|  |  |
| --- | --- |
| Логотип КБК копия | **Автономная некоммерческая организация профессионального образования**  **КАЛИНИНГРАДСКИЙ БИЗНЕС-КОЛЛЕДЖ** |

**Курсовая работа**

На тему: **«**ПУНКТ ПРОКАТА АВТОМОБИЛЕЙ**»**

По дисциплине: **МДК 11.01 Технология разработки и защиты баз данных**

Выполнили студенты

группы 23-ИСП-3/1:

Полвонов С. С. и Одинаев В. М.

Проверил(а):

Дубинин А. В.

Оценка:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Калининград**

**2025**

**Содержание**

[Введение 2](#_Toc137130790)

[1. Анализ предметной области 3](#_Toc137130791)

[1.1 Описание предметной области 3](#_Toc137130792)

[1.2 Определение функциональных требований 4](#_Toc137130793)

[2. Концептуальное (инфологическое) проектирование 4](#_Toc137130794)

[2.1 Описание сущностей 5](#_Toc137130795)

[2.2 Определение связей между сущностями 6](#_Toc137130796)

[2.3 Создание ER-модели предметной области 6](#_Toc137130797)

[3. Выбор СУБД 7](#_Toc137130798)

[4. Даталогическое проектирование 10](#_Toc137130799)

[4.1. Даталогическая модель 10](#_Toc137130800)

[5. Физическое проектирование 13](#_Toc137130801)

[5.1 Таблицы 13](#_Toc137130802)

[5.2 Хранимые процедуры 16](#_Toc137130803)

[5.3 Разработка интерфейса 18](#_Toc137130804)

[Заключение 25](#_Toc137130805)

[Список использованных источников 26](#_Toc137130806)

# **Введение**

В последние годы наблюдается активный рост популярности и значимости службы проката автомобилей. Увеличение числа туристов в регионе, а также растущий интерес к гибкости передвижения среди граждан обусловливает высокий спрос на услуги краткосрочной и долгосрочной аренд транспортных средств. В современном быстро меняющемся мире, где время и мобильность становятся критически важными факторами, создание эффективной базы данных для управления пунктами проката автомобилей представляется особенно актуальным.

Одной из ключевых проблем, с которыми сталкиваются компании в этой сфере, является необходимость оптимизации процессов управления парком автомобилей, учёта клиентов и обработки платежей. Наличие централизованной базы данных позволяет значительно повысить эффективность работы, упростить взаимодействие с клиентами и предоставить качественные услуги. Разработка эффективной системы на базе современных информационных технологий не только способствует улучшению обслуживания клиентов, но и помогает увеличить прибыль бизнеса за счёт оптимизации затрат и повышения точности учёта.

Цель данной курсовой работы состоит в разработке базы данных для пункта проката автомобилей, которая позволит автоматизировать процессы учёта, управления и обработки данных.

Для достижения данной цели необходимо решить ряд задач:

1. Исследовать требования к функциональности базы данных для пункта проката автомобилей.

2. Проанализировать существующие методы проектирования баз данных и выбрать наиболее подходящий для данной задачи.

3. Спроектировать структуру базы данных, включая все необходимые таблицы и их взаимосвязи.

4. Реализовать базу данных с использованием выбранной СУБД (системы управления базами данных) и протестировать её функциональность.

5. Оформить документацию, описывающую разработанную базу данных и её возможности.

В процессе работы будут рассмотрены различные методы разработки баз данных, такие как ER-диаграммы для моделирования сущностей и их взаимосвязей, реляционная модель для структурирования данных, а также применение SQL для реализации запросов и обработки данных в базе. Это позволит не только изучить теоретические аспекты, но и применить их на практике, создавая эффективный инструмент для управления прокатом автомобилей.

Таким образом, данная курсовая работа охватывает важные аспекты проектирования и разработки баз данных, представляя собой необходимый шаг к оптимизации процессов в сфере проката автомобилей.

В достижении поставленных задач будут использованы следующие инструменты:

1. Microsoft Visual Studio — Это стартовая площадка для написания, откладки и сборки кода, а также последующие публикации приложения.

2.Microsoft Visio — это программное решение, которое помогает визуализировать бизнес-процессы, которые связаны с определёнными данными.

3.ERwin —программное обеспечение для проектирования и документирования баз данных.

4.Microsoft SQL Server — современная система управления базами данных.

1. **Анализ предметной области**

## **1.1 Описание предметной области**

База данных проката автомобилей является ключевым инструментом для организаций, предоставляющих услуги аренды. Она помогает эффективно управлять информацией о доступных автомобилях, отслеживать автопарк и предоставлять клиентам актуальную информацию о доступных вариантах аренды.

В базе содержится подробная информация о каждом автомобиле: марка, модель, год выпуска, тип кузова, цвет и технические характеристики. Также фиксируется статус автомобиля (доступен, арендован, на обслуживании) и дата последней аренды или обслуживания.

Кроме данных об автомобилях, база хранит информацию о клиентах, включая их имя, контактные данные и информацию о водительских правах.

База данных также включает информацию о тарифах и условиях аренды: ставки, сроки аренды, возрастные ограничения и требования к стажу вождения. Она позволяет выполнять поиск доступных автомобилей, бронирование, регистрацию новых клиентов, обновление статуса автомобилей и аренды, а также генерировать отчеты о прокате и доходах.

Таким образом, база данных проката автомобилей обеспечивает эффективное управление, надежное хранение и учет информации о транспортных средствах, клиентах и операциях аренды:

- Аренда автомобилей на длительный и краткосрочный срок

- Аренда автомобилей с водителем

- Корпоративная аренда

- Доставка и возврат автомобилей

- Различные типы автомобилей

- Дополнительные услуги

- Онлайн-бронирование

## **1.2 Определение функциональных требований**

Транзакционные требования (задачи учета):

1. Ввод и корректировка сведений о клиентах, автомобилях.
2. Регистрация выдачи/возврата в прокат автомобиля.
3. Ведение системы скидок клиентам.

Справочные требования (оперативные запросы):

1. Сведения о конкретном клиенте, сколько раз пользовался услугами фирмы, когда и какие автомобили брал в прокат.
2. Наличие свободных, отданных на прокат, заказанных заранее автомобилей (по маркам, моделям).
3. Прайс-лист услуг.

Справочно-расчётные требования (аналитические запросы):

1. Сумма выручки за период или на определённую дату.
2. Самые популярные автомобили для проката.
3. Анализ и расчёт количества аренд автомобилей по маркам и моделям.
4. **Концептуальное (инфологическое) проектирование**

Цель этапа концептуального проектирования – создание концептуальной модели данных исходя из представлений пользователей. Инфологическое проектирование прежде всего связано с попыткой представления семантики предметной области в модели БД.

Для ее достижения выполняется ряд процедур:

1. Определение сущностей и связей между ними
2. Описание атрибутов, ключевых атрибутов и внешних ключевых атрибутов
3. Создание ER-модели предметной области

Основными конструктивными элементами инфологических моделей являются сущности, связи между ними и их свойства (атрибуты).

Сущность (Объект) – Идентифицируемый различными способами объект, отличающихся от других объектов.

Атрибут – свойство, характеризующее сущность.

Ключевой атрибут – позволяет однозначно идентифицировать сущность.

Связь – ассоциация между сущностями

**2.1 Описание сущностей**

В процессе обследования предметной области были следующие выявлены сущности, представленные в таблице 1.

Таблица 1

Список сущностей

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование сущности** | **Краткое описание (атрибуты)** |
| 1 | **Автомобили** | Сущность "Автомобили" описывает автомобили, доступные для аренды. Атрибуты:  - **ID\_Автомобиля** — уникальный идентификатор.  - **Марка** — производитель автомобиля.  - **Модель** — модель автомобиля.  - **Стоимость\_за\_день** — стоимость аренды за день.  - **Статус** — текущее состояние автомобиля (доступен/занят). |
| 2 | **Клиенты** | Сущность "Клиенты" описывает клиентов, которые арендуют автомобили. Атрибуты:  - **ID\_Клиента** — уникальный идентификатор.  - **Паспорт\_ID** — паспортные данные клиента.  - **Имя** — имя клиента.  - **Телефон** — номер телефона клиента.  - **Электронная\_почта** — почта клиента.  - **Скидка** — процент скидки. |
| 3 | **Заказы** | Сущность "Заказы" описывает аренду автомобилей клиентами. Атрибуты:  - **ID\_Заказа** — уникальный идентификатор заказа.  - **Клиент\_ID** — внешний ключ на клиента.  - **Автомобиль\_ID** — внешний ключ на автомобиль.  - **Дата\_Начала** — дата начала аренды.  - **Дата\_Окончания** — дата окончания аренды.  - **Базовая\_Стоимость** — предварительная стоимость аренды.  - **Стоимость\_Со\_Скидкой** — стоимость с учетом скидки.  - **Итоговая\_Стоимость** — итоговая стоимость аренды. |
| 4 | **Скидки** | Сущность "Скидки" описывает скидки для клиентов. Атрибуты:  - **ID\_Скидки** — уникальный идентификатор скидки.  - **Клиент\_ID** — внешний ключ на клиента.  - **Процент\_Скидки** — процент скидки. |
| **№ п/п** | **Наименование сущности** | **Краткое описание (атрибуты)** |
| 5 | **История заказов** | Сущность "История заказов" хранит информацию о завершенных арендах. Атрибуты:  - **ID\_Истории** — уникальный идентификатор записи.  - **Клиент\_ID** — внешний ключ на клиента.  - **Автомобиль\_ID** — внешний ключ на автомобиль.  - **Дата\_Аренды** — дата начала аренды.  - **Дата\_Возврата** — дата возврата автомобиля.  - **Итоговая\_Стоимость** — итоговая стоимость аренды. |
| 6 | **Заказы\_Услуги** | Сущность "Заказы\_Услуги" описывает дополнительные услуги, предоставляемые клиентам. Атрибуты:  - **Заказ\_ID** — внешний ключ на заказ.  - **Услуга\_ID** — внешний ключ на услугу.  - **Стоимость\_Услуги** — стоимость услуги. |
| 7 | **Дополнительные услуги** | Сущность "Дополнительные услуги" описывает услуги, доступные клиентам. Атрибуты:  - **ID\_Услуги** — уникальный идентификатор услуги.  - **Название** — название услуги.  - **Описание** — описание услуги.  - **Стоимость** — стоимость услуги. |

## **2.2. Определение связей между сущностями**

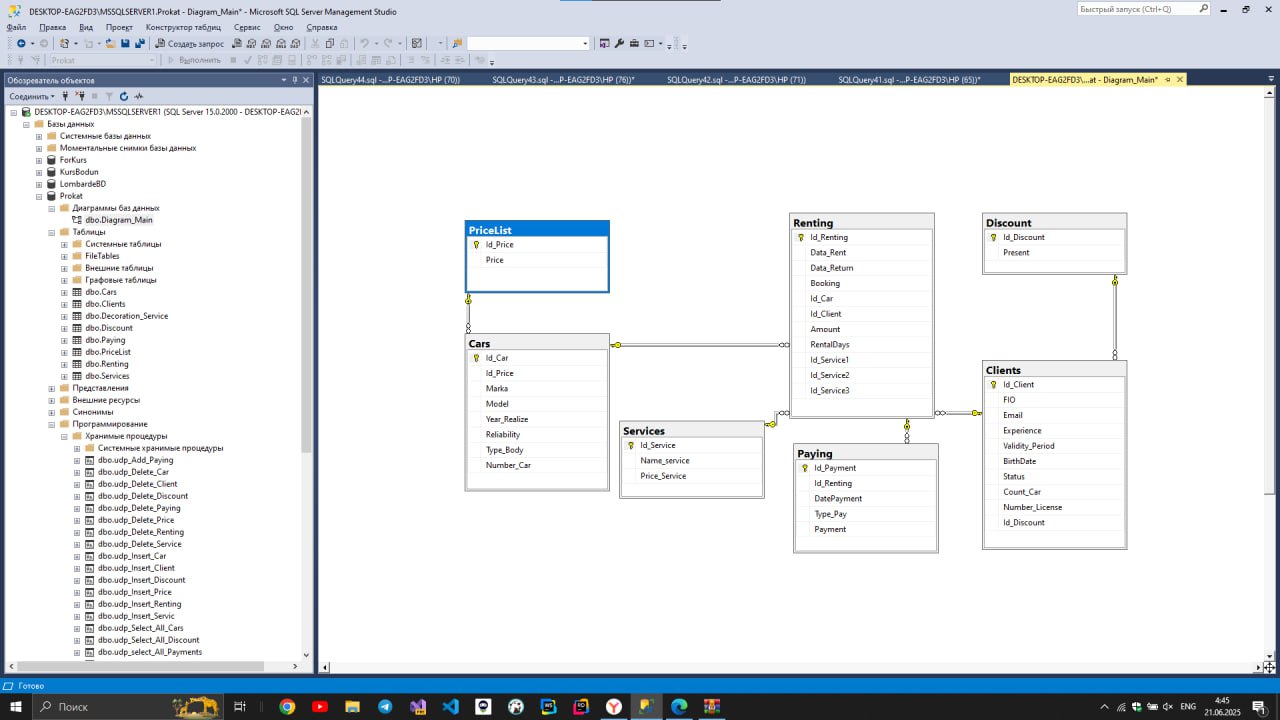


Рис. 1. Схема базы данных

Выше представлены связи между сущностями (см. рис. 1)

# **2.3 Создание ER-модели предметной области**

Исходя из выявленных сущностей, их атрибутов и связей между сущностями была разработана инфологическая модель, учитывающая все функциональные требования пользователя (Рис. 2).

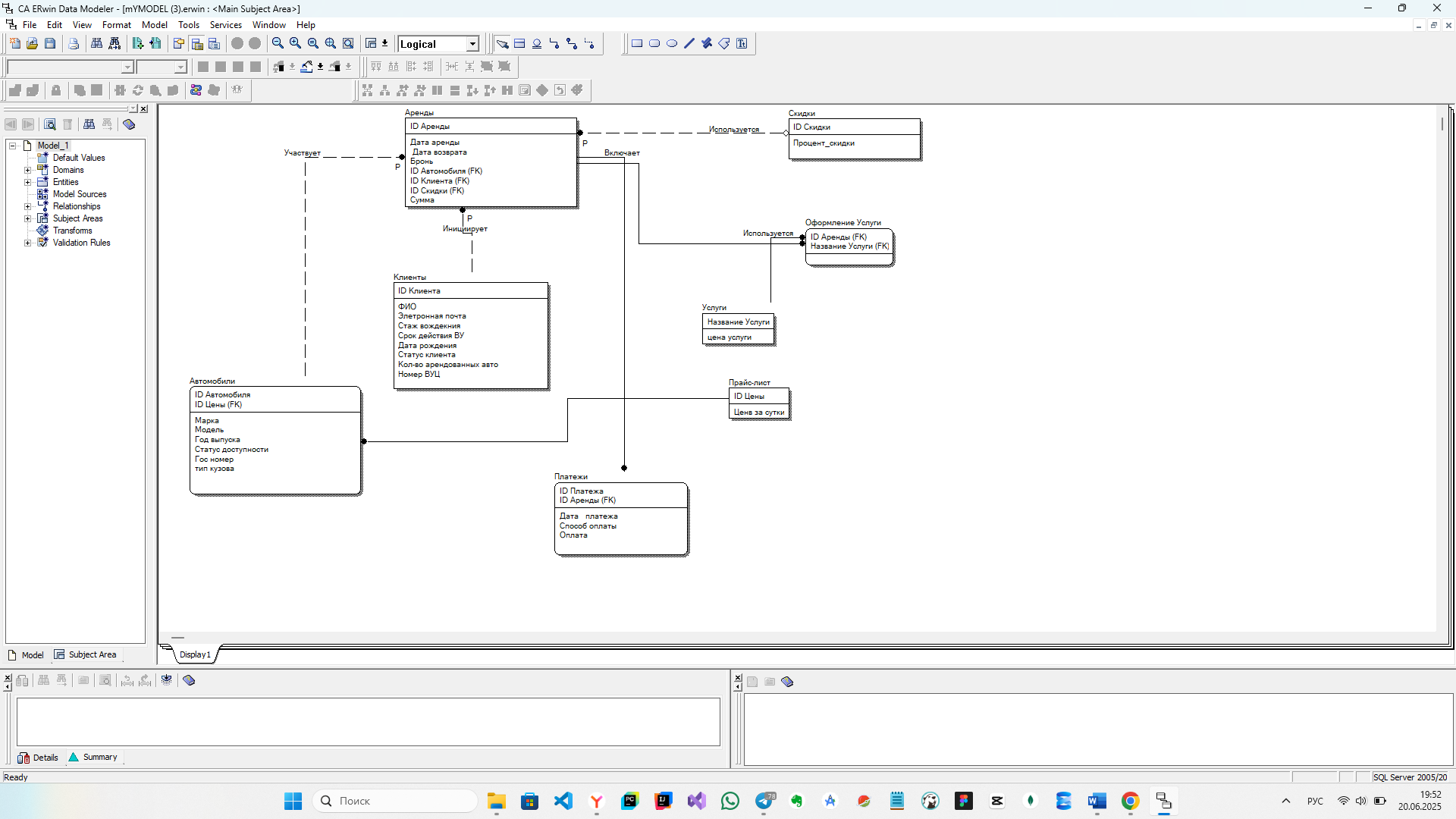


Рис. 2. ER-модель предметной области

Список ограничений:

**1. Таблица: Клиенты**

**id: Примарный ключ, уникальный идентификатор клиента. Значение не может быть NULL.**

**паспорт\_id: Не может быть пустым (NOT NULL), уникальные паспортные данные.**

**телефон: Не может быть пустым (NOT NULL), должно быть уникальным для каждого клиента.**

**электронная\_почта: Не может быть пустым (NOT NULL), должна быть уникальной.**

**скидка: Значение должно быть в диапазоне от 0 до 100 (включительно). Используется ограничение CHECK для проверки корректности значения скидки.**

**2. Таблица: Заказы**

**id: Примарный ключ, уникальный идентификатор заказа. Значение не может быть NULL.**

**клиент\_id: Внешний ключ, который ссылается на таблицу Клиенты. Обеспечивает связь с клиентом, сделавшим заказ. Значение не может быть NULL.**

**автомобиль\_id: Внешний ключ, который ссылается на таблицу Автомобили. Обеспечивает связь с арендованным автомобилем. Значение не может быть NULL.**

**дата\_начала: Не может быть пустым (NOT NULL), дата начала аренды.**

**дата\_окончания: Не может быть пустым (NOT NULL), дата окончания аренды. Дата окончания не может быть раньше даты начала.**

**базовая\_стоимость: Не может быть пустым (NOT NULL), стоимость аренды без учета скидки.**

**стоимость\_со\_скидкой: Не может быть пустым (NOT NULL), расчет стоимости аренды с учетом скидки.**

**итоговая\_стоимость: Не может быть пустым (NOT NULL), итоговая стоимость аренды, включая все скидки и возможные дополнительные расходы.**

**Таблица: Автомобили**

**id: Примарный ключ, уникальный идентификатор автомобиля. Значение не может быть NULL.**

**марка: Не может быть пустым (NOT NULL), марка автомобиля.**

**модель: Не может быть пустым (NOT NULL), модель автомобиля.**

**стоимость\_за\_день: Не может быть пустым (NOT NULL), стоимость аренды автомобиля за день.**

**статус: Не может быть пустым (NOT NULL), должен быть одним из двух значений: "доступен" или "занят".**

**Таблица: Скидки**

**id: Примарный ключ, уникальный идентификатор скидки. Значение не может быть NULL.**

**клиент\_id: Внешний ключ, который ссылается на таблицу Клиенты. Обеспечивает связь скидки с клиентом. Значение не может быть NULL.**

**процент\_скидки: Не может быть пустым (NOT NULL), процент предоставляемой скидки. Значение должно быть в пределах от 0 до 100.**

Таблица: История заказов

id: Примарный ключ, уникальный идентификатор записи истории. Значение не может быть NULL.

клиент\_id: Внешний ключ, который ссылается на таблицу Клиенты. Обеспечивает связь с клиентом, сделавшим заказ. Значение не может быть NULL.

автомобиль\_id: Внешний ключ, который ссылается на таблицу Автомобили. Обеспечивает связь с арендованным автомобилем. Значение не может быть NULL.

дата\_аренды: Не может быть пустым (NOT NULL), дата начала аренды.

дата\_возврата: Не может быть пустым (NOT NULL), дата возврата автомобиля.

итоговая\_стоимость: Не может быть пустым (NOT NULL), итоговая стоимость аренды.

Таблица: Заказы\_Услуги

заказ\_id: Внешний ключ, который ссылается на таблицу Заказы. Обеспечивает связь с заказом. Значение не может быть NULL.

услуга\_id: Внешний ключ, который ссылается на таблицу Дополнительные услуги. Обеспечивает связь с услугой. Значение не может быть NULL.

стоимость\_услуги: Не может быть пустым (NOT NULL), стоимость предоставленной услуги.

Таблица: Дополнительные услуги

id: Примарный ключ, уникальный идентификатор услуги. Значение не может быть NULL.

название: Не может быть пустым (NOT NULL), название услуги.

описание: Не может быть пустым (NOT NULL), описание услуги.

стоимость: Не может быть пустым (NOT NULL), стоимость услуги.

# **3.Выбор СУБД**

СУБД содержит в своём составе программные средства создания баз данных, средства работы с данными и сервисные средства. С помощью средств создания БД проектировщик, используя язык описания данных, переводит логическую модель БД в физическую структуру, а на языке манипуляции данными разрабатывает программы, реализующие основные операции с данными (в реляционных БД – это реляционные операции). При проектировании привлекаются визуальные средства, т. е. объекты, и программа-отладчик, с помощью которой соединяются и тестируются отдельные блоки разработанной программы управления конкретной БД.

В данном проекте выбор системы управления баз данных представлял сложную задачу и являлся одним из важных этапов при разработке приложений баз данных. Выбранный программный продукт должен был удовлетворять как текущим, так и будущим потребностям предприятия, для которого создавалась данная БД, при этом следует учитывать финансовые затраты на приобретение необходимого оборудования, самой системы, разработку необходимого программного обеспечения на ее основе, а также обучение персонала. Кроме того, необходимо убедиться, что новая СУБД способна принести предприятию реальные выгоды, а не расходы.

Очевидно, наиболее простой подход при выборе СУБД основан на оценке того, в какой мере существующие системы удовлетворяют основным требованиям создаваемого проекта информационной системы. Более сложным и дорогостоящим вариантом является создание испытательного проекта на основе нескольких СУБД и последующий выбор наиболее подходящего из кандидатов. Но и в этом случае необходимо ограничивать круг возможных систем, опираясь на некие критерии отбора. Вообще говоря, перечень требований к СУБД, используемых при анализе той или иной информационной системы, может изменяться в зависимости от поставленных целей. Тем не менее можно выделить несколько групп критериев:

* Моделирование данных
* Особенности архитектуры и функциональные возможности
* Контроль работы системы
* Особенности разработки приложений
* Производительность
* Надёжность
* Требования к рабочей среде
* Смешанные критерии

Следуя критериям в качестве среды реализации БД, были рассмотрены следующие СУБД: Oracle, MS SQL Server, PostgreSQL, My SQL, MongoDB.

Результаты сравнительного анализа представлены в таблице 2.

В ходе анализы были выделены характерные особенности рассматриваемых СУБД.

Oracle Database

Преимущества:

* Самые свежие инновации и впечатляющий функционал уже внедрены в этом продукте, поскольку компания Oracle стремится держать планку даже на фоне других разработчиков СУБД.
* Oracle является крайне надёжной, фактически это эталон надёжности среди подобных систем.

Недостатки:

* Стоимость Oracle может оказаться непомерно высокой, особенно для небольших организаций.
* Система может потребовать значительных ресурсов уже сразу после установки, поэтому возможно потребуется модернизировать оборудование для внедрения Oracle.

SQL Server

Преимущества:

* Продукт очень прост в использовании
* Текущая версия работает быстро и стабильно
* Движок предоставляет возможность регулировать и отслеживать уровни производительности, которые помогают снизить использование ресурсов.
* Можно получить доступ к визуализации на мобильных устройствах.
* SQL Server очень хорошо взаимодействует с другими продуктами Microsoft.

Недостатки:

* Цена для юридических лиц оказывается неприемлемой для большей части организаций
* Даже при тщательной настройке производительности SQL Server способен задействовать все доступные ресурсы
* Проблемы с использованием службы интеграции для импорта файлов

PostgreSQL

Преимущества:

* Является масштабируемым решением и позволяет обрабатывать терабайты данных.
* Существует множество предопределенных.
* PostgreSQL предоставляет несколько клиентских интерфейсов для доступа к данным

Недостатки:

* Документация непонятная, поэтому, возможно, ответы на некоторые вопросы придется искать в интернете.
* Конфигурация может смутить неподготовленного пользователя.
* Скорость работы может падать во время проведения пакетных операций или выполнения запросов чтения.

MongoDB

Преимущества:

* Скорость и простота в использовании
* Движок поддерживает json и другие традиционные документы NoSQL.
* Данные любой структуры могут быть сохранены/прочитаны быстро и легко.

Недостатки:

* SQL не используется в качестве языка запросов.
* Инструменты для перевода SQL-запросов в MongoDB доступны, но их следует рассматривать именно как дополнение.
* Программа установки может занять много времен

Сравнительный анализ СУБД

Для проекта была выбрана СУБД SQL Server, так как

1. SQL Server предоставляет широкие возможности для аналитики и создания отчетов.

2. SQL Server обеспечивает поддержку транзакций для целостности данных.

3. SQL Server легко использовать благодаря интуитивному интерфейсу и хорошей документации.

Все перечисленные факторы определили выбор данной СУБД в качестве среды для проектирования баз данных.

# **4. Даталогическое проектирование**

* 1. **Даталогическая модель**

При помощи программы ERwin Data Modeler была создана даталогическая модель БД на логическом и физическом уровне. В данной модели содержится информация обо всех ее объектах и атрибутах, типах данных атрибутов, связях.

* 1. **Нормализация таблиц**

База данных считается нормализованной, если ее таблицы (по крайней мере, большинство таблиц) представлены как минимум в третьей нормальной форме.

Все построенные таблицы были проанализированы и находятся в третьей нормальной форме, так как каждая таблица имеет первичный ключ, а неключевые атрибуты зависят только от первичного ключа или внешних ключей. Транзитивных зависимостей нет.

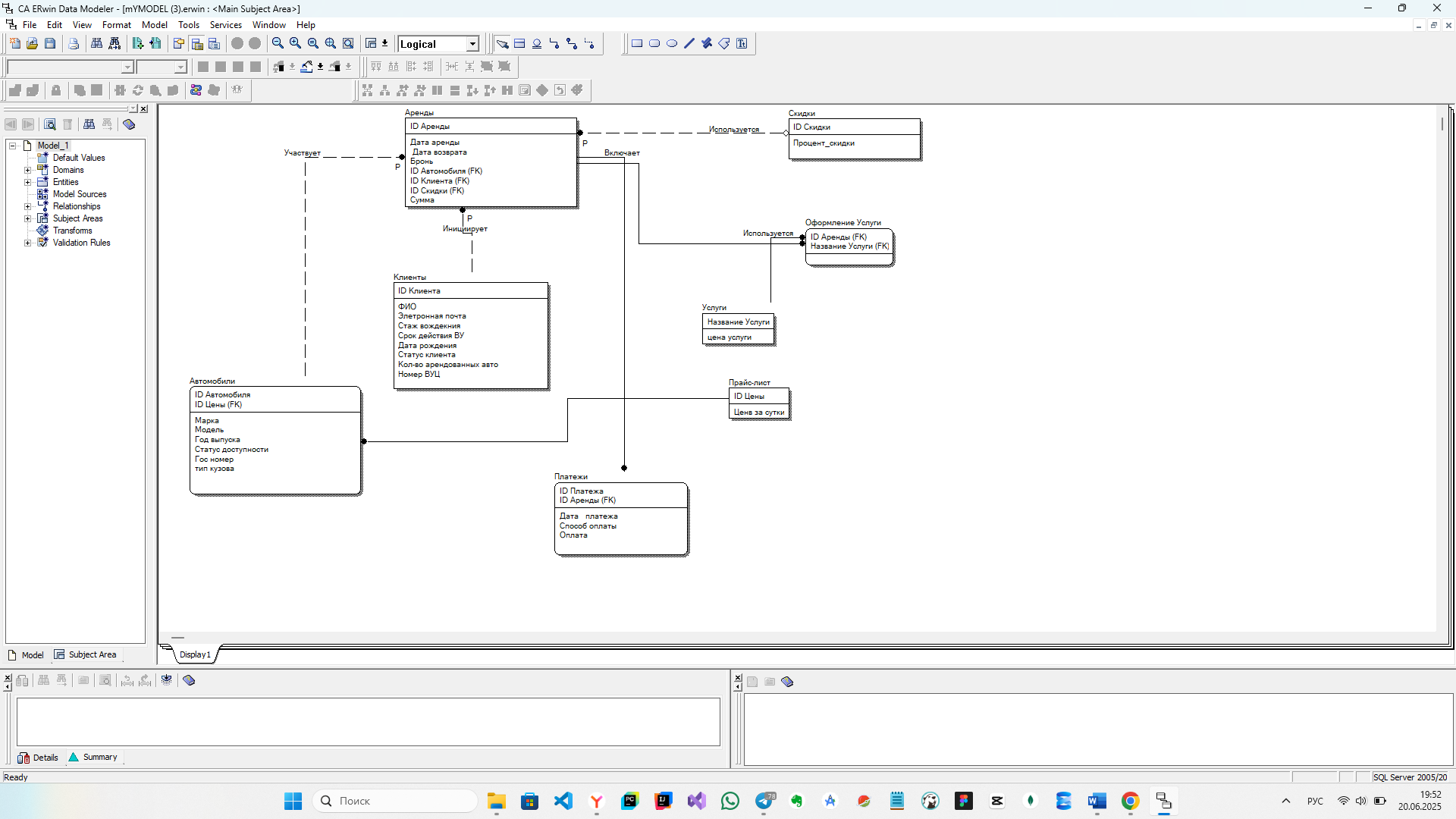


Рис. 3. ERwin модель БД на логическом уровне

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рис. 4. ERwin модель БД на физическом уровне

После нормализации отношений было произведено определение доменов (типов) данных, хранящихся в столбцах таблиц. Параллельно с заданием типа необходимо сформулировать ограничения целостности, связанные с типом, — перечень допустимых значений типа.

Исходя из особенностей данных и их функционального назначения, требуется задать способ представления и границы возможных изменений для каждого из столбцов таблиц. При этом необходимо ответить на вопрос: данные каких типов должны храниться в столбцах и какова их максимальная длина (например, если в столбце предполагается хранить процентные значения, то достаточно будет целого типа данных длиной 1 байт, так как диапазон возможных значений — от 0 до 255; если для данных столбца выбирается тип «строка символов», то желательно указать максимальный размер данных столбца и т. п.).

Далее представлены таблицы базы данных «ПУНКТ ПРОКАТА АВТОМОБИЛЕЙ» с типами данных столбцов и предлагаемыми ограничениями целостности (Таблица 3-8).

Таблица 2

«Автомобили»

|  |  |  |
| --- | --- | --- |
| Имя столбца | Тип данных | Описание |
| id | int | Примарный ключ, ID автомобиля |
| марка | varchar(100) | Марка автомобиля |
| модель | varchar(100) | Модель автомобиля |
| стоимость\_за\_день | decimal(10, 2) | Цена аренды за день |
| статус | varchar(50) | Статус (доступен/занят) |

Таблица 3

«Клиенты»

|  |  |  |
| --- | --- | --- |
| Имя столбца | Тип данных | Описание |
| id | int | Примарный ключ, ID клиента |
| паспорт\_id | varchar(20) | Паспортные данные |
| имя | varchar(255) | Имя клиента |
| телефон | varchar(20) | Номер телефона |
| электронная\_почта | varchar(255) | Почта клиента |
| скидка | decimal(5, 2) | Процент скидки |

Таблица 4

«Заказы»

|  |  |  |
| --- | --- | --- |
| Имя столбца | Тип данных | Описание |
| id | int | Примарный ключ, ID заказа |
| клиент\_id | int | Внешний ключ, идентификатор клиента |
| автомобиль\_id | int | Внешний ключ, ID автомобиля |
| дата\_начала | date | Дата начала аренды |
| дата\_окончания | date | Дата окончания аренды |
| базовая\_стоимость | decimal(10, 2) | Предварительная стоимость |
| стоимость\_со\_скидкой | decimal(10, 2) | Сумма со скидкой |
| Имя столбца | Тип данных | Описание |
| итоговая\_стоимость | decimal(10, 2) | Итоговая стоимость заказа |

Таблица 5

«Дополнительные услуги»

|  |  |  |
| --- | --- | --- |
| Имя столбца | Тип данных | Описание |
| id | int | Примарный ключ, ID услуги |
| название | varchar(255) | Название услуги |
| описание | text | Описание услуги |
| стоимость | decimal(10, 2) | Стоимость услуги |

Таблица 6

«Скидки»

|  |  |  |
| --- | --- | --- |
| Имя столбца | Тип данных | Описание |
| id | int | Примарный ключ, ID скидки |
| клиент\_id | int | Внешний ключ, ID клиента |
| процент\_скидки | decimal(5, 2) | Процент предоставляемой скидки |

Таблица 7

«История заказов»

|  |  |  |
| --- | --- | --- |
| Имя столбца | Тип данных | Описание |
| id | int | Примарный ключ, ID записи истории |
| клиент\_id | int | Внешний ключ, ID клиента |
| автомобиль\_id | int | Внешний ключ, ID автомобиля |
| дата\_аренды | date | Дата начала аренды |
| дата\_возврата | date | Дата возврата автомобиля |
| итоговая\_стоимость | decimal(10, 2) | Итоговая стоимость заказа |

Таблица 8

«Заказы\_Услуги»

|  |  |  |
| --- | --- | --- |
| Имя столбца | Тип данных | Описание |
| заказ\_id | int | Внешний ключ, ID заказа |
| услуга\_id | int | Внешний ключ, ID услуги |
| стоимость\_услуги | decimal(10, 2) | Стоимость предоставленной услуги |

**5. Физическое проектирование**

**5.1. Таблицы**

Ниже приведены изображения с физической структурой основных таблиц базы данных, содержащие их поля, типы данных и ключевые ограничения.

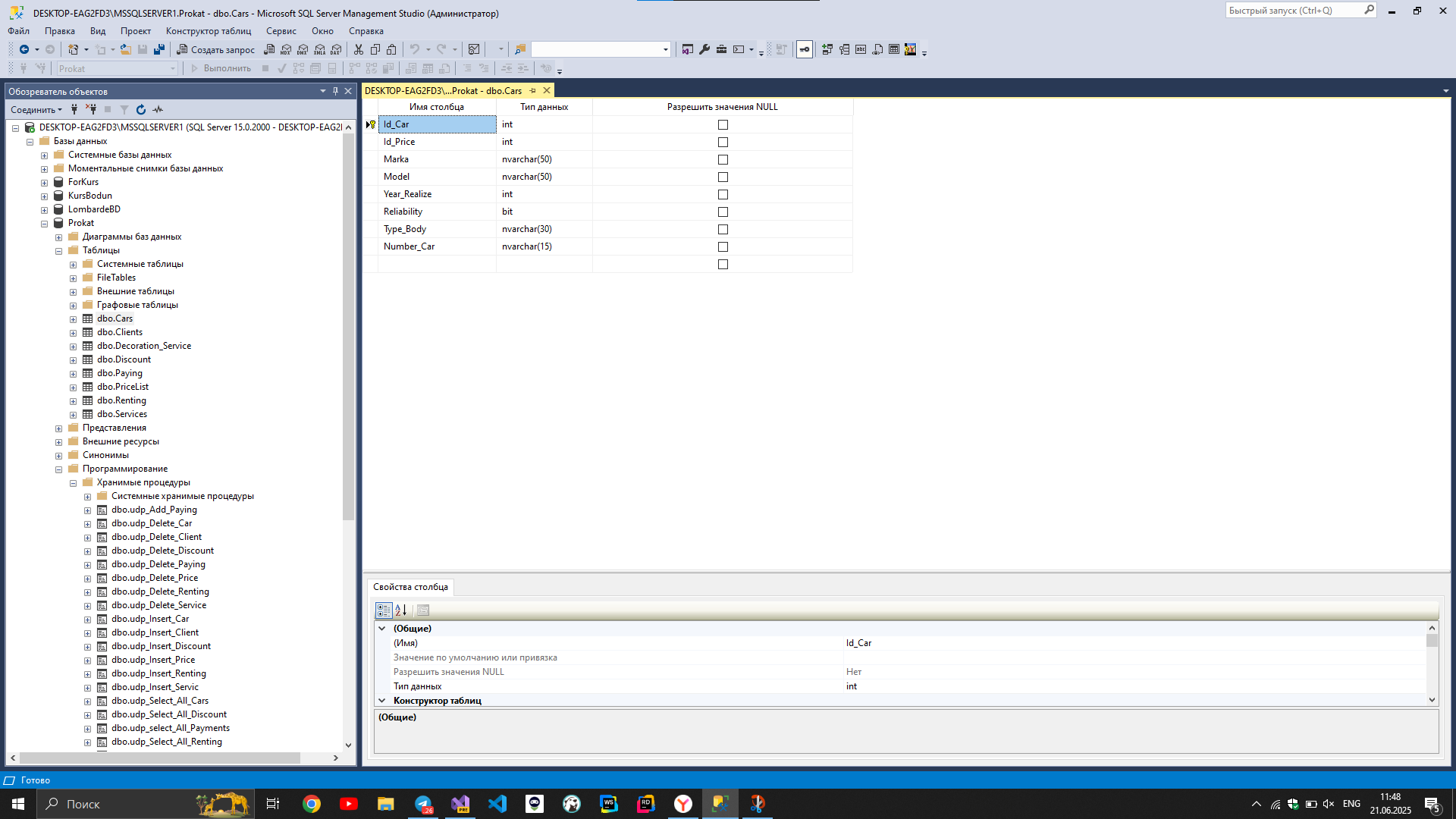


Рис. 5. Таблица «Cars»

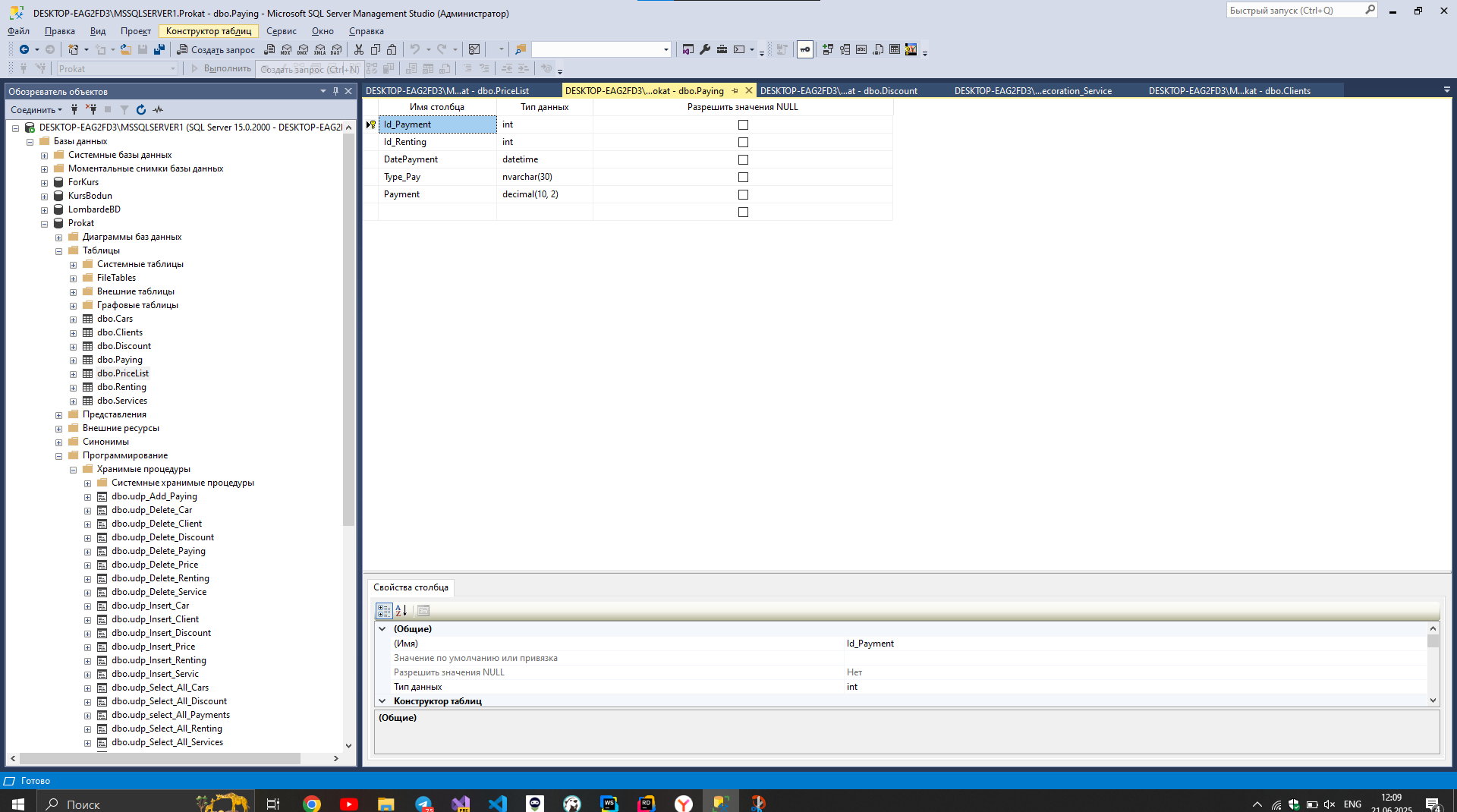


Рис. 6. Таблица «Payments»

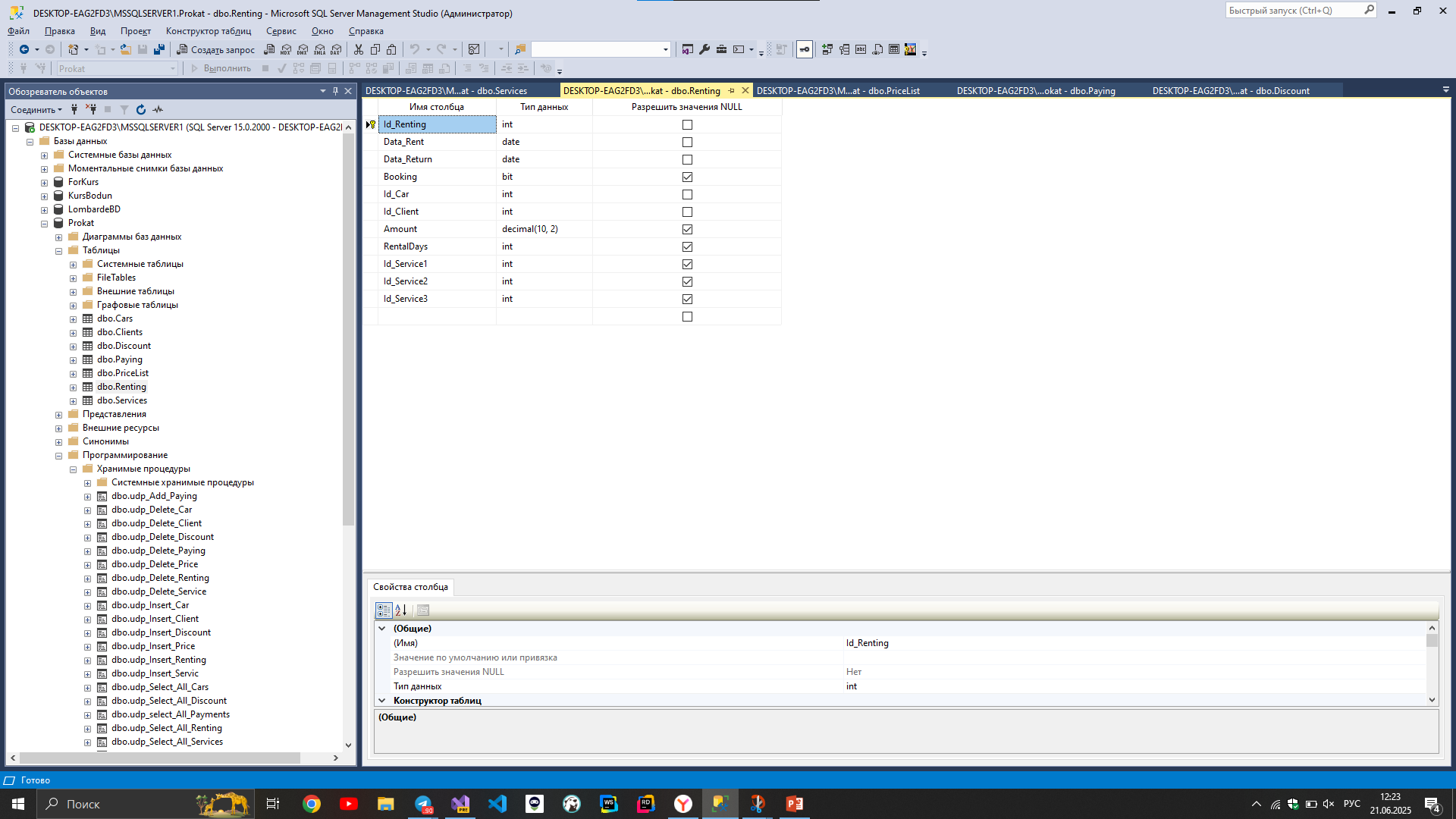


Рис. 7. Таблица «Renting»

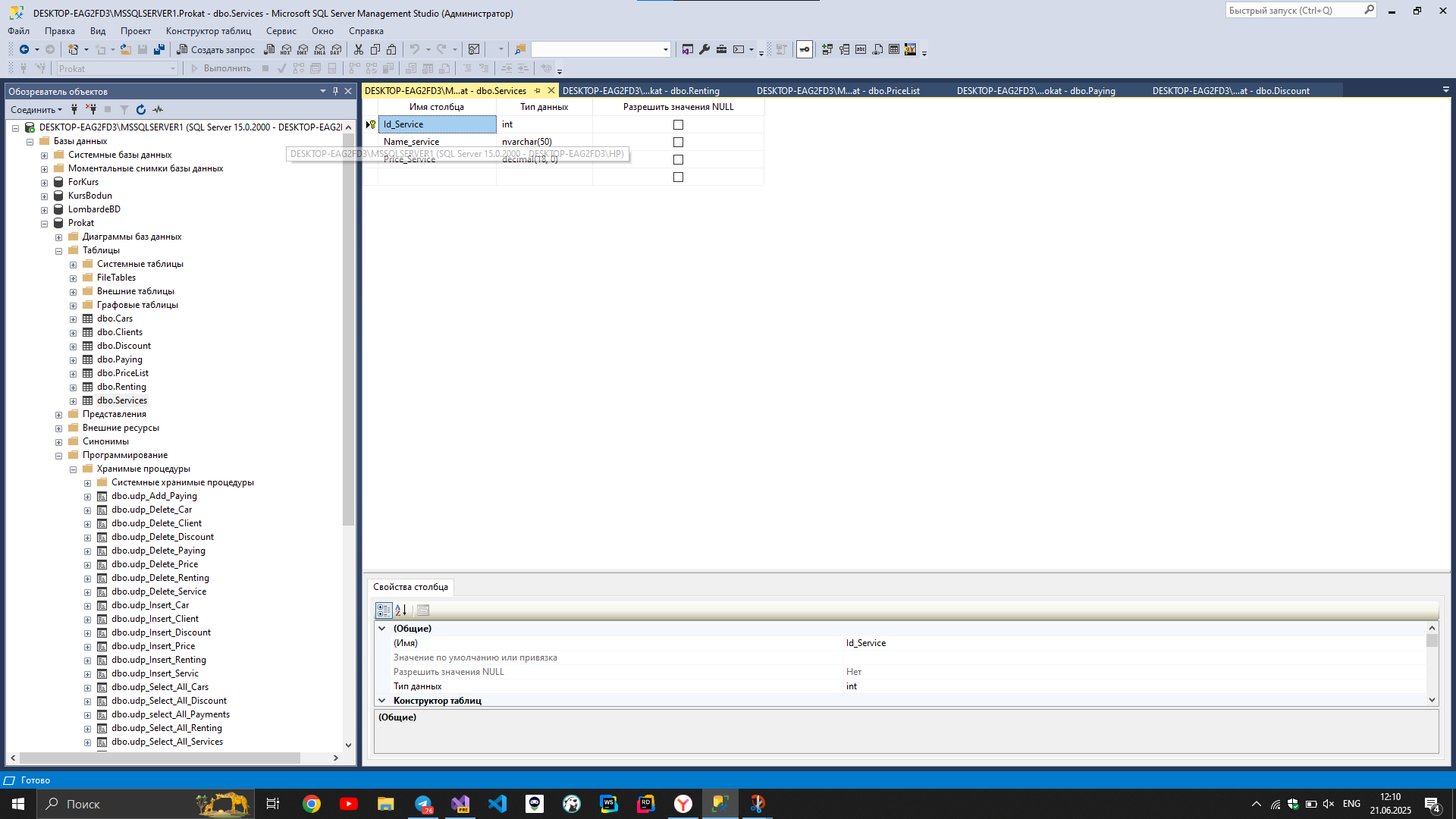


Рис. 8. Таблица «Service»

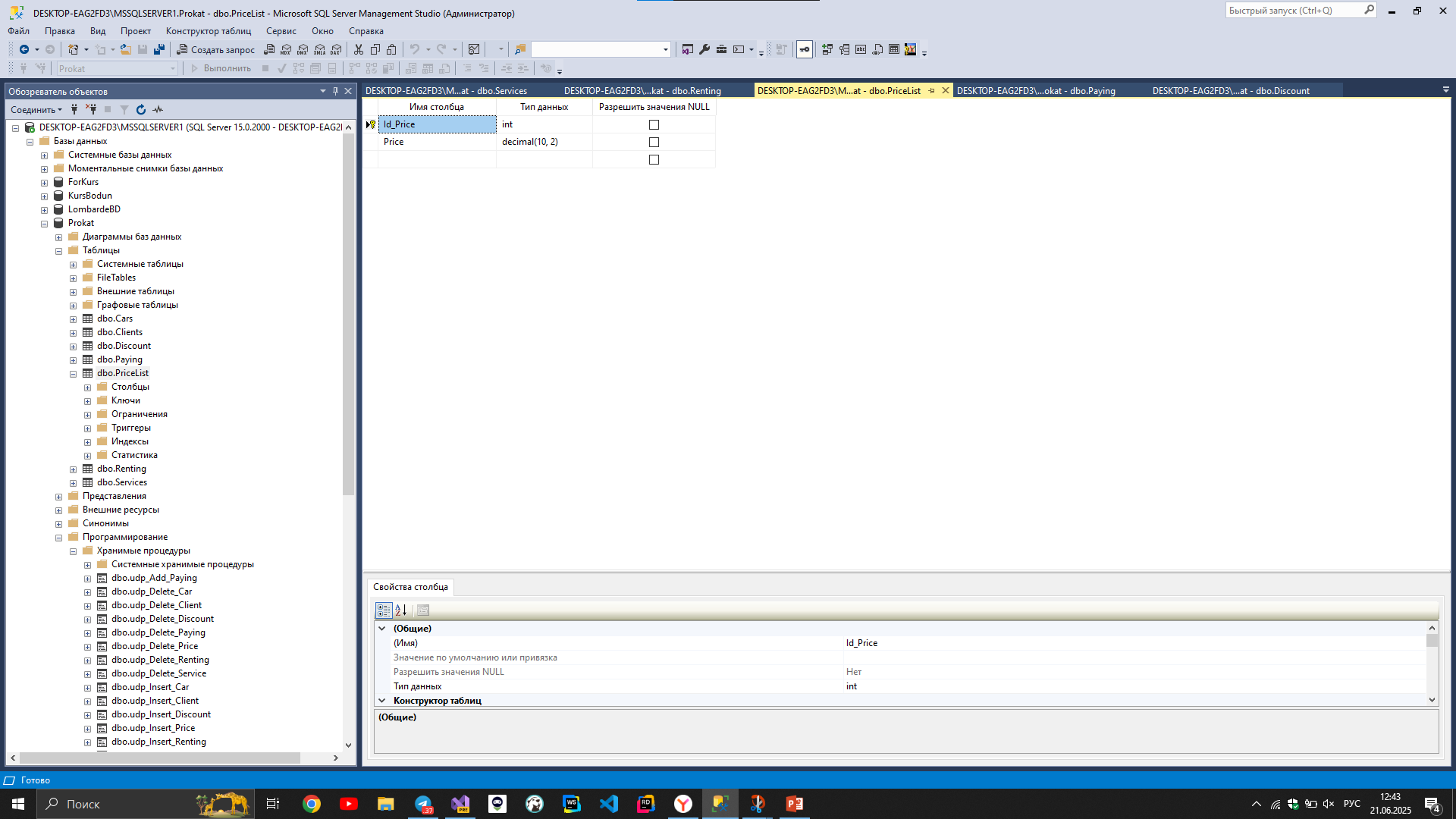


Рис. 9. Таблица «Pricelist»

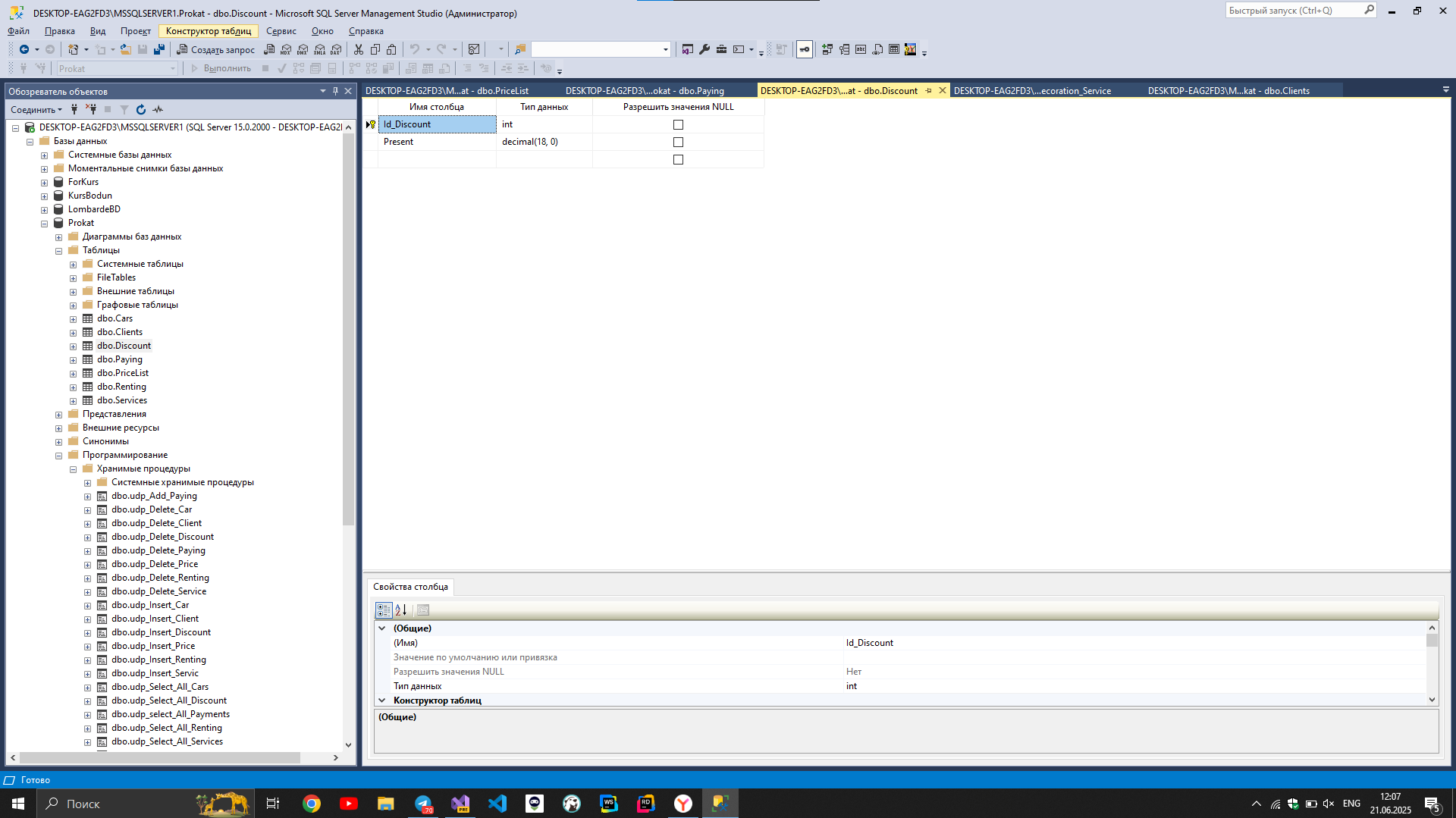


Рис. 10. Таблица «Discount»

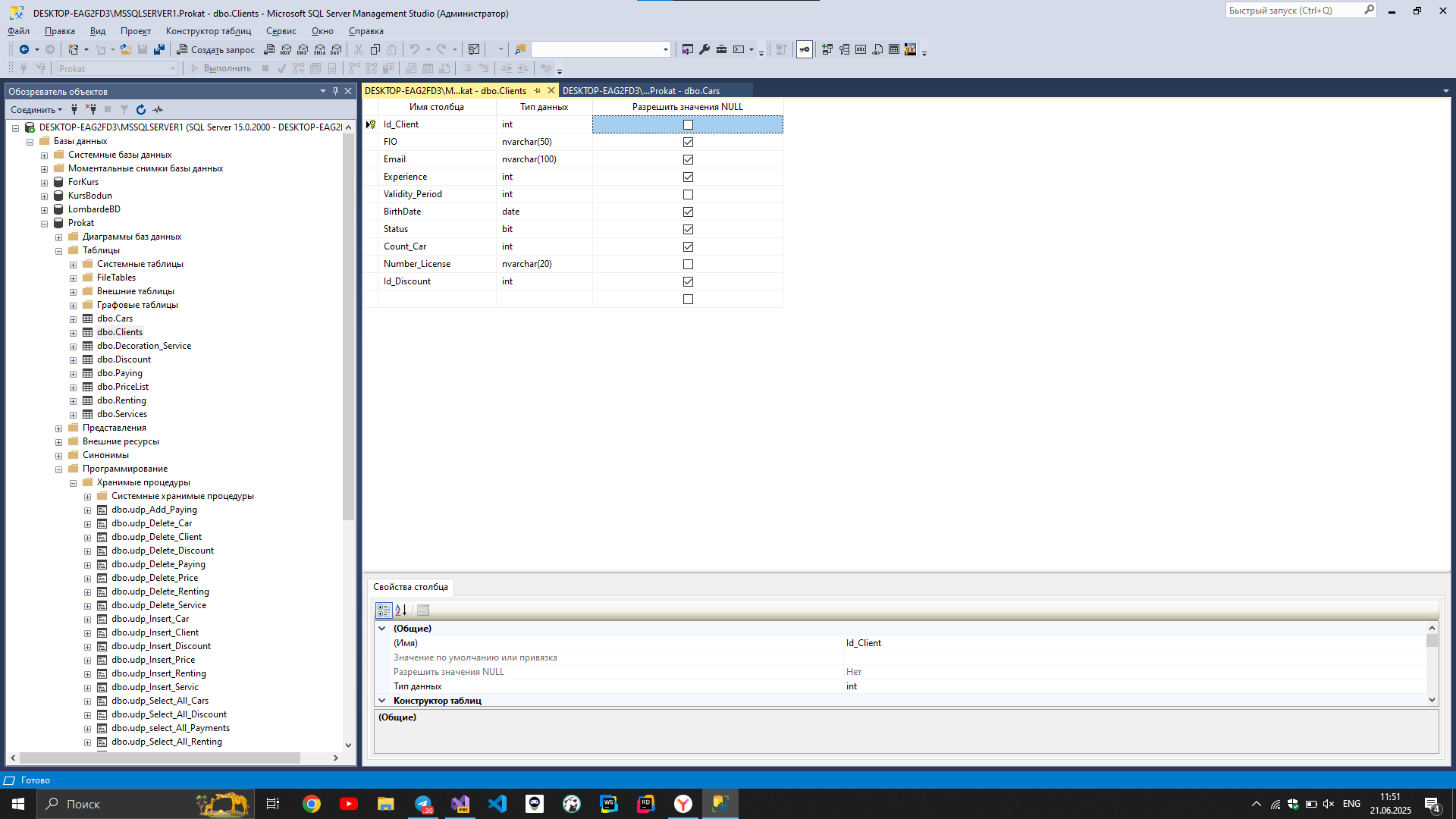


Рис. 11. Таблица «Client»

**5.2 Хранимые процедуры**

Хранимые процедуры SQL представляют собой исполняемый программный модуль, который может храниться в базе данных в виде различных объектов. Другими словами, это объект, в котором содержатся SQL-инструкции. Эти хранимые процедуры могут быть выполнены в клиенте прикладных программ, чтобы получить хорошую производительность.

При вызове процедуры она моментально обрабатывается самим сервером без лишних процессов и вмешательства пользователя. После этого можно осуществлять любые действия с информацией: удаление, исполнение, изменение. За все это отвечает DDL-оператор, который в одиночку совершает сложнейшие действия по обработке объектов. Причём все это происходит очень быстро, а сервер фактически не нагружается. Такая скорость и производительность позволяют очень быстро передавать большие объёмы информации от пользователя на сервер и наоборот

В ходе выполнения курсовой работы было разработано множество процедур (см. рис. 12 - 15)

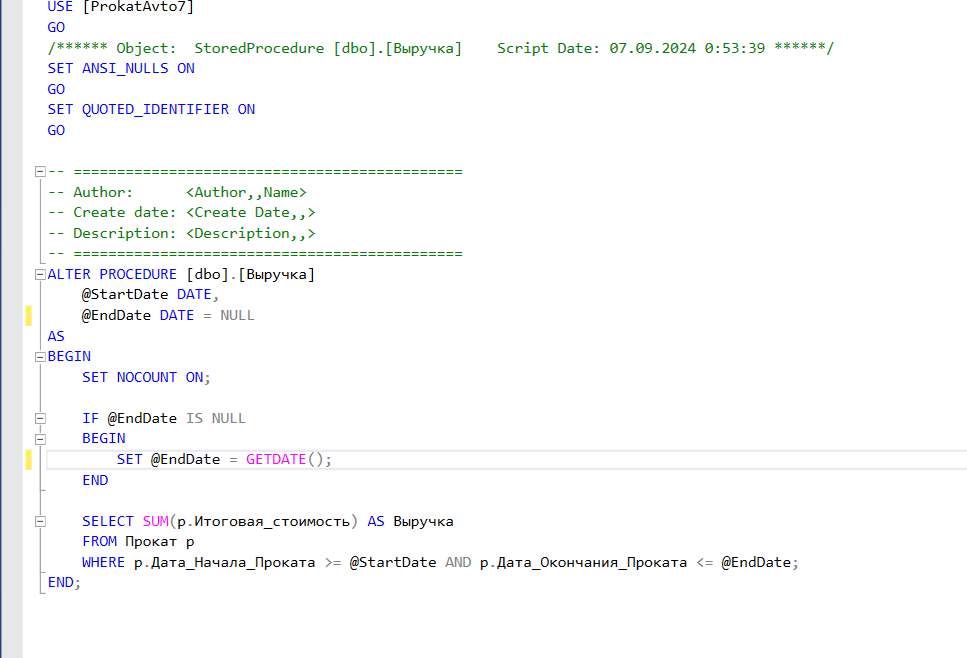


Рис. 12. Процедура «Выручка»

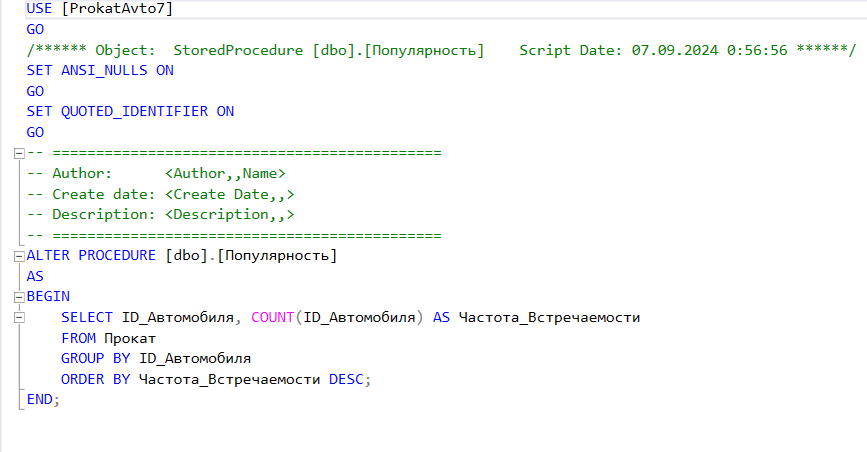


Рис. 13. Процедура «Популярность»

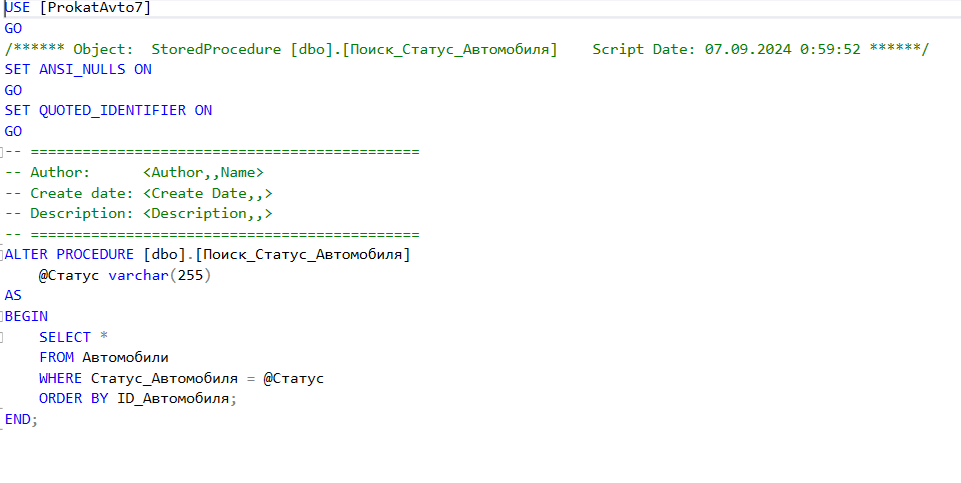


Рис. 14. Процедура «Поиск\_статус\_авотмобиля»

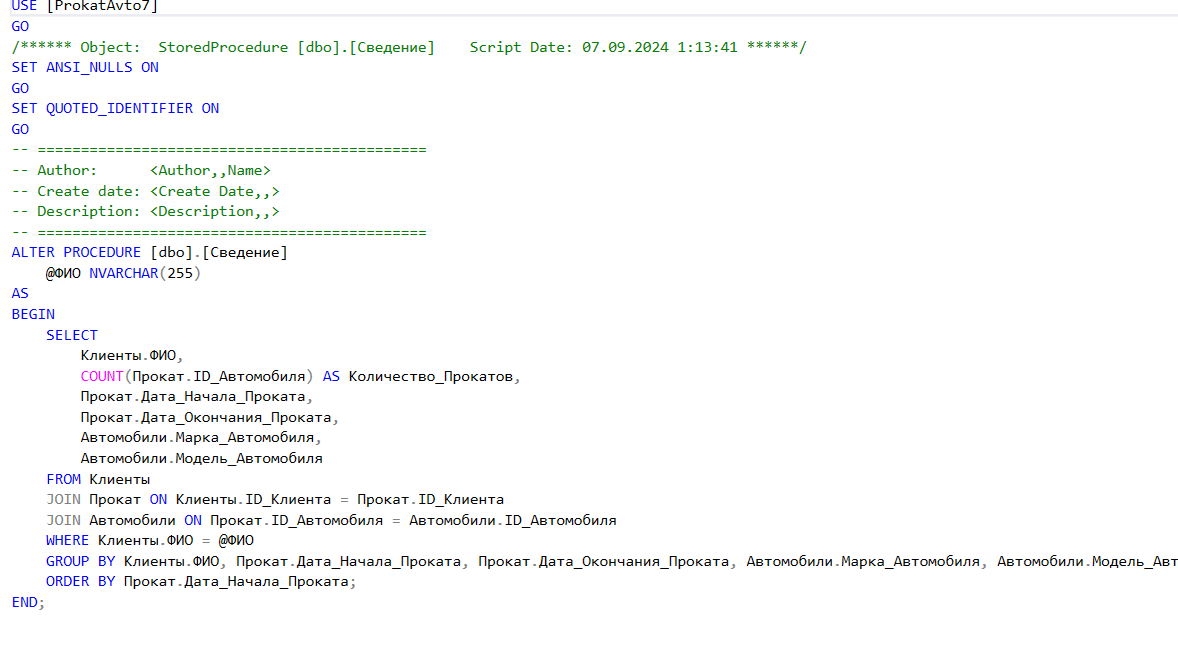


Рис. 15. Процедура «Сведения»

# **Заключение**

В данной курсовой работе была разработана информационная система «Пункт проката автомобилей», направленная на автоматизацию и упрощение ключевых процессов управления прокатом автомобилей. В ходе работы был проведён тщательный анализ поставленной задачи, что позволило чётко определить требования и функциональные возможности будущей системы.

Для реализации проекта были использованы современные программные средства: ERWin для моделирования базы данных, SQL Server 2022 в качестве системы управления базами данных, а также редактор кода Microsoft Visual Studio для разработки пользовательского интерфейса и логики приложения. Такой комплексный подход обеспечил создание эффективного и надёжного решения.

Основные функции разработанного приложения включают приём и учёт исполнения проката автомобилей, бронирование авто, оформление бланков счёта, а также удобный просмотр и анализ выручки. Это позволило значительно упростить работу сотрудников пункта проката, повысить скорость обработки данных и минимизировать ошибки, связанные с ручным вводом информации.

В процессе изучения и выбора технологий был проведён обзор существующих систем управления базами данных, что помогло обосновать выбор именно SQL Server 2022 как оптимального инструмента для данного проекта. Это решение обеспечило надёжность, масштабируемость и удобство поддержки базы данных.

Разработанное приложение обладает интуитивно понятным интерфейсом, что делает его доступным для пользователей с разным уровнем компьютерной грамотности. Благодаря этому сотрудники смогут быстро освоить работу с системой и эффективно использовать её возможности в повседневной деятельности.

Полученный опыт выполнения курсовой работы не только позволил создать полноценный прототип базы данных для пункта проката автомобилей, но и сформировал навыки анализа, проектирования и реализации информационных систем. Эти знания и умения будут полезны при разработке и сопровождении баз данных в любой организации, что значительно расширяет профессиональный потенциал автора.

Таким образом, данный проект является успешным примером комплексного подхода к автоматизации бизнес-процессов на основе современных технологий и может служить основой для дальнейшего развития и внедрения подобных решений в реальных условиях.