

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное автономное образовательное учреждение  
высшего образования  
**«Санкт-Петербургский государственный университет  
аэрокосмического приборостроения»**  
Кафедра №34 «Технологий защиты информации»

В.А. Мыльников

**Безопасность систем баз данных**

*Методические указания к выполнению лабораторных работ*

Санкт-Петербург, 2019

Мыльников В.А. Безопасность систем баз данных: Методические указания к выполнению лабораторных работ/ГУАП. – Санкт-Петербург, 2019. – 47 с.

Методические указания предназначены студентам специальностей 10.05.03 «Информационная безопасность автоматизированных систем» и 10.05.05 «Безопасность информационных технологий в правоохранительной сфере».

В практикуме приведены 6 лабораторных работ направленных на проектирование и анализ с точки зрения безопасности структуры данных, конфигурация системы управления базой данных и использование криптографических методов для шифрования данных. В качестве практических рекомендаций и примеров используются наиболее распространенные версии систем управления баз данных.

Рецензент

к.т.н., доцент, Наталья Александровна Марковская ГУАП

## Введение

Методические указания к лабораторным работам по дисциплине «Безопасность систем баз данных» содержит цикл работ, направленных на закрепление студентами теоретических знаний, полученных при изучении данного курса, и применения их в решении практических задач.

Необходимость в использовании безопасных информационных систем обусловлена спецификой деятельности организации и обработкой конфиденциальной информации.

Цель курса лабораторных работ по дисциплине «Безопасность систем баз данных» заключается в формировании знаний и умений постановки и решения практических задач проектирования, разработки и эксплуатации баз данных в программных приложениях с учетом требований безопасности.

Для успешного выполнения лабораторных работ необходимо владение базовыми дисциплинами «Основы информационной безопасности», «Базы данных».

В результате изучения дисциплины студенты должны знать:

- принципы построения баз данных в ИС;
- модели представления данных;
- основные операции над данными в ИС;
- основы проектирования баз данных в ИС.
- структурные элементы базы данных;
- основные технологические этапы решения задач в системе управления базами данных (СУБД).

Во время выполнения лабораторных работ студент должен проявить умения:

- проводить анализ предметной области;
- выполнять адаптацию модели данных к требованиям безопасности;
- осуществлять конфигурацию СУБД для работы в безопасном режиме;
- осуществлять обоснованный выбор СУБД с точки зрения защиты данных.

Лабораторные работы сопровождаются краткой теорией, что делает более удобным их выполнение.

При выполнении лабораторных работ предусмотрена самостоятельная реализация изучаемых методов, что способствует их глубокому изучению.

Учебным планом специальности 10.05.03 «Информационная безопасность автоматизированных систем» и 10.05.05 «Безопасность информационных технологий в правоохранительной сфере» предусмотрены лабора-

торные работы по предмету «Безопасность систем баз данных» в седьмом учебном семестре в количестве 34 часа (2 часа в неделю).

## **1. Общие сведения**

### **1.1. Содержание отчета по лабораторной работе**

Отчет по лабораторной работе представляет собой практический отчет, являющийся итогом проведенной студентом работы, представленный для защиты преподавателю. К отчетам по лабораторным работам предъявляется ряд требований, основным из которых является полное описание проделанной работы, позволяющее судить о полученных результатах, степени выполнения заданий и профессиональной подготовке студентов.

В отчет по лабораторной работе должны быть включены следующие составляющие:

- титульный лист (приложение 1);
- цель работы;
- краткие теоретические сведения;
- демонстрация реализации (диаграммы, текст программы, SQL-сценарий, таблицы, графики и т.д.);
- анализ результатов работы программы;
- выводы.

Цель работы должна отражать тему лабораторной работы, а также индивидуальные задачи, поставленные студенту для реализации в рамках лабораторной работы. По объему цель работы в зависимости от сложности и многозадачности работы составляет от нескольких строк до 0,5 страницы.

В кратких теоретических сведениях излагается общее описание изучаемого в работе процесса или проблемы. Материал раздела не должен копировать содержание методического пособия или учебника по данной теме, а ограничивается изложением основных понятий, требующихся для выполнения задания. Объем этого раздела не должен превышать трети всего отчета.

Демонстрация реализации может быть представлена в виде рисунков (скриншотов экрана компьютера), диаграмм ER-моделей, описанием структуры базы данных, хранимых процедур и триггеров, результирующим SQL-сценарием.

Раздел отчета «Анализ результатов работы» должен содержать подробный анализ полученных результатов. Необходимо убедиться в подтверждении теоретических сведений или разобраться в причинах их расхождений.

В выводах по лабораторной работе кратко излагаются результаты, которые удалось или не удалось достичь в ходе выполненной работы.

Отчет по лабораторной работе оформляется на писчей бумаге стандартного формата А4 на одной стороне листа, которые сшиваются в скоросшивателе или переплетаются. Допускается оформление отчета по лабораторной работе только в электронном виде.

## 1.2. Защита лабораторной работы

К защите лабораторной работы студент обязан:

- предоставить полностью оформленную лабораторную работу с заполненным отчетом и результирующим SQL-сценарием или диаграммой;
- знать необходимый теоретический материал;
- знать цели моделирования, уметь приводить примеры использования модели для конкретной предметной области;
- свободно ориентироваться в разработанной диаграмме, коде и функциях представленной реализации;
- уметь решать практические задачи по теме данной работы.

Лабораторная работа оценивается по критериям, представленным в табл. 1.1. Отдельно оцениваются отчет, программное приложение и ответы студента.

В соответствии с табл. 1.2 выставляется итоговая оценка за лабораторную работу на основании суммы набранных студентом баллов.

Таблица 1.1

Критерии оценки выполненной лабораторной работы

Раздел	Баллы	Критерии оценки выполнения раздела
Отчет	0	Отчет имеет неверную структуру, содержит не все разделы, неверно описывает используемую модель и не отражает полученные результаты
	1	Отчет содержит небольшие неточности и недоработки
	2	Отчет полный, структурированный, верно описывает используемую модель и отражает полученные результаты
Реализация	0	Реализация отсутствует
	1	Содержатся ошибки, неверно работает на отдельных наборах входных данных
	2	Реализация полностью соответствует теме работы

Таблица 1.1 (окончание)

## Критерии оценки выполненной лабораторной работы

Защита	0	Студент не ответил правильно на вопросы преподавателя, допускает грубые ошибки при ответе и после дополнительной подготовки не может их исправить
	1	Студент при допуске к работе допустил ошибки при ответе на вопросы преподавателя, но затем исправил их
	2	Студент правильно и уверенно отвечает на вопросы, знает теоретический материал и умеет применять его к решению практических задач

Таблица 1.2

## Шкала соотнесения баллов и оценок

Оценка	Количество баллов
«3» удовлетворительно	2-3
«4» хорошо	4
«5» отлично	5-6

## 2. Содержание лабораторных работ

### 2.1. Лабораторная работа 1.

#### Разработка требований к информационной системе

##### Цель работы

Составить и проанализировать требования к информационной системе, разработать техническое задание на разработку программного обеспечения.

##### Задачи

1. Сформировать и выполнить анализ требований к разрабатываемой информационной системе.
2. Разработать информационную модель будущей системы на базе диаграммы прецедентов (Use-Case diagram).

##### Теоретические основы занятия

Разработка требований – это процесс, включающий мероприятия, необходимые для создания и утверждения документа, содержащего спецификацию системных требований. Различают четыре основных этапа процесса разработки требований:

- анализ технической осуществимости создания системы;
- формирование и анализ требований;
- специфицирование требований и создание соответствующей документации;
- аттестацию этих требований.

Но поскольку в процессе разработки системы в силу разнообразных причин требования могут меняться, управление требованиями, т.е. процесс управления изменениями системных требований, является необходимой составной частью деятельности по их разработке.

Процесс формирования и анализа требований проходит через ряд этапов. Ими являются:

1. Анализ предметной области. Аналитики должны изучить предметную область, где будет эксплуатироваться система.
2. Сбор требований. Это – процесс взаимодействия с лицами, формирующими требования. Во время этого процесса продолжается анализ предметной области.
3. Классификация требований. На этом этапе бесформенный набор требований преобразуется в логически связанные группы требований.
4. Разрешение противоречий. На этом этапе определяются и разрешаются противоречия различного рода.

5. Назначение приоритетов. В любом наборе требований одни из них будут более важны, чем другие. На этом этапе совместно с лицами, формирующими требования, определяются наиболее важные требования.
6. Проверка требований. На этом этапе определяются их полнота, последовательность и непротиворечивость.

Пользовательские требования должны описывать внешнее поведение системы, основные функции и сервисы, предоставляемые системой, ее нефункциональные свойства. Необходимо выделить опорные точки зрения и сгруппировать требования в соответствии с ними. Пользовательские требования можно оформить как простым перечислением, так и используя нотацию вариантов использования. Рассмотрим пример анализа требований для системы поддержки заказа и учета товаров.

Система для отдела продаж предусматривает взаимодействие с клиентом, обновления списка доступных товаров. При помощи системы составляются заявки клиентов. Каждый заказ может содержать несколько позиций из списка продуктов, в каждой позиции указываются наименование товара и его количество в заказе. Система по требованию пользователя формирует и выдает на печать следующую справочную информацию.

Первым шагом в формировании требований является идентификация опорных точек зрения. Во всех методах формирования требований, основанных на использовании точек зрения.

Следующей стадией процесса формирования требований будет идентификация опорных точек зрения и сервисов (табл. 2.1). Сервисы должны соответствовать опорным точкам зрения. Но могут быть сервисы, которые не поставлены с ними в соответствие. Это означает, что на начальном этапе обсуждения некоторые опорные точки зрения не были идентифицированы. Один и тот же сервис может быть соотнесен с несколькими точками зрения.

Таблица 2.1

Сервисы, соотнесенные с точками зрения

Покупатель	Подача заявки на покупку товара Оплата товара по договору Отправка партии покупателю
Руководитель	Составление накладной
Менеджер	Обработка заявки Проверка заявки на наличие ошибок Составления договора об оплате Отправка договора об партии
Заведующий складом	Формирование партии Отгрузка партии



### Разработка диаграммы прецедентов

Способ использования системы пользователями определяет проектное решение, которое будет положено в основу разрабатываемой системы.

Прецедент – это конструкция, позволяющая описать систему с точки зрения потенциальных пользователей.

Прецедент представляет собой набор сценариев использования системы. Каждый сценарий описывает последовательность действий. Каждая последовательность действий инициируется пользователем, другой системой, аппаратным средством или в какой-либо момент времени. Сущности, инициирующие сценарии, называются исполнителями (*actor*).

Результат прецедента должен быть полезен исполнителю, инициировавшему этот прецедент, либо какому-то другому исполнителю.

Представление модели прецедентов следующее:

- один исполнитель инициирует прецедент;
- другой исполнитель получает новое качество от его реализации.

При выполнении анализа прецедентов определяются границы системы и ее связь с окружающим миром. Исполнители обычно находятся вне системы, а прецеденты – внутри. Для обозначения границ системы используется прямоугольник, внутри которого указывается имя системы. Исполнители, прецеденты и соединительные линии образуют модель прецедентов.

### Построение диаграммы прецедентов

Рассмотрим построение диаграммы прецедентов на примере приложения Sybase Power Designer.

1. Создайте диаграмму *Use-Case*.

2. Далее создайте три прецедента, используя инструмент *Use-Case*. Чтобы изменить имя прецедента и другие свойства, нажмите дважды левой кнопкой мыши на объект. В открывшемся окне в поле *Name* введите наименование объекта в понятной форме, а в поле *Code* введите название объекта, которое затем будет использоваться в программном коде.

3. Создайте взаимосвязи между объектами. Взаимосвязи могут идти в двух направлениях:

- от исполнителя к прецеденту (связь *Primary actor*);
- от прецедента к исполнителю (связь *Secondary actor*).

Создайте взаимосвязи от инициирующих исполнителей к прецедентам и от прецедентов к принимающим исполнителям с помощью инструмента *Association*. Для этого подведите указатель к исполнителю, нажмите левую клавишу мыши и, удерживая ее, подведите к прецеденту.

4. Чтобы повторно использовать шаги одного прецедента в другом, применяется *включение*. Графически включение обозначается в виде соединительной пунктирной линии со стрелкой, указывающей на тот класс,

от которого зависит другой (инструмент *Dependency*). В свойствах данного объекта необходимо в поле *Stereotype* указать слово <<включает>>.

5. Можно расширить исходный (базовый) прецедент за счет добавления новых шагов. Данный тип взаимоотношений называется *расширением*.

6. Расширение может происходить только на заданных точках последовательности шагов базового прецедента. Такие места называются точками расширения. Подобно включению, расширение отображается линией зависимости (пунктир со стрелкой) со стереотипом <<расширяет>> (инструмент *Dependency*).

В создаваемой модели на основе отдела продаж были выделены следующие роли (Актеры): Менеджер, Покупатель, Заведующий складом, Руководитель (рис. 2.1).

Субъекты элементов модели: «Отправка заявки на товары», «Обработка заявки», «Проверка истории покупателя», «Согласование заявки», «Переоформление заявки», «Отправка договора и счета», «Оформление счета на оплату», «Отправка квитанции оплаты счета», «Оформление товарной накладной», «Формирование партии товаров», «Отгрузка партии товаров».

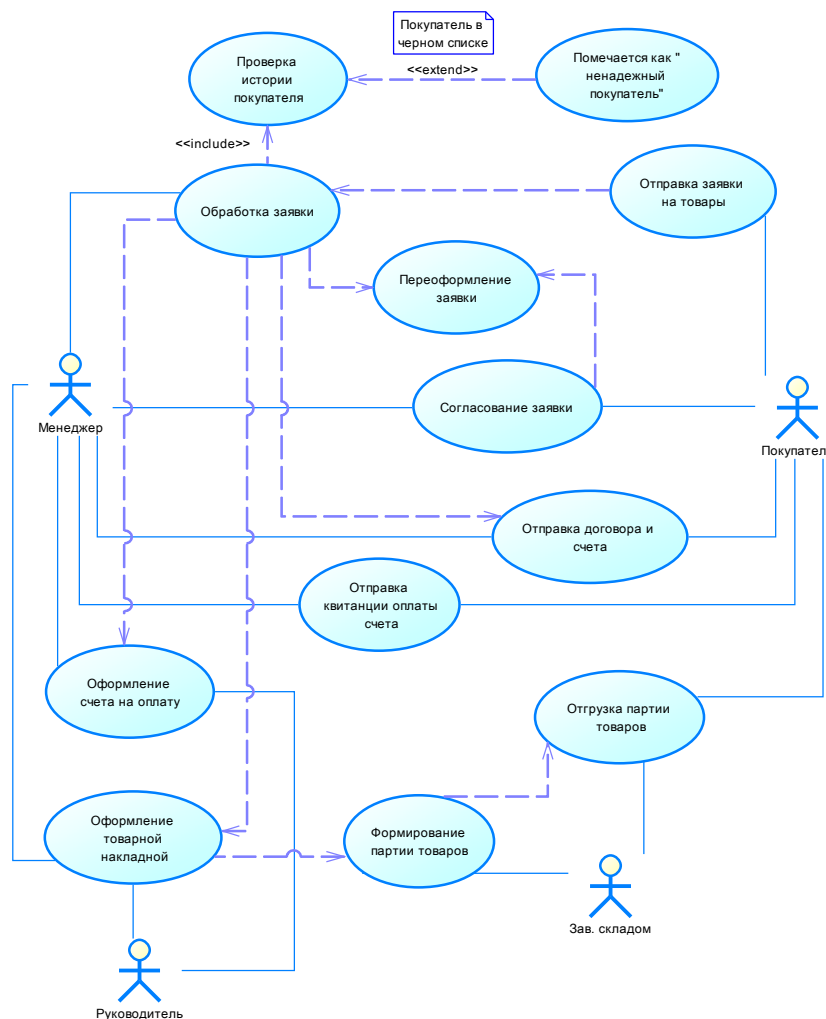


Рис. 2.1. Пример диаграммы прецедентов

Система разбита на несколько субъектов, поэтому моделировать будем фрагмент «Отправка заявки на товары», он включает в себя прецедент «Проверка истории покупателя» и расширение «Помечается как ненадежный покупатель». Основной успешный сценарий представлен на рис. 2.2. Для предотвращения выхода системы из строя, был написан сценарий расширения, представленный на рис. 2.3.

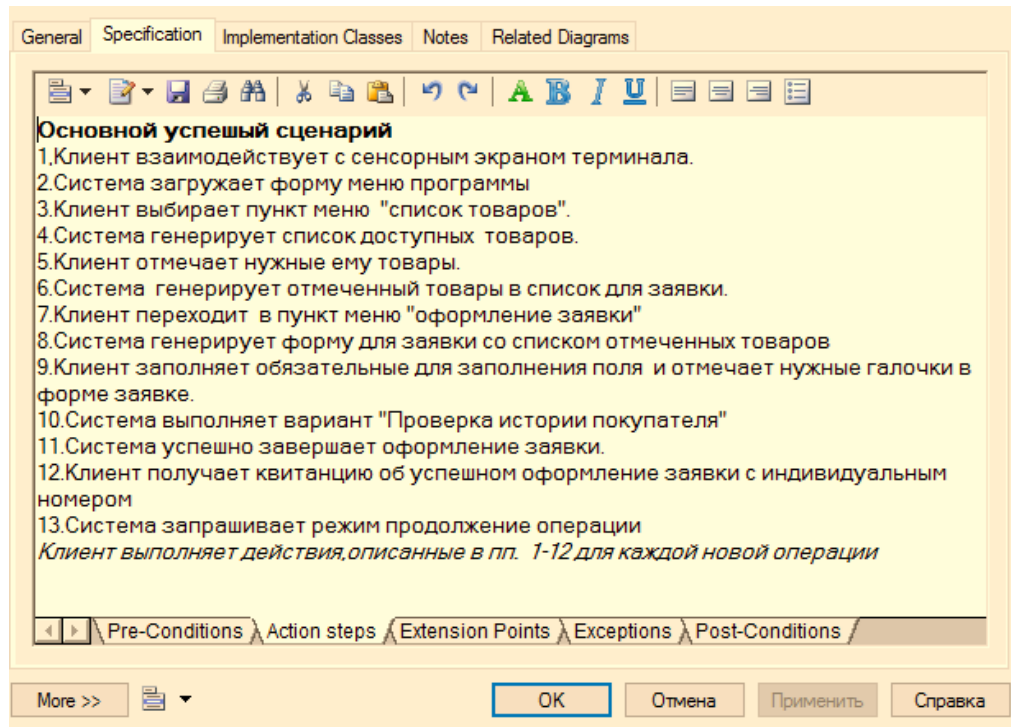


Рис. 2.2. Основной сценарий «Отправка заявки на товары»

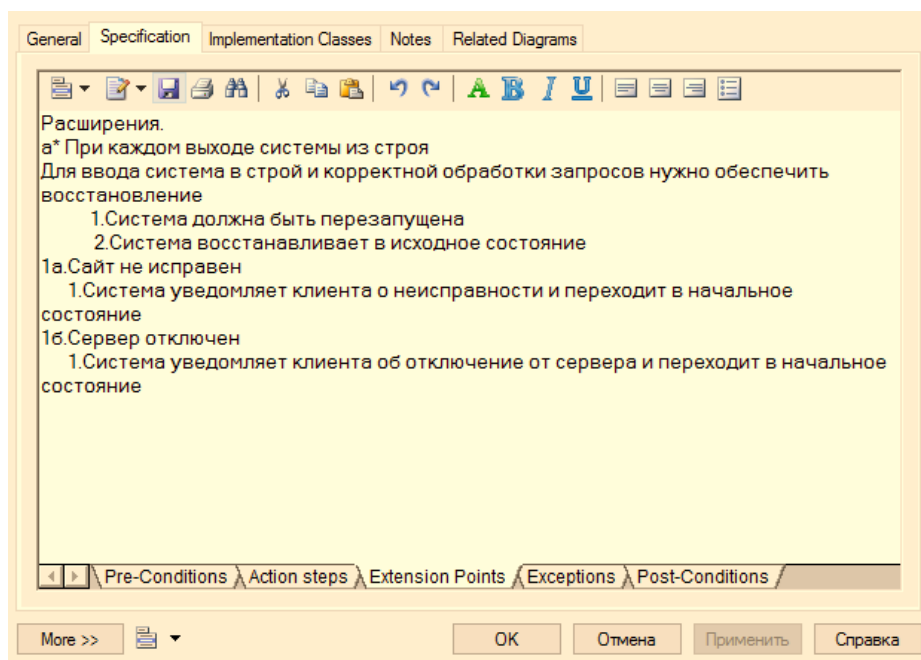


Рис. 2.3. Сценарий расширение «Отправка заявки на товары»

В построенной диаграмме рассмотрен случай, когда расширение прецедента может быть сложным по структуре, поэтому оно выделено в отдельный прецедент. Его сценарий и расширение представлены на рис. 2.4 и рис. 2.5.

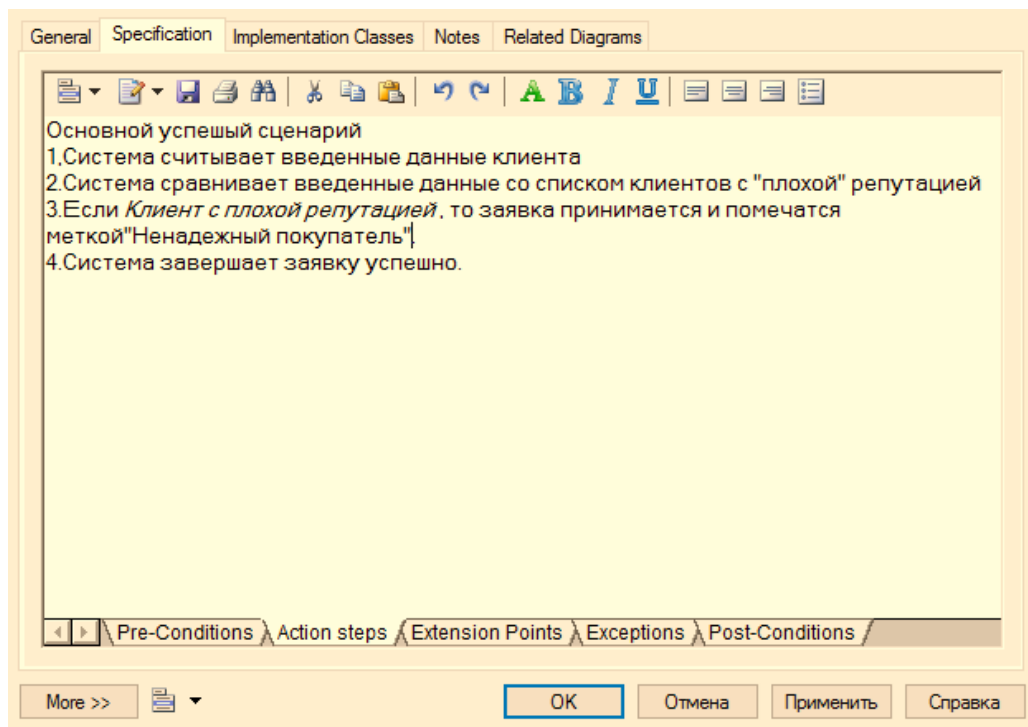


Рис. 2.4. Основной сценарий «Проверка истории покупателя»

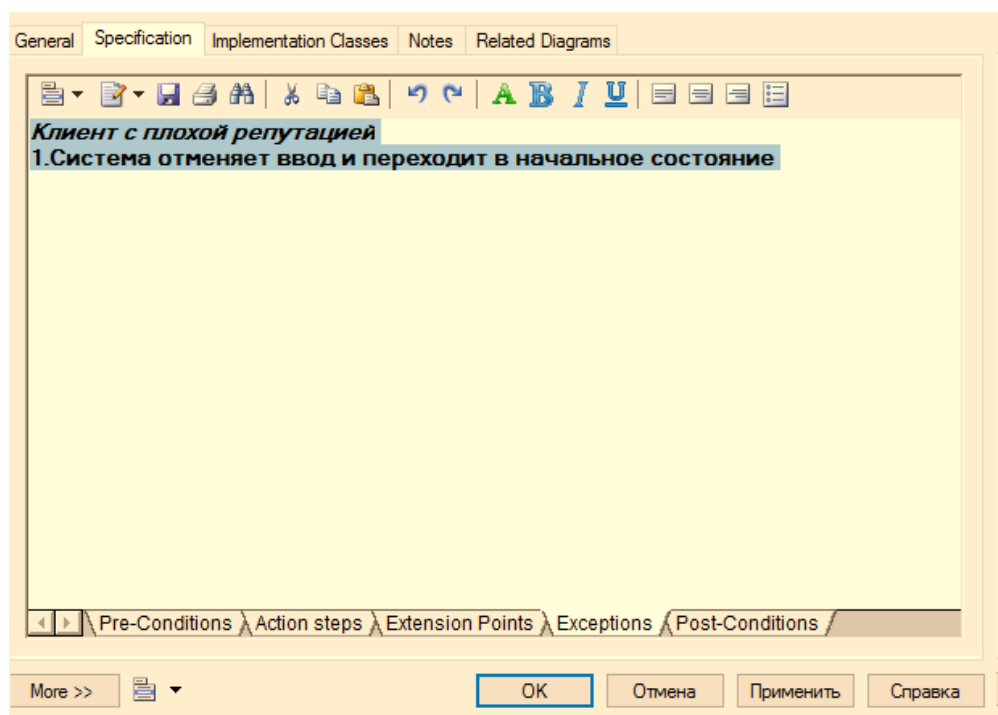


Рис. 2.5 Сценарий расширения «Проверка истории покупателя»

### **Контрольные вопросы**

1. К каким основным группам можно отнести требования к информационным системам?
2. При решении задачи анализа осуществимости проекта на какие основные вопросы следует ответить?
3. Объясните, почему требования, сформулированные разными лицами, могут быть противоречивыми?
4. Назовите основные элементы диаграммы вариантов использования.
5. Что в диаграммах вариантов использования называется актером?
6. Поясните применение отношения ассоциации на диаграмме.

## **2.2. Лабораторная работа 2. Разработка структуры базы данных с учетом снижения рисков**

### Цель занятия

Приобретение навыков по исследованию баз данных на предмет устранения рисков информационной безопасности.

### Задача

Разработать результирующую логическую и физическую модель диаграммы «сущность-связь» для разрабатываемой информационной системы.

### План занятия

1. Построение модели логического уровня диаграммы «сущность-связь».
2. Разработка модели физического уровня, выбор типа СУБД.
3. Определить возможные существенные риски.
4. Предложить способы снижения рисков.
5. Генерация SQL-сценария для создания структуры базы данных.

### Теоретические основы занятия

В данной работе необходимо разработать логическую и физическую модель базы данных. При составлении диаграммы необходимо ее привести к 3НФ, определиться с составом первичных ключевых полей, обосновать необходимость идентифицирующей внешней связи.

Выполнение указанных требований необходимо для проверки случаев ввода и изменения записей без нарушения ссылочной целостности, отсутствия дубликатов.

Для более сложных ситуаций, где требуется заранее выполнять сложные проверки, необходимо предусмотреть создание триггеров и сопутствующих хранимых процедур. В качестве примеров подобных случаев можно представить контроль остатков товаров при заказе, бронирование/приобретение билета с проверкой свободных посадочных мест и т.д.

В таком случае контроль операций по вводу, изменению и удалению данных требует предварительных проверок и последующих дополнительных изменений в различных таблицах базы данных. В зависимости от сложности требуется разработать механизм транзакций.

На примере нескольких сотрудников необходимо разработать несколько представлений, которые в дальнейшем могут быть использованы на формах приложений разрабатываемой системы, оперативных и аналитических отчетов.

Дополнительно представления могут оказать помощь при настройке доступа к объектам базы данных, в зависимости от версии используемой СУБД.

Перечень отчетов зависит от предметной области варианта задания, его необходимо уточнить и согласовать с преподавателем.

#### Экспериментальная часть занятия

Для выполнения лабораторной работы и разработки логической и физической модели данных возможно использовать приложение Sybase Power Designer или ErWin Data Modeler.

Для рассматриваемого варианта упрощенный пример логической модели, представлена на рис. 2.6, физическая модель - на рис. 2.7.

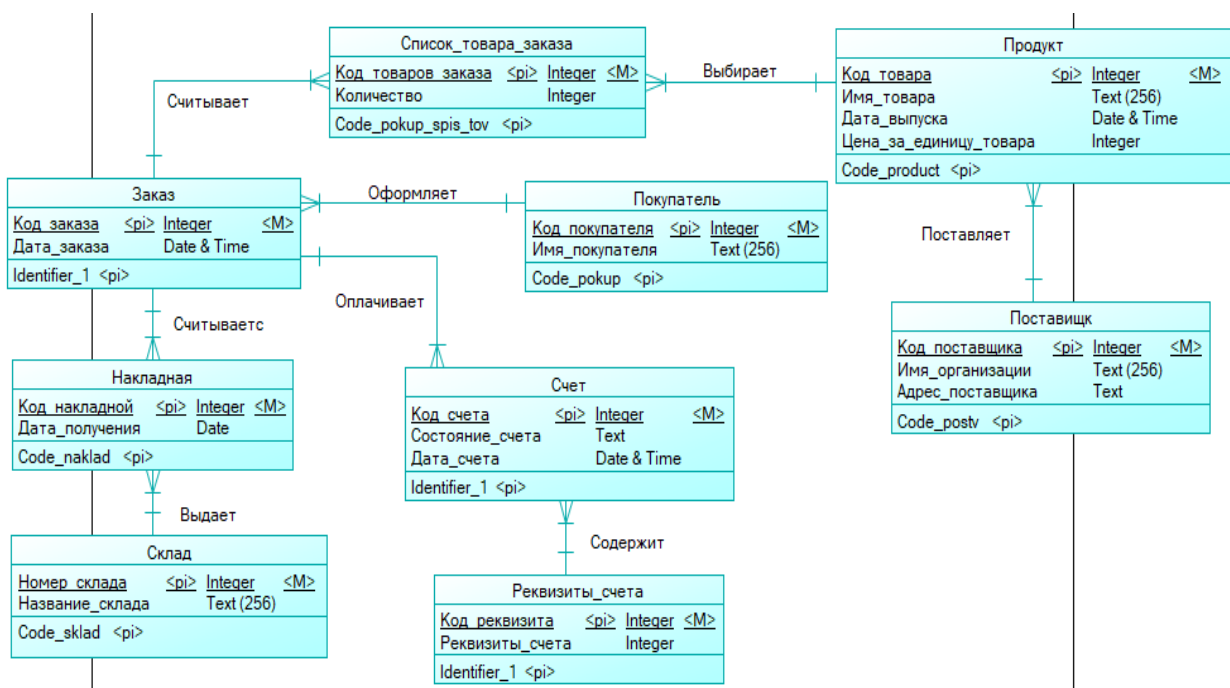


Рис. 2.6. Логическая модель данных Power Designer

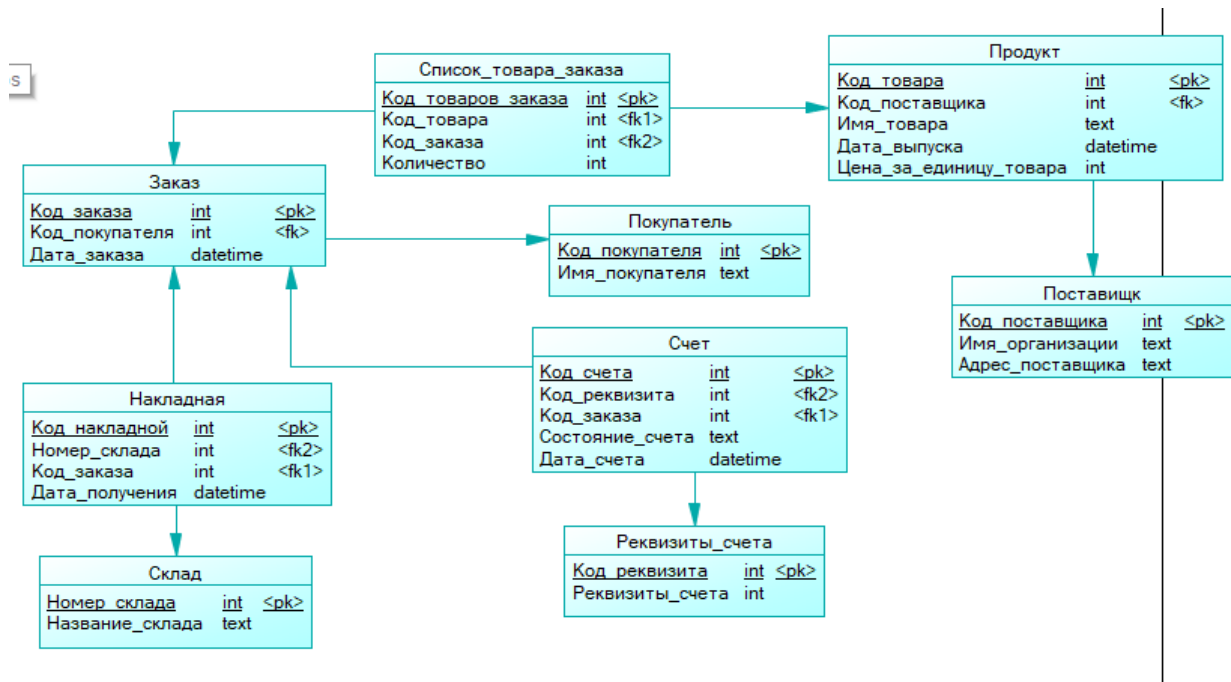


Рис. 2.7. Физическая модель данных Power Designer

Логическая модель данных состоит из сущностей, как:

- накладная;
- продукт;
- склад;
- поставщик;
- счета;
- реквизиты;
- список товаров заказа;
- покупатель;
- заказ.

Каждая из сущностей содержит атрибуты, имеются основные (primary key) и внешние ключи (foreign key). Для примера можно рассмотреть одну из основных сущностей модели.

Сущность «Счет» содержит в себе атрибуты:

- код счета – первичный ключ;
- состояние счета;
- Дата изменения счета.

Атрибуты как внешние ключи в логической модели данных не отображаются. Данные атрибуты будут видны после построения физической модели данных.

Физическую модель необходимо создавать после успешной проверки логической модели, в зависимости от используемого средства проектирования может создаваться автоматически.

В физической модели данных в таблицах уже присутствуют дополнительные атрибуты в качестве внешних ключевых полей. Например, в таблице «Накладная» появляются дополнительные поля:

- код накладной;
- номер склада;
- код заказа;
- дата получения.

В качестве результата реализации в отчете необходимо представить ER-модель и сгенерированный SQL-сценарий. Оценка модели выполняется в соответствии со списком функций по разрабатываемой системе. Поэтому достаточность атрибутов сущностей, нормализация отношений зависят от специфики предметной области задания.

Для удобства работы с базой данных были созданы следующие представления (табл. 2.2).

В данной лабораторной работе не предполагается представлений для заведующего складом, поэтому все представления в системе создаются для менеджера.

Таблица 2.2

Список представлений для разрабатываемой системы

Название представления	Сотрудник	Участвующие элементы
Информация о заказе	Менеджер	Поставщик Список товаров заказа Счет Склад Дата получения
Информация о товаре	Менеджер	Наименование Стоимость единицы товара Количество Дата выпуска
Информация о финансах	Менеджер	Реквизиты Состояние Дата изменения состояния

### Контрольные вопросы

1. Что такое сущность, атрибут сущности?
2. Что такое связь, виды связей, мощность связи?
3. Что такое первичный ключ?
4. Что такое внешний ключ?



### **2.3. Лабораторная работа 3. Разработка стратегии информационной безопасности базы данных**

#### Цель занятия

Ознакомление с основными принципами обеспечения безопасности базы данных.

#### Задача

Разработать стратегию информационной безопасности базы данных для разрабатываемой системы.

#### План занятия

Выполнить совместный анализ диаграммы прецедентов и модели базы данных.

Внести изменения в модель базы данных для упрощения организации доступа к объектам базы данных.

Разработать схему директивного доступа к объектам базы данных (таблица, столбец, представление и т.д.) для объектов взаимодействия диаграммы прецедентов.

#### Теоретическая часть

Разработка стратегии направлена на то, чтобы наиболее важные цели соответствующей деятельности достигались при наиболее рациональном расходовании имеющихся ресурсов.

Процесс выработки стратегии обеспечения информационной безопасности баз данных в самом общем виде может быть определен как поиск компромисса между уровнем обеспечения информационной безопасности и необходимыми для достижения этих целей ресурсами.

#### Практическая часть

Анализ диаграммы прецедентов и модели базы данных. Проектируемая база данных предназначена для оперативной работы сотрудников отдела продаж, удаленного доступа системного администратора для проверки работоспособности СУБД и доступа руководящего персонала различного уровня для просмотра отчетов.

Основной предполагаемой проблемой для данной базы данных является хищение информации из базы данных штатными сотрудниками.

Сотрудники компании, имеющие непосредственный доступ к базе данных:

1. руководитель;
2. менеджер;
3. администратор базы данных.

Второй предполагаемой проблемой является получение информации в незащищенном канале подключения к серверу СУБД.

Для защиты базы данных от хищения информации предлагается:

1. Использовать двухфакторную аутентификацию.
2. Шифрование канала подключения при работе с сервером СУБД.

3. Шифрование конфиденциальных сведений с помощью алгоритмов шифрования.
4. Ведение журнала событий при работе с данными.
5. Четкое разграничение доступа к данным между сотрудниками.
6. Планирование схемы обслуживания СУБД.

Для двухфакторной аутентификации предлагается использовать логин/пароль в связке с SQL-сертификатами. Сгенерированный сертификат устанавливается на компьютер сотрудника в «Личном хранилище сертификатов» в профиле учетной записи.

Для шифрования трафика требуется обязательным образом работать по защищенному соединению с применением механизма SSL/TLS. Для этого необходимо сгенерировать сертификаты на стороне сервера СУБД, который будет предоставляться клиентскому компьютеру при инициализации подключения.

Для конфиденциальных сведений предлагается использовать шифрование данных. Для этого необходимо подготовить структуру данных на физическом уровне, чтобы поля могли хранить зашифрованные данные. В зависимости от используемой СУБД выбирается один из доступных методов шифрования.

Журнал событий необходим для регистрации всех событий при работе с данными на уровне подключения к СУБД, изменения структуры объектов базы данных, просмотра и изменения данных. Данные сведения в дальнейшем позволят повысить эффективность работы благодаря использованию утилит оптимизации настроек сервера СУБД по изменению объема кэш-памяти для оперативных запросов, временных таблиц, но и повысить уровень сохранности данных.

Схема обслуживания сервера СУБД требует анализа предметной области и разработки предположений о размере базы данных, интенсивности роста объема данных, критичности данных. Это требуется для составления плана резервного копирования данных – инкрементная и полная резервная копия таблиц/базы данных и их периодичность. Также требуется составить регламент обслуживания базы данных – периодичность очистки данных и журнала событий базы данных от удаленных объектов, последующего сжатия базы данных, обновление индексов и т.д.

В данной работе рассмотрим один из пунктов требований. Остальные рассматриваются в следующих работах.

#### Схема директивного доступа к объектам базы данных

Определение области работы сотрудников базы данных заключается в контроле уровней доступа сотрудника к изменению, редактированию или удалению данных с помощью распределения создания ролей директивного доступа. При создании учетной записи для нового сотрудника заранее определяется его роль на сервере СУБД.

В табл. 2.3 представлены данные о доступе для роли «Руководитель».

В таблице представлены сокращения: С – создания записей, R – чтения данных, U – обновления данных, D – удаления записей. В данном случае роль имеет полный доступ ко всем данным.

Таблица 2.3

## Схема доступа роли «Руководитель»

Таблица	Столбец	Представление	Уровни доступа
Накладная	Все	Нет	C,R,U,D
Продукт			C,R,U,D
Список товаров заказа			C,R,U,D
Склад			C,R,U,D
Поставщик			C,R,U,D
Покупатель			C,R,U,D
Счет			C,R,U,D
Реквизиты			C,R,U,D
Заказ			C,R,U,D

В табл. 2.4 представлена схема доступа для роли «Менеджер».

Таблица 2.4

## Схема доступа роли «Менеджер»

Таблица	Столбец	Представление	Уровни доступа
Накладная		Информация о заказе	C,R,U,D
Заказ		Информация о заказе	C,R,U,D
Продукт		Информация о товаре	C,R,U,D
Список товаров заказа		Информация о заказе	C,R,U,D
Склад		Информация о товаре	R
Поставщик		Информация о товаре	R
Покупатель		Информация о заказе	C,R,U,D
Счет		Информация о финансах	R
Реквизиты		Информация о финансах	R

**Контрольные вопросы**

1. Сформулируйте понятие угрозы информационной системе.
2. Приведите несколько примеров реализации угроз.
3. В чем состоит существо логического вывода конфиденциальных значений атрибутов на основе функциональных зависимостей?
4. Сформулируйте угрозу доступности баз данных, вытекающую из свойств первичных ключей и поддержки ссылочной целостности.

## 2.4. Лабораторная работа 4. Организация разграничения доступа к объектам базы данных

### Цель занятия

Ознакомление с основными механизмами организации доступа к объектам базы данных SQL сервера.

### Задача

Организовать доступ к объектам базы данных.

### План занятия

1. Спроектировать механизм сетевой безопасности базы данных (удаленный доступ, удаленные соединения, режим проверки подлинности, политика паролей).
2. Создание ролей директивного доступа.
3. Представить в отчете SQL-сценарий настройки привилегий.

### Теоретическая часть

Когда различным сотрудникам требуется выполнять одинаковые задачи в базе данных, можно воспользоваться механизмом ролей.

Участниками роли в СУБД могут быть:

- группы и учетные записи операционной системы;
- пользователи сервера СУБД.

Рассмотрим архитектуру безопасности на примере сервера СУБД Microsoft SQL Server.

Компонент Database Engine включает несколько системных ролей, которые имеют заранее предопределенных разрешений. Кроме ролей, определяемых пользователями, существует два типа предопределенных ролей:

- фиксированные серверные роли;
- фиксированные роли базы данных.

### Фиксированные серверные роли

Фиксированные серверные роли определяются на уровне сервера и поэтому находятся вне баз данных, принадлежащих серверу баз данных. В табл 2.5 приводится список фиксированных серверных ролей и краткое описание действий, которые могут выполнять члены этих ролей.

Участников фиксированной серверной роли можно добавлять и удалять двумя способами:

- используя среду Management Studio;
- используя инструкции языка Transact-SQL.

Таблица 2.5

## Фиксированные серверные роли

Фиксированная серверная роль	Описание
<i>sysadmin</i>	Выполняет любые действия в системе баз данных
<i>serveradmin</i>	Конфигурирует параметры сервера
<i>setupadmin</i>	Устанавливает репликацию и управляет расширенными процедурами
<i>securityadmin</i>	Управляет регистрационными именами и разрешениями для инструкции CREATE DATABASE и чтением журналов логов
<i>processadmin</i>	Управляет системными процессами
<i>dbcreator</i>	Создает и модифицирует базы данных
<i>diskadmin</i>	Управляет файлами на диске

Фиксированные роли базы данных

Фиксированные роли определяются на уровне базы данных и поэтому существуют в каждой созданной базе данных. В табл. 2.6 приводится список фиксированных ролей и краткое описание действий.

Таблица 2.6

## Фиксированные роли базы данных

Фиксированная роль базы данных	Описание
db_owner	Пользователи, которые могут выполнять почти все действия в базе данных
db_accessadmin	Пользователи, которые могут добавлять и удалять пользователей
db_datareader	Пользователи, которые могут просматривать данные во всех таблицах пользователей базы данных
db_datawriter	Пользователи, которые могут добавлять, изменять или удалять данные во всех пользовательских таблицах базы данных
db_ddladmin	Пользователи, которые могут выполнять инструкции DDL в базе данных
db_securityadmin	Пользователи, которые могут управлять всеми действиями в базе данных, связанными разрешениями безопасности

Таблица 2.6 (окончение)

## Фиксированные роли базы данных

Фиксированная роль базы данных	Описание
<i>db_backupoperator</i>	Пользователи, которые могут выполнять резервное копирование базы данных
<i>db_denydatareader</i>	Пользователи, которые не могут просматривать любые данные в базе данных
<i>db_denydatawriter</i>	Пользователи, которые не могут изменять никакие данные в базе данных

Механизм сетевой безопасности данных

Для организации удаленного доступа к базе данных необходима выполнить первоначальную настройку, представленную на рис. 2.8.

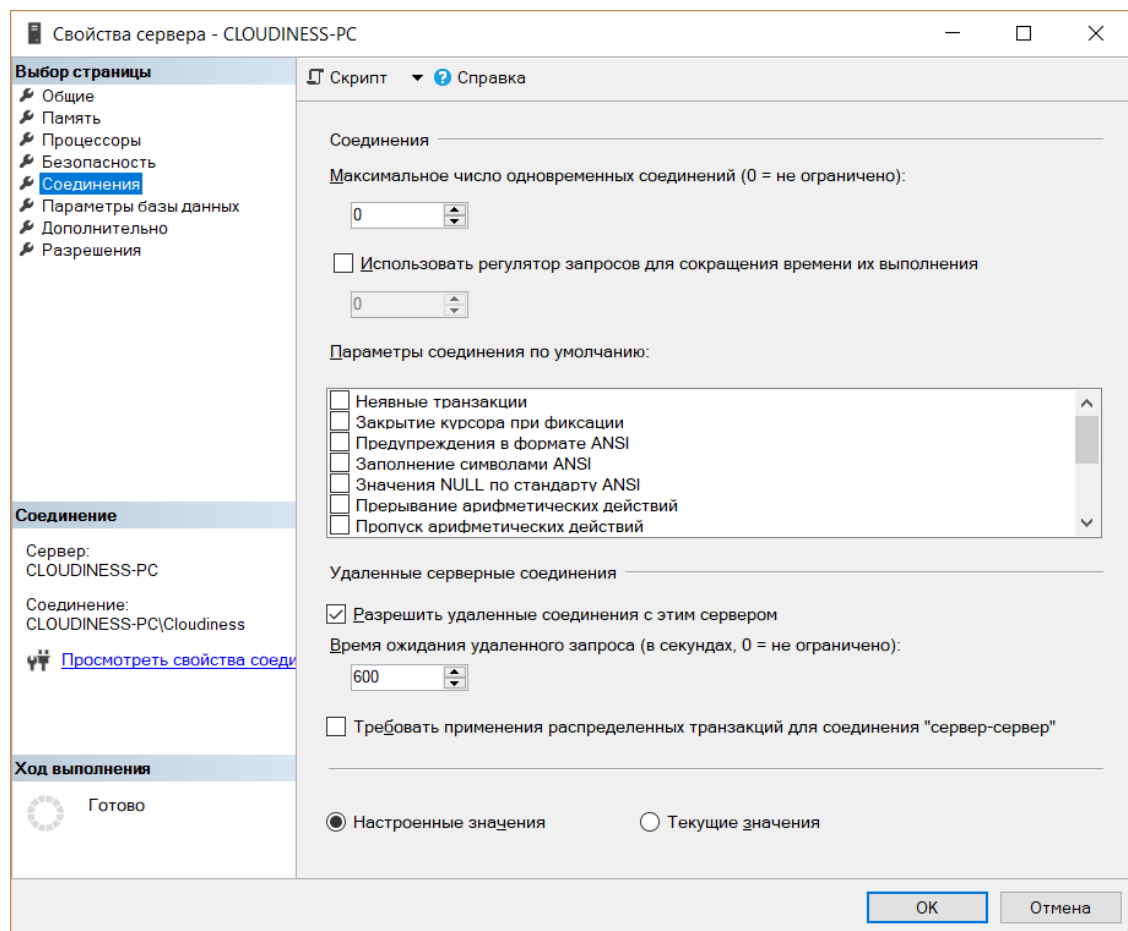


Рис. 2.8. Настройка удаленных соединений

Для удаленного подключения к СУБД необходимо включить параметр «Разрешить удаленные соединения с этим сервером». Данный параметр находится во вкладке «Соединения».

На рис. 2.9 показана настройка протоколов для установления удаленного соединения к базе данных. Для того чтобы база данных работала с удаленными подключениями необходимо в настройках сети Microsoft Sql Server для нашего сервера «требуется включить протокол «TCP/IP»».

Для работы сервера СУБД в автономном режиме, например в режиме обработки данных с помощью сервера приложений или веб-сервера, требуется активировать либо протокол «Общая память» (обеспечивает самое высокое быстродействие) или «Именованные каналы» (подключение через socket-файл). Данные механизмы исключают проверку безопасности сетевого шлюза и обеспечивают наибольшую скорость обмена пакетами, что является ограничением для сетевых протоколов.

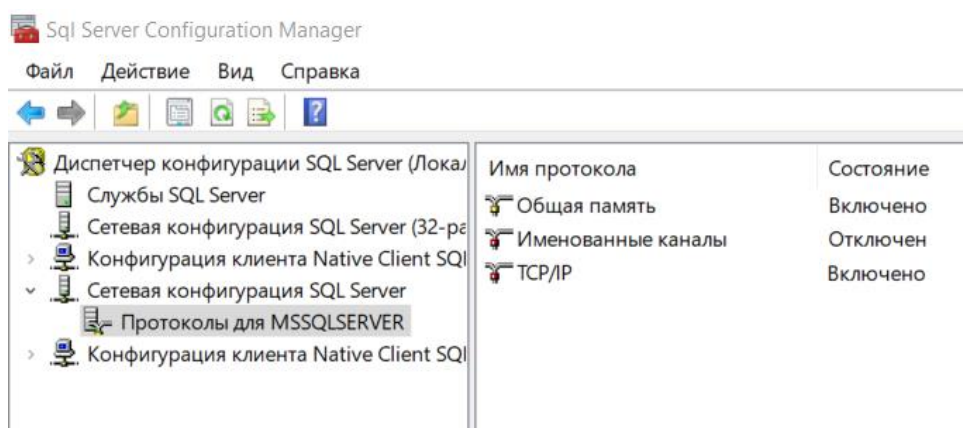


Рис. 2.9. Настройка протоколов подключения к серверу СУБД

На рис. 2.10 изображена настройка аутентификации сервера.

На вкладке «Безопасность» требуется настроить проверку подлинности в разделе «Серверная проверка подлинности». При смешанной проверке учетных записей необходимо переключиться на режим «Проверка подлинности SQL Server и Windows».

Дополнительно требуется скорректировать регистрацию в журнале событий в разделе «Аудит входа». При необходимости отслеживания доступа через удаленное подключение и причин утечек данных рекомендуется фиксировать «Все попытки входа».

В данном пример оформления работы продемонстрировано создание двух ролей с заранее разработанными правилами доступа.

#### Создание учетных записей

На рис. 2.11 показано создание пользователей с помощью языка Transact- SQL.

Командой CREATE LOGIN создается учетная запись и пароль для работы с базой данных. Командой «CREATE USER» создается пользователь с присоединением к нему ранее созданной учетной записи.

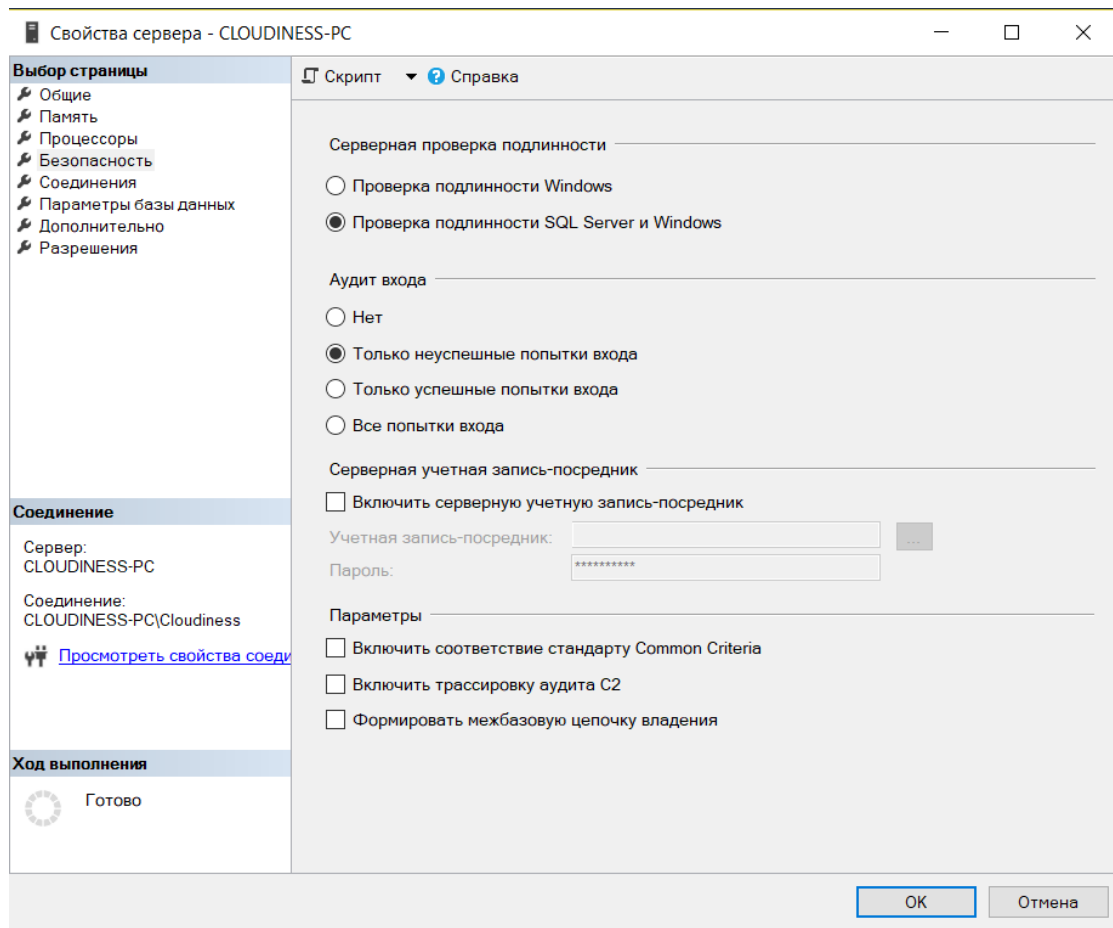


Рис. 2.10. Настройка аутентификации сервера СУБД

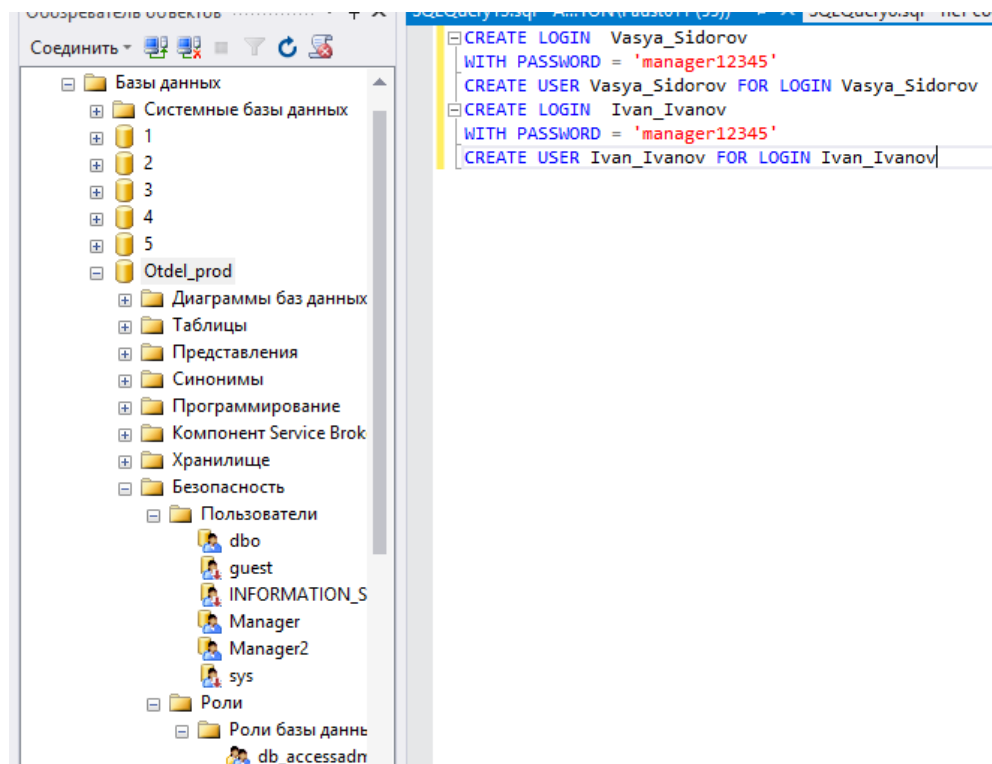


Рис. 2.11. Создание учетных записей



## Создание ролей

На рисунке 2.12 показано создание ролей доступа к базе данных. Для каждого пользователя создается определенная роль с набором привилегий. Роли создавались в соответствии с табл. 2.3 и табл. 2.4.

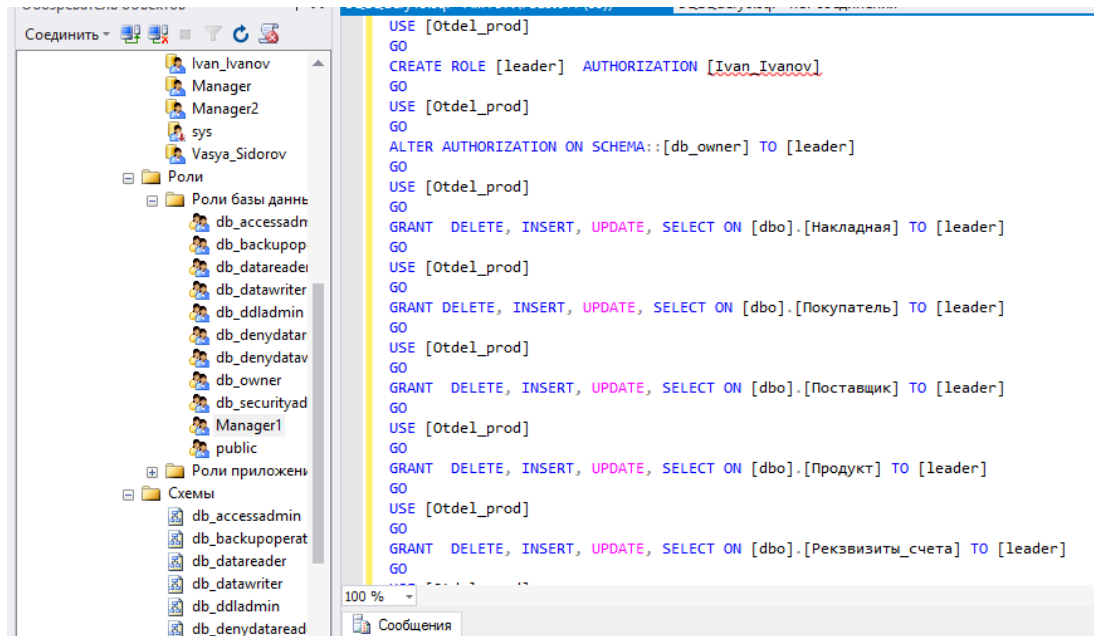


Рис. 2.12. Создание роли «Руководитель»

На рис. 2.13 можно увидеть результат создания необходимых ролей с определенными привилегиями.

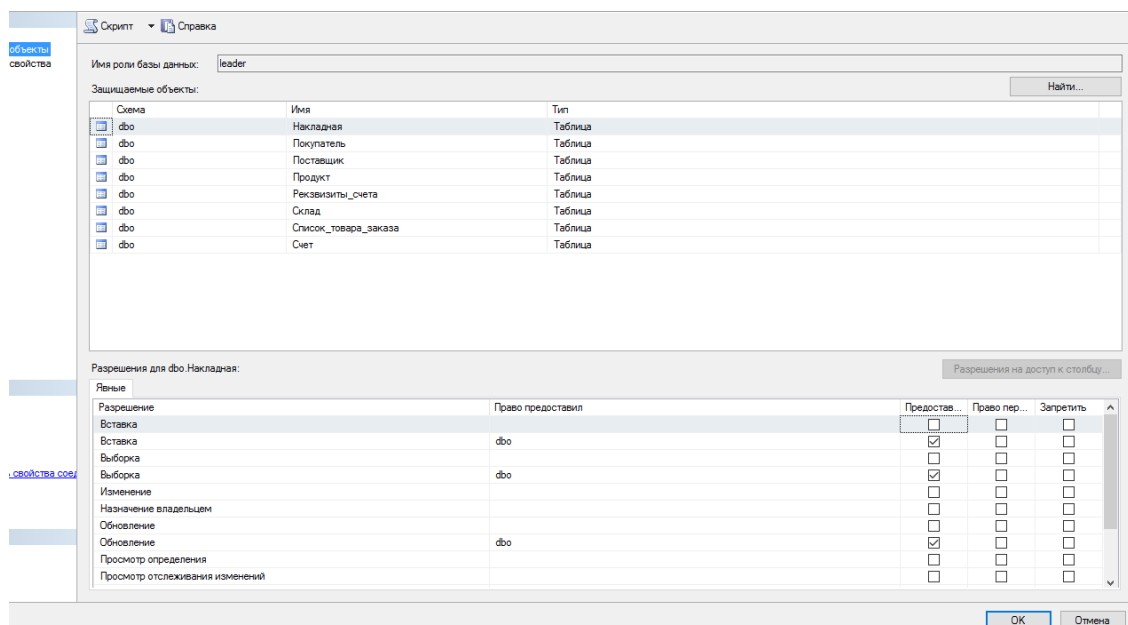


Рис. 2.13. Роли базы данных

### **Контрольные вопросы**

1. Дайте характеристику объектов СУБД используемые для построения субъектно-объектных моделей защиты информации.
2. Объясните необходимость нескольких вариантов документального оформления политики безопасности для различных уровней управления.
3. Выполнение каких условий необходимо обеспечить для эффективной реализации политики безопасности?
4. Приведите примеры применения принципа минимально возможных привилегий. Почему применение данного принципа иногда входит в противоречие с принципом психологической приемлемости?

## **2.5. Лабораторная работа 5. Методы резервного копирования и восстановления объектов базы данных**

### Цель занятия

Ознакомление с методами резервного копирования и восстановления для защиты базы данных.

### Задача

Разработать схему резервного копирования и восстановления базы данных.

### План занятия

1. Разработать схему резервного копирования базы данных с использованием полного и инкрементного способа.
2. Настроить и проверить работоспособность резервного копирования.
3. Выполнить проверку восстановления базы данных из резервной копии.

### Теоретическая часть

Методы резервного копирования.

Существуют различные методы резервного копирования базы данных: полное и инкрементное резервное копирование, резервное копирование журнала транзакций, группы файлов и файла данных. Каждый из них имеет свои режимы и возможности работы.

Полное резервное копирование (full backup) предусматривает резервное копирование всех данных базы данных, группы файлов или файла данных.

Инкрементное резервное копирование (differential backup) предусматривает резервное копирование только тех данных, которые изменились с момента последнего резервного копирования.

Резервное копирование журнала транзакций используется для резервного копирования и усечения журнала транзакций.

Резервное копирование журнала транзакций является определяющей задачей для сервера СУБД, т.к. данные журнала транзакций используются в сочетании с резервными копиями базы данных.

Резервное копирование групп файлов и файла данных используется для создания резервной копии определенной группы файлов или файла данных в базе данных. В качестве примеров можно привести файлы метаданных, импорта/экспорта данных, выгрузки для .

Все виды резервного копирования выполняются для определенной базы данных. Дополнительно рекомендуется выполнять резервное копирование системных баз данных, содержащих метаданные сервера СУБД.

#### Полное резервное копирование

Если имеется несколько баз данных, то следует создать резервные копии для каждой из них. Полное резервное копирование является наиболее распространенным методом для баз данных небольшого и среднего размера. Основным критерием является длительность его выполнения, поэтому можно предусмотреть резервное копирование групп файлов.

#### Разностное резервное копирование

Разностное резервное копирование выполняется только для той информации, которая изменилась с момента последнего резервного копирования. Это происходит быстрее и занимает меньше места. Недостатком является то, что восстановление с разностных копий происходит сложнее и занимает больше времени. Для восстановления с разностной копии требуется восстановление полной резервной копии и всех последующих разностных копий.

#### Практическая часть

Продemonстрируем создание резервных копий базы данных. Для полного резервного копирования была выбрана база данных созданная для отдела продаж.

На рис. 2.14 можно увидеть, что база данных заполнена данными для проверки работы полного и инкрементного способа копирования данных.

	Код_накладной	Номер_склада	Код_счета	Код_покупате...	Код_товаров...	Дата_получен...
	1	2	1	1	2	2015-11-22 00:0...
»»	NULL	NULL	NULL	NULL	NULL	NULL

Рис. 2.14 Одна из заполненных таблиц базы данных

Для создания полной резервной копии необходимо открыть диалоговое окно для резервного копирования, представленного на рис. 2.15. В данном диалоговом окне необходимо выбрать, куда будет сохраняться копия базы данных и способы записи.

На рис. 2.16 представлена ситуация, когда злоумышленник удалил таблицу «Накладная», в списке объектов она отсутствует.

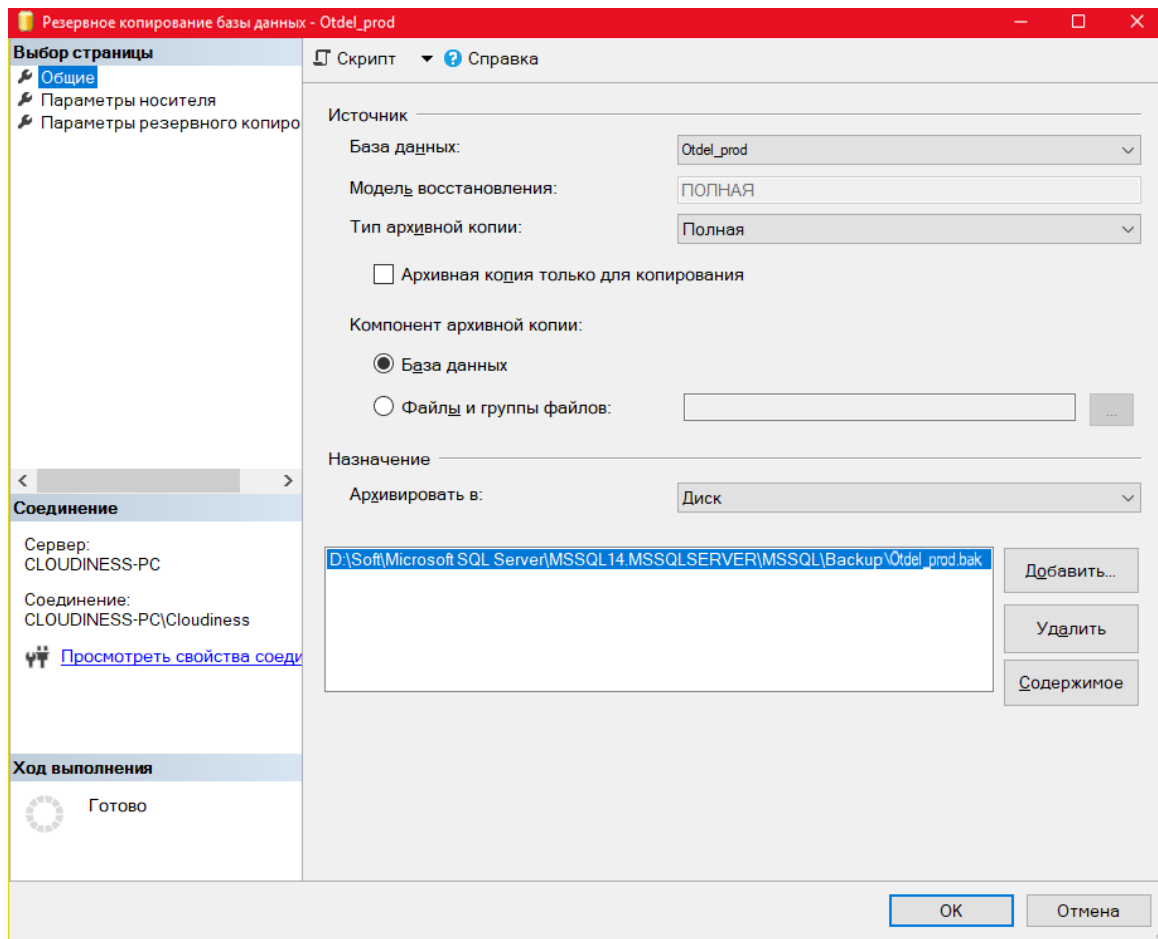


Рис. 2.15. Диалоговое окно создания резервной копии.

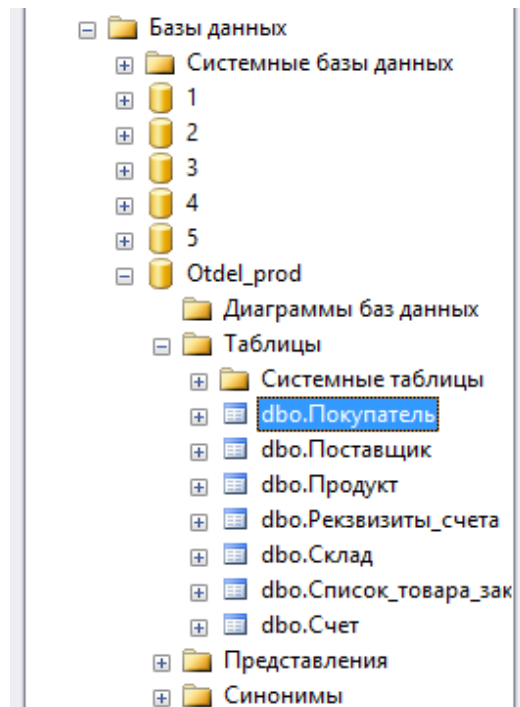


Рис. 2.16. Удаление таблицы «Накладная» из списка объектов

Пытаемся выполнить восстановление данных. Переходим в диалоговое окно с выбором файла для восстановления (рис. 2.17).

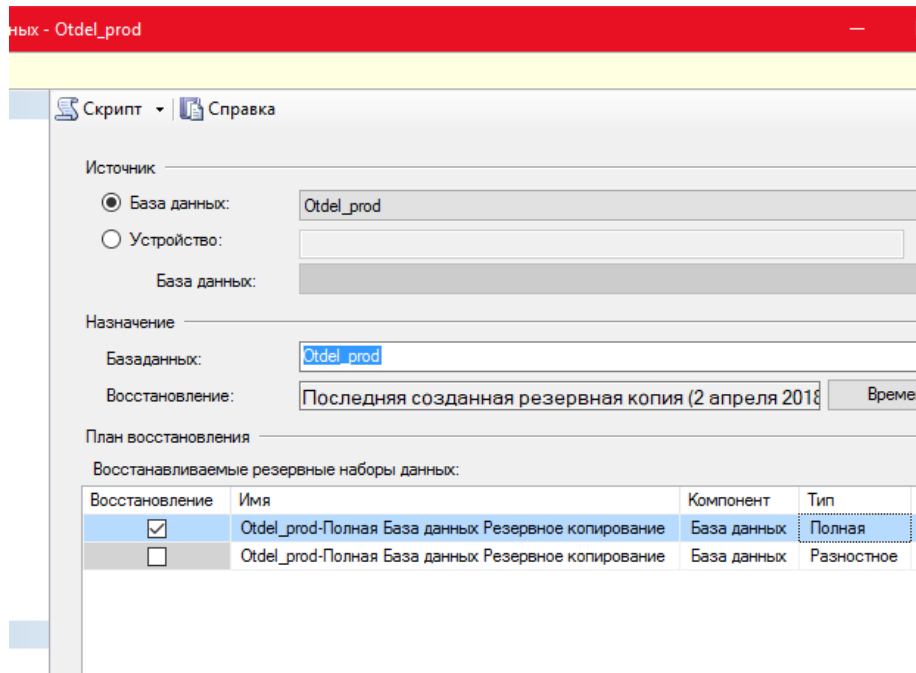


Рис. 2.17. Окно для выбора файла восстановления

После восстановления базы данных таблица «Накладная» появилась в списке объектов (рис. 2.18).

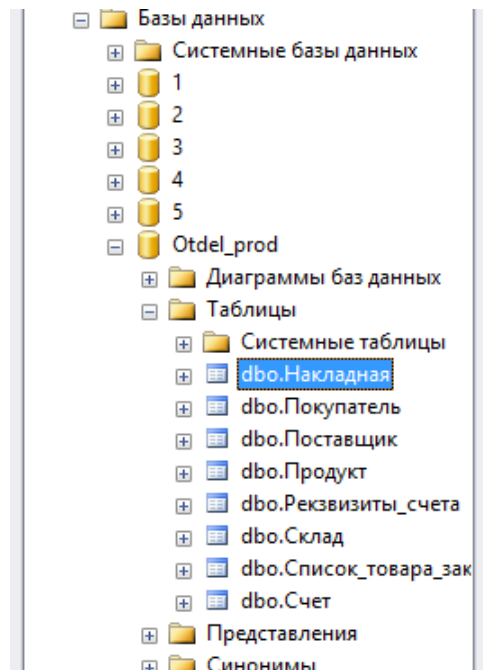
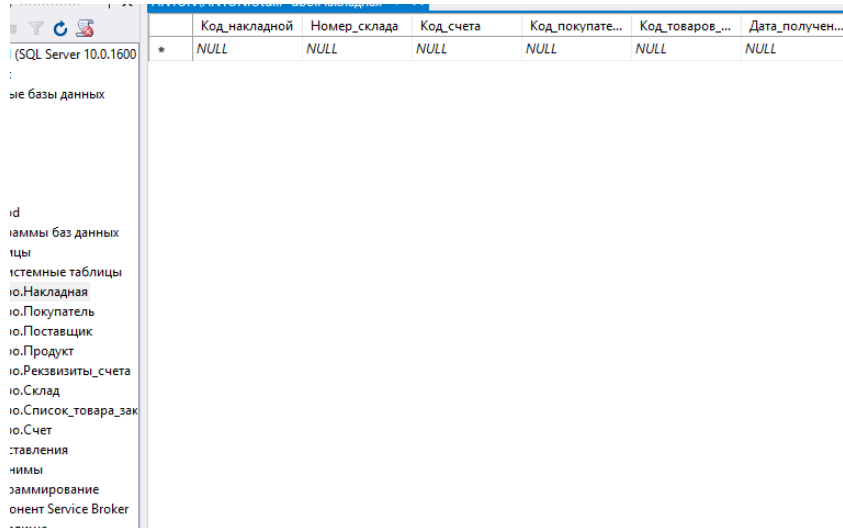


Рис. 2.18. Восстановление таблицы «Накладная»

Выполним восстановления данных с инкрементной копии, которая изменилась с момента последнего полного резервного копирования. Пример данного восстановления показан на рис. 2.19 – рис. 2.21.

На рис. 2.19 после восстановления из полной копии в таблице «Накладная» данные отсутствуют.



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Server Explorer' pane displays the database structure for 'Оtsel\_prod'. The 'Tables' folder is expanded, showing a list of tables including 'Накладная'. The main pane displays the 'Накладная' table, which contains a single row with all NULL values in the following columns: 'Код\_накладной', 'Номер\_склада', 'Код\_счета', 'Код\_покупате...', 'Код\_товаров...', and 'Дата\_получен...'.

Код_накладной	Номер_склада	Код_счета	Код_покупате...	Код_товаров...	Дата_получен...
NULL	NULL	NULL	NULL	NULL	NULL

Рис. 2.19. Таблица «Накладная» после восстановления из полной копии

В диалоговом окне для восстановления базы, мы уже выбираем разностный способ восстановления базы. Результат представлен ниже.

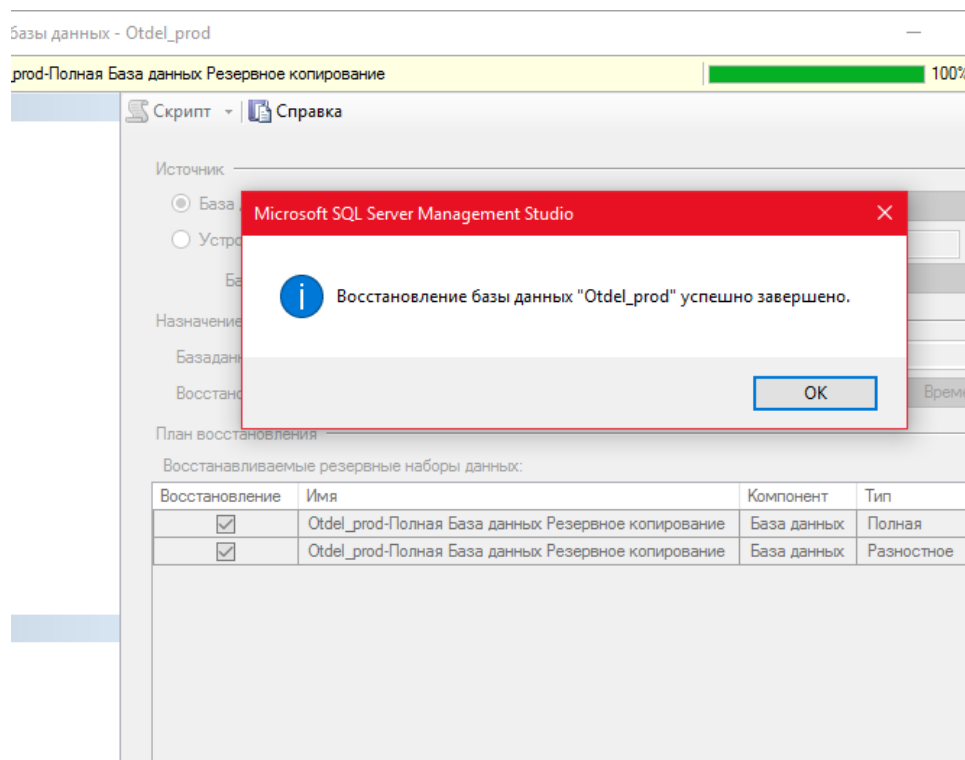
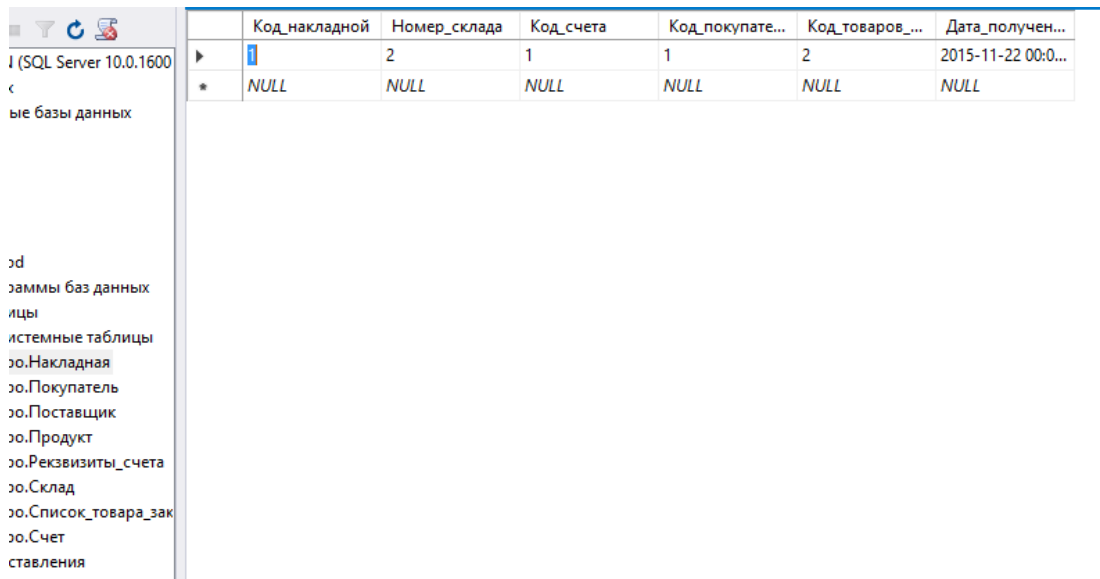


Рис. 2.20. Успешное восстановление данных

В результате на рис. 2.21 данные были успешно восстановлены.



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Данные базы данных' (Database Data) folder is expanded, showing the 'Накладная' (Receipt) table. The table is selected, and its data is displayed in the right pane. The table has seven columns: 'Код\_накладной' (Receipt Code), 'Номер\_склада' (Warehouse Number), 'Код\_счета' (Account Code), 'Код\_покупате...' (Buyer Code), 'Код\_товаров\_...' (Goods Code), and 'Дата\_получен...' (Date Received). The data is as follows:

	Код_накладной	Номер_склада	Код_счета	Код_покупате...	Код_товаров_...	Дата_получен...
▶	1	2	1	1	2	2015-11-22 00:0...
*	NULL	NULL	NULL	NULL	NULL	NULL

Рис. 2.21. Таблица «Накладная» после последнего восстановления

### Контрольные вопросы

1. Что такое полное резервное копирование?
2. Что такое инкрементное резервное копирование?
3. Какие критерии влияют на составление схемы резервного копирования?
4. Каким образом и кем должна определяться ценность информационного ресурса?

## 2.6. Лабораторная работа 6. Защита базы данных от SQL-инъекций

### Цель занятия

Получение навыков по защите баз данных от SQL-инъекций.

### Задача

Разработать и продемонстрировать модель защиты объектов базы данных от SQL-инъекций.

### План занятия

1. Ознакомиться с теоретическим материалом.
2. Прodelать пример на тестовом макете.
3. Обосновать три SQL-инъекции, представить их реализацию и методы защиты.

### Теоретические сведения

SQL-инъекция (внедрение SQL-кода) – один из распространённых способов взлома сайтов и программ, работающих с базами данных, основанный на внедрении в запрос произвольного SQL-кода.

SQL-инъекция, в зависимости от типа используемой СУБД и условий внедрения, может дать атакующему возможность выполнить произвольный запрос к базе данных (например, прочитать содержимое любых таблиц, удалить, изменить или добавить данные), получить возможность чтения и/или записи локальных файлов и выполнения произвольных команд на атакуемом сервере.

SQL-инъекции в большинстве случаев используются для изменения, удаления, добавления или чтения информации из БД путем внедрения в запросы к этой БД произвольного SQL-кода.

Обычно сначала производят поиск уязвимостей скриптов атакуемого сервера и восстановление структуры БД (названий таблиц, полей, записей) путем формирования различных запросов, которые могут привести к аномальной реакции сервера (такой, например, как сообщение об ошибке). Затем, когда структура БД восстановлена, с ней производят любые необходимые манипуляции. Универсальной защиты БД от любых SQL-инъекций нет, но существуют некоторые стандартные методы контроля над запросами: экранирование спецсимволов, максимальное усечение длины запроса, использование параметризованных запросов, разграничение доступа и т.д. Ниже представлен простейший пример SQL-инъекции для запроса:

```
SELECT * FROM news WHERE id_news = 10
```

```
SELECT * FROM news WHERE id_news = 10 OR 1=1
```

Приведенный пример SQL-инъекции, где в условие выборки вместо конкретного значения подставлено выражение — **OR 1=1**.



Это выражение всегда истинно, поэтому результатом запроса будет выборка всех записей этой таблицы.

Таким же образом можно использовать команды INSERT, DELETE, DROP и т.д., что позволяет проводить над БД незащищенного сервера любые необходимые действия.

Для защиты данного запроса от подобной SQL-инъекции, необходимо, чтобы:

- 1) Параметр – **id\_news** принимал только целочисленные значения;
- 2) Длина запроса была ограничена (для параметра – **id\_news** – двузначное число).

#### Пример реализации SQL-инъекции

Для реализации примера необходимо использовать макет с разработанной базой данной и разработать приложение по генерации SQL-инъекций.

На запрос вида

`SELECT* FROM Пользователи WHERE id = 1 or 1=@@version`

сервер отвечает сообщением об ошибке (рис. 2.22), в котором приведены сведения о версии сервера, версии операционной системы и т.д.

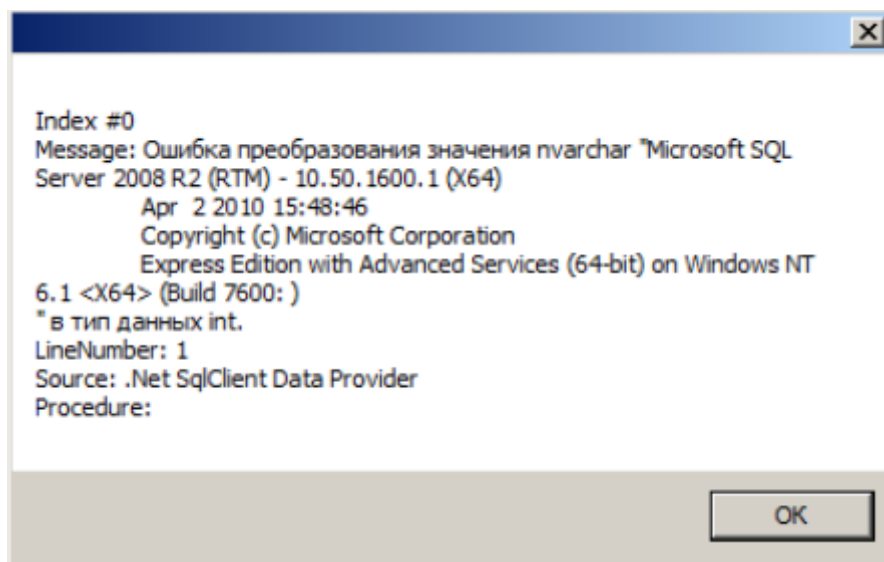


Рис. 2.22. Запрос на вывод информации о сервере

Чтобы защитить БД от подобной инъекции, можно использовать экранирование спецсимволами (например, кавычки). Этот способ защиты представлен ниже:

Текст\_Вставки = “1 or 1=1”

Текст\_Запроса = "SELECT\* FROM Пользователи WHERE id =" +  
Текст\_Вставки + " ' " .

Попытка обойти данный способ защиты возможен с помощью текста вставки следующего вида

Текст\_Вставки = "1' or 1=1".

Нарушение работы сервера, вызванное обращением к командной строке с последующим отключением процесса (SQL-сервера) командой—taskkill возможно следующей вставкой.

Текст\_Вставки = "1;exec xp\_cmdshell 'cmd/c taskkill /f /IM sqlservr.exe'".

Реализация метода защиты от SQL-инъекции

При разработке стратегии защиты от SQL-инъекций необходимо предусмотреть следующие базовые операции семантического разбора параметров запроса:

- операции, изменяющие структуру таблиц;
- запросы к служебным таблицам и базам;
- запросы с использованием доступа к файловой системе;
- запросы с использованием команд операционной системы;
- административные запросы, например, SHOW TABLES, SHOW CREATE TABLE;
- операции сравнения, всегда возвращающие ИСТИНУ, например, 1=1, field=field и т.д.;
- комментарии внутри запроса;
- использование логического оператора **OR** в запросе;
- операции над таблицами, содержащими личную информацию, например, users, accounts, payments;
- передача в запросе пустого пароля, например, password="", pwd="", passwd="".

В результате необходимо разработать клиентское приложение, способное продемонстрировать распознавание и блокировку атаки по любому из трех вариантов угрозы. В отчете требуется предоставить текст программы и результаты запросов с успешными блокировками атак.

### Контрольные вопросы

1. Опишите сущность угрозы, связанной с SQL- инъекцией.
2. Назовите базовые операции семантического разбора запроса для защиты от SQL- инъекций.
3. Назовите языки программирования, где присутствуют встроенные механизмы защиты от SQL- инъекций.

## 2.7. Лабораторная работа 7. Методы шифрования объектов базы данных

### Цель занятия

Ознакомление с методами шифрования объектов базы данных.

### Задача

Разработать и продемонстрировать модель защиты объектов базы данных с использованием симметричных, асимметричных ключей и сертификатов.

### План занятия

1. Подготовить модель базы данных для использования метода шифрования на уровне ячеек.
2. Создать необходимые ключи и сертификаты.
3. Подготовить SQL-сценарии для шифрования и расшифровки данных на уровне ячеек.
4. Выполнить сравнительный анализ различных методов шифрования.
5. Выполнить настройку прозрачного шифрования данных.

### Теоретические сведения

Модель шифрования MS SQL Server предоставляет собой функции управления ключами шифрования, соответствующие стандарту ANSI X9.17 (табл. 2.6). В этом стандарте определены несколько уровней ключей шифрования, где часть ключей используется для шифрования других ключей предназначенных для шифрования данных.

Таблица 2.6

Уровни шифрования ключей SQL Server и ANSI X9.17

Уровень SQL Server	Уровень ANSI X9.17	Описание
Service master key	Главный ключ	Ключ верхнего уровня, создается с применением Windows DP API
Database master key	Ключ шифрования ключей данных	Используется для шифрования ключей данных, используется только один ключ для каждой базы данных.
Симметричные, асимметричные ключи и сертификаты	Ключ данных	Используются для шифрования данных, применяются на уровне сеанса пользователей.

Главный ключ службы (Service master key, SMK) — ключ верхнего уровня и предок всех ключей в SQL Server. SMK — асимметричный ключ, шифруемый с использованием Windows Data Protection API (DPAPI). SMK автоматически создается, когда шифруется какой-нибудь объект, и привязан к учетной записи службы SQL Server. SMK используется для шифрования главного ключа базы данных (Database master key, DMK).

Второй уровень иерархии ключей шифрования — DMK. С его помощью шифруются симметричные ключи, асимметричные ключи и сертификаты. Каждая база данных располагает лишь одним DMK.

Следующий уровень содержит симметричные ключи, асимметричные ключи и сертификаты. Симметричные ключи — основное средство шифрования в базе данных. Microsoft рекомендует шифровать данные только с помощью симметричных ключей. Кроме того, в SQL Server 2008 и более новых версиях есть сертификаты уровня сервера и ключи шифрования базы данных для прозрачного шифрования данных. На рис. 2.23 показана иерархия ключей шифрования для SQL Server 2008 и более новых версий.

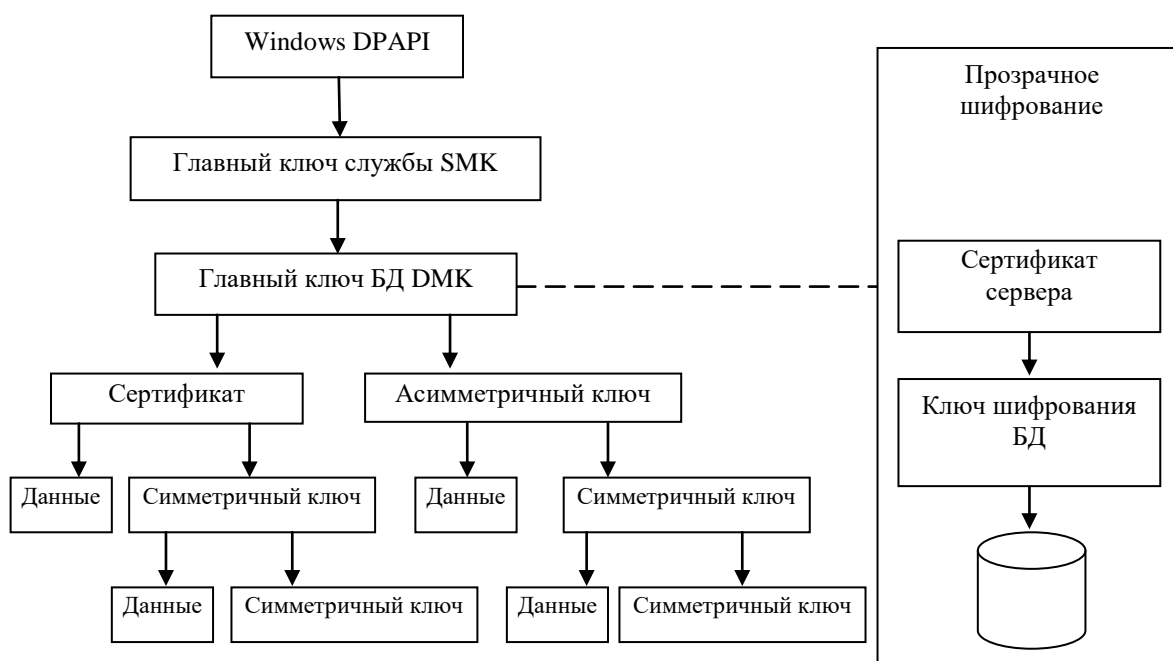


Рис. 2.23. Иерархия ключей шифрования в MS SQL Server

### Шифрование на уровне ячеек

Начиная с MS SQL Server 2005, можно шифровать или расшифровывать данные на сервере. Делать это можно различными способами.

- Пароль. Это наименее надежный способ, так как для шифрования и расшифровки данных используется одна и та же парольная фраза.

Если хранимые процедуры и функции не зашифрованы, то доступ к парольной фразе возможен через метаданные.

- Сертификат. Этот способ обеспечивает надежную защиту и высокое быстродействие. Сертификат можно связать с пользователем; подписать его необходимо с помощью главного ключа базы данных DMK.
- Симметричный ключ. Достаточно надежен, удовлетворяет большинству требований к безопасности данных и обеспечивает достаточное быстродействие. Для шифрования и расшифровки данных используется один ключ.
- Асимметричный ключ. Обеспечивает надежную защиту, так как применяются различные ключи для шифрования и расшифровки данных. Однако это негативно влияет на быстродействие. Специалисты Microsoft не рекомендуют использовать его для шифрования крупных значений. Асимметричный ключ может быть подписан с использованием главного ключа базы данных DMK или создан с помощью пароля.

MS SQL Server имеет встроенные функции для шифрования и расшифровки на уровне ячеек.

Функции шифрования:

- ENCRYPTBYKEY, использует симметричный ключ для шифрования данных;
- ENCRYPTBYCERT, использует открытый ключ сертификата для шифрования данных;
- ENCRYPTBYPASSPHRASE, использует парольную фразу для шифрования данных;
- ENCRYPTBYASYMKEY, использует асимметричный ключ для шифрования данных.

Функции расшифровки:

- DECRYPTBYKEY, использует симметричный ключ для расшифровки данных;
- DECRYPTBYCERT, использует открытый ключ сертификата для расшифровки данных;
- DECRYPTBYPASSPHRASE, использует парольную фразу для расшифровки данных;
- DECRYPTBYASYMKEY, использует асимметричный ключ для расшифровки данных;
- DECRYPTBYKEYAUTOASYMKEY, использует асимметричный ключ, который автоматически расшифровывает сертификат.

SQL Server располагает двумя системными представлениями, с помощью которых можно получить метаданные для всех симметричных и асимметричных ключей, существующих в экземпляре SQL Server:

sys.symmetric\_keys - возвращает метаданные для симметричных ключей;

sys.asymmetric\_keys - возвращает метаданные для асимметричных ключей;

sys.openkeys - информация о ключах шифрования, открытых в текущем сеансе пользователя.

### Практическая часть

1. Пример таблицы с подготовленным атрибутом с типом данных «varbinary» для хранения зашифрованных данных.

```
USE [test]
GO
CREATE TABLE [dbo].[CreditCard]
([ID] [int] PRIMARY KEY,
[CardNum] [varbinary](max))
GO
```

2. Создание главного ключа базы данных DMK для базы данных «Test».

```
USE [test]
GO
CREATE MASTER KEY ENCRYPTION BY PASSWORD='Str0nGPa$$w0rd'
GO
```

3. Проверка существования главного ключа.

```
USE [test]
GO
select * from sys.key_encryptions
```

4. Создание асимметричного ключа данных.

```
USE [test]
GO
CREATE ASYMMETRIC KEY MyAsymmetricKey
WITH ALGORITHM = RSA_2048
ENCRYPTION BY PASSWORD = 'StrongPa$$w0rd!'
```

GO

5. Создание симметричного ключа данных, зашифрованного асимметричным ключом.

```
USE [test]
GO
CREATE SYMMETRIC KEY MySymmetricKey
WITH ALGORITHM = AES_256
ENCRYPTION BY ASYMMETRIC KEY MyAsymmetricKey
GO
```

6. Открытие симметричного ключа для шифрования в сеансе пользователя.

```
USE [test]
GO
OPEN SYMMETRIC KEY MySymmetricKey
DECRYPTION BY ASYMMETRIC KEY MyAsymmetricKey
WITH PASSWORD = 'StrongPa$$w0rd!'
GO
```

7. Просмотр открытых сертификатов.

```
USE [test]
GO
SELECT * FROM [sys].[openkeys]
GO
```

8. Пример заполнения значений с использованием симметричного ключа.

```
USE [test]
GO
DECLARE @SymmetricKeyGUID AS [uniqueidentifier]
SET @SymmetricKeyGUID = KEY_GUID('MySymmetricKey')
IF (@SymmetricKeyGUID IS NOT NULL)
BEGIN
INSERT INTO [dbo].[CreditCard]
VALUES (01, ENCRYPTBYKEY(@SymmetricKeyGUID,
N'9876-1234-8765-4321'))
INSERT INTO [dbo].[CreditCard]
VALUES (02, ENCRYPTBYKEY(@SymmetricKeyGUID,
N'9876-8765-8765-1234'))
INSERT INTO [dbo].[CreditCard]
```

```
VALUES (03, ENCRYPTBYKEY(@SymmetricKeyGUID,
N'9876-1234-1111-2222'))
END
```

#### 9. Создание сертификата для симметричного ключа данных.

```
USE [test]
GO
CREATE CERTIFICATE [CertToEncryptSymmetricKey]
WITH SUBJECT = 'Сертификат для симметричного ключа.'
```

#### 10. Создание симметричного ключа данных на базе сертификата.

```
USE [test]
GO

CREATE SYMMETRIC KEY [SymmetricKeyEncryptedWithCert]
WITH ALGORITHM = AES_256
ENCRYPTION BY CERTIFICATE [CertToEncryptSymmetricKey]
```

#### 11. Открытие симметричного ключа в сеансе и проверка его статуса.

```
USE [test]
GO
OPEN SYMMETRIC KEY [SymmetricKeyEncryptedWithCert]
DECRYPTION BY CERTIFICATE [CertToEncryptSymmetricKey]
GO
SELECT * FROM [sys].[openkeys]
GO
```

#### 12. Шифрование данных с использованием симметричного ключа на базе сертификата

```
USE [test]
GO
DECLARE @SymmetricKeyGUID AS [uniqueidentifier]
SET @SymmetricKeyGUID =
KEY_GUID('SymmetricKeyEncryptedWithCert')
IF (@SymmetricKeyGUID IS NOT NULL)
BEGIN
INSERT INTO [dbo].[CreditCard]
VALUES (04, ENCRYPTBYKEY(@SymmetricKeyGUID,
N'9876-1234-8765-4321'))
INSERT INTO [dbo].[CreditCard]
VALUES (05, ENCRYPTBYKEY(@SymmetricKeyGUID,
N'9876-8765-8765-1234'))
INSERT INTO [dbo].[CreditCard]
VALUES (06, ENCRYPTBYKEY(@SymmetricKeyGUID,
```



N'9876-1234-1111-2222'))

END

13. Просмотр данных, зашифрованных различными способами через представление. При условии открытия всех необходимых симметричных ключей (рис. 2.24).

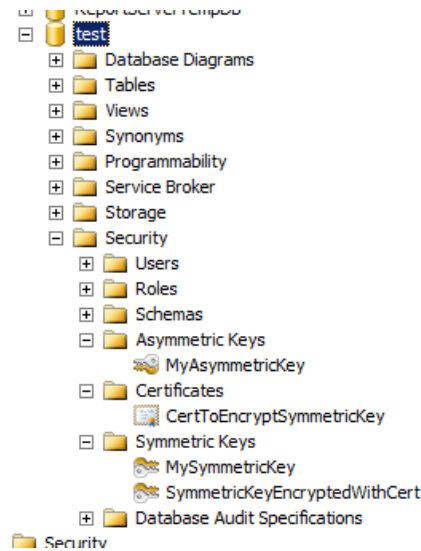


Рис. 2.24. Список ключей базы данных

```
USE [test]
GO
SELECT [ID],
CONVERT([nvarchar](32), DECRYPTBYKEY(CardNum))
AS [CreditCard]
FROM [dbo].[CreditCard]
GO
```

Результат запроса представлен ниже.

	ID	CardNum
1	1	0x002D8D73E9458D4AB933740A4FA1837C01000009221E991D968559F42B8560896F9D09856EC9EB5DD863DF983C061F3EF6A06A39E10543B57C0E80C81910F26B6FD5C4EA0050B7D5656D8277A01B12AA92BCC45
2	2	0x002D8D73E9458D4AB933740A4FA1837C01000000872B98A3C380D74CA0C5FD9DF93099180AEA15EB6DF2110BAC3747987F094ED8BC5F942A198DB206225BA10F99CFF1A5D330D212410D2DA151B0BF552B567EEB
3	3	0x002D8D73E9458D4AB933740A4FA1837C01000000E8D68E782DF48B7CFE3E89F4AAF814A7C00183E9F0D2B68343B21BBFC18FA339D0166CA820E3246F19BD1913AB6C6E6EB828165FB8D32EA3B9A37850B033EBF6
4	4	0x00C2415F7132574093B6AD38FABE1BE0100000050074E60AD4F09ED18763969164B2E862331A906FB3B108DCA3422A6E30CA8E2E5CF4895597B3BD551479A278BF405DB743E5F7B17A42AF7C6F81A6AC3A3BF33
5	5	0x00C2415F7132574093B6AD38FABE1BE01000000D16C8A9135F43A5747ED9232EA35210FA76A4CA91CBFEF8D7FBF181DF4485092088FB55C2FD7A8D41F29ECE50C2BCF668A5BE6A110F42259A868B9B14829C3FC
6	6	0x00C2415F7132574093B6AD38FABE1BE01000000CFF0BC9CC8E69CB2F6DACB9A92E59E851C1F98152AA33C67354FF4DF2C5DCBD85C91B66ED9C7291E001E9E1EE2572885999179103EBAD85F5C7BB988ABD92A80

а)

	ID	CreditCard
1	1	9876-1234-8765-4321
2	2	9876-8765-8765-1234
3	3	9876-1234-1111-2222
4	4	9876-1234-8765-4321
5	5	9876-8765-8765-1234
6	6	9876-1234-1111-2222

б)

Рис. 2.25. Результат запроса без расшифровки (а) и с расшифровкой (б)

## Использование прозрачного шифрования

1. Создание главного ключа и сертификата для сервера выполняется следующим сценарием.

```
USE [master]
GO
```

```
CREATE MASTER KEY ENCRYPTION BY
PASSWORD = '$tr0ngPa$$w0rd1'
GO
```

```
CREATE CERTIFICATE EncryptedDBCert
WITH SUBJECT = 'Certificate to encrypt EncryptedDB';
GO
```

Далее необходимо убедиться в его наличии для продолжения работы (рис. 2.26).

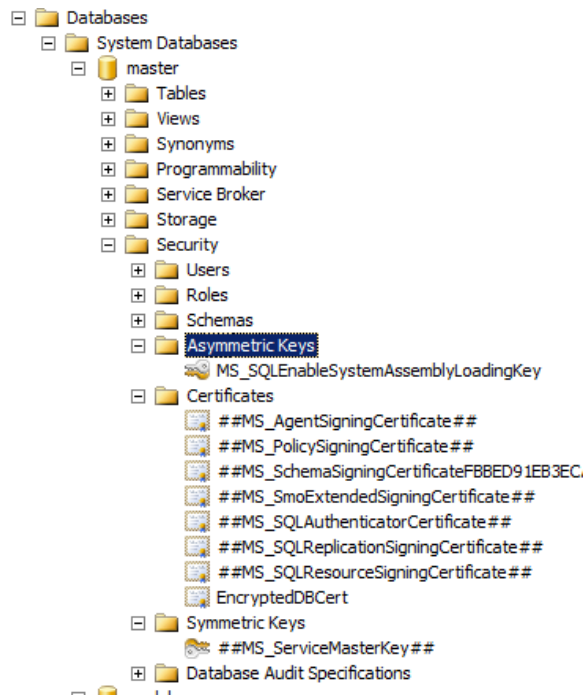


Рис. 2.26. Проверка создания ключа и сертификата

2. Активация прозрачного шифрования всей базы данных. Функционирует только в редакции MS SQL Server Enterprise или Developer.

Для проверки редакции сервера используется следующий запрос.

```
SELECT SERVERPROPERTY('productversion'), SERVERPROPERTY ('productlevel'),
SERVERPROPERTY ('edition')
```

Если версия СУБД удовлетворяет необходимым требованиям, то продолжаем работу.

```
USE [master]
GO
```

```
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_256
ENCRYPTION BY SERVER CERTIFICATE [EncryptedDBCert]
GO
ALTER DATABASE [test]
SET ENCRYPTION ON
GO
```

### 3. Проверка состояния шифрования всех баз данных на сервере.

```
USE [master]
GO
SELECT db.[name]
, db.[is_encrypted]
, dm.[encryption_state]
, dm.[percent_complete]
, dm.[key_algorithm]
, dm.[key_length]
FROM [sys].[databases] db
LEFT OUTER JOIN [sys].[dm_database_encryption_keys] dm
ON db.[database_id] = dm.[database_id];
GO
```

### Контрольные вопросы

1. Какие методы шифрования данных могут поддерживать СУБД?
2. Назовите рекомендуемые методы шифрования.
3. Какие данные рекомендуется подвергать шифрованию?
4. Какие преимущества дает применение прозрачного шифрования?

## Список литературы

1. Давыдова Е.М., Сопов М.А. Безопасность систем баз данных: Учебно-методические указания по лабораторным работам. Часть 2 / ТУ-СУР. – Томск, 2012. – 29 с.
2. Советов Б. Я. Базы данных: теория и практика : учеб. для студентов вузов / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовский . –2-е изд. – Москва : Юрайт, 2012 . –464 с.
3. Малыхина М. П. Базы данных : основы, проектирование, использование: [учеб. пособие] / М. П. Малыхина. – Санкт-Петербург : БХВ-Петербург, 2004 .—512 с.
4. Зыков Р.И. Системы управления базами данных / Р.И. Зыков. –М.: Лаборатория книги, 2012. -162 с. : табл., схем. -ISBN 978-5-504-00394-8; [Электронный ресурс]. - URL:<http://biblioclub.ru/index.php?page=book&id=142314>.
5. Мельников В.В. Безопасность информации в автоматизированных системах / В.В. Мельников. - М.: Финансы и статистика, 2003. -368 с. - ISBN 978-5-279-02560-2; [Электронный ресурс]. -URL: <http://biblioclub.ru/index.php?page=book&id=221458>.
6. Кашалов В. GreenSQL: Защита SQL-серверов от инъекций; [Электронный ресурс]. -URL: <https://habr.com/ru/post/117375/>.
7. Фарук Б. Модель шифрования SQL Server; [Электронный ресурс]. - URL: <http://www.interface.ru/home.asp?artId=36233/>.

**Приложение 1**

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**  
федеральное государственное автономное образовательное учреждение высшего образования  
«Санкт-Петербургский государственный университет  
аэрокосмического приборостроения»

КАФЕДРА № 34

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

---

должность, уч. степе-  
нь, звание

---

подпись, дата

---

инициалы, фами-  
лия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

**НАЗВАНИЕ ЛАБОРАТОРНОЙ РАБОТЫ**

по курсу: БЕЗОПАСНОСТЬ СИСТЕМ БАЗ ДАННЫХ

РАБОТУ ВЫПОЛНИЛ  
СТУДЕНТ  
ГР. №

---

подпись, дата

---

инициалы, фа-  
милia

Санкт-Петербург 20\_\_

## Приложение 2

### Примерные темы для лабораторных работ

1. Разработка базы данных учета компьютерной техники и периферийных устройств.
2. Разработка базы данных электронного документооборота.
3. Разработка базы данных для информационной системы аудита информационной безопасности.
4. Разработка базы данных для информационной системы учета готовой продукции.
5. Разработка базы данных для информационной системы контроля и управления доступом.
6. Разработка базы данных для системы заказов розничной/оптовой торговли.
7. Разработка базы данных система безопасности жилого комплекса.
8. Разработка базы данных для информационной системы мониторинга состояния городского района.
9. Разработка базы данных для автоматизированной системы оценки уровня знаний.
10. Разработка базы данных агентства помощи в трудоустройстве.
11. Разработка базы данных выпускников вуза.
12. Разработка базы данных производственная практика.
13. Разработка базы данных по учету учебно-методических комплексов вуза.
14. Разработка базы данных делопроизводства кафедры/факультета.
15. Разработка базы данных по учету текущей успеваемости студентов.
16. Разработка базы данных для мониторинга посещений сайта.
17. Разработка базы данных для информационной системы обнаружения вторжений.
18. Разработка базы данных для учета и контроля лицензий на услуги/товары/продукцию.
19. Разработка базы данных деятельности кафе/столовой/ресторана.
20. Разработка базы данных для учета товаров и материалов на складе организации.
21. Разработка базы данных телефонного справочника для частных и юридических лиц.
22. Разработка базы данных для камеры хранения.
23. Разработка базы данных для кассы авиа/железнодорожного/ автотранспорта.
24. Разработка базы данных деятельности медицинского учреждения.

- 25.Разработка базы данных деятельности аптеки.
- 26.Разработка базы данных для туристического агентства.
- 27.Разработка базы данных библиотеки.
- 28.Разработка базы данных по работе с электронными картами клиентов и счетами банка.
- 29.Разработка базы данных для продажи и проката видеопродукции.
- 30.Разработка базы данных сервиса покупки/продажи/проката автомобильной техники.
- 31.Разработка базы данных деятельности страховой компании.
- 32.Разработка базы данных для кадровой службы.
- 33.Разработка базы данных для агентства недвижимости.
- 34.Разработка базы данных ГИБДД для учета транспортных средств.
- 35.Разработка базы данных сервисного центра по гарантийному и пост гарантийному обслуживанию.
- 36.Разработка базы данных для организации спортивных мероприятий.
- 37.Разработка базы данных для организации концертных и культурно-массовых мероприятий.
- 38.Разработка базы данных паспортного стола отдела внутренних дел.
- 39.Разработка базы данных для управляющей компании жилищного комплекса.
- 40.Разработка базы данных по согласованной с преподавателем теме.