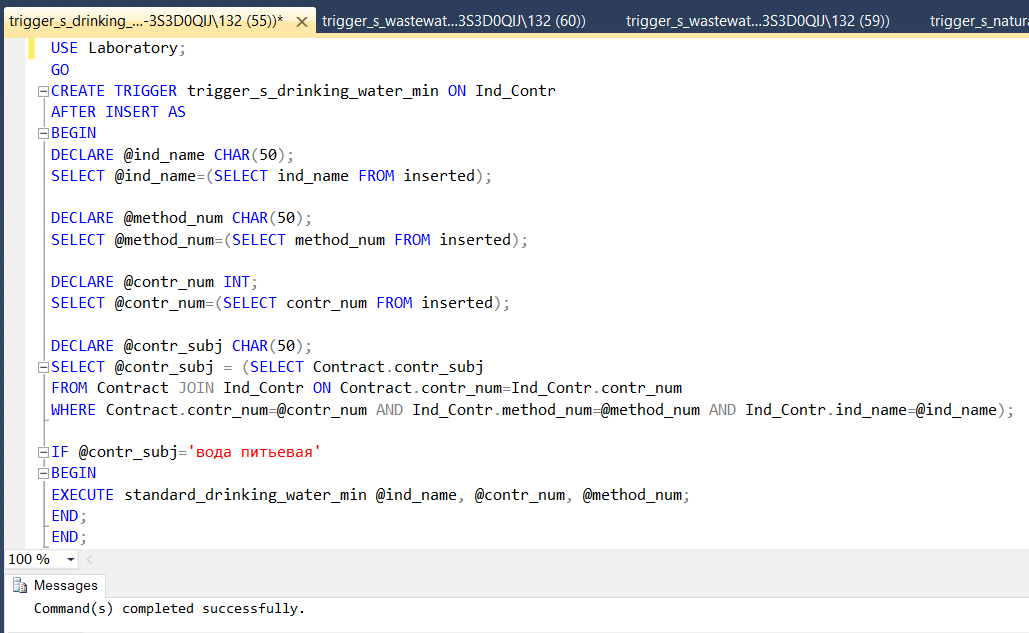


Рис. 35. Создание триггера trigger\_s\_drinking\_water\_min.

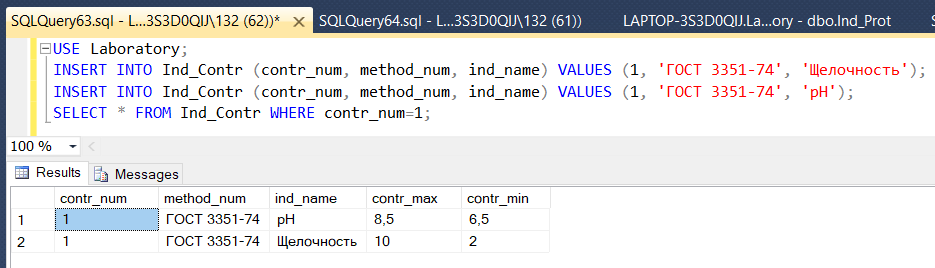


Рис. 36. Срабатывание триггера trigger\_s\_drinking\_water\_min.

Второй триггер: trigger\_s\_drinking\_water\_max.

Код создания триггера:

GO

CREATE TRIGGER trigger\_s\_drinking\_water\_max ON Ind\_Contr

AFTER INSERT AS

BEGIN

DECLARE @ind\_name CHAR(50);

SELECT @ind\_name=(SELECT ind\_name FROM inserted);

DECLARE @method\_num CHAR(50);

SELECT @method\_num=(SELECT method\_num FROM inserted);

DECLARE @contr\_num INT;

SELECT @contr\_num=(SELECT contr\_num FROM inserted);

DECLARE @contr\_subj CHAR(50);

SELECT @contr\_subj = (SELECT Cоntract.contr\_subj

FROM Cоntract JOIN Ind\_Contr ON Cоntract.contr\_num=Ind\_Contr.contr\_num

WHERE Cоntract.contr\_num=@contr\_num AND Ind\_Contr.method\_num=@method\_num AND Ind\_Contr.ind\_name=@ind\_name);

IF @contr\_subj='вода питьевая'

BEGIN

EXECUTE standard\_drinking\_water\_max @ind\_name, @contr\_num, @method\_num;

END;

END;

Создание триггера изображено на рис. 37.

Код для вызова срабатывания триггера:

INSERT INTO Ind\_Contr (contr\_num, method\_num, ind\_name) VALUES (1, 'ГОСТ 3351-74', 'Щелочность');

INSERT INTO Ind\_Contr (contr\_num, method\_num, ind\_name) VALUES (1, 'ГОСТ 3351-74', 'pH');

SELECT \* FROM Ind\_Contr WHERE contr\_num=1;

Срабатывание триггера на рис. 38.

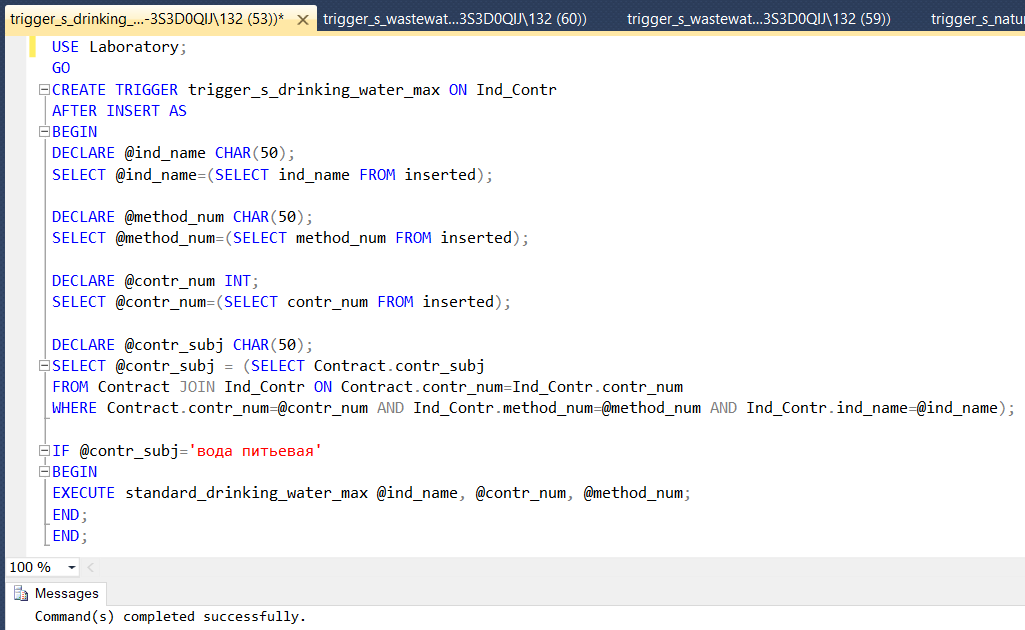


Рис. 37. Создание триггера trigger\_s\_drinking\_water\_max.

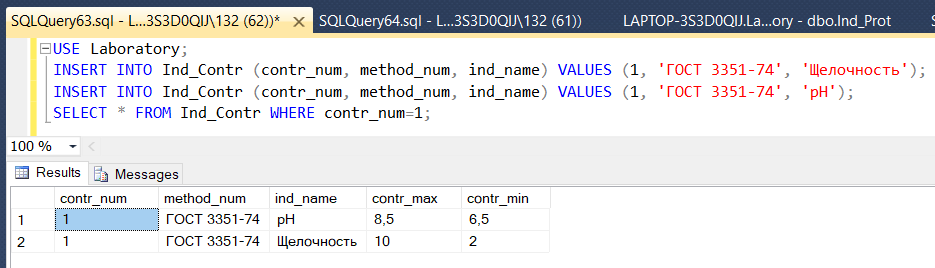


Рис. 38. Срабатывание триггера trigger\_s\_drinking\_water\_max.

Третий триггер: trigger\_s\_natural\_water\_min.

Код создания триггера:

GO

CREATE TRIGGER trigger\_s\_natural\_water\_min ON Ind\_Contr

AFTER INSERT AS

BEGIN

DECLARE @ind\_name CHAR(50);

SELECT @ind\_name=(SELECT ind\_name FROM inserted);

DECLARE @method\_num CHAR(50);

SELECT @method\_num=(SELECT method\_num FROM inserted);

DECLARE @contr\_num INT;

SELECT @contr\_num=(SELECT contr\_num FROM inserted);

DECLARE @contr\_subj CHAR(50);

SELECT @contr\_subj = (SELECT Cоntract.contr\_subj

FROM Cоntract JOIN Ind\_Contr ON Cоntract.contr\_num=Ind\_Contr.contr\_num

WHERE Cоntract.contr\_num=@contr\_num AND Ind\_Contr.method\_num=@method\_num AND Ind\_Contr.ind\_name=@ind\_name);

IF @contr\_subj='вода природная'

BEGIN

EXECUTE standard\_natural\_water\_min @ind\_name, @contr\_num, @method\_num;

END;

END;

Создание триггера изображено на рис. 39.

Код для вызова срабатывания триггера:

INSERT INTO Ind\_Contr (contr\_num, method\_num, ind\_name) VALUES (2, 'ГОСТ 3351-74', 'Щелочность');

INSERT INTO Ind\_Contr (contr\_num, method\_num, ind\_name) VALUES (2, 'ГОСТ 3351-74', 'pH');

SELECT \* FROM Ind\_Contr WHERE contr\_num=2;

Срабатывание триггера на рис. 40.

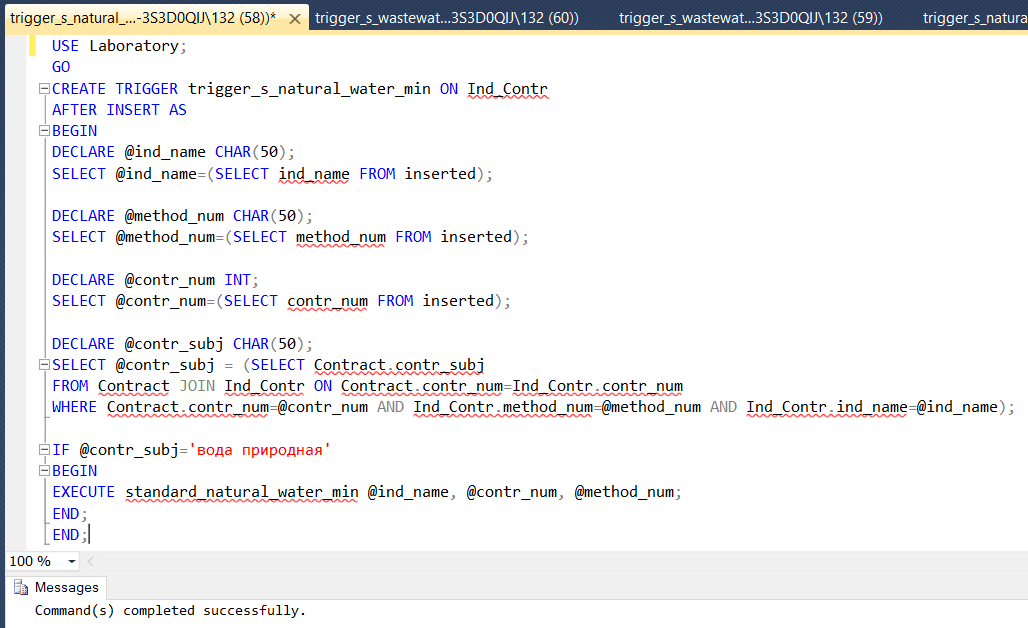


Рис. 39. Создание триггера trigger\_s\_natural\_water\_min.

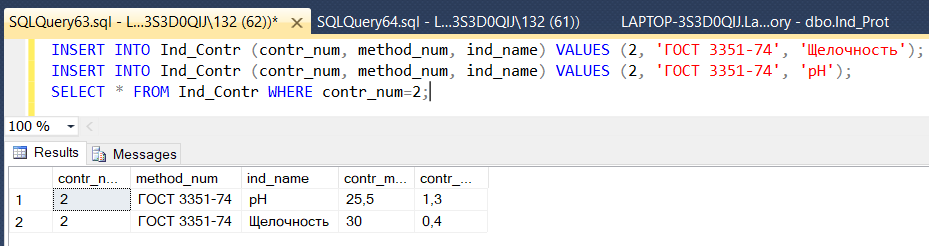


Рис. 40. Срабатывание триггера trigger\_s\_natural\_water\_min.

Четвертый триггер: trigger\_s\_natural\_water\_max.

Код создания триггера:

GO

CREATE TRIGGER trigger\_s\_natural\_water\_max ON Ind\_Contr

AFTER INSERT AS

BEGIN

DECLARE @ind\_name CHAR(50);

SELECT @ind\_name=(SELECT ind\_name FROM inserted);

DECLARE @method\_num CHAR(50);

SELECT @method\_num=(SELECT method\_num FROM inserted);

DECLARE @contr\_num INT;

SELECT @contr\_num=(SELECT contr\_num FROM inserted);

DECLARE @contr\_subj CHAR(50);

SELECT @contr\_subj = (SELECT Cоntract.contr\_subj

FROM Cоntract JOIN Ind\_Contr ON Cоntract.contr\_num=Ind\_Contr.contr\_num

WHERE Cоntract.contr\_num=@contr\_num AND Ind\_Contr.method\_num=@method\_num AND Ind\_Contr.ind\_name=@ind\_name);

IF @contr\_subj='вода природная'

BEGIN

EXECUTE standard\_natural\_water\_max @ind\_name, @contr\_num, @method\_num;

END;

END;

Создание триггера изображено на рис. 41.

Код для вызова срабатывания триггера:

INSERT INTO Ind\_Contr (contr\_num, method\_num, ind\_name) VALUES (2, 'ГОСТ 3351-74', 'Щелочность');

INSERT INTO Ind\_Contr (contr\_num, method\_num, ind\_name) VALUES (2, 'ГОСТ 3351-74', 'pH');

SELECT \* FROM Ind\_Contr WHERE contr\_num=2;

Срабатывание триггера на рис. 42.

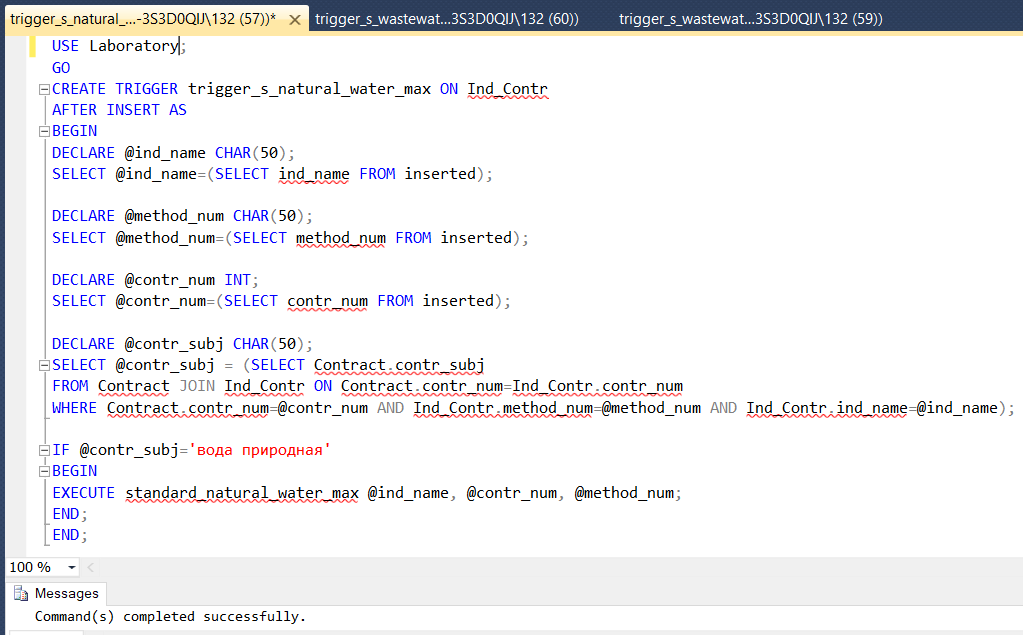


Рис. 41. Создание триггера trigger\_s\_natural\_water\_max.

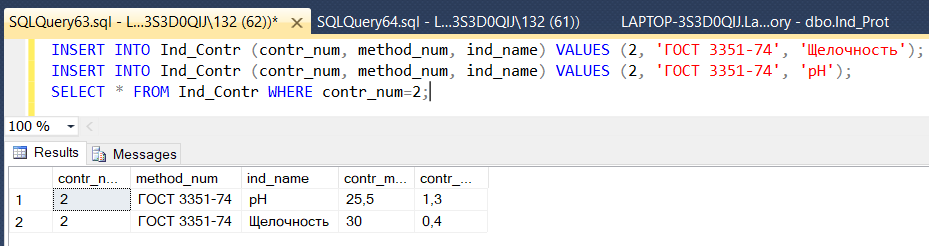


Рис. 42. Срабатывание триггера trigger\_s\_natural\_water\_max.

Пятый триггер: trigger\_s\_wastewater\_min.

Код создания триггера:

GO

CREATE TRIGGER trigger\_s\_wastewater\_min ON Ind\_Contr

AFTER INSERT AS

BEGIN

DECLARE @ind\_name CHAR(50);

SELECT @ind\_name=(SELECT ind\_name FROM inserted);

DECLARE @method\_num CHAR(50);

SELECT @method\_num=(SELECT method\_num FROM inserted);

DECLARE @contr\_num INT;

SELECT @contr\_num=(SELECT contr\_num FROM inserted);

DECLARE @contr\_subj CHAR(50);

SELECT @contr\_subj = (SELECT Cоntract.contr\_subj

FROM Cоntract JOIN Ind\_Contr ON Cоntract.contr\_num=Ind\_Contr.contr\_num

WHERE Cоntract.contr\_num=@contr\_num AND Ind\_Contr.method\_num=@method\_num AND Ind\_Contr.ind\_name=@ind\_name);

IF @contr\_subj='вода сточная'

BEGIN

EXECUTE standard\_wastewater\_min @ind\_name, @contr\_num, @method\_num;

END;

END;

Создание триггера изображено на рис. 43.

Код для вызова срабатывания триггера:

INSERT INTO Ind\_Contr (contr\_num, method\_num, ind\_name) VALUES (3, 'ГОСТ 3351-74', 'Щелочность');

INSERT INTO Ind\_Contr (contr\_num, method\_num, ind\_name) VALUES (3, 'ГОСТ 3351-74', 'pH');

SELECT \* FROM Ind\_Contr WHERE contr\_num=3;

Срабатывание триггера на рис. 44.

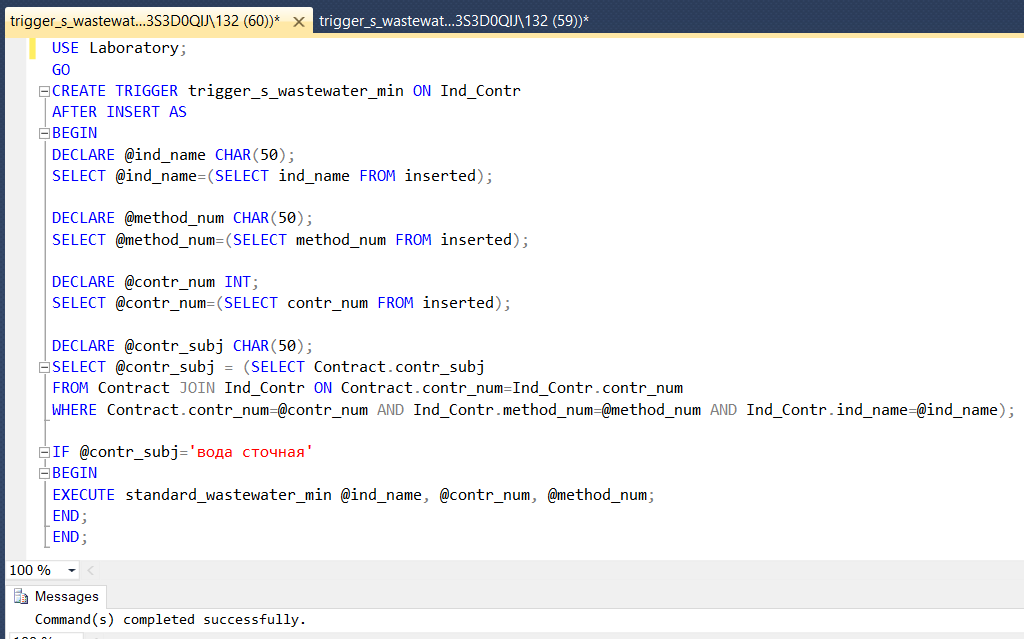


Рис. 43. Создание триггера trigger\_s\_wastewater\_min.

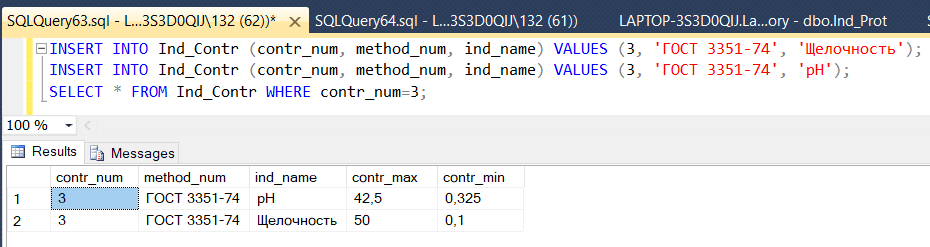


Рис. 44. Срабатывание триггера trigger\_s\_wastewater\_min.

Шестой триггер: trigger\_s\_wastewater\_max.

Код создания триггера:

GO

CREATE TRIGGER trigger\_s\_wastewater\_max ON Ind\_Contr

AFTER INSERT AS

BEGIN

DECLARE @ind\_name CHAR(50);

SELECT @ind\_name=(SELECT ind\_name FROM inserted);

DECLARE @method\_num CHAR(50);

SELECT @method\_num=(SELECT method\_num FROM inserted);

DECLARE @contr\_num INT;

SELECT @contr\_num=(SELECT contr\_num FROM inserted);

DECLARE @contr\_subj CHAR(50);

SELECT @contr\_subj = (SELECT Cоntract.contr\_subj

FROM Cоntract JOIN Ind\_Contr ON Cоntract.contr\_num=Ind\_Contr.contr\_num

WHERE Cоntract.contr\_num=@contr\_num AND Ind\_Contr.method\_num=@method\_num AND Ind\_Contr.ind\_name=@ind\_name);

IF @contr\_subj='вода сточная'

BEGIN

EXECUTE standard\_wastewater\_max @ind\_name, @contr\_num, @method\_num;

END;

END;

Создание триггера изображено на рис. 45.

Код для вызова срабатывания триггера:

INSERT INTO Ind\_Contr (contr\_num, method\_num, ind\_name) VALUES (3, 'ГОСТ 3351-74', 'Щелочность');

INSERT INTO Ind\_Contr (contr\_num, method\_num, ind\_name) VALUES (3, 'ГОСТ 3351-74', 'pH');

SELECT \* FROM Ind\_Contr WHERE contr\_num=3;

Срабатывание триггера на рис. 46.

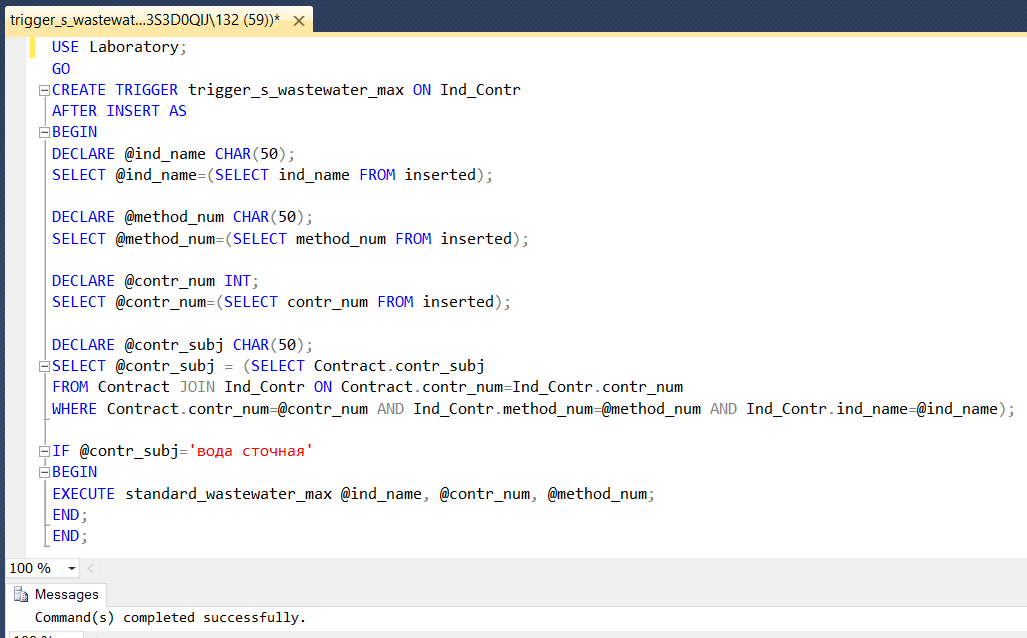


Рис. 45. Создание триггера trigger\_s\_wastewater\_max.

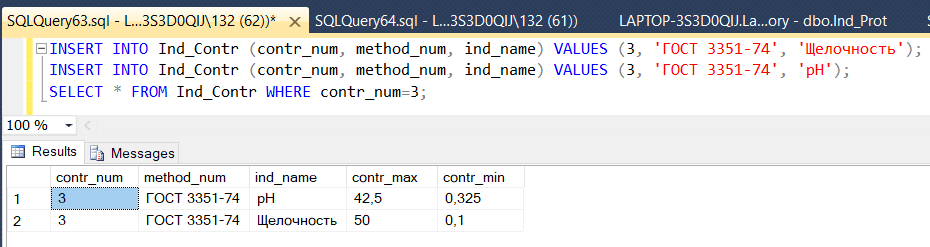


Рис. 46. Срабатывание триггера trigger\_s\_wastewater\_max.

Седьмой триггер: contract\_date.

Код создания триггера:

GO

CREATE TRIGGER contract\_date ON Cоntract

AFTER INSERT AS

BEGIN

DECLARE

@contr\_num INT,

@contr\_date DATE;

SET @contr\_num = (SELECT MAX(contr\_num) FROM Cоntract)

SET @contr\_date = (SELECT contr\_date FROM Cоntract WHERE contr\_num=@contr\_num)

IF @contr\_date > GETDATE()

BEGIN

PRINT 'Дата составления договора не может быть позже сегодняшней даты!'

UPDATE Cоntract SET contr\_date=GETDATE() WHERE contr\_num=@contr\_num

END

END

Создание триггера изображено на рис. 47.

Код для вызова срабатывания триггера:

INSERT INTO Cоntract (contr\_num, contr\_date, contr\_subj, contr\_sample\_amount, сust\_id) VALUES (1, '06-20-2027', 'вода питьевая, 1, 1);

SELECT \* FROM Cоntract;

Срабатывание триггера на рис. 48.

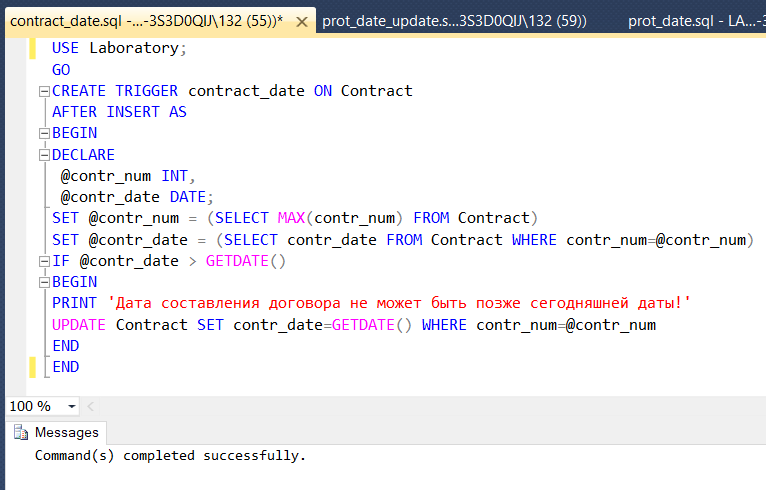


Рис. 47. Создание триггера contract\_date.

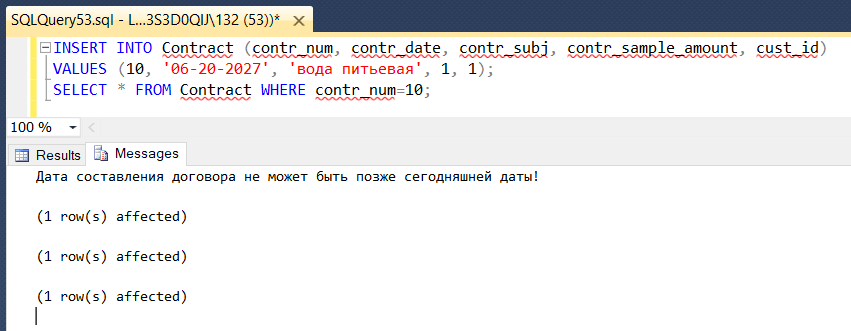


Рис. 48, а. Срабатывание триггера contract\_date.

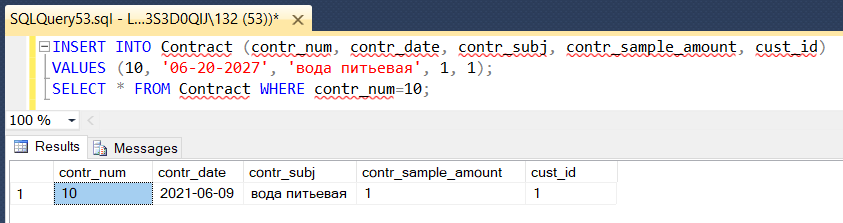


Рис. 48, б. Срабатывание триггера contract\_date.

Восьмой триггер: prot\_act\_date.

Код создания триггера:

GO

CREATE TRIGGER prot\_act\_date ON ActProtocol

AFTER INSERT, UPDATE AS

BEGIN

DECLARE @act\_date DATE;

DECLARE @prot\_date DATE;

DECLARE @sample\_num INT;

SET @sample\_num = (SELECT sample\_num FROM ActProtocol WHERE prot\_date=(SELECT prot\_date FROM inserted));

SET @act\_date = (SELECT sample\_date FROM ActProtocol WHERE sample\_num=@sample\_num);

SET @prot\_date = (SELECT prot\_date FROM inserted);

IF @prot\_date < @act\_date

BEGIN

PRINT 'Дата составления протокола не может быть раньше даты составления акта отбора!'

UPDATE ActProtocol SET prot\_date=GETDATE() WHERE sample\_num=@sample\_num;

END

END

Создание триггера изображено на рис. 49.

Код для вызова срабатывания триггера:

INSERT INTO ActProtocol (sample\_num, sample\_address, sample\_date, prot\_date, contr\_num)

VALUES (1, 'г. Москва, ул. Маяковского, д. 14', '02-02-2021', '01-01-2021', 1);

UPDATE ActProtocol SET prot\_date='01-01-2021' WHERE sample\_num = 1;

SELECT \* FROM ActProtocol;

Срабатывание триггера на рис. 50.

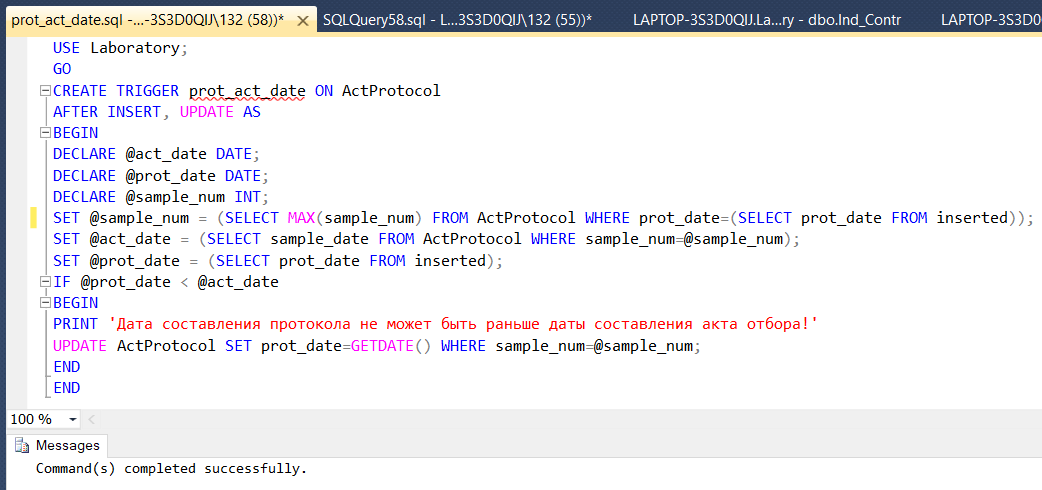


Рис. 49. Создание триггера prot\_act\_date.

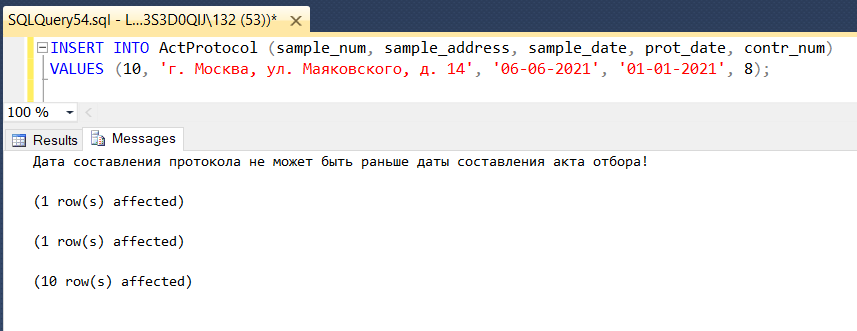


Рис. 50, а. Срабатывание триггера prot\_act\_date.

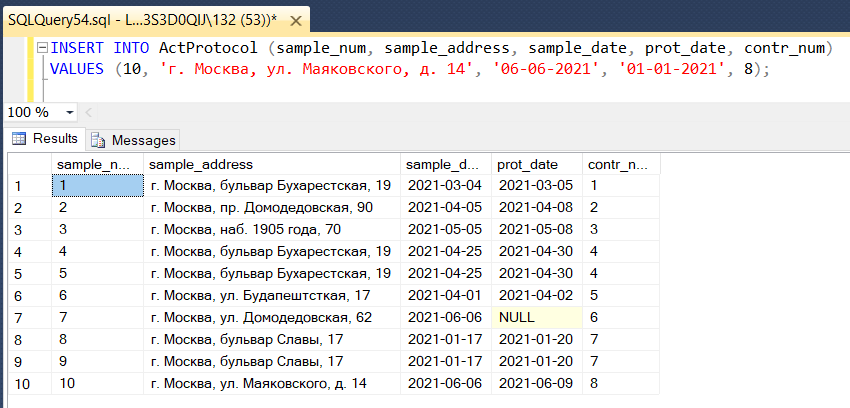


Рис. 50, б. Срабатывание триггера prot\_act\_date.

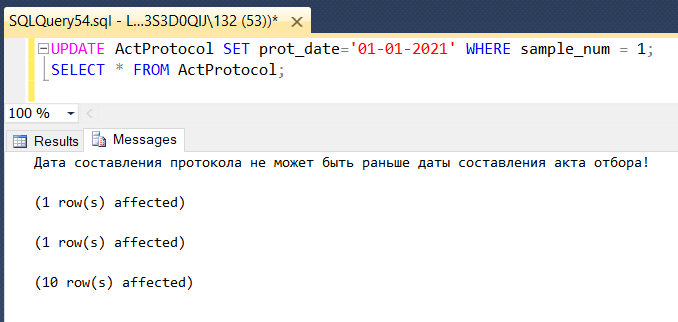


Рис. 50, в. Срабатывание триггера prot\_act\_date.

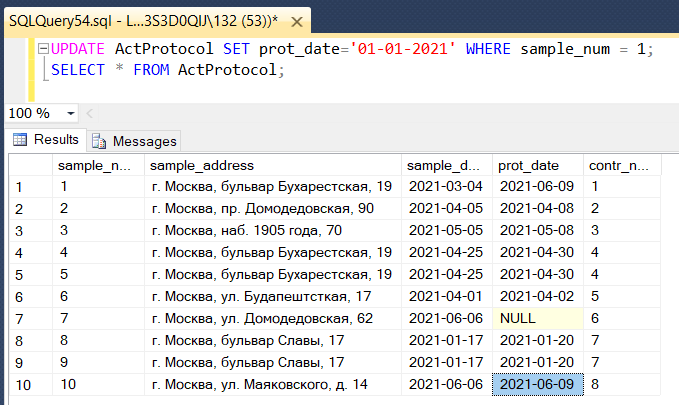


Рис. 50, г. Срабатывание триггера prot\_act\_date.

Девятый триггер: ind\_names.

Код создания триггера:

USE Laboratory;

GO

CREATE TRIGGER ind\_names ON Ind\_Prot AFTER INSERT AS

BEGIN

DECLARE @sample\_num INT;

SELECT @sample\_num=(SELECT sample\_num FROM inserted);

DECLARE @contr\_num INT;

SELECT @contr\_num = (SELECT contr\_num FROM ActProtocol WHERE sample\_num=@sample\_num);

IF (SELECT ind\_name FROM inserted WHERE sample\_num=@sample\_num) !=

ANY (SELECT ind\_name FROM Ind\_Contr WHERE contr\_num=@contr\_num)

PRINT 'Этот показатель не анализировался по указанному договору!'

ROLLBACK

END

Создание триггера изображено на рис. 51.

Код для вызова срабатывания триггера:

INSERT INTO Ind\_Prot (sample\_num, ind\_name, ind\_result)

VALUES (8, 'Сухой остаток', 700);

Срабатывание триггера на рис. 52.

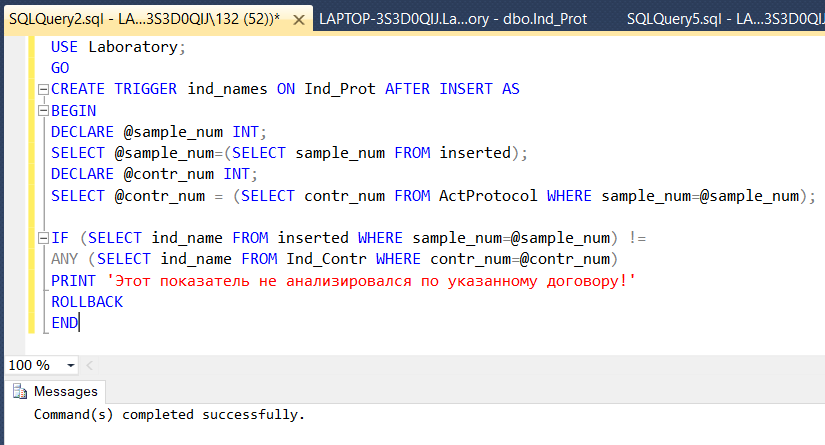


Рис. 51. Создание триггера prot\_act\_date.

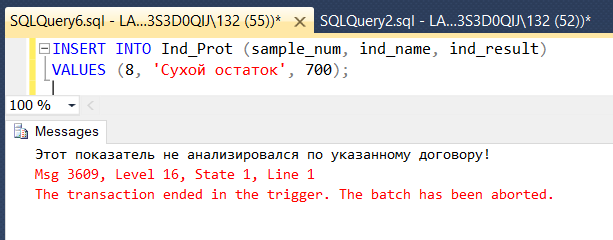


Рис. 52. Срабатывание триггера prot\_act\_date.

Разработка прототипа пользовательского интерфейса.

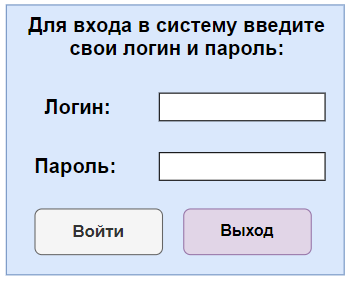


Рис. 53. Окно авторизации в систему.



Рис. 54. Главное окно начальника лаборатории.



Рис. 55. Главное окно сотрудников.



Рис. 56. Окно с таблицей Заказчики.

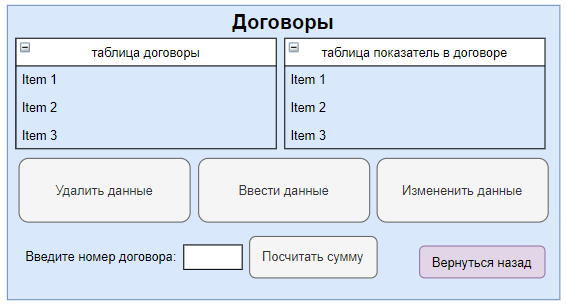


Рис. 57. Окно с таблицей Договоры и вспомогательной таблицей Показатель в договоре.



Рис. 58. Окно с таблицей Показатели для начальника лаборатории.



Рис. 59. Окно с таблицей Методики для начальника лаборатории.

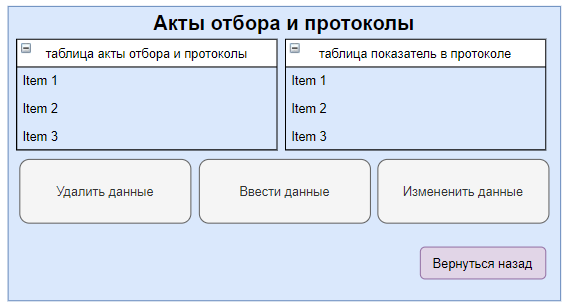


Рис. 60. Окно с таблицей Акты отбора и протоколы и вспомогательной таблицей Показатель в протоколе.