МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра «Информационно-вычислительные системы»

Курсовой проект

по дисциплине «Автоматизированные систем специального назначения»

на тему «Разработка базы данных для автоматизированной системы со сведениями о поставке газовых баллонов для автомобилей»

ПГУ 1.090501.9

Специальность 09.05.01 «Применение и эксплуатация автоматизированных систем специального назначения»

Специализация №1 «Эксплуатация вычислительных машин комплексов, систем и сетей специального назначения»

Выполнил: студент гр. 15ВО2

_____Новиков В.В.

Руководитель: к.т.н. доцент кафедры ИВС

_____Чувыкин Б.В.

Работа защищена с оценкой Удовлетворительно

Преподаватели Чувыкин Б.В.

Данова Н.А.

Дата защиты 26.05.2019г.

2019

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра Информационно-вычислительные системы

"УТВЕРЖДАЮ"
Зав. кафедрой
"13" 02 2019 г.

З А Д А Н И Е
на курсовое проектирование по курсу

Автоматизированные системы специального назначения

Студенту Новикову Виктору Викторовичу Группа 15ВО2

Тема проекта Разработка базы данных для автоматизированной системы по учету сведений о замене газовых баллонов для автомобилей

Разработать базу данных для СУБД Линтер.

В базе данных должна храниться следующая информация: номер машины, переодичность замены, количество баллонов, имя клиента, фамилия клиента, отчество клиента, адрес клиента, дата замены, количество заменяемых баллонов , фамилия экспедитора, имя экспедитора, отчество экспедитора.

Разработку базы данных осуществлять для СУБД Линтер с использованием CASE-средства OpenModelSphere.

Литература:

1. Осипов, Д.Л. InterBase и Delphi. Клиент-серверные базы данных [Электронный ресурс] — Электрон. дан. — Москва : ДМК Пресс, 2015. — 536 с.
2. Макарова Н.В. Информатика. Базовый курс [Текст]: учеб. пособие для бакалавров / Н. В. Макарова, В. Б. Волков. - СПб.: Питер, 2012. – 576 с. (49 экз.)
3. СУБД ЛИНТЕР. Официальный сайт. Документация. — Режим доступа: http://www.linter.ru/ru/documentation/
4. Руководящий документ. Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации». – Режим доступа: http://fstec.ru/component/content/article/114-deyatelnost/tekushchaya/tekhnicheskaya-zashchita-informatsii/normativnye-i-metodicheskie-dokumenty/spetsialnye-normativnye-dokumenty/384-rukovodyashchij-dokument-reshenie-predsedatelya-gosekhkomissii-rossii-ot-30-marta-1992-g

Объем работы по курсу

1. Расчетная часть

1.1.Разработка физической модели БД с использованием CASE-средства OpenModelSphere.

1.2.Создание таблиц БД для СУБД Линтер.

1.3.Создание ограничений целостности.

1.4.Разработка хранимых процедур и триггеров.

1.5.Разработка SQL-запросов.

1.6.Написание пояснительной записки.

2. Графическая часть

Графическая часть не предусмотрена

3. Экспериментальная часть

3.1. Создание демонстрационной БД для СУБД ЛИНТЕР.

3.2. Тестирование БД системы.

Срок выполнения проекта по разделам

1. П.1.1	к	*27.02.19г.*
2.П.1.2.	к	*06.03.19г.*
3.П.1.3.	к	*13.03.19г.*
4.П.1.4.	к	*20.03.19г.*
5.П.1.5.	к	*27.03.19г.*
6.П.2.1.	к	*17.04.19г.*
7.П.3.1.	к	*24.04.19г.*
8.П.3.2.	к	*08.05.19г.*
9.П.1.6.	к	*22.05.19г.*

Дата выдачи задания "13" февраля 2019 г.

Дата защиты проекта "22" мая 2019 г.

Руководитель к. т. н., доц. *Долгова И.А.*

Задание получил "13" февраля 2019 г.

Студент

Реферат

Пояснительная записка 115 листов, 144 рисунка, 16 таблиц, 4 источника, 1 приложение.

Объектом исследования является процесс поставки газовых баллонов для автомобилей.

Цель работы – разработать базу данных для автоматизированной системы со сведениями о поставке газовых баллонов для автомобилей.

В результате проделанной работы разработана база данных со сведениями о поставке газовых баллонов для автомобилей, с помощью которого можно вести учет информации о клиентах, автомобилях, мастерах и о замене.

База данных спроектирована с помощью Case-средства Open Modal Sphere 3.2. База данных разработана для СУБД Линтер «Бастион».

Содержание

Введение.....5
1 Техническое задание.....7
1.1 Основание для разработки.....7
1.2 Назначение разработки.....7
1.3 Требования к функциональным характеристикам.....7
1.4 Требования к составу и параметрам технических средств.....7
1.5 Требования к программной совместимости.....8
1.6 Требования к программной документации.....9
1.7 Стадии и этапы разработки.....10
1.8 Порядок контроля и приёмки.....11
2 Концептуальное проектирование базы данных.....12
2.1 Разработка модели предметной области.....12
2.2 Разработка контекстной диаграммы.....15
2.3 Разработка детализирующей диаграммы.....15
2.4 Разработка концептуальной модели.....17
3 Логическое проектирование базы данных.....20
4 Физическое проектирование базы данных.....25
5 Создание базы данных.....44
6 Создание таблиц базы данных.....51
7 Создание ограничений целостности.....57
8 Создание хранимых процедур и триггеров.....61
9 Разработка SQL-запросов.....71
10 Выборка из связанных таблиц.....81
11 Резервное копирование и восстановление базы данных.....96
12 Тестирование базы данных.....101
Заключение.....109
Список использованных источников.....110
Приложение А.....111

Введение

ЛИНТЕР - система управления базами данных (СУБД), полностью разработанная российской компанией РЕЛЭКС, прошедшая сертификацию в Министерстве обороны РФ, ФСТЭК России и гарантирующая надежную защиту информации от уровня «персональные данные» до уровня «совершенно секретно»./1/

СУБД ЛИНТЕР доверяют управление атомными реакторами, добычу нефти и полезных ископаемых, безопасность воздушного движения, управление бизнес-процессами, работу станков, бытовой техники, мобильных и других приложений. Она не раз отмечалась на отраслевых выставках и конференциях, в рейтингах различных журналов, освещающих сферу ИКТ, признавалась лауреатом, продуктом года и лучшим программным продуктом в своей категории, а самое главное получила множество положительных отзывов от пользователей.

ЛИНТЕР ценят за: беспрецедентную защиту данных (среди систем подобного класса), высочайшую отказоустойчивость, в том числе в случаях сбоев оборудования,поддержку большинства современных операционных систем, возможность работы в условиях жестких ресурсных ограничений и в критичных приложениях,поддержку функций реального времени, простоту встраиваемости в программное обеспечение, широкий набор утилит и интерфейсов, профессионализм и оперативность службы технической поддержки/1/

Семейство СУБД ЛИНТЕР включает в себя две параллельно развивающиеся версии, каждая из которых ориентирована на выполнение определённых функций:

- обеспечение высочайшего уровня защиты данных –ЛИНТЕР БАСТИОН

- решение основных задач, стоящих перед современной СУБД –ЛИНТЕР СТАНДАРТ

Отдельного внимания заслуживает версия СУБД ЛИНТЕР БАСТИОН, сертифицированная ФСТЭК России и Министерством обороны РФ по 2 и 3

классу защищённости от несанкционированного доступа (НСД) к информации соответственно и по 2 уровню контроля отсутствия недекларированных возможностей (НДВ)./2/

ЛИНТЕР БАСТИОН обладает уникальным комплексом средств защиты, которые позволяют использовать СУБД в информационных системах персональных данных (ИСПДн) до 1 класса включительно и обеспечивают выполнение требований Федерального закона № 152-ФЗ, что подтверждено ФСТЭК России./3/

В данной работе описывается база данных о сведениями о замене газовых баллонов на автомобилях. При разработке приложения использовалась СУБД Линтер «Бастион». База данных спроектирована с помощью CASE — средства Open ModelSphere 3.2. База данных разработана для СУБД ЛИНТЕР «БАСТИОН»

1 Техническое задание

1.1 Основание для разработки

База данных должна быть разработана на основании задания на курсовое проектирование, выданного к.т.н. доцентом кафедры ИВС Долговой И.А. и утвержденного заведующей кафедры ИВС к.т.н., доцентом Бобрышевой Г.В.

1.2 Назначение разработки

Разрабатываемая база данных должна быть предназначена для учета информации о клиентах, об автомобилях, о замене и экспедиторах.

1.3 Требования к функциональным характеристикам

База данных должна выполнять следующие функции:

- вести учет информации о клиентах;
- вести учет информации об автомобилях;
- вести учет информации о замене;
- вести учет информации об экспедиторах.

Входными данными должна быть информация, вводимая пользователем.

Выходными данными должна быть информация, выводимая из базы данных по запросам пользователя.

1.4 Требования к составу и параметрам технических средств

База данных должна быть предназначена для использования как на локальном компьюторе, так и в сети. Локальный компьютер или рабочая станция в сети должны иметь следующие технические характеристики:

- тактовая частота процессора - не менее 2,1 МГц;

- оперативная память – не менее 4096 Мбайт;

- на жестком диске при установке используется около 250 Мбайт;

- объем жестокого диска должен быть не менее 1024 Мбайт.

Локальный компьютер или сервер в сети должны иметь следующие технические характеристики:

- тактовая частота процессора - не менее 2,1 МГц;

- оперативная память – не менее 4096 Мбайт;

- на жестком диске при установке используется около 250 Мбайт;

- объем жестокого диска должен быть не менее 1024 Мбайт

1.5 Требования к программной совместимости

База данных должна быть предназначена для использования как на локальном компьютере, так и в сети.

Если программа будет использоваться на локальном компьютере, то на нем должен быть установлен СУБД Линтер , размещен файл с базой данных. База данных будет использоваться на рабочей станции сети, то на сервере баз данных должна быть установлена СУБД Линтер и размещен файл с базой данных. На рабочей станции в сети должен быть размещен файл содержащий клиент-серверное приложение.

Разработанная база данных должна работать под управлением операционной системы Windows 10.

1.6 Требования к программной документации

Программная документация к разрабатываемому приложению должна содержать следующие разделы:

– техническое задание;

– описание проектирования базы данных;

– описание проектирования приложения;

– описание программы;

– программа и методика испытаний;

– описание применения;

– текст программы.

1.7 Стадии и этапы разработки

Стадии и этапы разработки проекта указаны в таблице 1.

Таблица 1 - Стадии выполнения проекта

№ п/п	Стадии разработки	Этап работ	Срок исполнения	Исполнитель
1	Расчётная часть	1.1 Разработка физической модели БД с использованием CASE-средства OpenModelSphere.	27.02.19г.	Новиков В.В.
		1.2 Создание таблиц БД для СУБД Линтер.	06.03.19г.	Новиков В.В.
		1.3 Создание ограничений целостности.	13.03.19г.	Новиков В.В.
		1.4 Разработка хранимых процедур и триггеров.	20.03.19г.	Новиков В.В.
		1.5 Разработка SQL-запросов.	27.03.19г.	Новиков В.В.
		1.6 Написание пояснительной записки.	22.05.19г.	Новиков В.В.
2	Графическая часть	Графическая часть не предусмотрена.	17.04.19г.	Новиков В.В.
3	Экспериментальная часть	3.1 Создание демонстрационной БД для СУБД Линтер.	24.04.19г.	Новиков В.В.
		3.2 Тестирование БД системы.	08.05.19г.	Новиков В.В.

1.8 Порядок контроля и приемки

Для проверки выполнения базы данных всех её функций в соответствие с техническим заданием следует разработать тестовые примеры и протестировать реакцию программы на возможные действия пользователя: запуск базы данных, заполнение полей ввода информации, выполнение хранимых процедур и SQL запросов.

База данных соответствует требованиям технического задания, если она выполняет все заданные функции при различных наборах входных данных из тестовых примеров.

Приёмка базы данных должна осуществляться при выполнение всех ее функций, а также при наличии полной документации на базу данных (пояснительная записка с приложениями: текстом сценария базы данных).

2 Проектирование базы данных

2.1 Разработка модели предметной области

Предметной областью для данной работы является учет сведений о ремонтной мастерской, в которой производится замена газовых баллонов на автомобилях. Разрабатываемое приложение поможет автоматизировать работу конторы.

Современная фирма по замене газовых баллонов — это хранилище огромного количества информации о замене газовых баллонов, клиентах, экспедиторах и автомобилях. Для упрощения работы поиска информации о клиентах, автомобилях, экспедиторах и замене газовых баллонов, создается база данных. Основная цель проектирования базы данных — это сокращение избыточности хранимых данных, а следовательно, экономия объема используемой памяти. Клиент может иметь несколько автомобилей, в тоже время у каждой машины один владелец.

Экспедитор может ремонтировать несколько автомобилей, в документации которую он заполняет указывается дата замены, количество газовых баллонов, переодичность замены и так же автомобиль. Поскольку каждый автомобиль имеет свой номер, количество газовых баллонов и владельца, а каждый клиент имеет Фамилию, имя, отчество и адрес, то для учета и систематизации этих данных создается множество документации. Данный вид учета усложняет поиск и контроль информации о ремонте автомобилей, но благодаря базам данных этот вид работы становится более быстрым и удобным. База данных позволяет экспедитору сообщить клиенту о завершении работ, администратору быстро оформить запрос на необходимый автомобиль, проконтролировать дату ремонта автомобиля, количество газовых баллонов.

В настоящее время процесс учета газовых баллонов для ремонта автомобилей, частично автоматизирован с помощью пакета Microsoft Office. Данный

вариант упрощает способ хранения информации, так как не придется вести огромную сводку данных. Так же будет уменьшено время для поиска нужного автомобиля. Но при этом есть такие недостатки как, сложность создания системы, контроль за целостностью данных при редактировании или удалении информации, возможны ошибки при добавлении, изменении или удалении информации, при осуществлении поиска, остается не оптимальным по времени. Кроме того, необходимо будет нанимать специалиста для создания системы учета продажи товаров и сотрудника для контроля работоспособности данной системы и исправления ошибок при их возникновении.

Таким образом, актуальной является задача автоматизации системы сведений о наличии товара для продажи, путем разработки автоматизированной системы или использования готовой автоматизированной системы.

Для автоматизации бизнес-процессов в розничной торговле фирма "1С" выпустила прикладное решение "1С:Автосервис 8". Решение позволяет вести учет в нескольких организациях – юридических лицах. Предусмотрено также ведение учета в нескольких автосервисах, которые не являются юридическими лицами. /4/

"1С:Автосервис 8" – отраслевое решение, которое предназначено для автоматизации управленческого учета в небольших автосервисах, автомойках, станциях технического обслуживания автомобилей.

Конфигурация "Автосервис" разработана на основе типовой конфигурации "Управление нашей фирмой", редакции 1.6 системы программ "1С:Предприятие 8" с сохранением всех возможностей и механизмов типового решения, учитывает специфику предприятий авто бизнеса и обеспечивает следующие возможности:

функции CRM-системы: Ведение базы клиентов. Фиксация всех контактов с клиентами: входящие и исходящие звонки, письма, встречи и прочее. Отправка e-mail и SMS сообщений непосредственно из программы.

Автосервис:

- использование специфичных для автосервиса справочников: модели автомобилей, нормо-часы, цеха, виды ремонта;

- предварительная запись на ремонт и планирование загрузки автосервиса;

- формирование необходимых печатных форм документа "Заказ-наряд": акт осмотра, рабочий лист, заказ-наряд, квитанция к заказ-наряду;

- контроль деятельности автосервиса из любой точки земного шара через Интернет, в том числе и с помощью планшетных компьютеров или коммуникаторов.

Недостатками данной системы являются: учет в программе ведется в одной валюте – рублях/4/, дорогая лицензия, первичное обучение персонала для эффективного использования автоматизированной системы, прием на работу специалиста на должность администратора, который будет следить за корректной работой системы, заключение дорогого контракта с фирмой 1С на обслуживание систем "1С:Автосервис 8", также, если у канторы по замене запчастей уже была база данных, в которой велся учет, то для работы в "1С:Автосервис 8" придется потратить дополнительные ресурсы на переработку базы данных.

Таким образом, актуальной является задача процесса учета информации о замене газовых баллонов, решаемая путем разработки автоматизированной системы.

Таким образом, использование базы данных в фирме имеет следующие преимущества:

1) устранение рутинных ручных операций, неизбежных при обработке информации;

2) существенное ускорение процессов обработки и преобразование данных;

3) повышение точности учетных и отчетных данных;

4) высвобождение времени работников для решения творческих задач.

2.2 Разработка контекстной диаграммы

В результате проведенного анализа предметной области построена контекстная диаграмма, приведенная на рисунке 2.1.

Рисунок 2.1 — Контекстная диаграмма

На основе анализа предметной области были выделены внешние объекты «Экспедитор», «Клиент», а также обобщенный процесс «Учет поставок газовых баллонов для автомобилей».

Внешний объект «Экспедитор» связан выходным потоком «Сведения об экспедиторах» и входным потоком «Сведения о клиентах» с обобщенным процессом «Учет поставок газовых баллонов для автомобилей».

Внешний объект «Клиент» связан выходным потоком «Сведения об автомобилях» и входным потоком «Сведения о замене» с обобщенным процессом «Учет поставок газовых баллонов для автомобилей».

Таким образом разработана контекстная диаграмма.

2.3 Разработка детализирующей диаграммы

На основе контекстной диаграммы в результате декомпозиции обобщенного процесса «Учет поставок газовых баллонов для автомобилей» была построена детализирующая диаграмма представленная на рисунке 2.2.

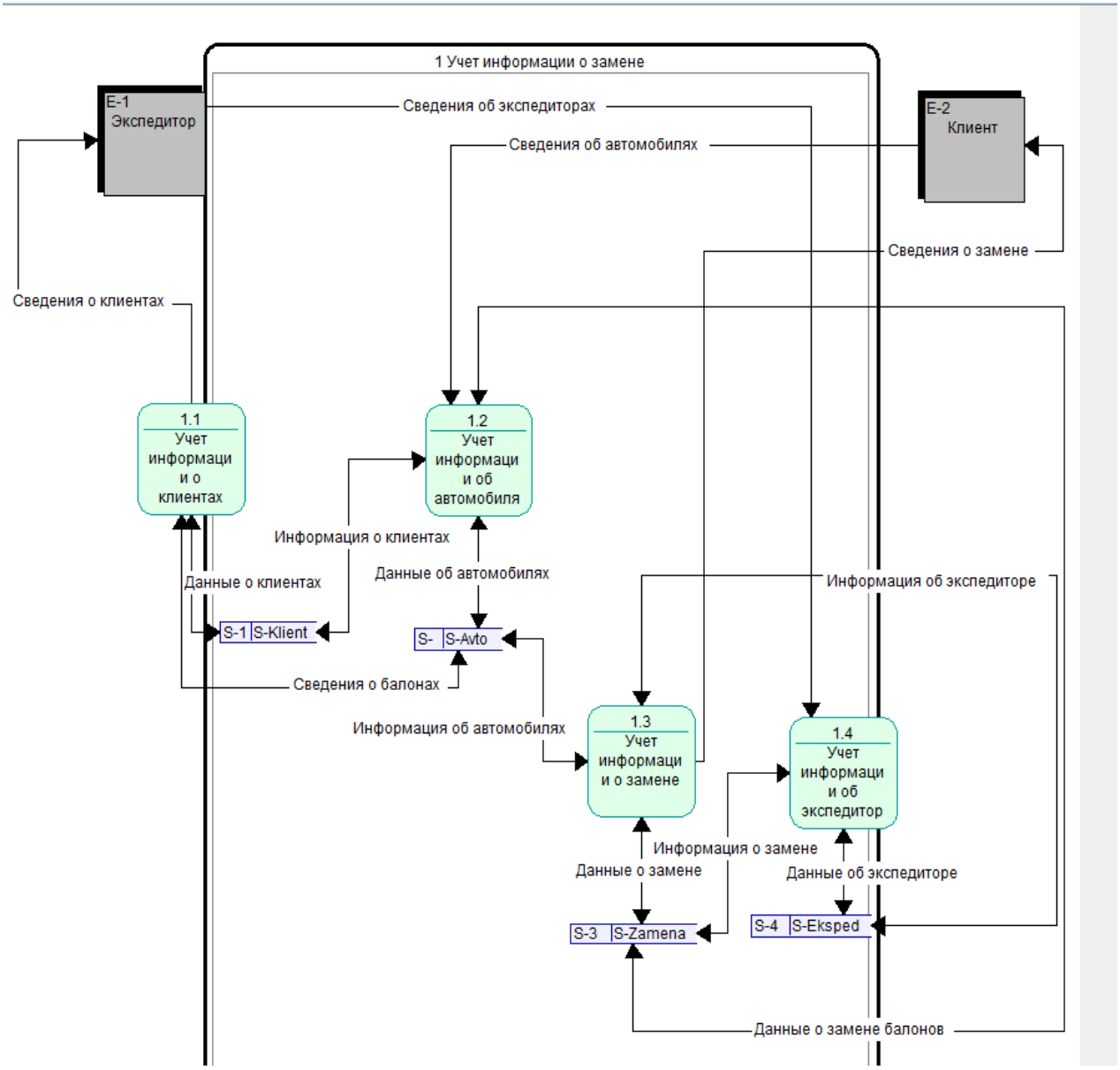


Рисунок 2.2 — Детализирующая диаграмма

Были выделены следующие процессы: «Учет информации о клиентах», «Учет информации об автомобиля», «Учет сведений о замене» и «Учет информации об экспедиторе». Также на основе анализа предметной области в модель были включены накопители: «klient», «avto», «zamena» и «eksped».

Внешний объект «Экспедитор» связан с процессом «Учет информации об экспедитор» и «Учет информации о клиентах выходным потоком «Сведения об экспедиторах» и входным потоком «Сведения о клиентах». Внешний объект «Клиент» связан с процессом «Учет информации об автомобиля» и «Учет сведений о замене» выходным потоком «Сведения об автомобилях» и входным потоком «Сведения о замене». Накопитель «klient» связан с процессом «Учет информации о клиентах» двунаправленным потоком «Данные о клиентах» и с процессом «Учет информации об автомобиля» двунаправленным потоком «Информация о клиентах». Накопитель «avto» связан с процессом «Учет информации об автомобиля» двунаправленным потоком «Данные об автомобилях», с процессом «Учет сведений о замене» двунаправленным потоком «Информация об автомобилях» и с процессом «Учет информации о клиентах» двунаправленным потоком «Сведения об автомобилях». Накопитель «zamena» связан с процессом «Учет информации о замене» двунаправленным потоком «Данные о замене», с процессом «Учет сведений об экспедитор» двунаправленным потоком «Информация о замене» и с процессом «Учет информации об автомобиля» двунаправленным потоком «Данные о замене запчастей». Накопитель «eksped» связан с процессом «Учет информации об экспедитор» двунаправленным потоком «Данные об экспедиторе» и с процессом «Учет сведений о замене» двунаправленным потоком «Информация об экспедиторе».

Таким образом, разработана детализирующая диаграмма.

На основе анализа предметной области были разработаны следующие структуры данных:

1) «avto»;

2) «eksped»;

3) «klient»;

4) «zamena»;

Описание структур данных модели приведено в таблицах 1-4.

Таблица 1 — Описание структуры данных «avto»

Имя атрибута	Назначение
navto	Идентификатор машины
nomer	Номер машины
kolb	Количество газовых баллонов
pzam	Периодичность замены

Таблица 2 — Описание структуры данных «klient»

Имя атрибута	Назначение
id_kl	Идентификатор клиента
fam	Фамилия
klname	Имя
otch	Отчество
kladres	Адрес клиента

Таблица 3 — Описание структуры данных «zamena»

Имя атрибута	Назначение
nzam	Идентификатор замены
zdate	Дата замены
kolzb	Количество газовых баллонов

Таблица 4 — Описание структуры данных «eksped»

Имя атрибута	Назначение
id_eksped	Идентификатор экспедитора
fam	Фамилия
ekname	Имя
otch	Отчество

2.4 Разработка концептуальной диаграммы

На основе анализа предметной области разработана концептуальная модель данных, которая содержит 4 сущности. Концептуальная модель данных представлена на рисунке 2.3.

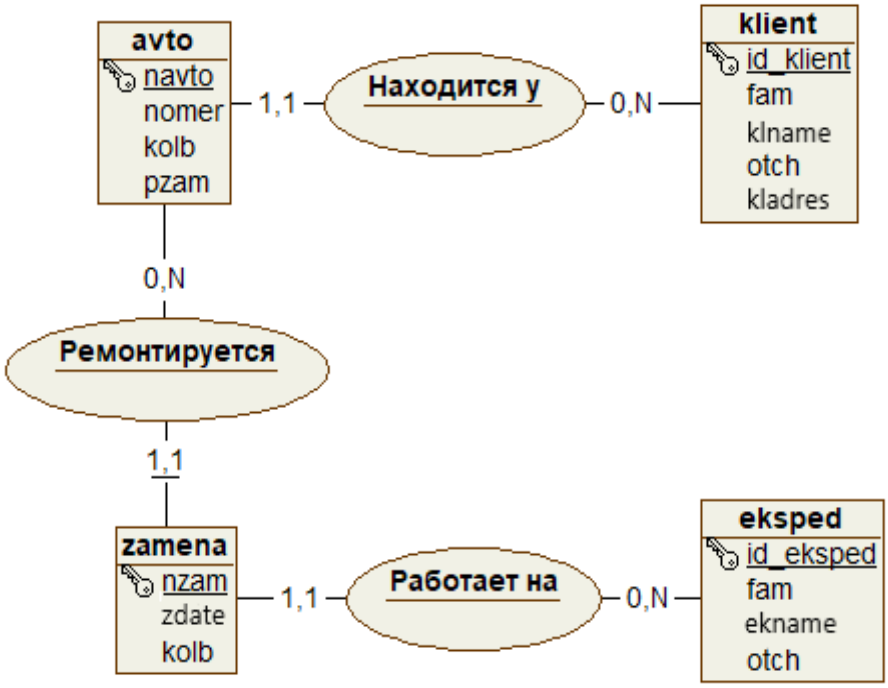


Рисунок 2.3 — Концептуальная модель данных

Описание сущностей и их атрибутов приведено в таблицах 5-8.

Таблица 5 - Описание атрибутов сущности avto

Имя атрибута	Назначение	Первичный ключ
navto	Номер автомобиля	PK
nomer	Гос. Номер автомобиля	-
kolb	Количество газовых баллонов	-
pzam	Периодичность замены	-

Таблица 6 - Описание атрибутов сущности klient

Имя атрибута	Назначение	Первичный ключ
Id_klient	Номер клиента	PK
fam	Фамилия клиента	-
klname	Имя клиента	-
otch	Отчество клиента	-
kladres	Адрес клиента	-

Таблица 7 - Описание атрибутов сущности eksped

Имя атрибута	Назначение	Первичный ключ
Id_eksped	Личный номер экспедитора	PK
fam	Фамилия экспедитора	-
ekname	Имя экспедитора	-
otch	Отчество экспедитора	-

Таблица 8 - Описание атрибутов сущности zamena

Имя атрибута	Назначение	Первичный ключ
nzam	Номер замены	PK
zdate	Дата замены	-
kolzb	Количество газовых баллонов	-

Сущность avto связана с сущностью klient с показателем кардинальности 1:1 — 0:N так как один автомобиль может находиться только у одного клиента, но в тоже время у одного клиента может находиться множество автомобилей.

Степень участия между сущностями не полная так как сущность avto может существовать без сущности klient.

Сущность avto связана с сущностью zamena с показателем кардинальности 0:N — 1:1 так как на одном автомобиле может заменяться несколько газовых баллонов и может не заменяться вообще, но в тоже время может заменяться только на одном автомобиле. Степень участия между сущностями полная так как сущность zamena не может существовать без сущности avto.

Сущность zamena связана с сущностью eksped с показателем кардинальности 1:1 — 0:N так как один экспедитор может обслуживать несколько машин и может не обслуживать их вовсе, но в тоже время одну машину может обслуживать только один экспедитор. Степень участия между сущностями не полная так как сущность eksped может существовать без сущности zamena.

В сущности avto в качестве первичного ключа выбран атрибут navto. В сущности klient в качестве первичного ключа выбран атрибут id_klient. В сущности zamena в качестве первичного ключа выбран атрибут nzam. В сущности eksped в качестве первичного ключа выбран атрибут id_eksped.

Таким образом разработана концептуальная модель данных.

3 Логическое проектирование базы данных

На основе построенной концептуальной модели данных разработана логическая модель данных, которая содержит 4 сущности. Логическая модель данных представлена на рисунке 3.1.

Так как между сущностями «Zamena» и «Avto» полная степень участия первичный ключ в сущности «Zamena» является составным и состоит из атрибутов: собственного первичного ключа «nzam» и внешнего ключа «navto», являющегося первичным ключом сущности «avto».

При преобразовании концептуальной модели данных в логическую, в модель были добавлены следующие внешние ключи:

- сущность «zamena» добавлен внешний ключ «navto», являющегося первичным ключом сущности «avto», а так же добавлен внешний ключ «id_eksped», являющегося первичным ключом сущности «eksped»;
- сущность «avto» добавлен внешний ключ «id_klient», являющегося первичным ключом сущности «klient».

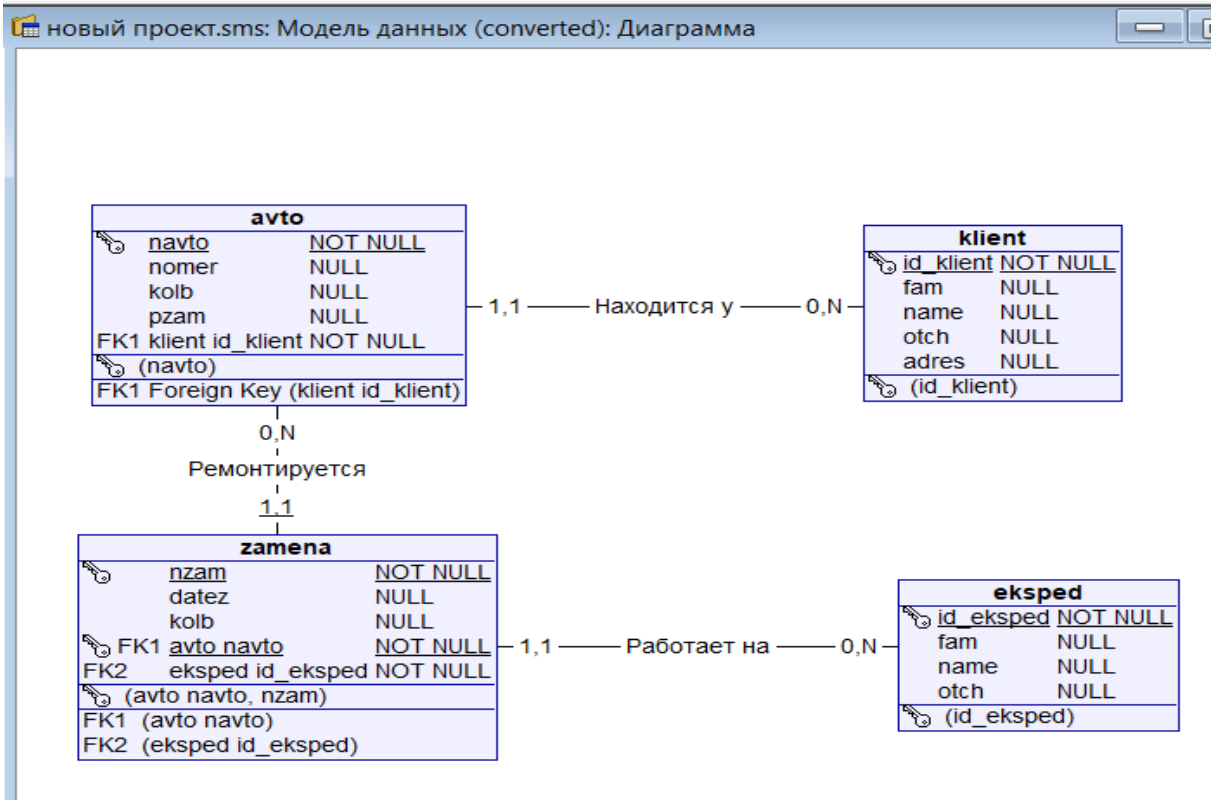


Рисунок 3.1 — Логическая модель данных

Описание атрибутов сущностей логической модели приведено в таблицах 9-12.

Таблица 9 - Описание атрибутов сущности «klient»

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
id_kl	Идентификатор клиента	PK	-
fam	Фамилия	-	-
klname	Имя	-	-
otch	Отчество	-	-
adres	Адрес клиента	-	-

Таблица 10 - Описание атрибутов сущности «avto»

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
navto	Идентификатор машины	PK	-
nomer	Номер машины	-	-
kolb	Количество газовых баллонов	-	-
pzam	Периодичность замены	-	-
id_kl	Идентификатор клиента	-	FK

Таблица 11 - Описание атрибутов сущности «zamena»

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
nzam	Идентификатор замены	PK	-
zdate	Дата замены	-	-
kolzb	Количество заменяемых газовых баллонов	-	-
navto	Идентификатор автомобиля	PK	FK
id_eksped	Идентификатор экспедитора	-	FK

Таблица 12 - Описание структуры данных «eksped»

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
id_eksped	Идентификатор экспедитора	PK	-
fam	Фамилия	-	-
ekname	Имя	-	-
otch	Отчество	-	-

Отношение находится в первой нормальной форме, если значения всех его атрибутов являются атомарными (однозначно определены).

Все атрибуты четырех сущностей атомарны и находятся в 1 НФ. Для подтверждения данного фактора необходимо рассмотреть каждую сущность:

1) сущность «avto» имеет ключевой атрибут «navto». Атрибуты «nomer», «kolb», «pzam» и «id_klient» данной сущности является простыми и неделимыми;

2) сущность «zamena» имеет ключевые атрибуты «nzam» и «navto». Атрибуты «datez», «kolb» и «id_eksped» данной сущности является простыми и неделимыми;

3) сущность «klient» имеет ключевой атрибут «id_klient». Атрибуты «fam», «name», «otch» и «adres» данной сущности является простым и неделимым.

4) сущность «eksped» имеет ключевой атрибут «id_eksped». Атрибуты «fam», «name», «otch» данной сущности является простым и неделимым.

Все отношения логической модели данных находятся в 2 НФ, так как их неключевые атрибуты функционально полно зависят от первичных ключей.

Отношение «avto» находится во 2НФ, так как имеет единственный ключевой атрибут «navto». Неключевые атрибуту «nomer», «kolb», «pzam» и «id_klient» данного отношения функционально полно зависит от данного первичного ключа. Следовательно, можно сделать вывод о том, что данное отношение находится в 2НФ.

Отношение «zamena» находится во 2НФ, так как имеет составной первичный ключ, состоящий из атрибутов из «nzam» и «navto». Неключевые атрибуты «datez», «kolb» и «id_eksped» данного отношения функционально полно зависят от данного первичного ключа в целом, а не от его части. Атрибуты «datez», «kolb» и «id_eksped» зависят от атрибутов «navto» и «nzam». Следовательно, можно сделать вывод о том, что данное отношение находится в 2НФ.

Отношение «klient» находится во 2НФ, так как имеет единственный ключевой атрибут «id_klient». Неключевой атрибут «fam», «name», «otch» и «adres» данного отношения функционально полно зависит от данного первичного ключа. Следовательно, можно сделать вывод о том, что данное отношение находится в 2НФ.

Отношение «eksped» находится во 2НФ, так как имеет единственный ключевой атрибут «id_eksped». Неключевой атрибут «fam», «name» и «otch» данного отношения функционально полно зависит от данного первичного ключа. Следовательно, можно сделать вывод о том, что данное отношение находится в 2НФ.

Ни в одной из сущностей логической модели данных не существует транзитивных зависимостей. Необходимо доказать, что сущности логической модели находятся в 3НФ.

В сущности «avto» неключевые атрибуты «nomer», «kolb», «pzam» и «id_klient» данного отношения не зависят друг от друга. Следовательно, можно сделать вывод о том, что данные сущности логической модели находится в 3НФ.

В сущности «zamena» неключевые атрибуты «datez», «kolb» и «id_eksped» данного отношения не зависят друг от друга. Следовательно, можно сделать вывод о том, что данные сущности логической модели находится в 3НФ.

В сущности «klient» неключевые атрибуты «fam», «name», «otch» и «adres» данного отношения не зависят друг от друга. Следовательно, можно

сделать вывод о том, что данные сущности логической модели находится в 3НФ.

В сущности «eksped» неключевые атрибуты «fam», «name» и «otch» данного отношения не зависят друг от друга. Следовательно, можно сделать вывод о том, что данные сущности логической модели находится в 3НФ.

Так как между сущностями «avto» и «zamena» полная степень участия первичный ключ в сущности «zamena» является составным и состоит из атрибутов: собственного первичного ключа «nzam» и внешнего ключа «navto», являющегося первичным ключом сущности «avto».

Так как сущность «klient» связана с сущностью «avto», в сущность «avto» добавлен внешний ключ «id_klient», являющегося первичным ключом сущности «klient».

Так как сущность «eksped» связана с сущностью «zamena», в сущность «zamena» добавлен внешний ключ «id_eksped», являющегося первичным ключом сущности «eksped».

Таким образом разработана реляционная модель данных.

4 Физическое проектирование базы данных

На основе разработанной логической модели данных была разработана физическая модель данных, выполнена команда меню «Сервис > Модель данных > Добавить/Удалить целевую систему», чтобы добавить к проекту новую целевую СУБД (используемую для создания базы данных).

В открывшемся окне, представленном на рисунке 4.1, нажата кнопка «Добавить».

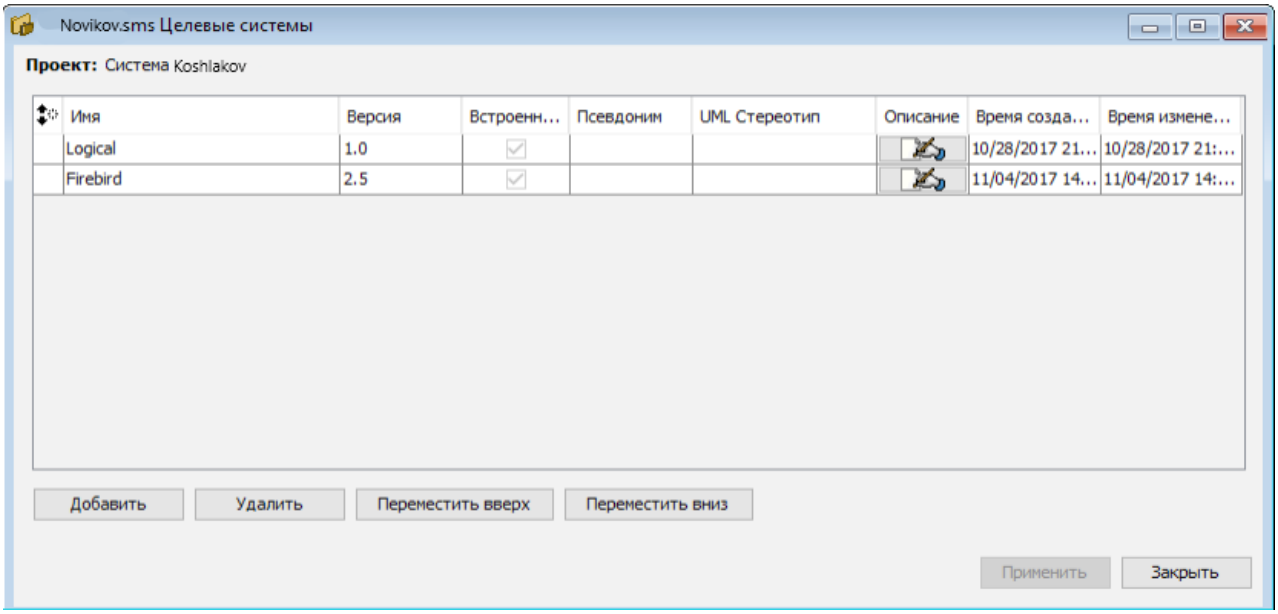


Рисунок 4.1 — Добавление к проекту целевой СУБД

Из списка доступных целевых систем выбрана система Linter 6.0, как представлено на рисунке 4.2.

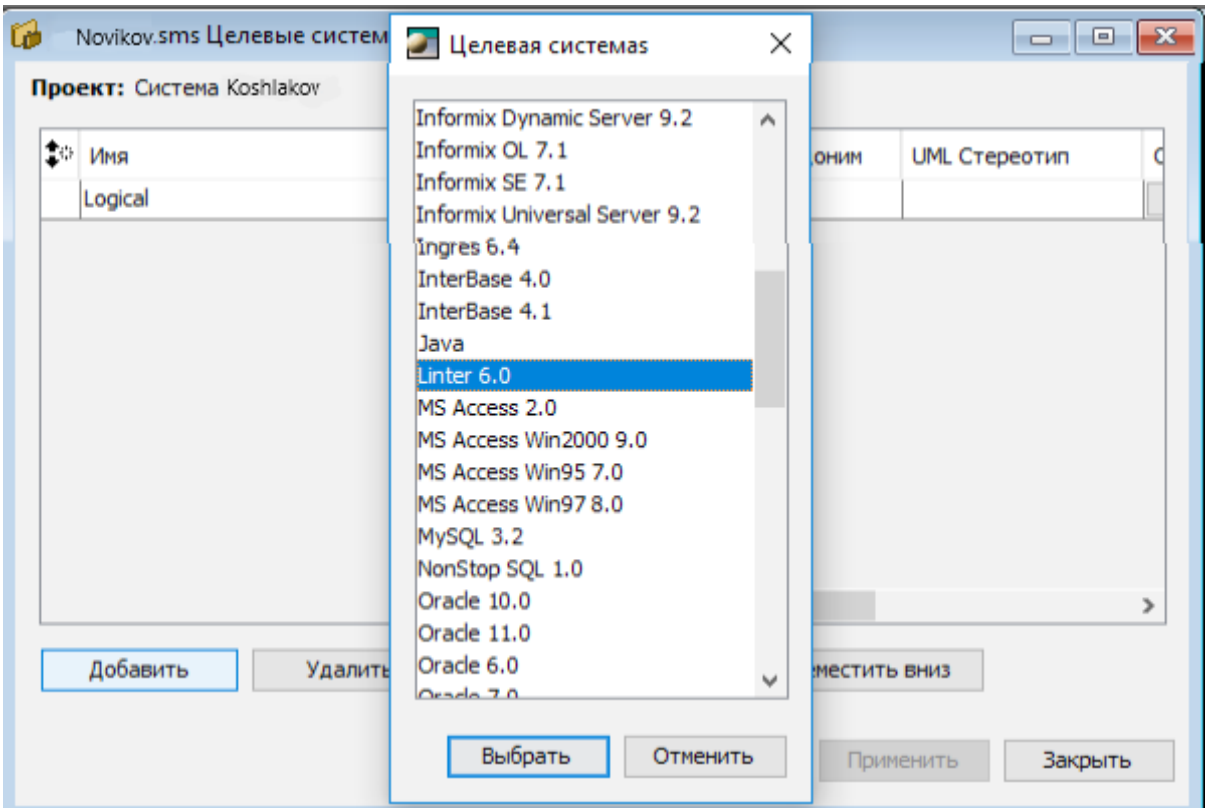


Рисунок 4.2 — Выбор целевой СУБД

Затем нажата кнопка «Выбрать», а затем - кнопка «Закрыть», чтобы закрыть окно со списком целевых систем, представленных на рисунке 4.3.

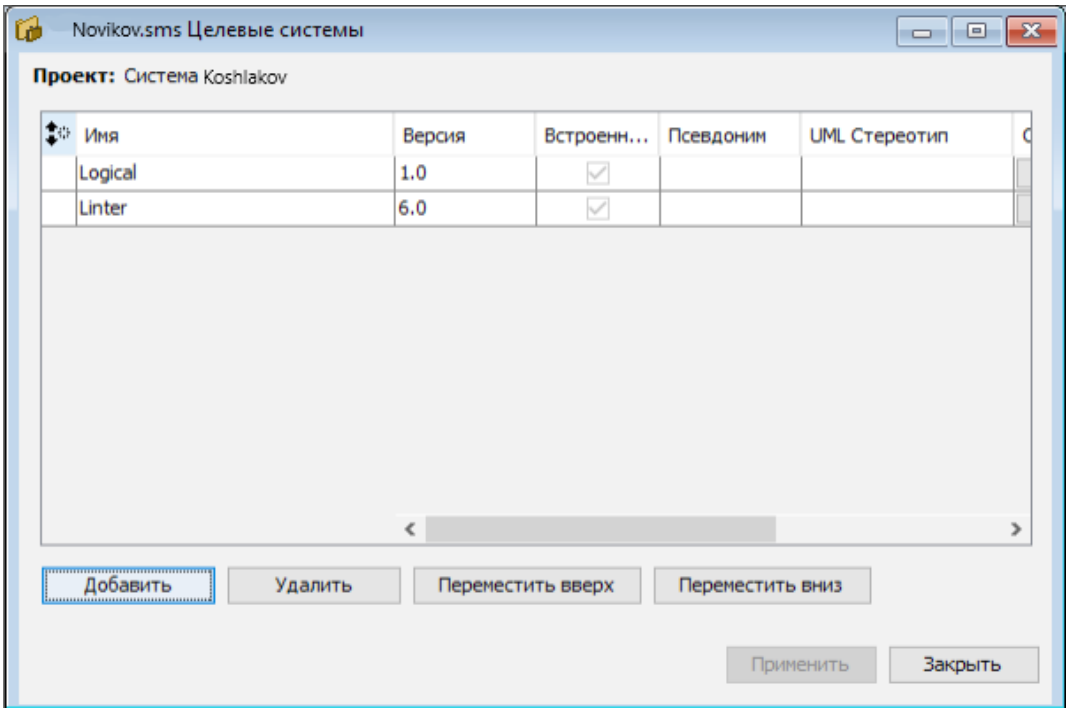


Рисунок 4.3 — Список целевых систем проекта

После назначения проекту целевой системы, могут быть использованы встроенные типы данных, поддерживаемые этой целевой системой, для определения типов данных в столбцах таблиц. В окне «Навигатор» в узле «Встроенные типы» выводится список встроенных типов данных, поддерживаемых СУБД Linter (Рисунок 4.4).

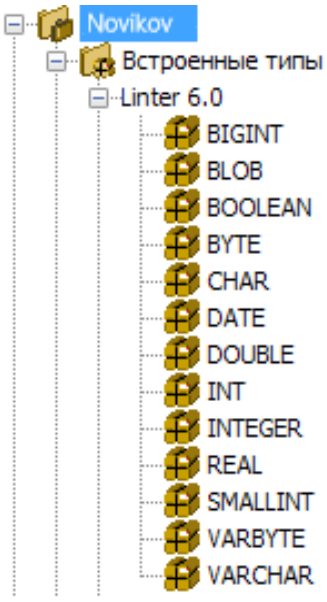


Рисунок 4.4 — Типы данных Linter

В окне «Навигатор» выбран узел «Novikov» и добавлена новая модель данных для СУБД Linter. Для этого нужно было нажато правой кнопкой по узлу «Novikov» и из всплывающего меню выбрана команда «Добавить > Модель данных > Linter 6.0». Затем необходимо переименована созданная модель в «Физическая модель данных <Linter 6.0>».

Далее раскрыт узел «Реляционная модель данных» и выделены все таблицы этой модели в узле «Таблицы», а затем скопированы в окно физической модели. Для этого нужно было щелкнуть правой кнопкой по выделенным строкам и из всплывающего меню выбрана команда «Копировать», как представлено на рисунке 4.5.

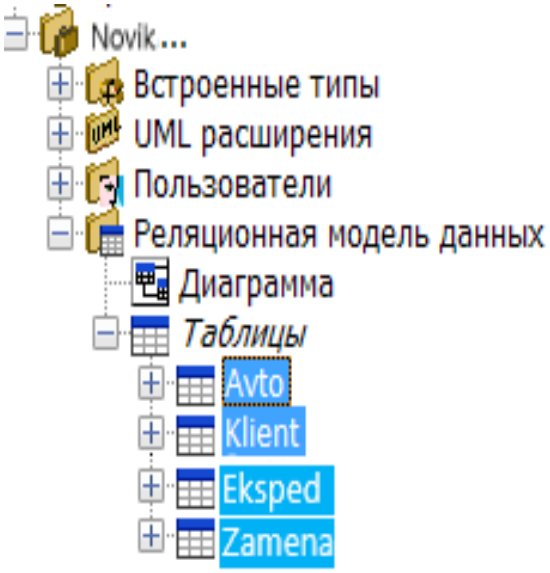


Рисунок 4.5 — Копирование таблиц

Модель данных представляет собой контейнер таблиц; для этого выбран узел «Физическая модель данных <Linter 6.0>» и вставлены скопированные таблицы. Для этого было нажато правой кнопкой по выбранному узлу и из всплывающего меню выбрана команда «Вставить».

После этого модель данных Linter содержит таблицы. Их столбцы имеют типы, встроенные в СУБД Linter.

Далее выбран узел «Диаграмма» для модели Физическая модель данных <Linter 6.0> в окне «Навигатор», затем нажато по нему правой кнопкой мыши и из всплывающего меню выбрана команда «Показать диаграмму».

По умолчанию, таблицы при этом не были отображены в графическом представлении. Далее была выбрана из всплывающего меню команду «Добавить отсутствующие объекты диаграммы». Результат представлен на рисунке 4.6.

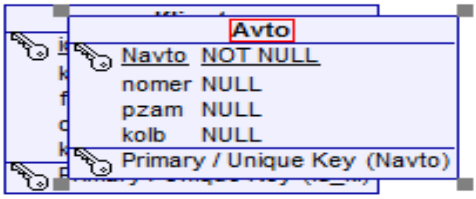


Рисунок 4.6 — Настройка модели

Графические представления таблиц перекрывают друг друга. Поэтому необходимо из всплывающего меню была выбрана команда «Упорядочить всю диаграмму». После этого все таблицы расположены в одну линию. Поэтому их необходимо было переупорядочить и создать между ними связи. Результат представлен на рисунке 4.7.

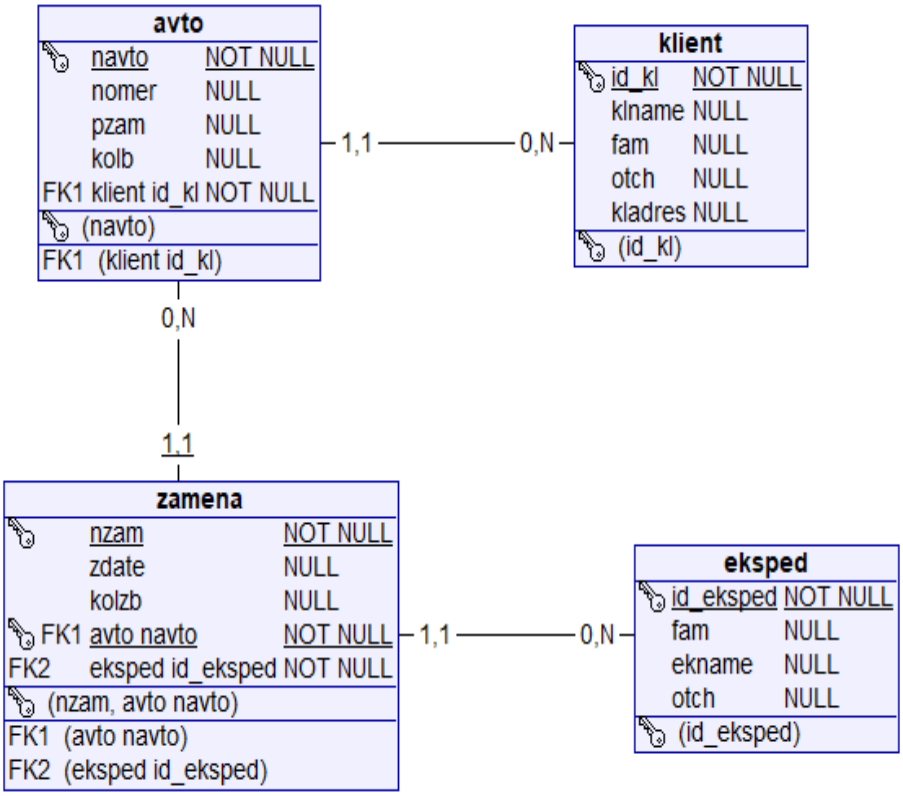


Рисунок 4.7 — Упорядоченная физическая модель данных

Для определения типов данных необходимо по очереди были выделен в таблицах реляционной модели имена атрибутов, а затем в окне «Редактор свойств» выбраны в столбце «Значение» свойства «Тип». В результате было открыто окно «Выбрать новое значение», как представлено на рисунке 4.8.

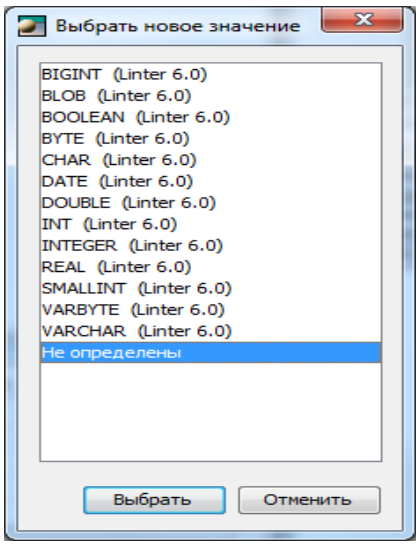


Рисунок 4.8 — Окно Выбрать новое значение

Из списка были выбраны необходимые типы данных, в соответствии с таблицами. После этого нажата на кнопка «Select» выпадающего списка. На диаграмме реляционной модели в соответствующих строках будут отображаться выбранные типы данных Linter.

Также были определены значение длины типа char и сделано то же самое для всех столбцов таблиц, как показано на диаграмме, представленной на рисунке 4.9. Для определения длины типа char выбрано в столбце «Значение» свойства «Длина» и введено необходимое значение.

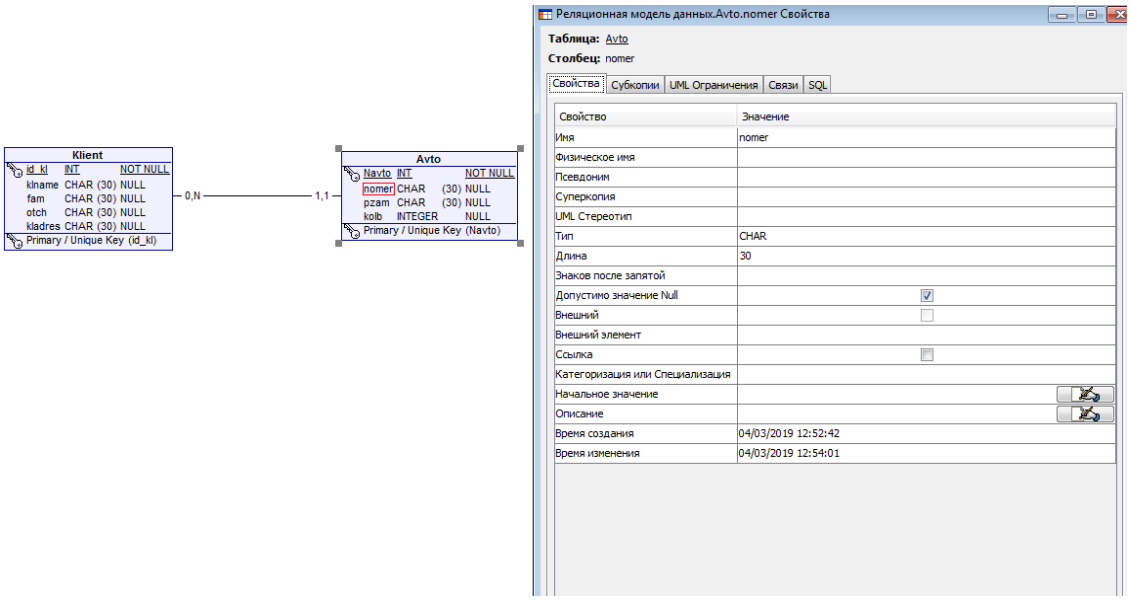


Рисунок 4.9 — Типы столбцов созданных таблиц

Далее сгенерированы внешние ключи. Для этого использовалась команду меню «Сервис > Модель данных > Сгенерировать внешние ключи».

В открывшемся окне, представленном на рисунке 4.10, выбрана на кнопка «Генерация внешних ключей».

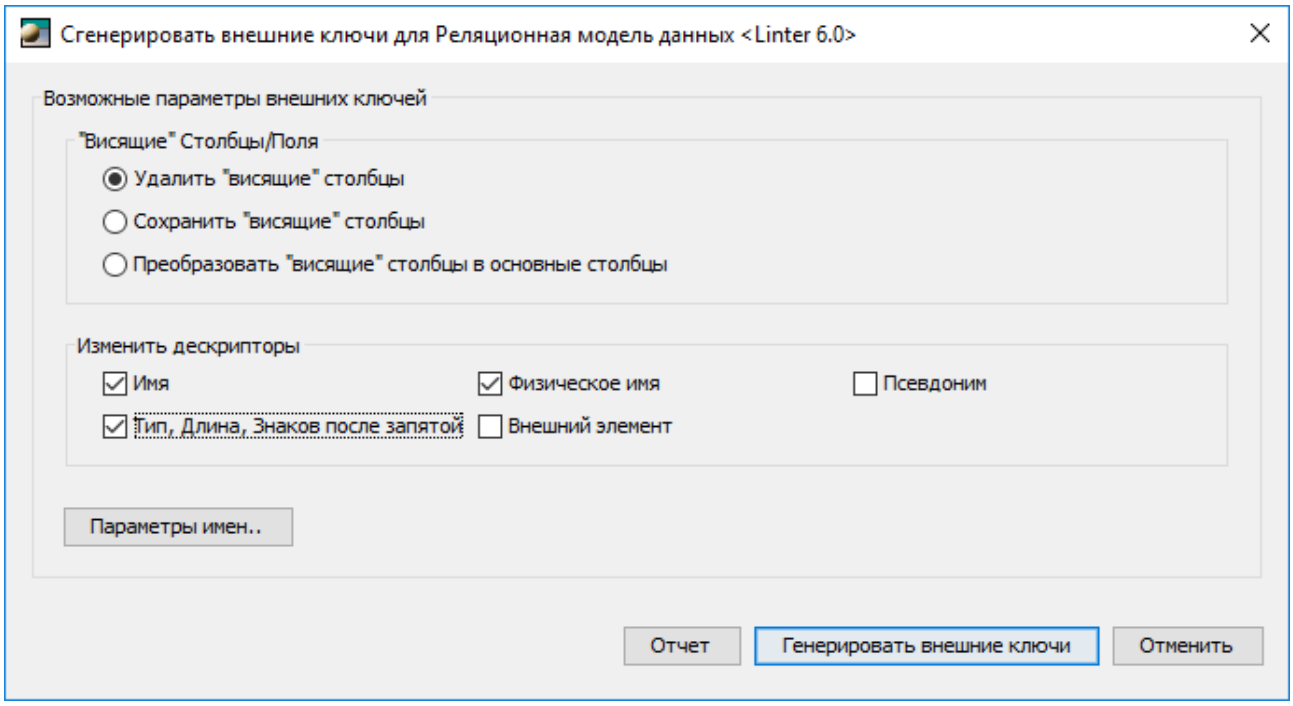


Рисунок 4.10 — Генерация внешних ключей

Как видно из отчета о генерации внешних ключей, представленном на рисунке 4.11, сгенерированы три новых внешних столбца и три новых внешних ключа.

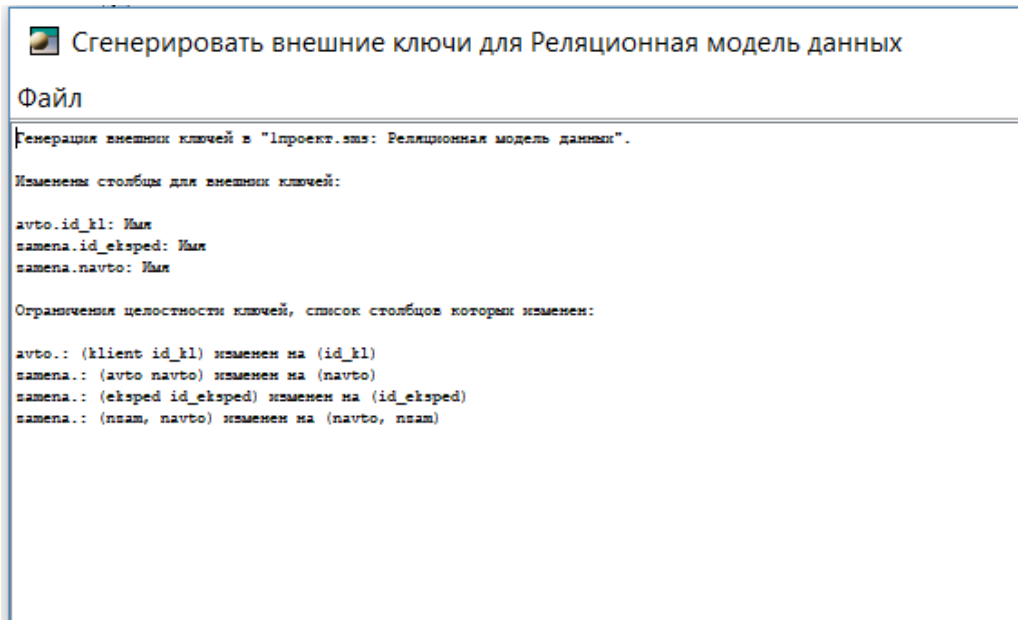


Рисунок 4.11 — Отчет о генерации внешних ключей

Модель после генерации внешних ключей показана на рисунке 4.12.

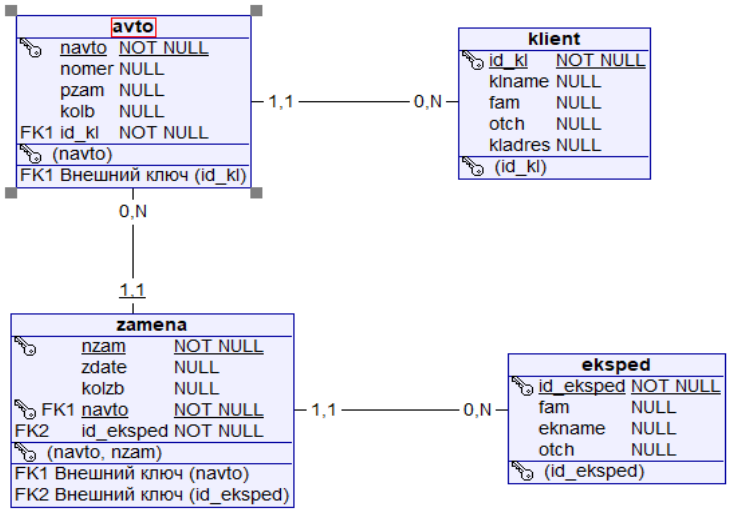


Рисунок 4.12 — Модель после генерации внешних ключей

После ввода вручную физических имен всех таблиц и их столбцов, остаются без физических имен первичные и внешние ключи всех таблиц физической модели.

Создать физические имена первичных и внешних ключей можно вручную или с помощью генератора физических имен системы Open ModelSphere. Однако, возможности автоматической генерации физических имен довольно ограничены, генерируемые таким образом имена далеки от совершенства и имеют большую длину. Поэтому необходимо воспользоваться ручным способом.

Для заданного варианта физические имена первичных и внешних ключей приведены в таблицах 8-9.

Таблица 8 - Первичные ключи

Имя таблицы	Имя ключа	Нулевое значение
KLIENT	PK_ID_KL	Недопустимо
AVTO	PK_NAVTO	Недопустимо
EKSPED	PK_ID_EKSPED	Недопустимо
ZAMENA	PK_NZAM	Недопустимо
	PK_FK_NAVTO	Недопустимо

Таблица 9 - Внешние ключи

Имя таблицы	Имя ключа	Нулевое значение
AVTO	FK_ID_KL	Недопустимо
ZAMENA	FK_ID_EKSPED	Недопустимо

Для выполнения генерации указанных физических имен с помощью средств ModelSphere выбран в окне «Навигатор» узел «Физическая модель данных <Linter 6.0>» и выполнена команда меню «Сервис > Генерировать > Физические имена». В результате было открыто окно «Генерация физических имен», как представлено на рисунке 4.13.

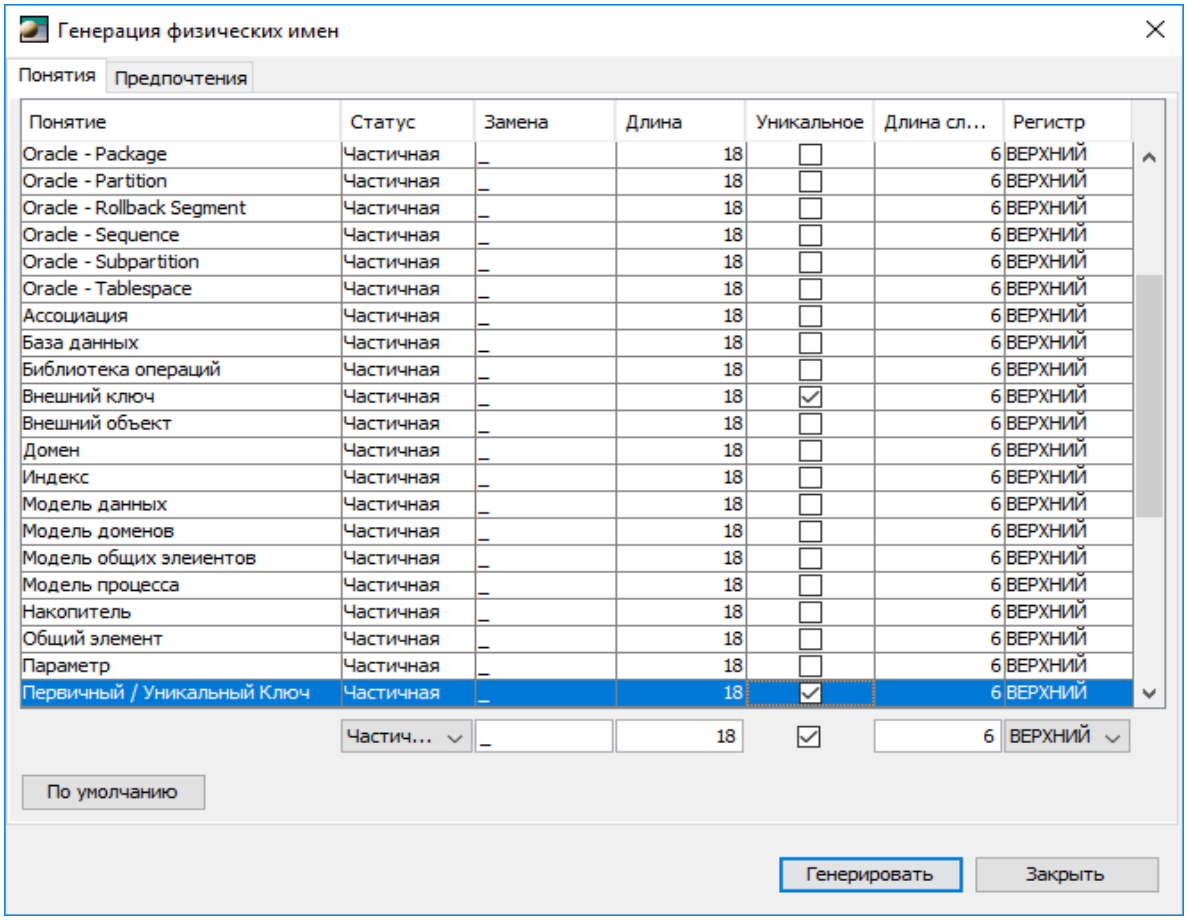


Рисунок 4.13 — Генерирование физических имен первичных и внешних ключей

В открывшемся окне в поле «Статус» можно увидеть слово «Частичная», которое означает, что только отсутствующие физические имена были модифицированы. Так как ранее физические имена всех атрибутов таблиц уже были определены. В поле «Замена» расположен символ замены недопустимых символов, таких как пробел. В открывшемся окне выделены строки с именами «Внешний ключ» и «Первичный/Альтернативный Ключ». Затем установлены в столбце «Уникальное» в нужных строках флажки, чтобы физические имена

были уникальными. Остальные настройки не изменялись. Затем нажата кнопка «Генерировать».

В результате сгенерированы физические имена всех первичных и внешних ключей физической модели данных, как представлено на рисунке 4.14. Кроме того, были сгенерированы физические имена для таблиц и ролей, которые ранее не были определены вручную.

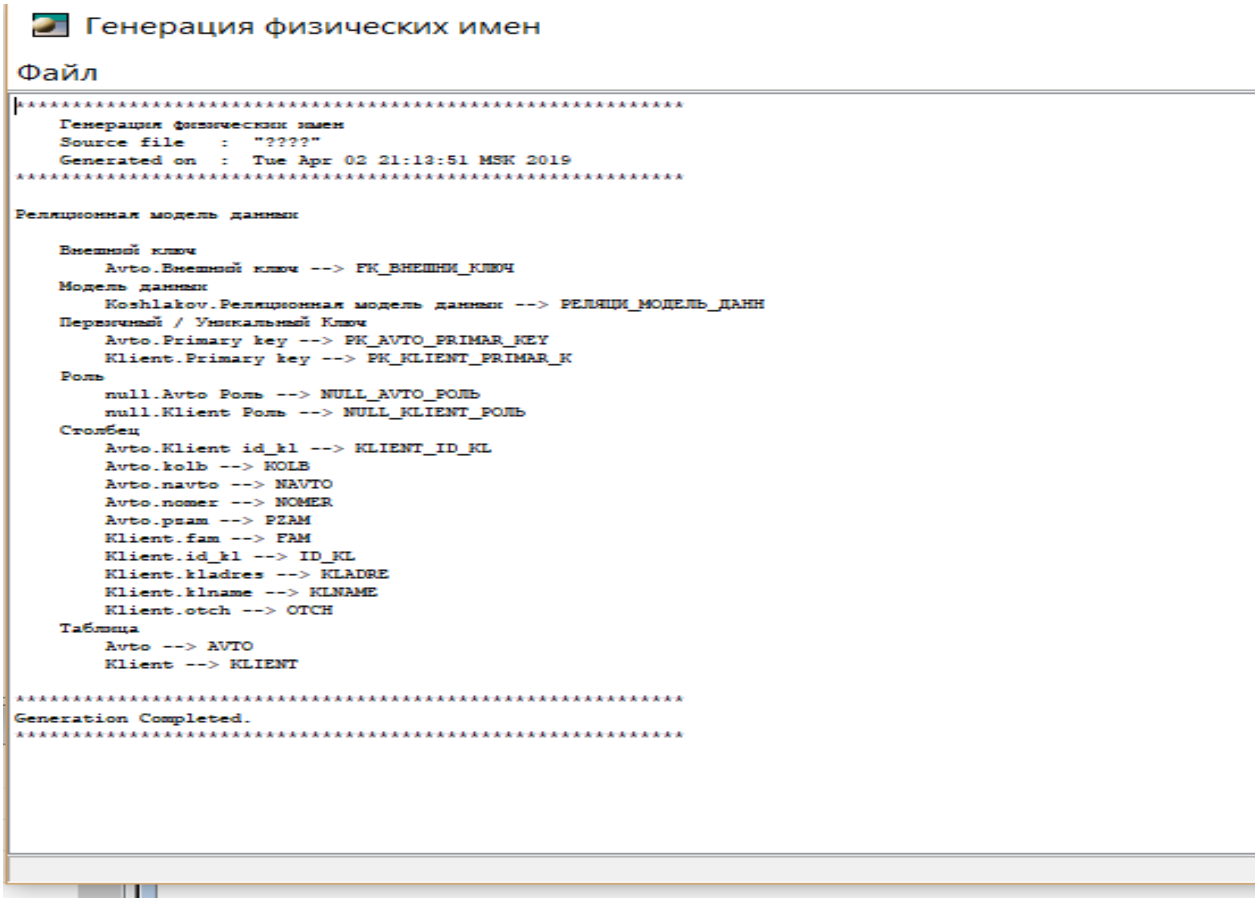


Рисунок 4.14 — Отчет о генерации физических имен

В отчете о генерации физических имен можно увидеть, что сгенерированные физические имена имеют значительно большую длину, чем те имена, которые были введены вручную. Кроме того, имена внешних ключей не содержат названий таблиц, в которых они используются.

Индексы в базе данных используются в следующих основных случаях:

1) Ускорение выполнения запросов;

2) Обеспечение уникальности значений в полях. Ограничение первичного ключа требует, чтобы во всей таблице не нашлось двух одинаковых значений полей, входящих в первичный ключ. Чтобы выполнить это условие, необходимо при каждой вставке новой записи производить поиск такого же значения, которое будет вставлено. Для поиска записей используется уникальный индекс;

3) Обеспечение ссылочной целостности. Ограничения внешних ключей используются для проверки того, чтобы вставляемые в таблицу значения обязательно существовали в другой таблице. При создании внешнего ключа индекс применяется для проверки условий внешнего ключа.

Для создания уникального индекса для первичного ключа в таблице щелкнули по первичному ключу (для таблицы KLIENT по строке Primary key (ID_KL)) правой кнопкой. Из контекстного меню выбрана команда «Добавить > Индекс». В нижней части изображения таблицы появится новая строка, в которой будет расположена запись о добавленном индексе (для таблицы KLIENT это будет X1 Primary key (ID_KL) UNIQUE). Далее была выбрана строка «Свойства». В результате было открыто окно со свойствами добавленного индекса, изображенное на рисунке 4.15.

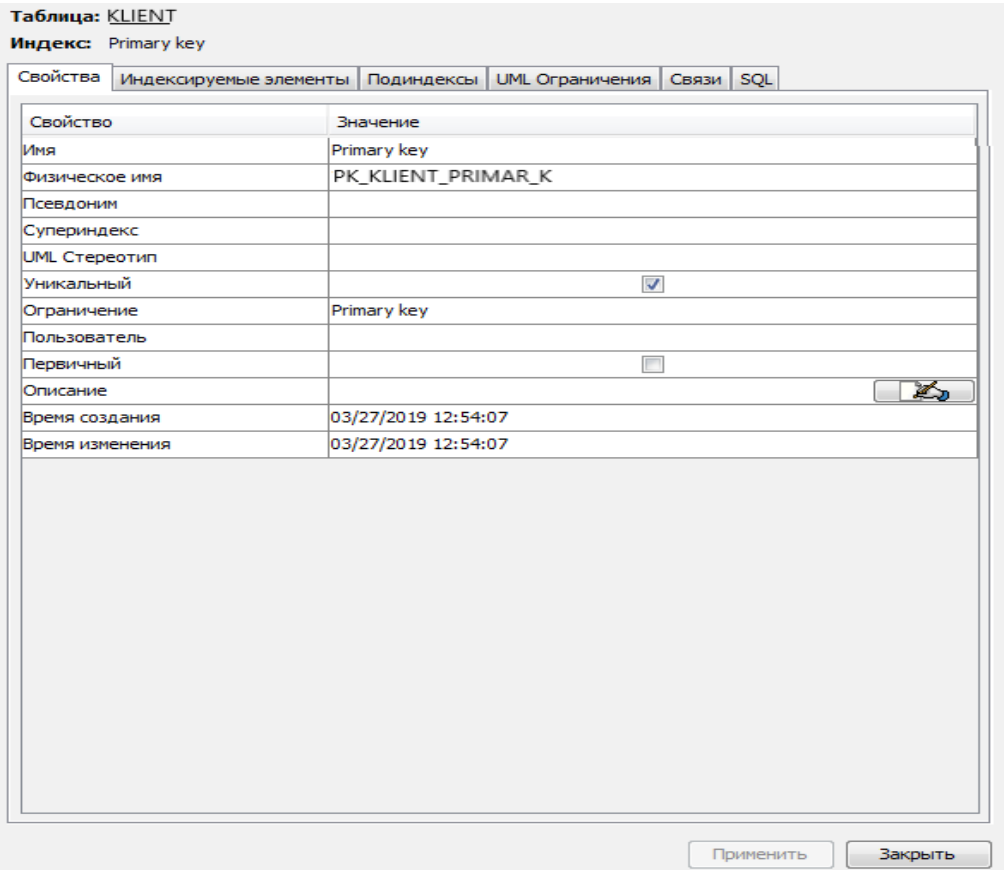


Рисунок 4.15 — Свойства уникального индекса для первичного ключа

На вкладке «Свойства» этого окна было изменено «Имя» этого индекса и вместо значения Primary key введено ID_KL (значение физического имени) и нажата кнопка «Применить». Затем открыта вкладка «Индексируемые элементы», представленная на рисунке 4.16, и в столбце «Сортировка» в строке с именем элемента необходимо щелкнуть два раза и выбрать из контекстного меню строку ASC (сортировка по возрастанию). После этого нажата кнопка «Применить», а затем кнопка «Закрыть».

На изображении таблицы будут видны внесенные изменения.

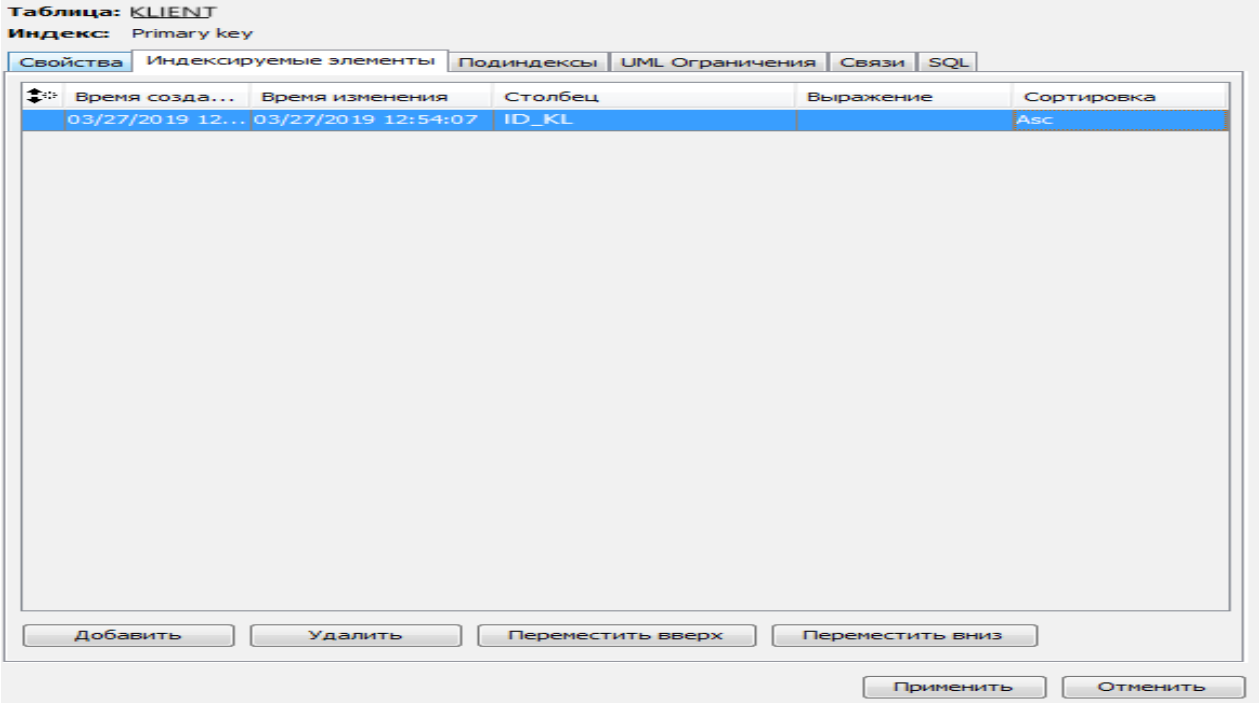


Рисунок 4.16 — Определение направления сортировки в индексе

Для создания индекса для внешнего ключа в таблице было нажато по внешнему ключу (для таблицы AVTO по строке FK1 Foreign key (ID_KL)) правой кнопкой. Из контекстного меню выбрана команда «Добавить > Индекс». В нижней части изображения таблицы появилась новая строка, в которой будет расположена запись о добавленном индексе (для таблицы AVTO это будет X2 Foreign key (ID_KL) NOT UNIQUE).

Далее было нажато правой кнопкой по этой записи и из контекстного меню выбрана строка «Свойства». В результате было открыто окно со свойствами добавленного индекса, представленное на рисунке 4.17.

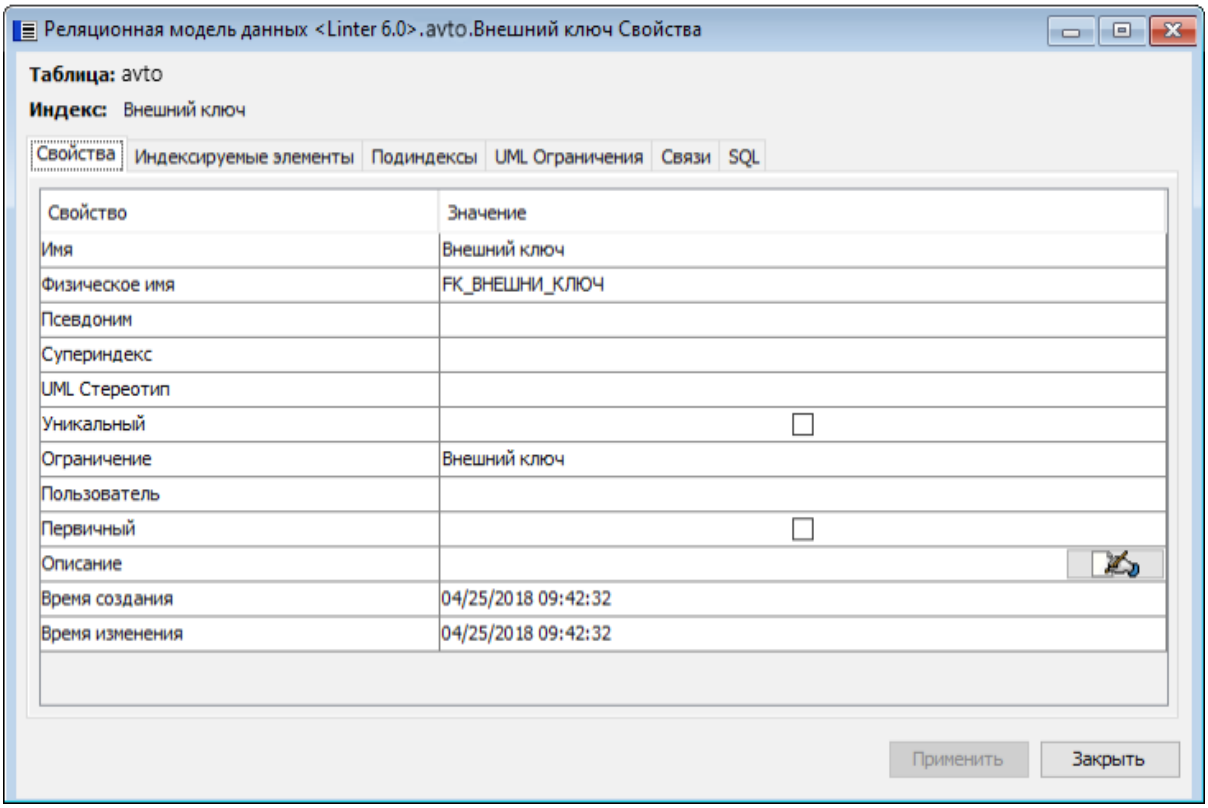


Рисунок 4.17 — Свойства индекса для внешнего ключа

На вкладке «Свойства» этого окна было изменено «Имя» этого индекса и вместо значения Foreign key введено FK_ID_KL (значение физического имени) и нажата кнопка «Применить». Затем открыта вкладка «Индексируемые элементы», представленная на рисунке 4.18, и в столбце «FK_ID_KL» в строке с именем элемента выбрана из контекстного меню строку ASC (сортировка по возрастанию). После этого нажата кнопка «Применить», а затем на кнопку «Закрыть».

На изображении таблицы станут видны внесенные изменения.

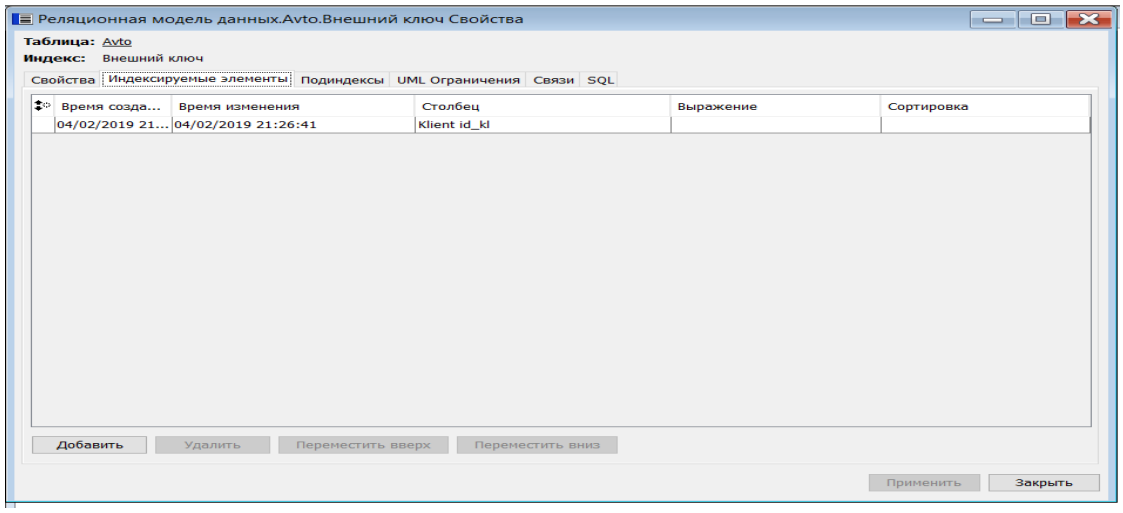


Рисунок 4.18 — Определение направления сортировки в индексе

Аналогичным образом добавлены ко всем таблицам физической модели индексы для первичных и внешних ключей. В результате физическая модель данных имеет вид, приведенный на рисунке 4.19.

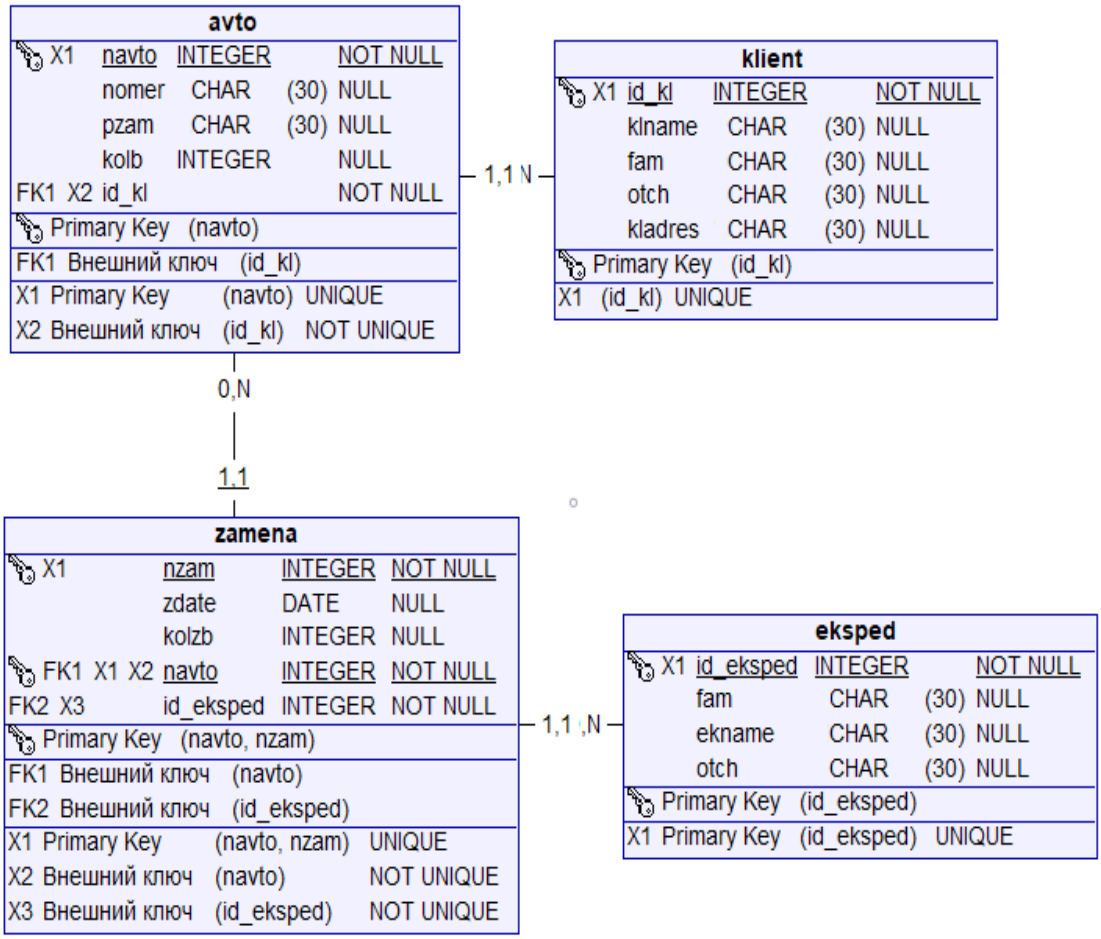


Рисунок 4.19 — Окончательный вид физической модели данных

На этапе физического проектирования были определены типы данных с учетом конкретной СУБД. Физическая модель данных содержит 3 таблицы. Описание полей таблиц физической модели приведено в таблицах 10-13.

Таблица 10 — Описание полей таблицы «KLIENT»

Наименование атрибута	Физическое имя	Нулевое значение	Ключевое поле	Тип данных
ID_KL	PK_ID_KL	Недопустимо	PK	INT
KLNAME	KLNAME	Допустимо		CHAR(30)
FAM	KLFAM	Допустимо		CHAR(30)
OTCH	KLOTCH	Допустимо		CHAR(30)
KLADRES	KLADRES	Допустимо		CHAR(30)

Таблица 11 — Описание полей таблицы «AVTO»

Наименование атрибута	Физическое имя	Нулевое значение	Ключевое поле	Тип данных
NAVTO	PK_NAVTO	Недопустимо	PK	INT
NOMER	NOMER	Допустимо		CHAR(30)
PZAM	PZAM	Допустимо		CHAR(30)
KOLB	KOLB	Допустимо		INTEGER
ID_KL	FK_ID_KL	Допустимо	FK	INT

Таблица 12 — Описание полей таблицы «EKSPED»

Наименование атрибута	Физическое имя	Нулевое значение	Ключевое поле	Тип данных
NAVTO	PK_NAVTO	Недопустимо	PK	INT
FAM	EKFAM	Допустимо		CHAR(30)
EKNAME	EKNAME	Допустимо		CHAR(30)
OTCH	EKOTCH	Допустимо		CHAR(30)

Таблица 13 — Описание полей таблицы «ZAMENA»

Наименование атрибута	Физическое имя	Нулевое значение	Ключевое поле	Тип данных
NZAM	PK_NZAM	Недопустимо	PK	INT
KOLZB	KOLZB	Допустимо		INTEGER
ZDATE	ZDATE	Допустимо		DATE
ID_EKSPED	FK_ID_EKSP	Допустимо	FK	INT
NAVTO	FK_NAVTO	Допустимо	FK,PK	INT

На основе физической модели данных был сгенерирован файл gendll.sql,, содержащий скрипт базы данных на языке sql. В полученном скрипте были удалены все комментарии, заключенные в «--/*/» Текст скрипта приведен в приложение А. Таким образом, была разработана физическая модель базы данных для СУБД Линтер.

5.Создание базы данных

В приложении "Администратор СУБД ЛИНТЕР" был выделен в дереве баз данных сервер sqledu03 и выполннена команда База > Создать.

С помощью вкладок диалогового окна "Свойства базы данных" была выполнена конфигурирование создаваемой базы данных (рисунок 5.1).

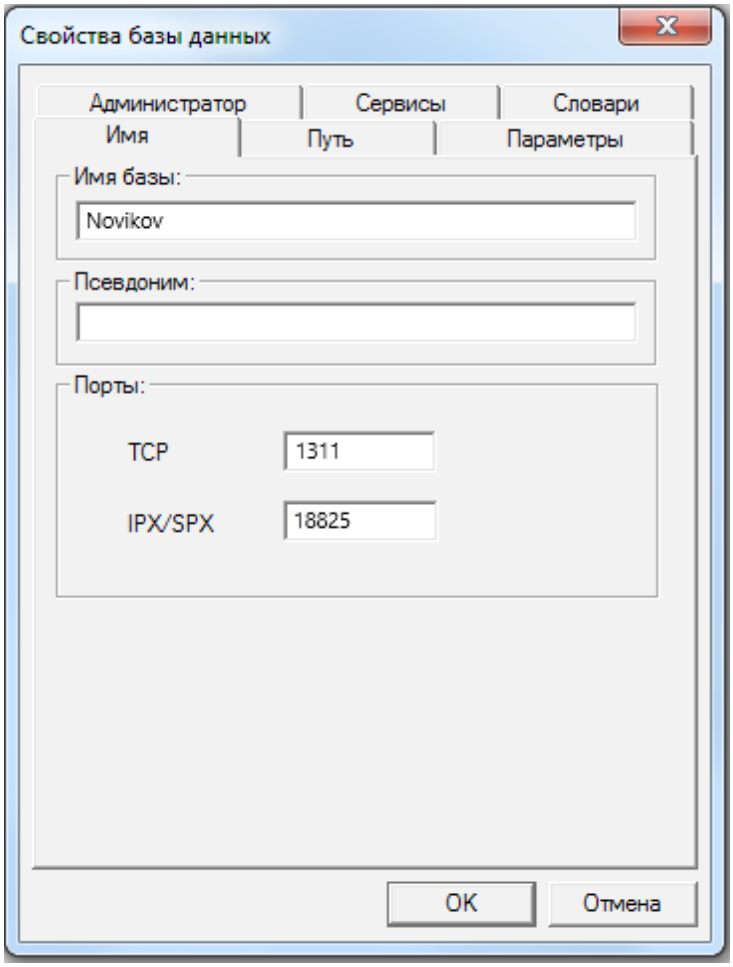


Рисунок 5.1 — Диалоговое окно «Создание базы данных»

Для конфигурирования создаваемой базы данных в окне "Свойства базы данных" были выполнены следующие действия:

На вкладке «Имя»:

1) в поле "Имя базы" (рисунок 5.1) было введено уникальное имя создаваемой базы данных на удаленной машине sqledu03 длиной не более 18 символов (Novikov);

2) в поле "TCP" (рисунок 5.1) введён номер TCP порта, который будет использоваться для подключения к серверу (1311). Номер TCP порта не совпадает с номерами портов других БД, установленных на сервере.

3) в поле "IPX/SPX" (рисунок 5.1) введён номер IPX/SPX порта, который может использоваться для подключения к серверу (18825). Номер IPX/SPX порта не совпадает с номерами портов других БД, установленных на сервере.

На вкладке «Путь» (рисунок 5.2) был введён путь и имя файла базы данных на том компьютере, где она будет находиться. В данном случае база данных создается на компьютере с именем sqledu03 в папке D:\Linter\LR1\Novikov\.

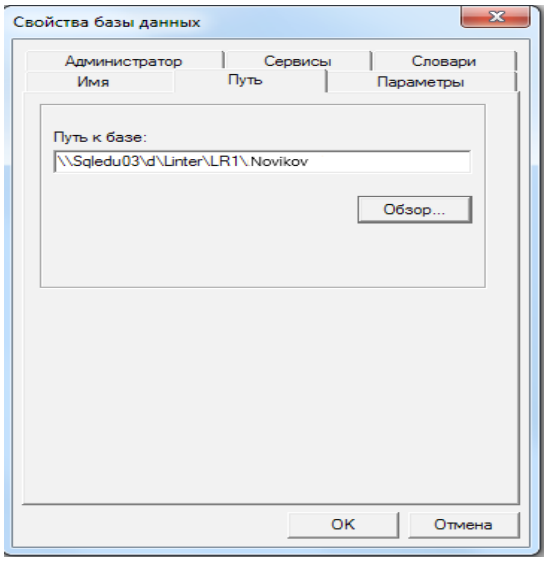


Рисунок 5.2 — Вкладка «Путь»

Для задания количественных параметров и параметров доступа для создаваемой БД перейти на вкладку Параметры. Эта вкладка используется для задания параметров доступа к создаваемой БД: количества таблиц, количества колонок (столбцов), числа регистрируемых в БД пользователей, а также возможности шифрования БД. Были использованы значения, указанные на рисунке 5.3.

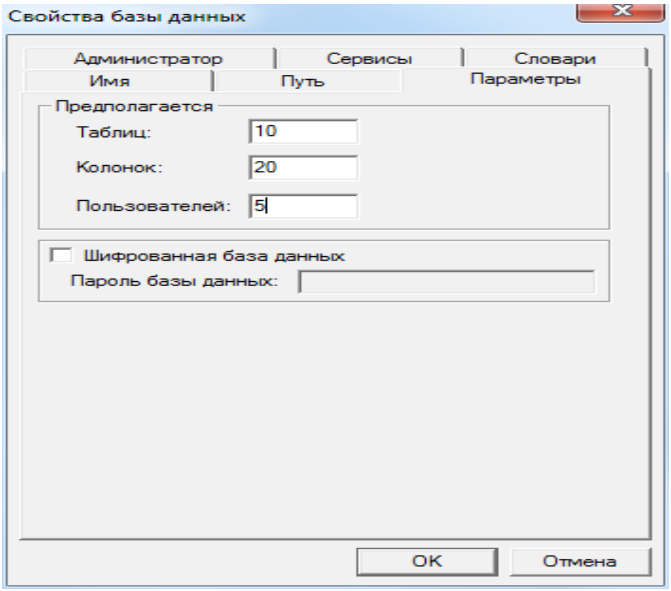


Рисунок 5.3 — Вкладка «Параметры»

Для назначения администратора создаваемой БД на вкладке Администратор были введены регистрационные данные администратора БД в поля "Имя", "Пароль" и "Проверка" (а именно TEAM039 и slave039). Для подтверждения пароля он был повторён в поле "Проверка".

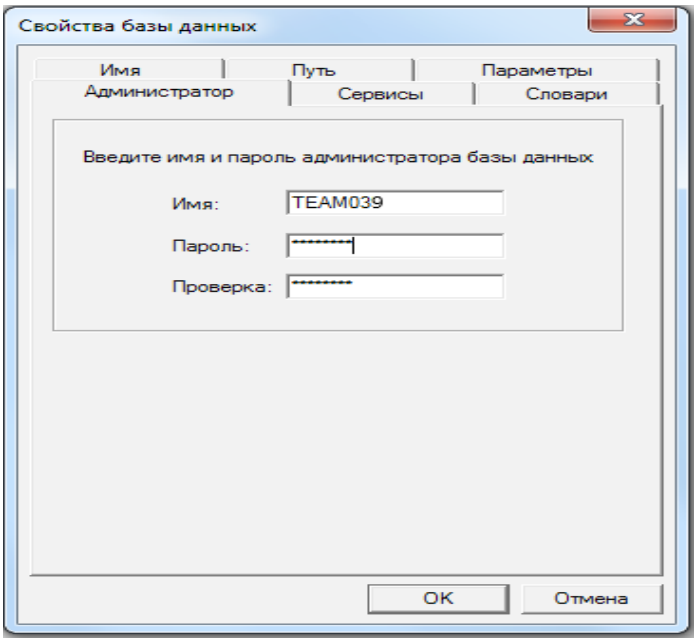


Рисунок 5.4 — Вкладка «Администратор»

На вкладке «Сервисы» значения параметров оставлены без изменения.

Для настройки дополнительных возможностей на вкладке «Словари» были установлены флажки для тех возможностей, которые должны быть включены в создаваемую БД в соответствии с рисунком 5.5.

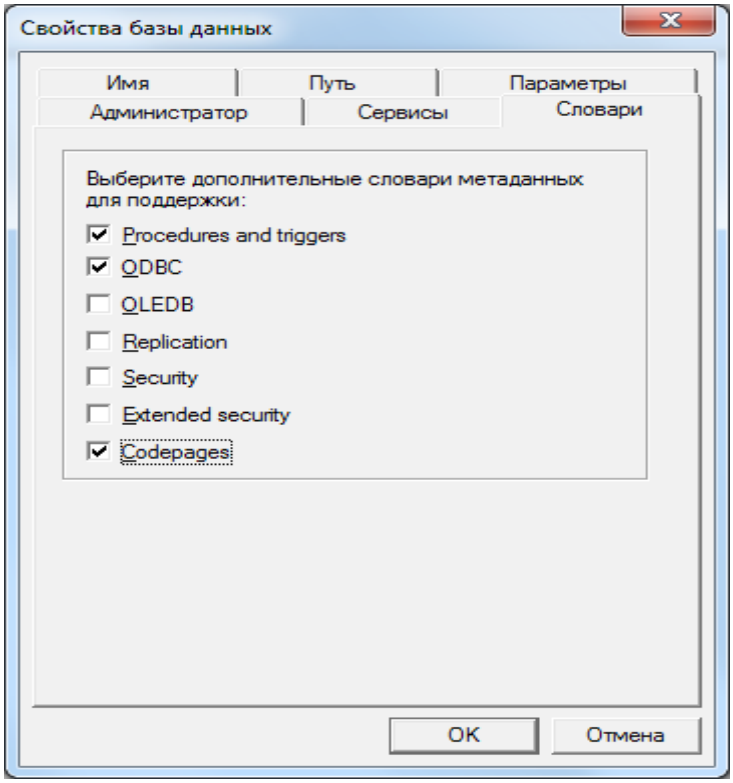


Рисунок 5.5 — Вкладка «Словари»

После нажатия на кнопку OK в окне Свойства базы данных, приложение "Администратор СУБД ЛИНТЕР" сгенерировало необходимый SQL-код для создания базы данных с теми свойствами, которые указаны в этом диалоговом окне и передало его СУБД для выполнения. После этого появилось диалоговое окно "Кодовая страница (Novikov)", в котором для операционной системы Windows были выбраны предлагаемые кодировки на рисунке 5.6.

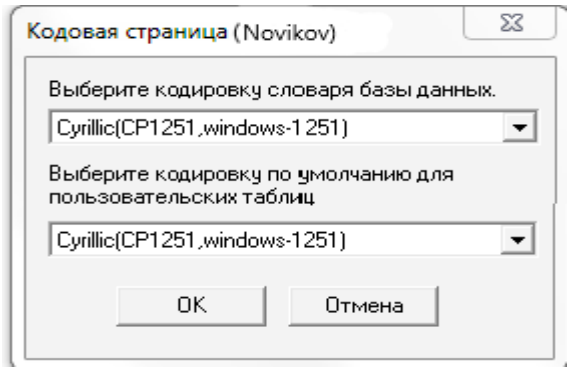


Рисунок 5.6 - Окно "Кодовая страница"

После этого было выведено диалоговое окно "Кодовая страница (Novikov)", в котором можно было выбрать кодировки словаря базы данных и пользовательских таблиц.

В главном окне приложения "Администратор СУБД ЛИНТЕР" в дереве баз данных появился узел с именем Novikov(рисунок 5.7).

Рисунок 5.7 - Дерево баз данных на сервере sqledu03 с новой базой данных

Таким образом была создана база данных Novikov для пользователя с именем TEAM039 и паролем slave039.

Чтобы подключиться к созданной базе данных, была выбрана база данных Novikov и выполнена команда меню База > Запустить.

Открылось окно "Запустить базу" (рисунок 5.8), в котором был установлен флажок "Запустить сетевой драйвер" и затем нажата кнопка ОК.

Затем было выведено диалоговое окно "Проверка доступа", в котором надо было ввести «Имя пользователя» и «Пароль» (рисунок 5.9).

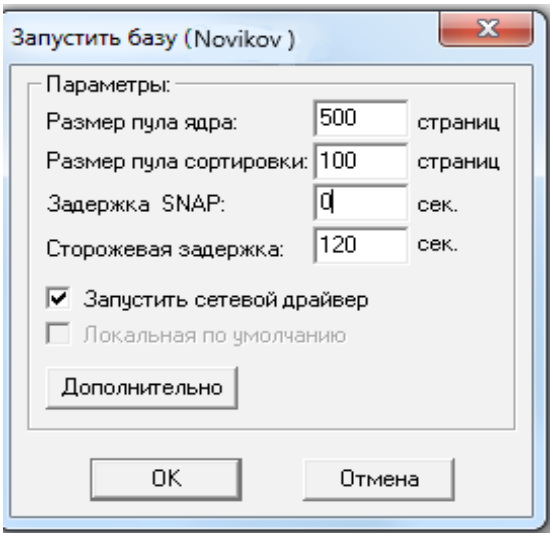


Рисунок 5.8 — Запуск базы данных

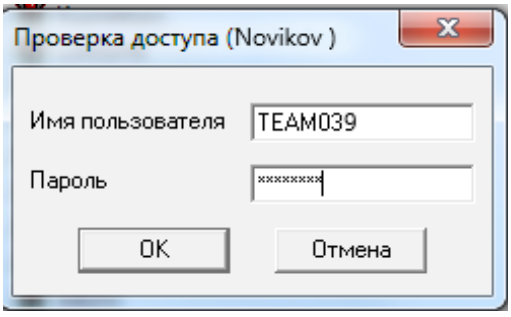


Рисунок 5.9 — Проверка доступа

После этого пиктограмма "Старт/стоп сервер" окне "Администратор СУБД ЛИНТЕР" стала находиться в активном состоянии.

Таким образом, было выполнено подключение к базе данных Novikov для пользователя с именем TEAM039 и паролем slave039.

6 Создание таблиц базы данных

Для создания таблиц базы данных, используя пиктограмму [Рабочий стол] на панели инструментов, запущено на выполнение приложение "Рабочий стол Линтера", и в открывшемся окне "Присоединение к Линтеру" (рисунок 6.1) введены имя пользователя «TEAM039», пароль «slave039» и из выпадающего списка «Сервер» выбрана база данных Novikov.

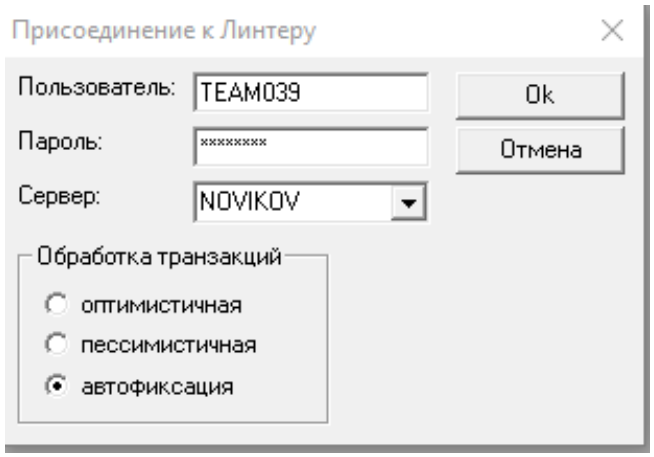


Рисунок 6.1 – Присоединение к Линтеру

Затем выполнена команда Запрос > Создать, после чего выделена строка «Простой запрос» и нажата кнопка ОК (рисунок 6.2).

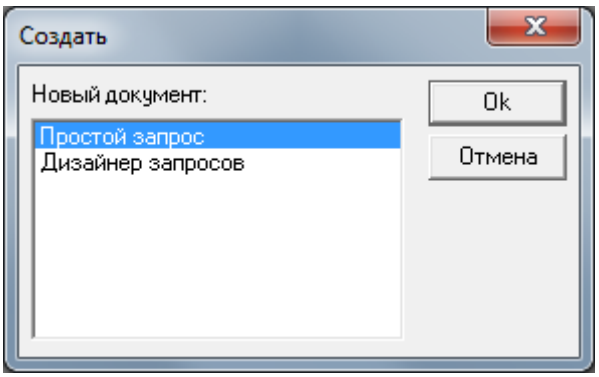


Рисунок 6.2 – Создание запроса

В результате было открыто окно "Запрос1"(рисунок 6.3). В верхнем разделе этого окна был введен скрипт для создания таблицы «KLIENT» (рисунок 6.4), «AVTO» (рисунок 6.5), «EKSPED» (рисунок 6.6) и «ZAMENA» (рисунок 6.7), так же был введен скрипт для создание индекса (рисунок 6.8).

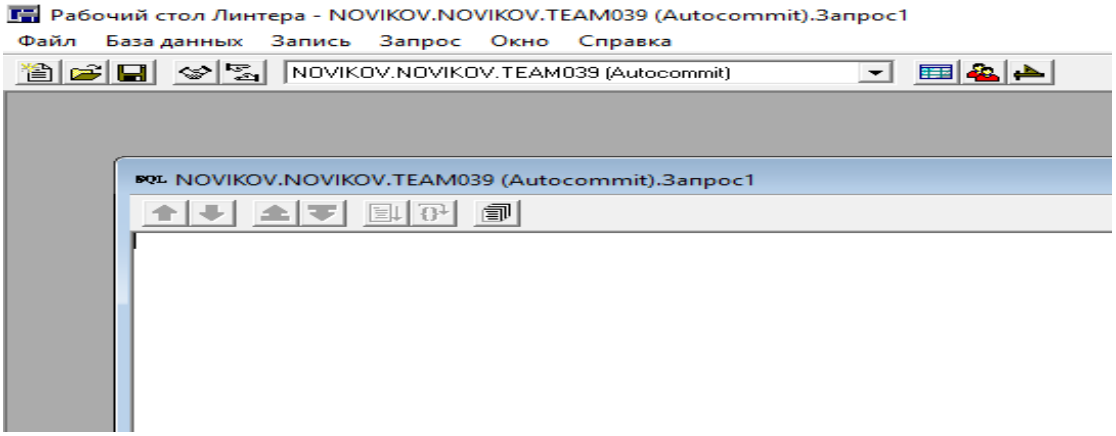


Рисунок 6.3 – Поле ввода запроса

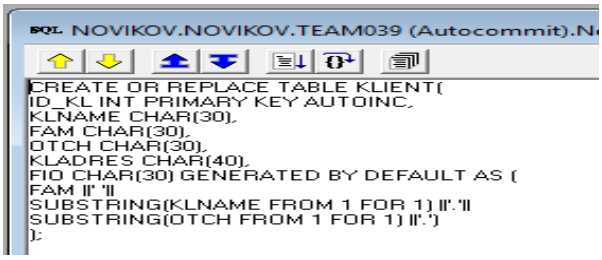


Рисунок 6.4 – Текст скрипта таблицы KLIENT

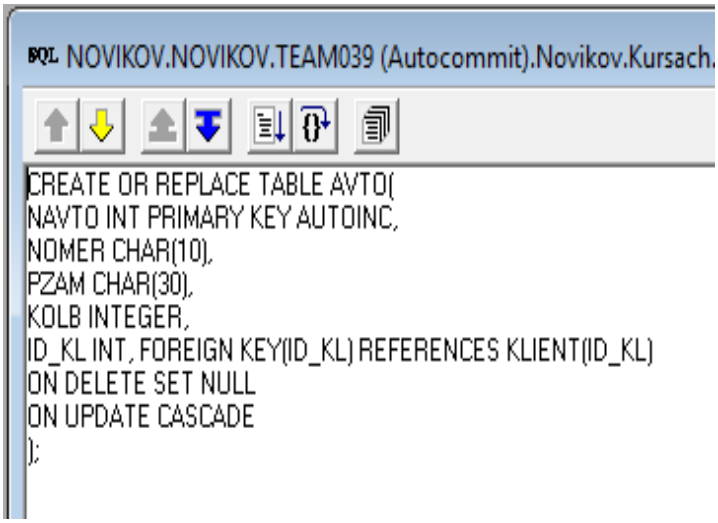


Рисунок 6.5 – Текст скрипта таблицы AVTO

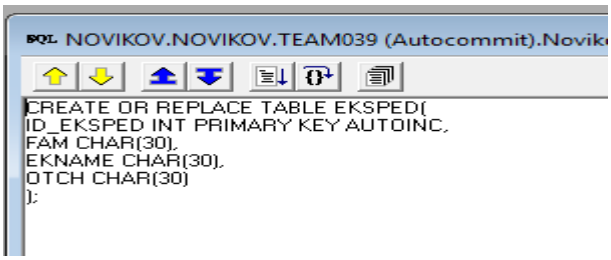


Рисунок 6.6 – Текст скрипта таблицы EKSPED

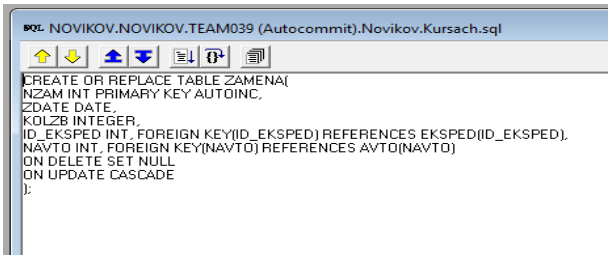


Рисунок 6.7 – Текст скрипта таблицы ZAMENA

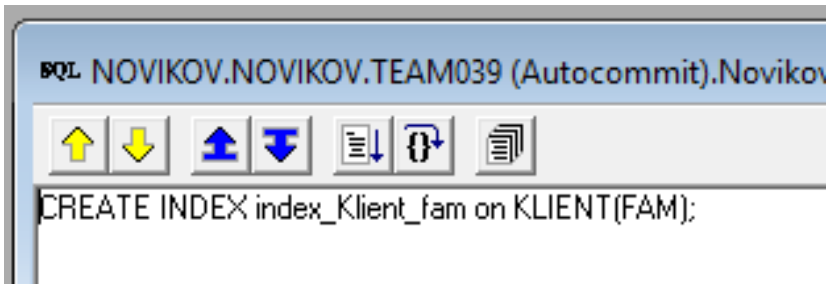
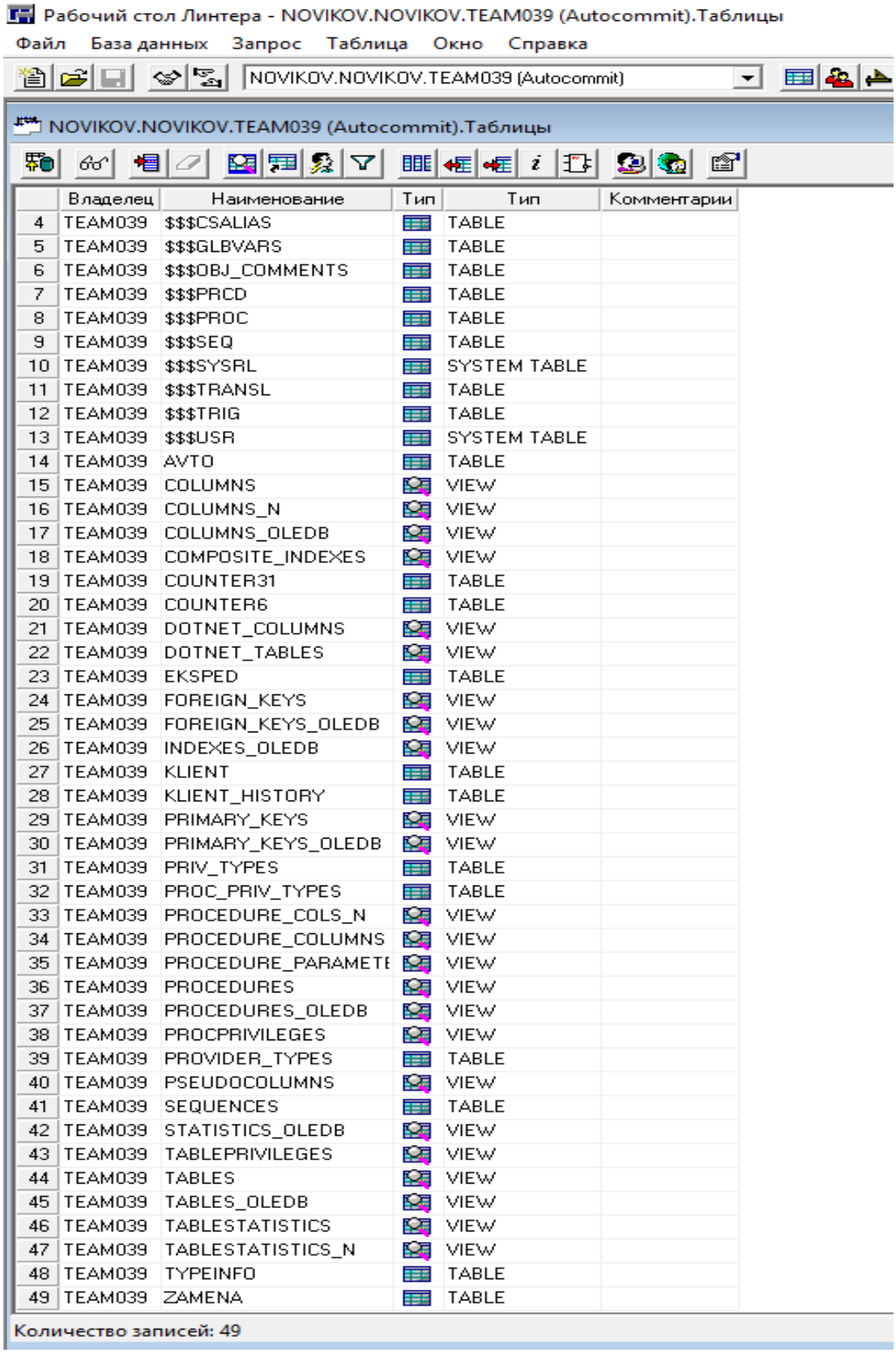


Рисунок 6.8 – Текст скрипта создания индекса

В таблицах созданы первичные ключи. Была выполнена команда «Выполнить текущий» для каждого скрипта. Чтобы убедиться, что таблицы были созданы была выполнена команда База данных>Таблицы, в результате чего было открыто окно "Таблицы" и выведен список всех таблиц базы данных Novikov, в котором присутствуют строки с именами созданных таблиц (рисунок 6.9).



	Владелец	Наименование	Тип	Тип	Комментарии
4	TEAM039	$$$CSALIAS		TABLE	
5	TEAM039	$$$GLBVARS		TABLE	
6	TEAM039	$$$OBJ_COMMENTS		TABLE	
7	TEAM039	$$$PRCD		TABLE	
8	TEAM039	$$$PROC		TABLE	
9	TEAM039	$$$SEQ		TABLE	
10	TEAM039	$$$SYSRL		SYSTEM TABLE	
11	TEAM039	$$$TRANSL		TABLE	
12	TEAM039	$$$TRIG		TABLE	
13	TEAM039	$$$USR		SYSTEM TABLE	
14	TEAM039	AVTO		TABLE	
15	TEAM039	COLUMNS		VIEW	
16	TEAM039	COLUMNS_N		VIEW	
17	TEAM039	COLUMNS_OLEDB		VIEW	
18	TEAM039	COMPOSITE_INDEXES		VIEW	
19	TEAM039	COUNTER31		TABLE	
20	TEAM039	COUNTER6		TABLE	
21	TEAM039	DOTNET_COLUMNS		VIEW	
22	TEAM039	DOTNET_TABLES		VIEW	
23	TEAM039	EKSPED		TABLE	
24	TEAM039	FOREIGN_KEYS		VIEW	
25	TEAM039	FOREIGN_KEYS_OLEDB		VIEW	
26	TEAM039	INDEXES_OLEDB		VIEW	
27	TEAM039	KLIENT		TABLE	
28	TEAM039	KLIENT_HISTORY		TABLE	
29	TEAM039	PRIMARY_KEYS		VIEW	
30	TEAM039	PRIMARY_KEYS_OLEDB		VIEW	
31	TEAM039	PRIV_TYPES		TABLE	
32	TEAM039	PROC_PRIV_TYPES		TABLE	
33	TEAM039	PROCEDURE_COLS_N		VIEW	
34	TEAM039	PROCEDURE_COLUMNS		VIEW	
35	TEAM039	PROCEDURE_PARAMETE		VIEW	
36	TEAM039	PROCEDURES		VIEW	
37	TEAM039	PROCEDURES_OLEDB		VIEW	
38	TEAM039	PROCPRIVILEGES		VIEW	
39	TEAM039	PROVIDER_TYPES		TABLE	
40	TEAM039	PSEUDOCOLUMNS		VIEW	
41	TEAM039	SEQUENCES		TABLE	
42	TEAM039	STATISTICS_OLEDB		VIEW	
43	TEAM039	TABLEPRIVILEGES		VIEW	
44	TEAM039	TABLES		VIEW	
45	TEAM039	TABLES_OLEDB		VIEW	
46	TEAM039	TABLESTATISTICS		VIEW	
47	TEAM039	TABLESTATISTICS_N		VIEW	
48	TEAM039	TYPEINFO		TABLE	
49	TEAM039	ZAMENA		TABLE	

Рисунок 6.9 – Созданные таблицы

Далее, чтобы увидеть какие поля были созданы в таблицах «KLIENT», «EKSPED», «ZAMENA» и «AVTO» в списке таблиц была выделена строка с именем «KLIENT», а затем открыто окно "Столбцы", в котором находится список с параметрами всех созданных столбцов (рисунок 6.10). Эти действия были выполнены и для таблиц «EKSPED», «ZAMENA» и «AVTO» (рисунки 6.11-6.13).

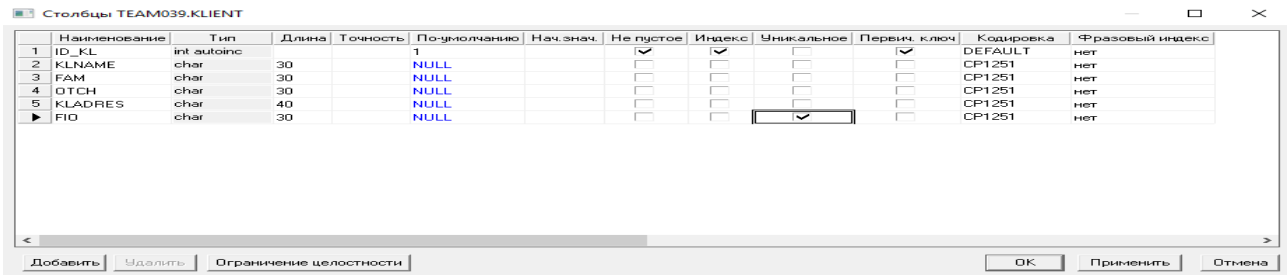


Рисунок 6.10 – Таблица KLIENT

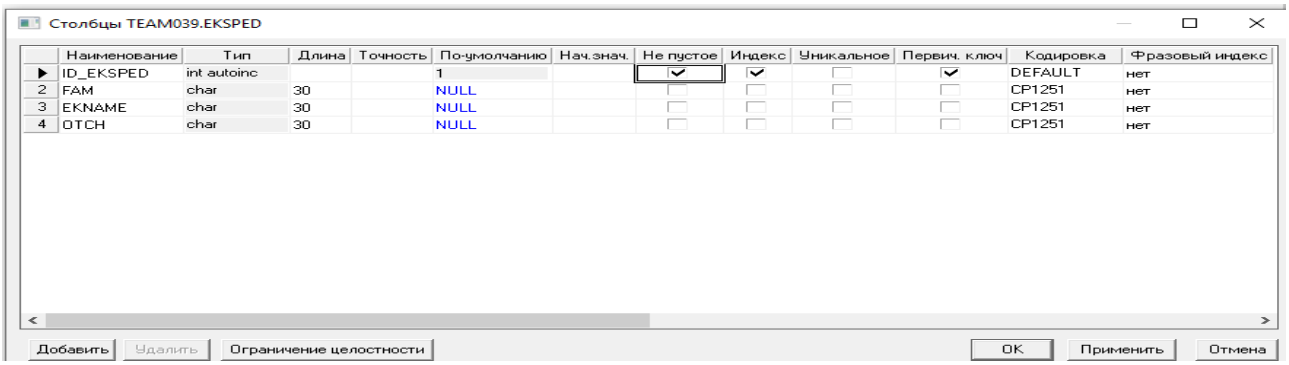


Рисунок 6.11 – Таблица EKSPED

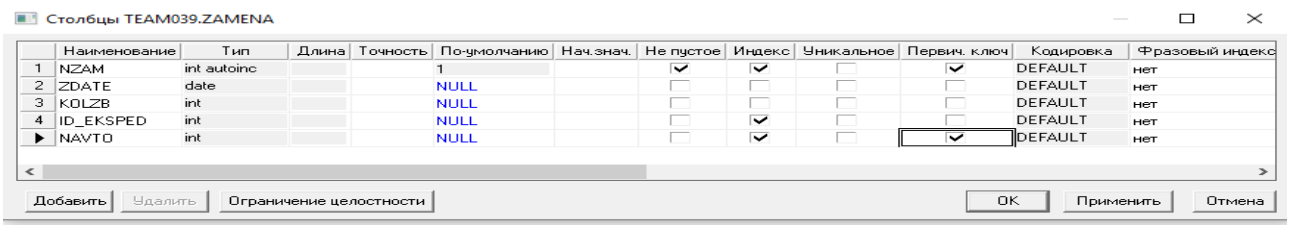


Рисунок 6.12 – Таблица ZAMENA

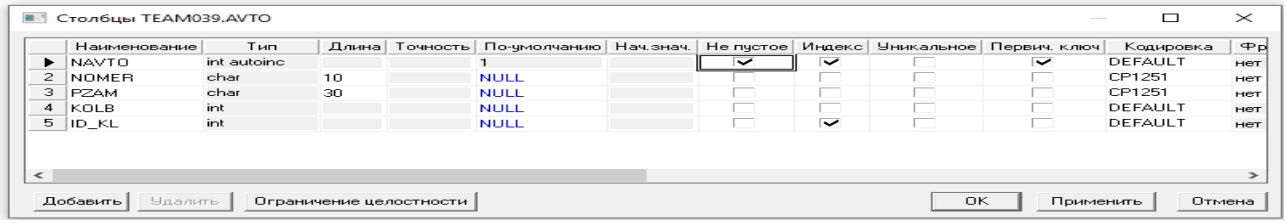


Рисунок 6.13 – Таблица AVTO

Таким образом, были созданы таблицы KLIENt, EKSPED AVTO и ZAMENA базы данных Novikov.

7 Создание ограничений целостности

Для создания ограничений целостности данных была выполнена команда Запрос > Создать (рисунок 7.1), после чего была выделена строка «Простой запрос» и нажата кнопка ОК.

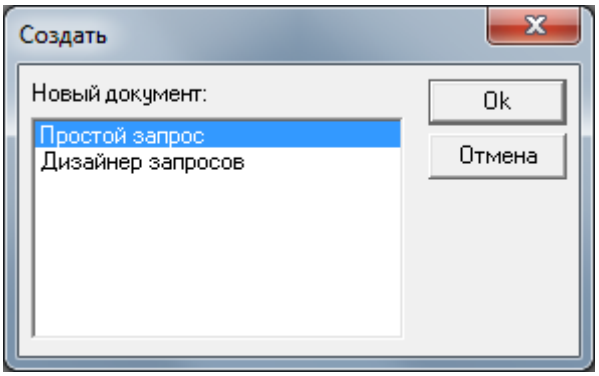


Рисунок 7.1 – Создание запроса

В результате было открыто окно "Запрос1". В верхнем разделе этого окна (рисунок 7.2) был введен скрипт для создания нового ограничения, создающего связь между таблицами KLIENT и AVTO с помощью внешнего ключа id_kl таблицы KLIENT.

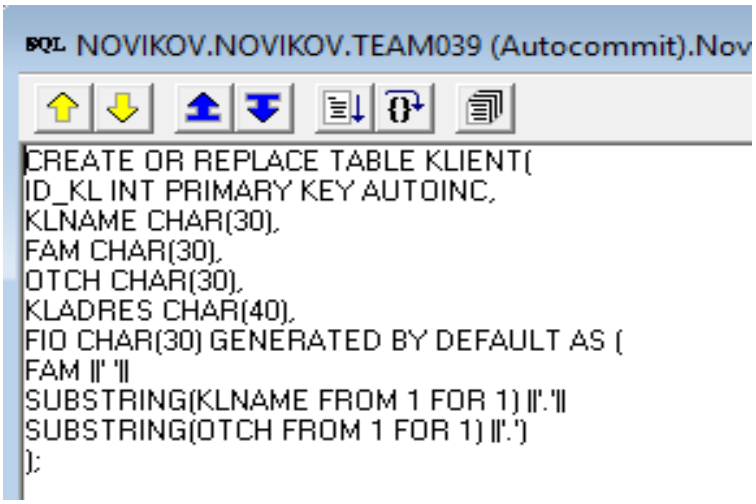


Рисунок 7.2 – Поле ввода запроса для связи таблиц KLIENT и AVTO

Была нажата кнопка "Таблицы" панели инструментов "Рабочего стола Линтера". В открывшемся окне "Таблицы" выбрана строка с названием таблицы AVTO и выполнена команду главного меню "Рабочего стола Линтера" Таблица > Свойства > Ссылки на. В открывшемся окне "Ссылки на другие таблицы" . Эти действия были проведены для таблицы AVTO (рисунок 7.3).

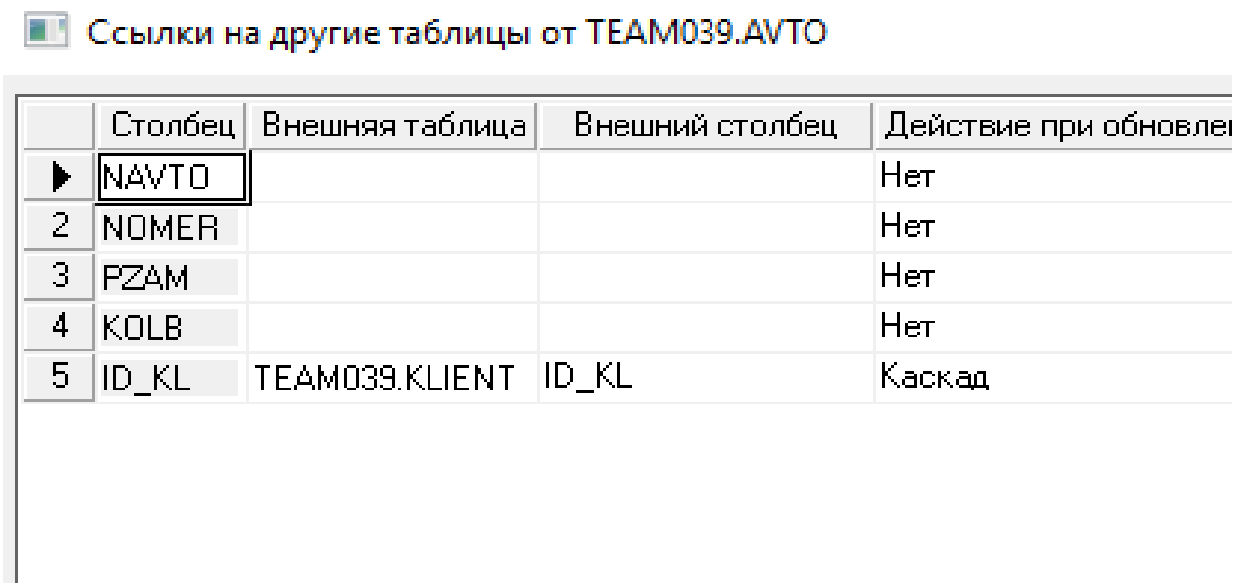


Рисунок 7.3 – Внешний ключ таблицы AVTO

После этого был введен скрипт для создания нового ограничения (рисунок 7.4), создающего связь между таблицами EKSPED, ZAMENA и AVTO с помощью внешнего ключа id_eksped таблицы EKSPED и внешнего ключа navto таблица AVTO.

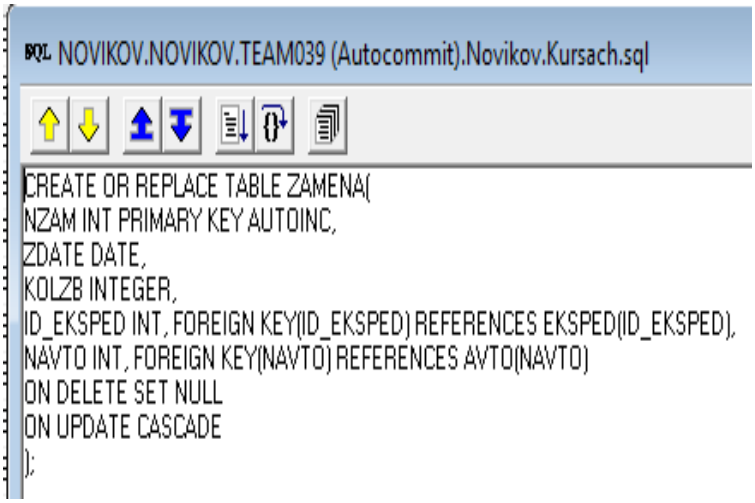


Рисунок 7.4 – Поле ввода запроса для связи таблиц EKSPED,ZAMENA и AVTO

Была нажата кнопка "Таблицы" панели инструментов "Рабочего стола Линтера". В открывшемся окне "Таблицы" выбрана строка с названием таблицы ZAMENA и выполнена команду главного меню "Рабочего стола Линтера" Таблица > Свойства > Ссылки на. В открывшемся окне "Ссылки на другие таблицы" . Эти действия были проведены для таблицы ZAMENA (рисунок 7.5).

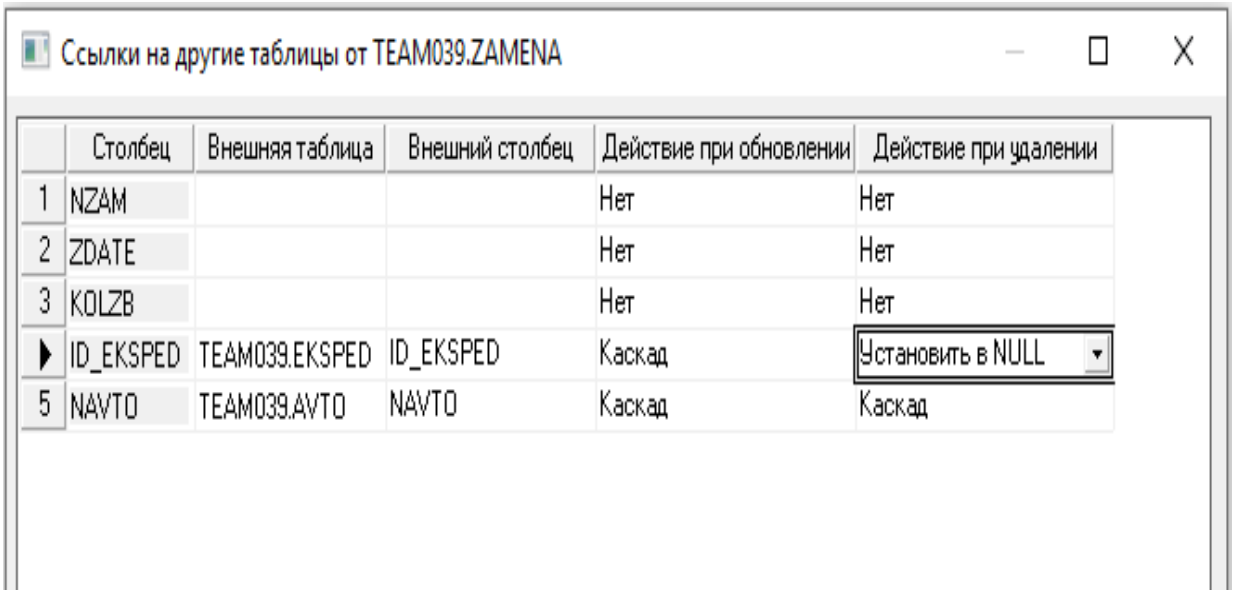


Рисунок 7.5 – Внешние ключи таблицы ZAMENA

После этого в верхнем разделе окна «Запрос» был введен скрипт для создания нового ограничения NOT NULL, приведенный на рисунке 7.6.

Рисунок 7.6 – Текст скрипта добавления ограничения NOT NULL

Далее был закрыт редактор скриптов и нажата кнопка "Таблицы" панели инструментов "Рабочего стола Линтера". В открывшемся окне "Таблицы" выбрана таблица ZAMENA. В открывшемся окне "Столбцы" (рисунок 7.7), проверена правильность выполнения последнего скрипта. В строке "KOLZB" установлен флажок в столбце "Не пустое".Для проверки составного первичного ключа в открывшемся окне "Столбцы" (рисунок 7.7), проверена правильность установки первичного ключа. В строках "NZAM" и "NAVTO" установлены флажки в столбце "Первичный ключ".

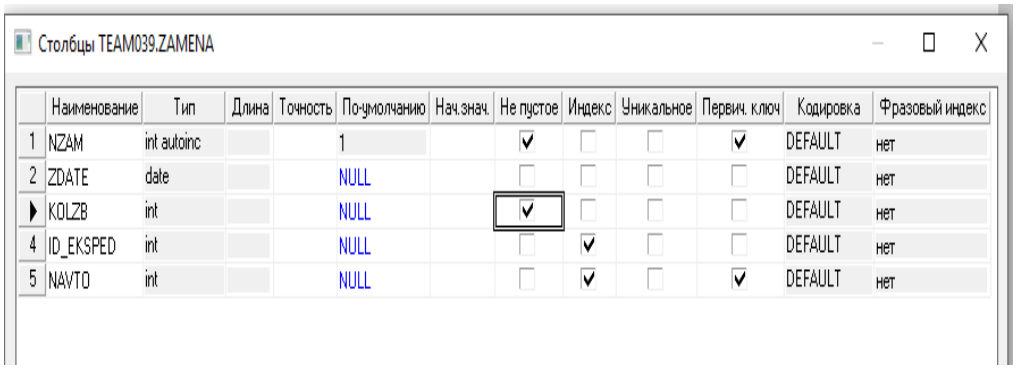


Рисунок 7.7– Проверка ограничения NOT NULLи составного первичного ключа

Затем в верхнем разделе окна «Запрос» был введен скрипт для создания нового ограничения UNIQUE для столбца FIO таблицы KLIENT, приведенный на рисунке 7.8.

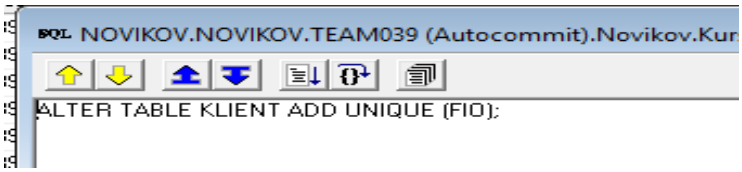
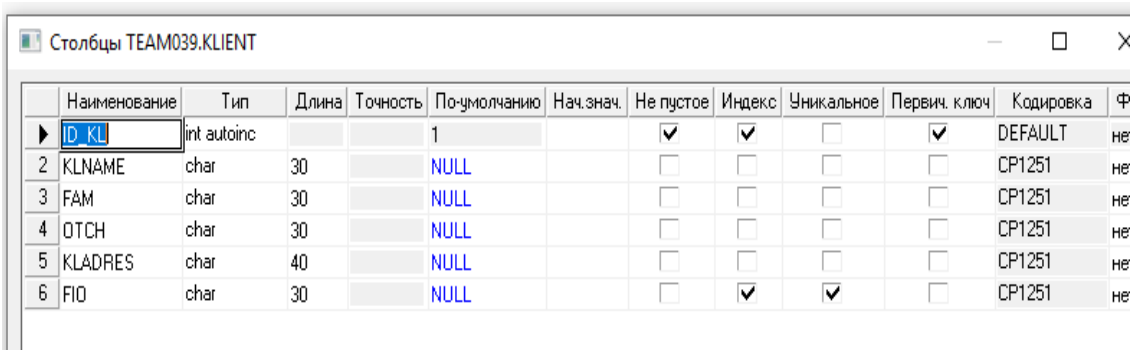


Рисунок 7.8 – Текст скрипта добавления ограничения UNIQUE

Далее закрыли редактор скриптов и нажали мышкой на кнопке "Таблицы" панели инструментов "Рабочего стола Линтера". В открывшемся окне "Таблицы" выбрали таблицу KLIENT. В открывшемся окне "Столбцы", проверили правильность выполнения последнего скрипта. В строке "FIO" установлен флажок в столбце "Уникальное" (рисунок 7.9).



	Наименование	Тип	Длина	Точность	По-умолчанию	Нач.знач.	Не пустое	Индекс	Уникальное	Первич. ключ	Кодировка	Ф
▶	ID_KL	int autoinc			1		☑	☑	☐	☑	DEFAULT	не
2	KLNAME	char	30		NULL		☐	☐	☐	☐	CP1251	не
3	FAM	char	30		NULL		☐	☐	☐	☐	CP1251	не
4	OTCH	char	30		NULL		☐	☐	☐	☐	CP1251	не
5	KLADRES	char	40		NULL		☐	☐	☐	☐	CP1251	не
6	FIO	char	30		NULL		☐	☑	☑	☐	CP1251	не

Рисунок 7.9 - Проверка добавления ограничения UNIQUE

Также в верхнем разделе окна «Запрос» был введен скрипт для создания нового ограничения CHECK для столбца kolb таблицы AVTO, приведенный на рисунке 7.10.

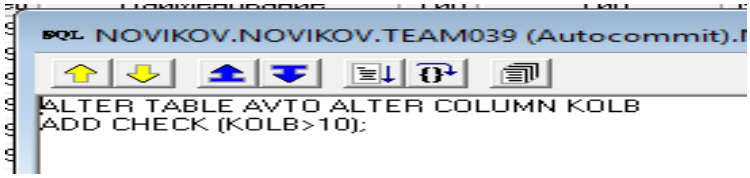


Рисунок 7.10 – Текст скрипта добавления ограничения CHECK

Таким образом были разработаны ограничения целостности базы данных.

8 Создание хранимых процедур и триггеров

Для создания генератора была выполнена команда Запрос > Создать (рисунок 6.3). Далее был введен текст скрипта для создания таблицы «SEQUENCES», используемой для определения очередного значения индекса записи в таблицах (рисунок 8.1).

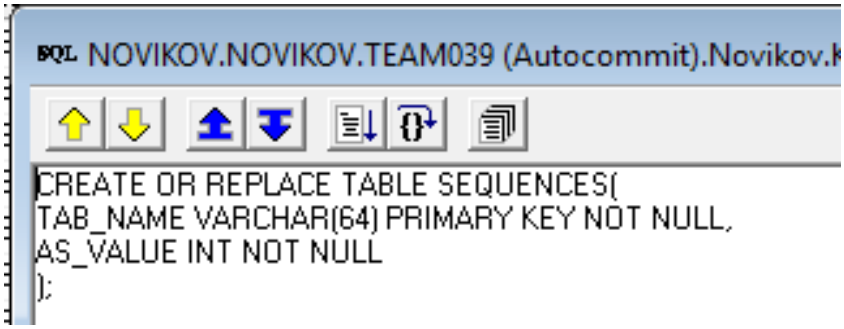


Рисунок 8.1 — Текст скрипта создания таблицы «SEQUENCES»

Здесь колонка TAB_NAME хранит имена таблиц, для которых генерируются уникальные значения, а колонка CUR_VALUE – текущее значение для генерации.

Далее была создана хранимая процедура для генерации первичного ключа в таблицах (рисунок 8.2).

В процедуре использована блокировка таблицы, чтобы оградить двух одновременно работающих пользователей от возможности модификации одних и тех же данных; надо иметь в виду, что работать это будет только в режиме обработки транзакций, отличном от autocommit.

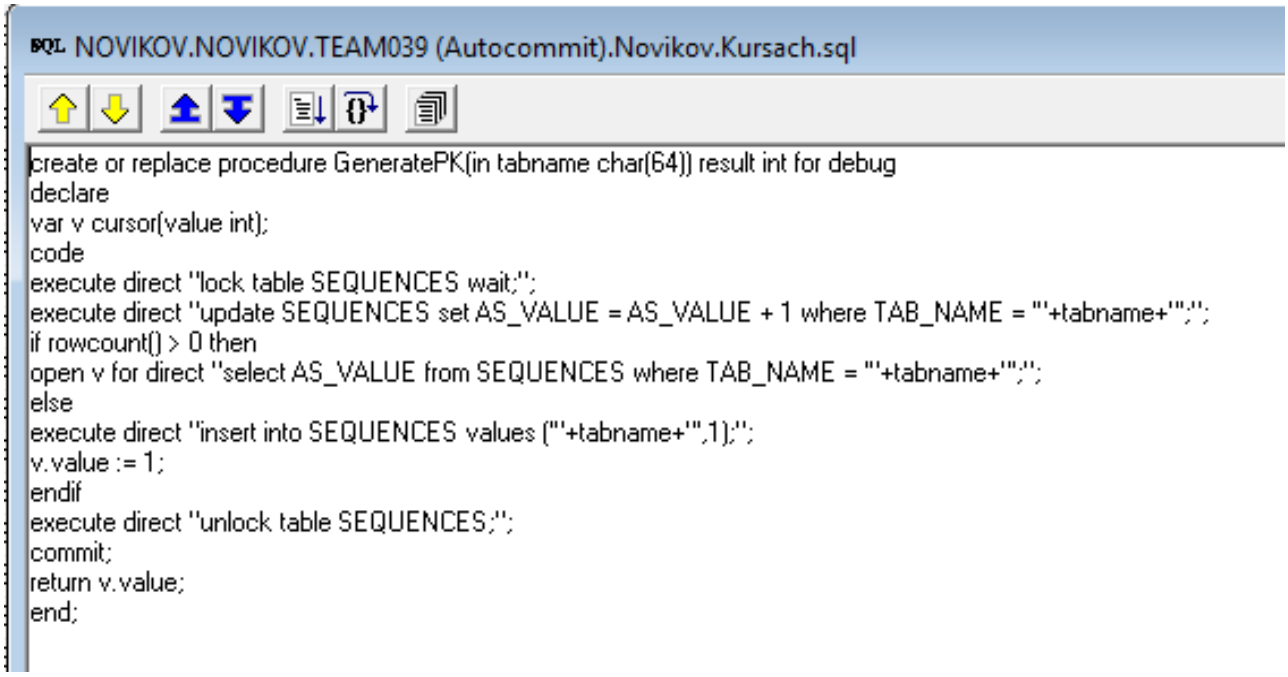


Рисунок 8.2 — Текст SQL-скрипта хранимой процедуры

Для занесения в таблицу значений из генератора, были созданы триггеры для каждой из четырех таблиц (рисунки 8.3-8.6). Триггер позволяет принудительно установить значение поля ID новой записи в таблице, какое бы значение не задал пользователь в исходном запросе.

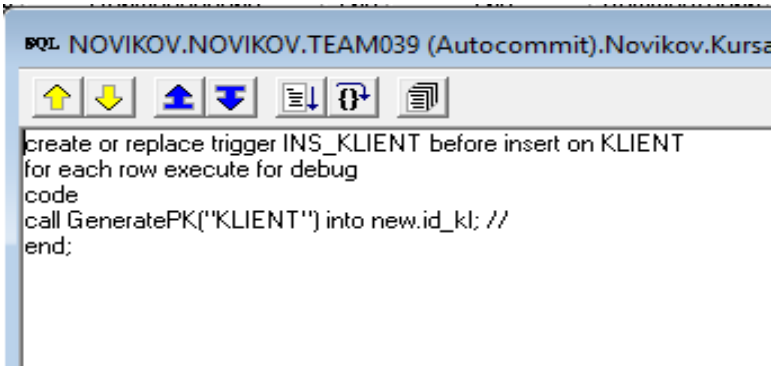


Рисунок 8.3 — Текст SQL-скрипта для создания триггера для таблицы KLIENT

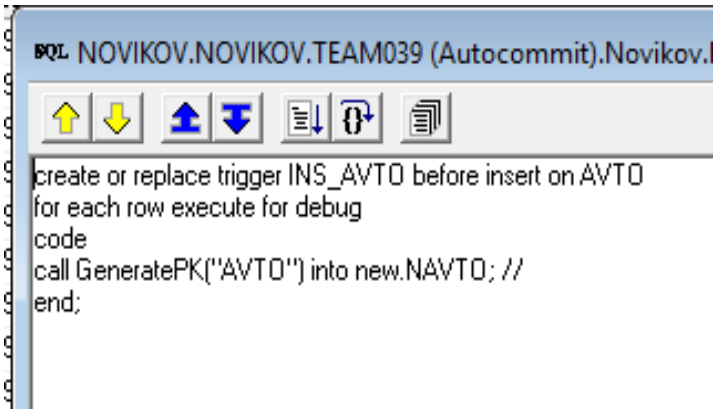


Рисунок 8.4 — Текст SQL-скрипта для создания триггера для таблицы AVTO

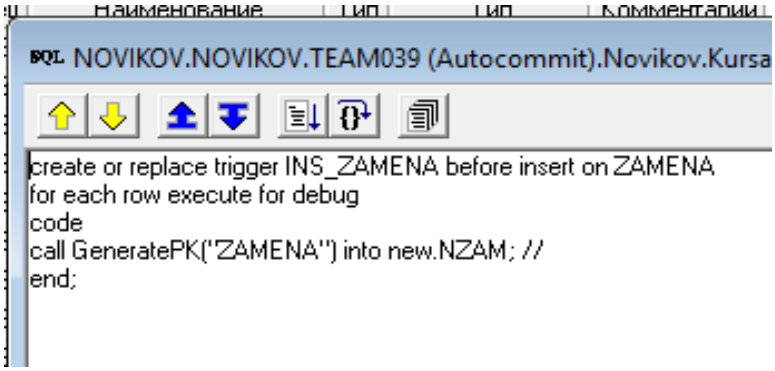


Рисунок 8.5 — Текст SQL-скрипта для создания триггера для таблицы ZAMEN

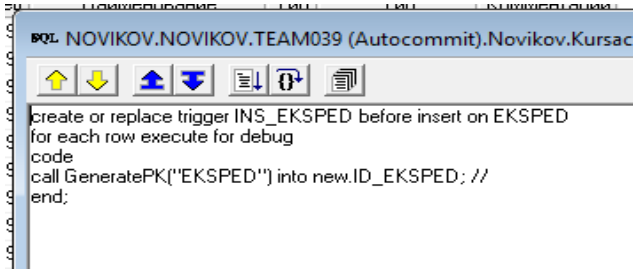


Рисунок 8.6 — Текст SQL-скрипта для создания триггера для таблицы EKSPED

Для хранения изменений в таблице «KLIENT», с помощью SQL-скрипта, была создана таблица «KLIENT_HISTORY» (рисунок 8.7). Таблица «KLIENT_HISTORY» хранит идентификатор и дату записи с любыми изменениями в таблице «KLIENT».

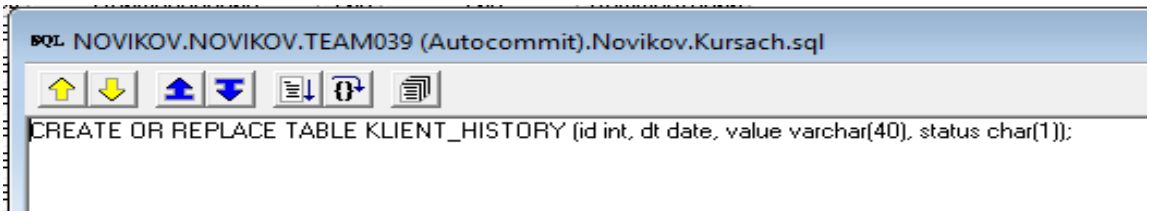


Рисунок 8.7 — Текст SQL-скрипта для создания таблицы «KLIENT_HISTORY»

После её создания, ей был присвоен составной первичный ключ, состоящий из двух полей: идентификатора и даты (рисунок 8.8).

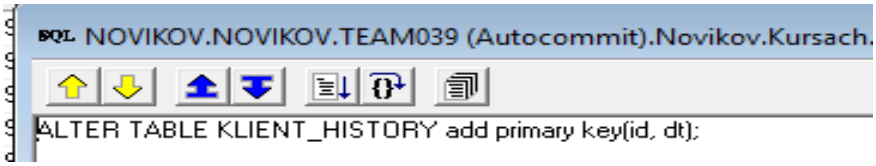


Рисунок 8.8 - Текст SQL-скрипта для присвоения первичного ключа

После завершения работы с таблицей «KLIEMT_HISTORY», для учета всех изменений в таблице «KLIENT» создается триггер «INS_KLIENT_HIST» (рисунок 8.9).

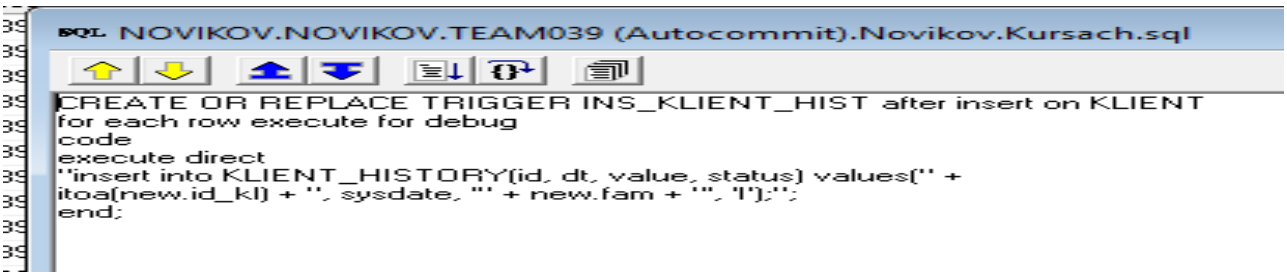
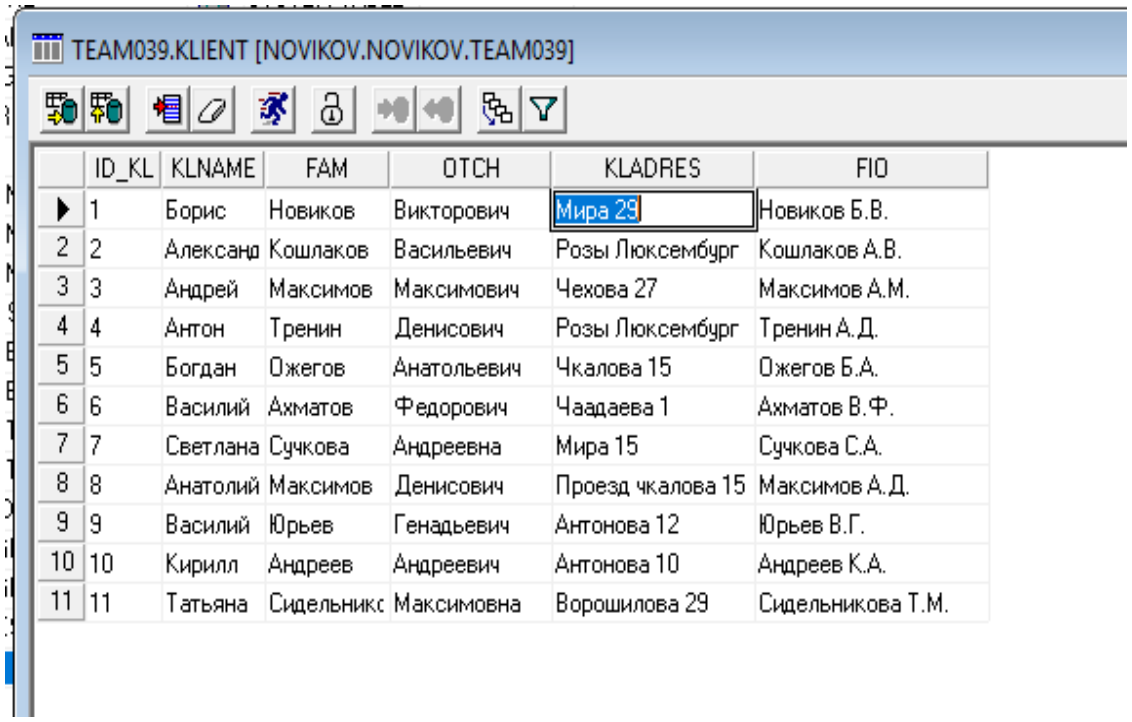


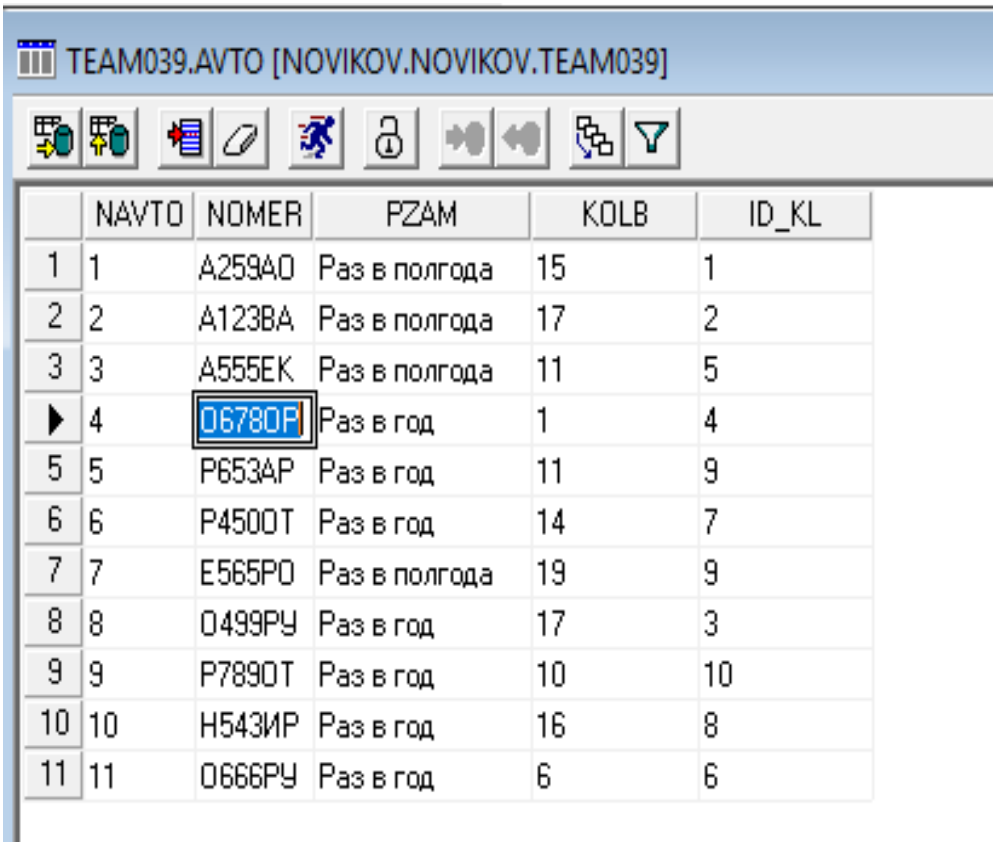
Рисунок 8.9 - Текст SQL-скрипта для создания триггера учета изменений в таблице «KLIENT»

Для проверки работы всех триггеров, таблиц и хранимой процедуры, были внесены записи в таблицы «KLIENT», «AVTO», «ZAMENA» и «EKSPED» с помощью интерфейса «Рабочего стола Линтера» и функции «Таблицы» (рисунок 8.10-8.14).



	ID_KL	KLNAME	FAM	OTCH	KLADRES	FIO
▶	1	Борис	Новиков	Викторович	Мира 29	Новиков Б.В.
2	2	Александ	Кошлаков	Васильевич	Розы Люксембург	Кошлаков А.В.
3	3	Андрей	Максимов	Максимович	Чехова 27	Максимов А.М.
4	4	Антон	Тренин	Денисович	Розы Люксембург	Тренин А.Д.
5	5	Богдан	Ожегов	Анатольевич	Чкалова 15	Ожегов Б.А.
6	6	Василий	Ахматов	Федорович	Чаадаева 1	Ахматов В.Ф.
7	7	Светлана	Сучкова	Андреевна	Мира 15	Сучкова С.А.
8	8	Анатолий	Максимов	Денисович	Проезд чкалова 15	Максимов А.Д.
9	9	Василий	Юрьев	Генадьевич	Антонова 12	Юрьев В.Г.
10	10	Кирилл	Андреев	Андреевич	Антонова 10	Андреев К.А.
11	11	Татьяна	Сидельникс	Максимовна	Ворошилова 29	Сидельникова Т.М.

Рисунок 8.10 — Добавление записи в таблицу «KLIENT»



	NAVTO	NOMER	PZAM	KOLB	ID_KL
1	1	А259АО	Раз в полгода	15	1
2	2	А123ВА	Раз в полгода	17	2
3	3	А555ЕК	Раз в полгода	11	5
▶	4	О678ОР	Раз в год	1	4
5	5	Р653АР	Раз в год	11	9
6	6	Р450ОТ	Раз в год	14	7
7	7	Е565РО	Раз в полгода	19	9
8	8	О499РУ	Раз в год	17	3
9	9	Р789ОТ	Раз в год	10	10
10	10	Н543МР	Раз в год	16	8
11	11	О666РУ	Раз в год	6	6

Рисунок 8.11 — Добавление записи в таблицу «AVTO»

Рисунок 8.12— Добавление записи в таблицу «EKSPED»

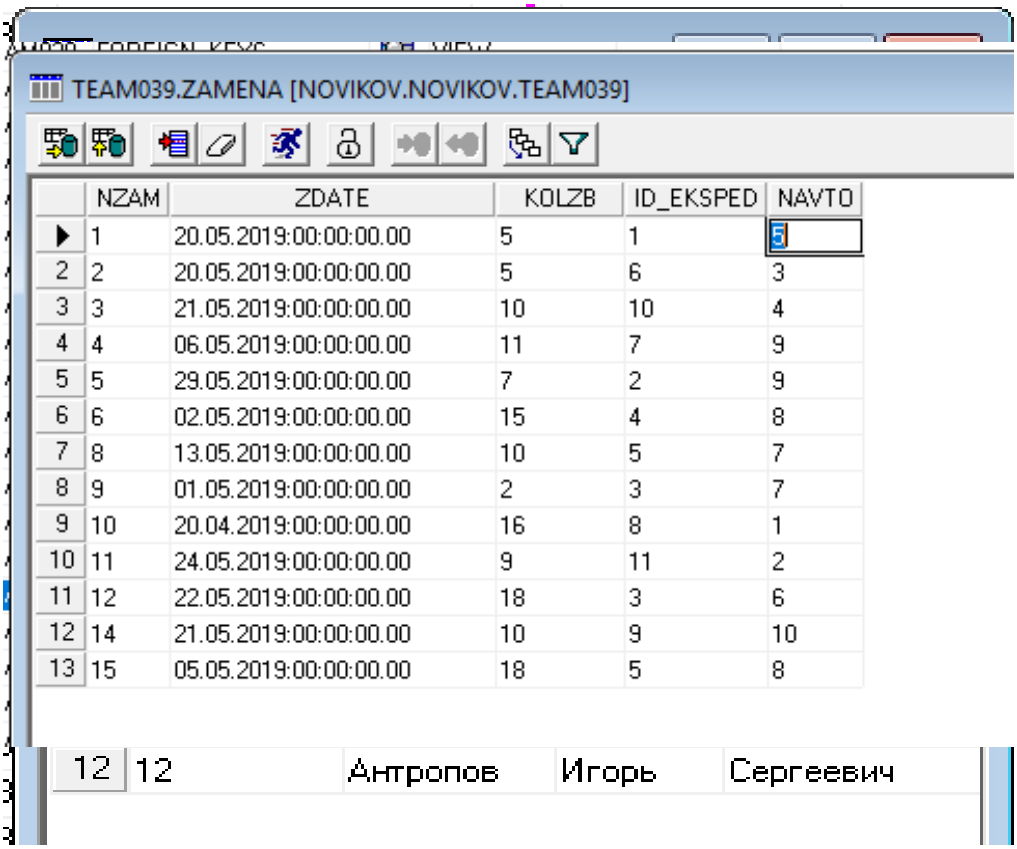


Рисунок 8.13— Добавление записи в таблицу «ZAMENA»

Также была добавлена запись в второстепенную таблицу «KLIENT» с помощью SQL-скрипта и выполнен просмотр с помощью функции «Рабочего стола Линтера» > «Таблицы» (рисунки 8.14-8.15).

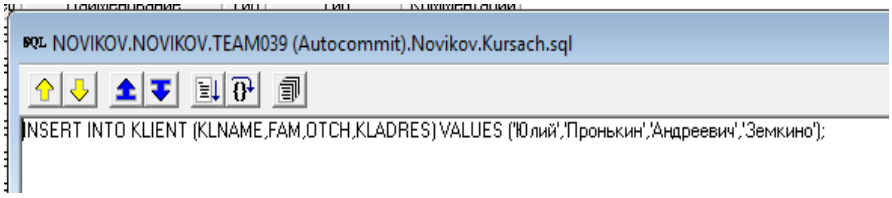
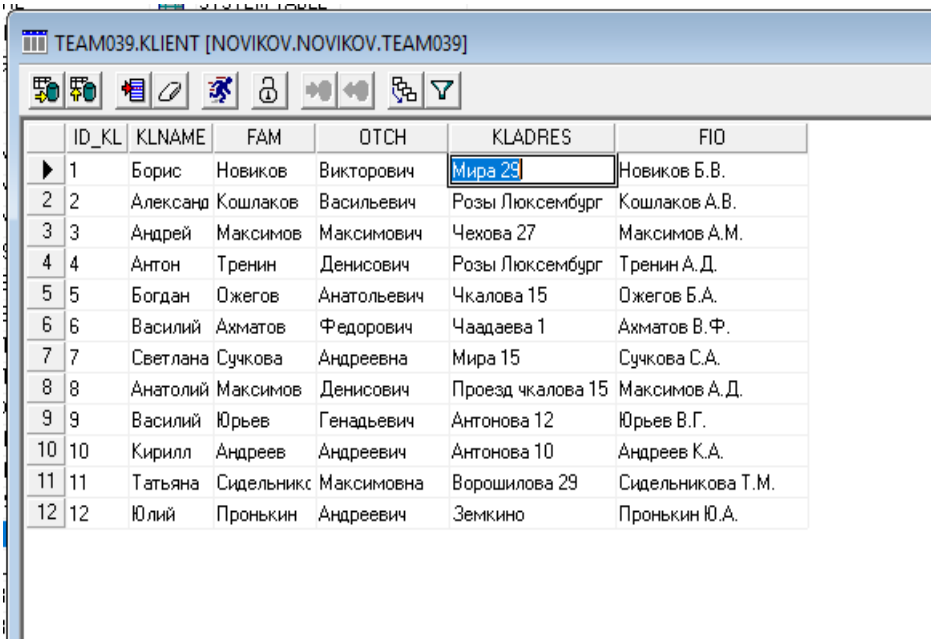


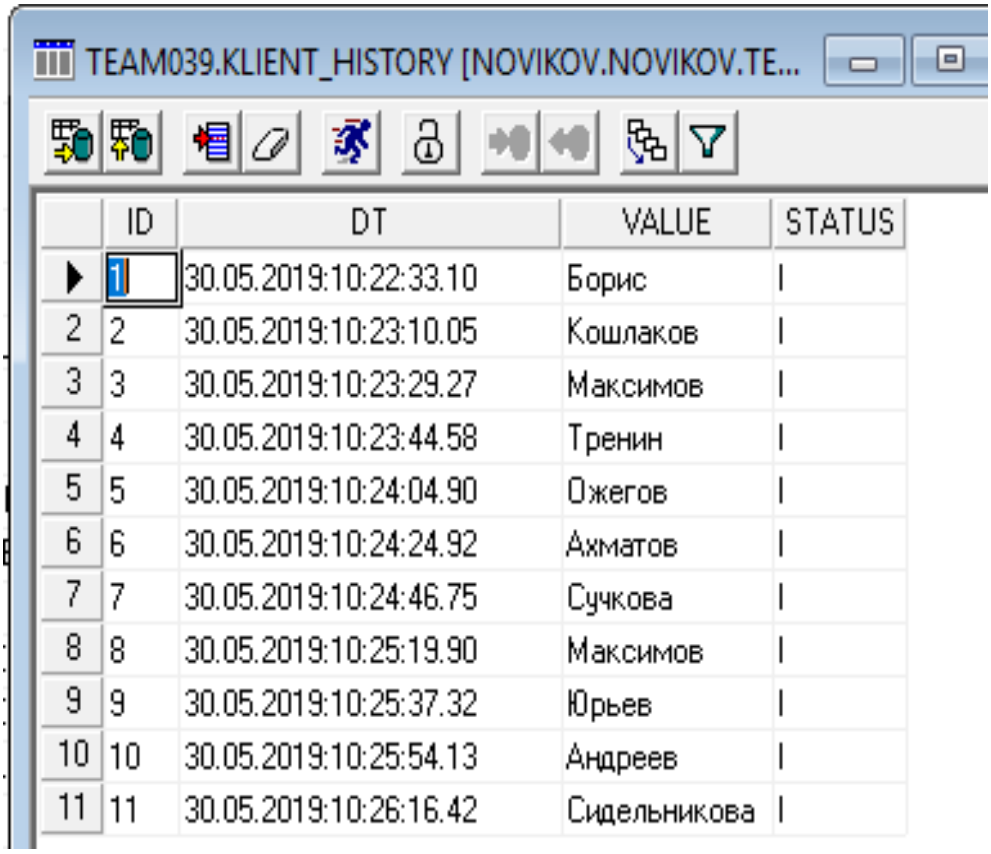
Рисунок 8.14 — Добавление записи в таблицу «KLIENT» с помощью SQL-скрипта



	ID_KL	KLNAME	FAM	OTCH	KLADRES	FIO
▶	1	Борис	Новиков	Викторович	Мира 29	Новиков Б.В.
2	2	Александ	Кошлаков	Васильевич	Розы Люксембург	Кошлаков А.В.
3	3	Андрей	Максимов	Максимович	Чехова 27	Максимов А.М.
4	4	Антон	Тренин	Денисович	Розы Люксембург	Тренин А.Д.
5	5	Богдан	Ожегов	Анатольевич	Чкалова 15	Ожегов Б.А.
6	6	Василий	Ахматов	Федорович	Чаадаева 1	Ахматов В.Ф.
7	7	Светлана	Сучкова	Андреевна	Мира 15	Сучкова С.А.
8	8	Анатолий	Максимов	Денисович	Проезд чкалова 15	Максимов А.Д.
9	9	Василий	Юрьев	Генадьевич	Антонова 12	Юрьев В.Г.
10	10	Кирилл	Андреев	Андреевич	Антонова 10	Андреев К.А.
11	11	Татьяна	Сидельникс	Максимовна	Ворошилова 29	Сидельникова Т.М.
12	12	Юлий	Пронькин	Андреевич	Земкино	Пронькин Ю.А.

Рисунок 8.15 — Просмотр добавленной записи в таблице «KLIENT»

После внесения данных в таблицу «KLIENT» в таблице «KLIENT_HISTORY» была добавлена запись с помощью созданного тригера. Просмотр таблицы «KLIENT_HISTORY» в окне «Таблицы» подтверждает работу созданного триггера и таблицы (рисунок 8.16).



	ID	DT	VALUE	STATUS
▶	1	30.05.2019:10:22:33.10	Борис	I
2	2	30.05.2019:10:23:10.05	Кошлаков	I
3	3	30.05.2019:10:23:29.27	Максимов	I
4	4	30.05.2019:10:23:44.58	Тренин	I
5	5	30.05.2019:10:24:04.90	Ожегов	I
6	6	30.05.2019:10:24:24.92	Ахматов	I
7	7	30.05.2019:10:24:46.75	Сучкова	I
8	8	30.05.2019:10:25:19.90	Максимов	I
9	9	30.05.2019:10:25:37.32	Юрьев	I
10	10	30.05.2019:10:25:54.13	Андреев	I
11	11	30.05.2019:10:26:16.42	Сидельникова	I

Рисунок 8.16 — Просмотр таблицы учета изменений «KLIENT_HISTORY»

После этого была добавлена хранимая процедура «DETALI» (рисунок 8.17) с помощью SQL-скрипта, которая изменяет данные количество деталей, если они больше 20 то устанавливается в 20, если меньше 20 то не изменяется и процедура «PERZAM»(Рисунок 8.19),которая изменяет переодичность замены, если оно не было установлено, в значение «Раз в год» . Результат выполнения этих хранимых процедур представлен на рисунке 8.18 и рисунке 8.20.

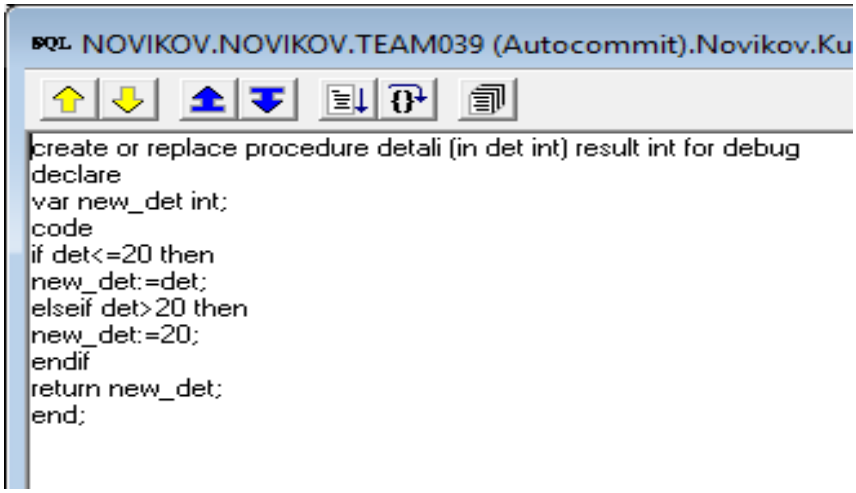


Рисунок 8.17 — Текст SQL-скрипта для создания хранимой процедуры DETALI

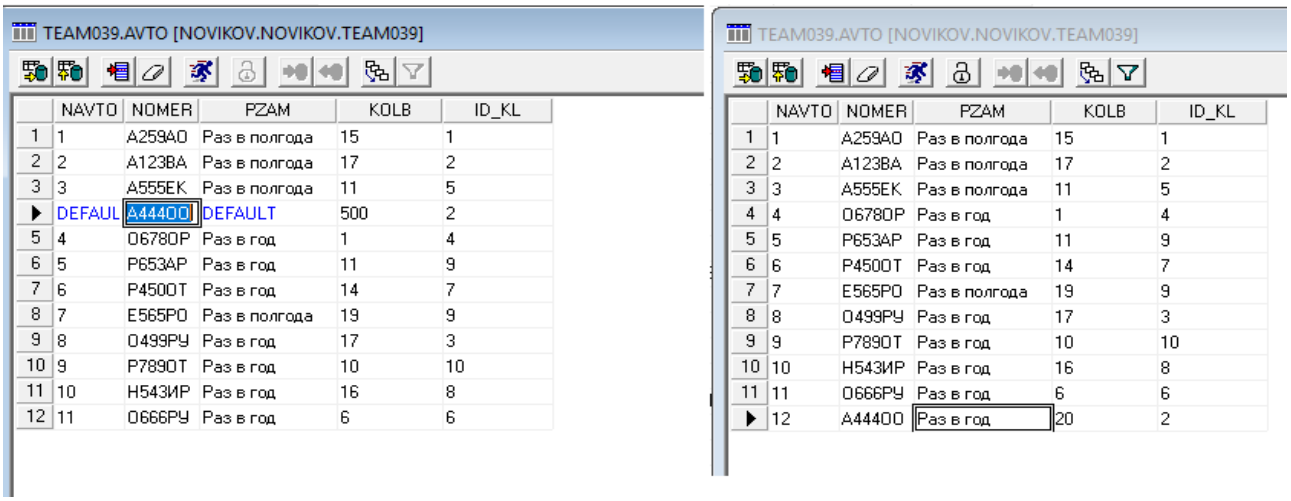


Рисунок 8.18 —Результат выполнения хранимой процедуры DETALI

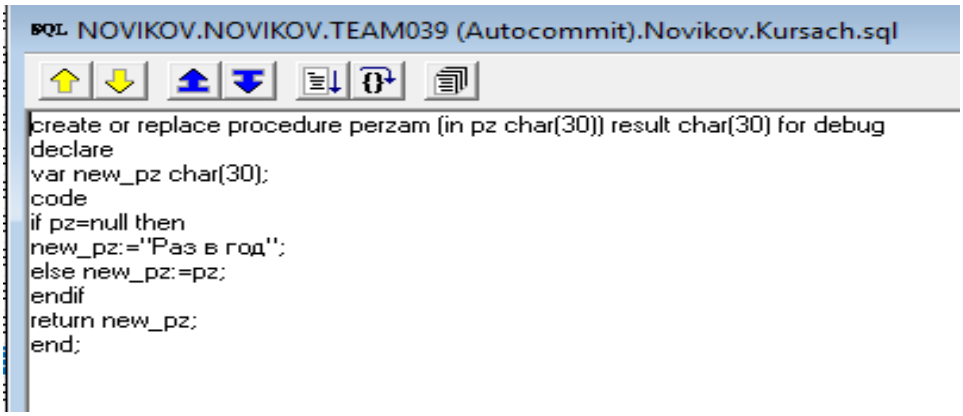


Рисунок 8.19 — Текст SQL-скрипта для создания хранимой процедуры PERZAM

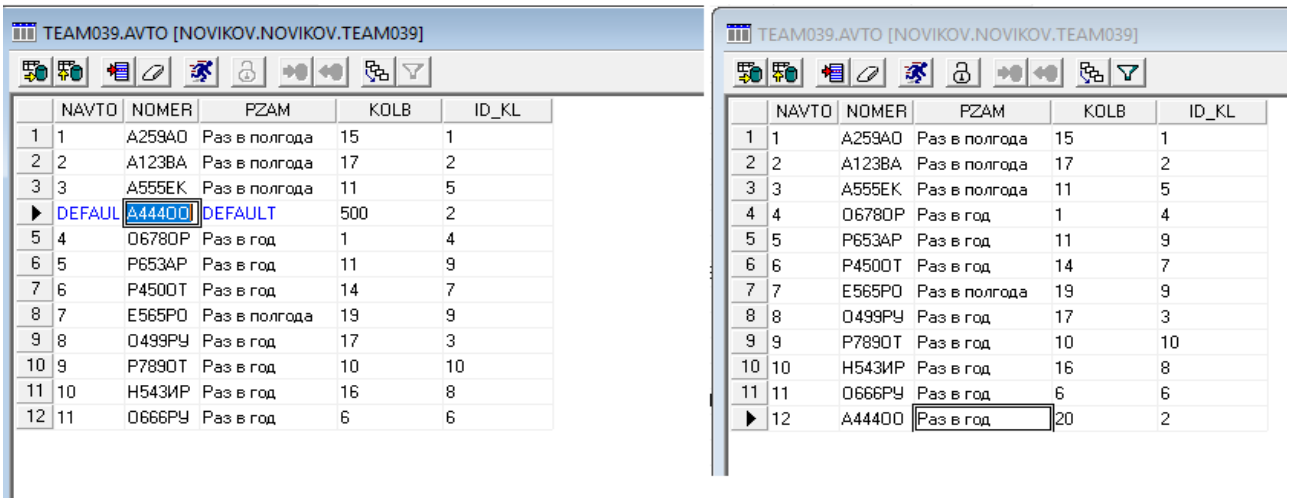


Рисунок 8.20 —Результат выполнения хранимой процедуры PERZAM

Для вызова процедур, были созданы триггеры для каждой из двух процедур (рисунки 8.21-8.22).

Рисунок 8.21 — Текст SQL-скрипта для создания триггера для процедуры perza

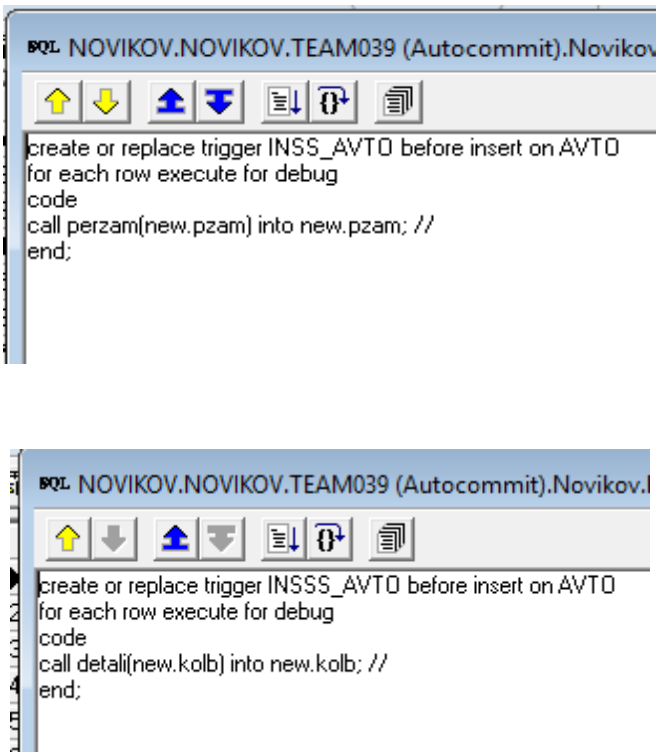


Рисунок 8.22 — Текст SQL-скрипта для создания триггера для процедуры detali

Таким образом, были созданы хранимые процедуры, генераторы и триггеры для таблиц KLIENT, EKSPED, ZAMENA и AVTO.

9 Разработка SQL-запросов

Для создания запросов была выполнена команда меню Запрос > Создать, в затем было открыто диалоговое окно "Создать → Простой запрос". В результате появилось окно редактора запросов (рисунок 9.1).

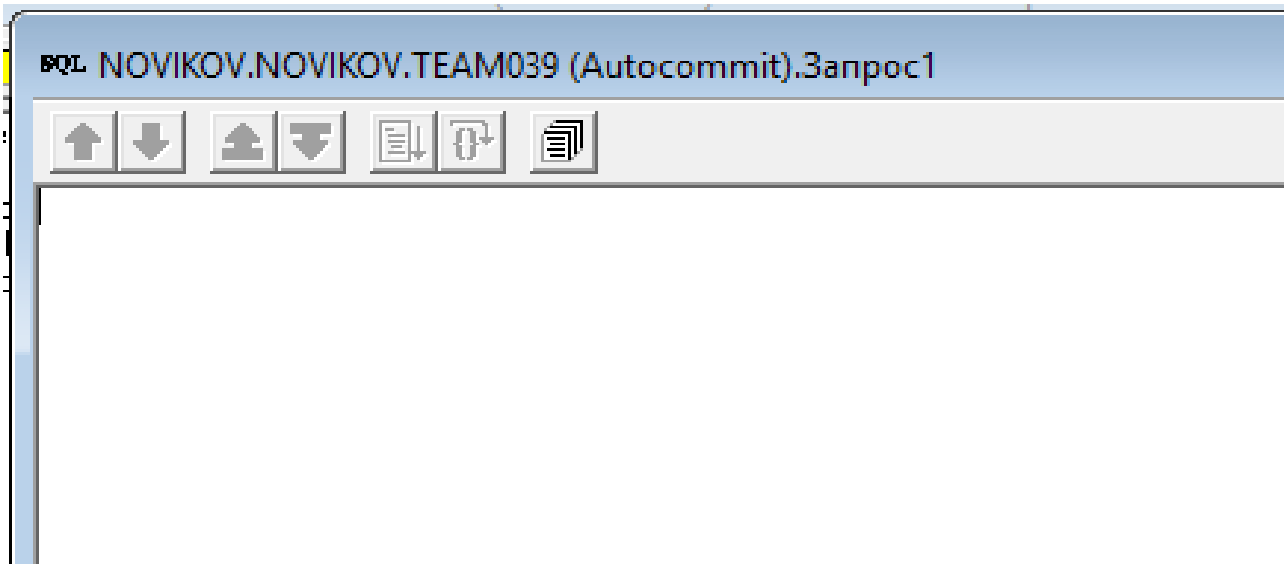
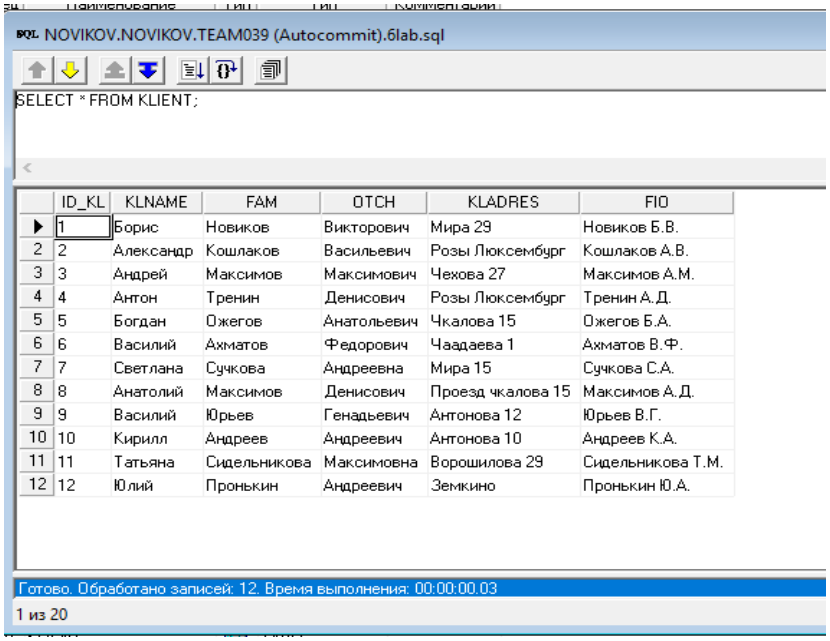


Рисунок 9.1 - Редактор запросов

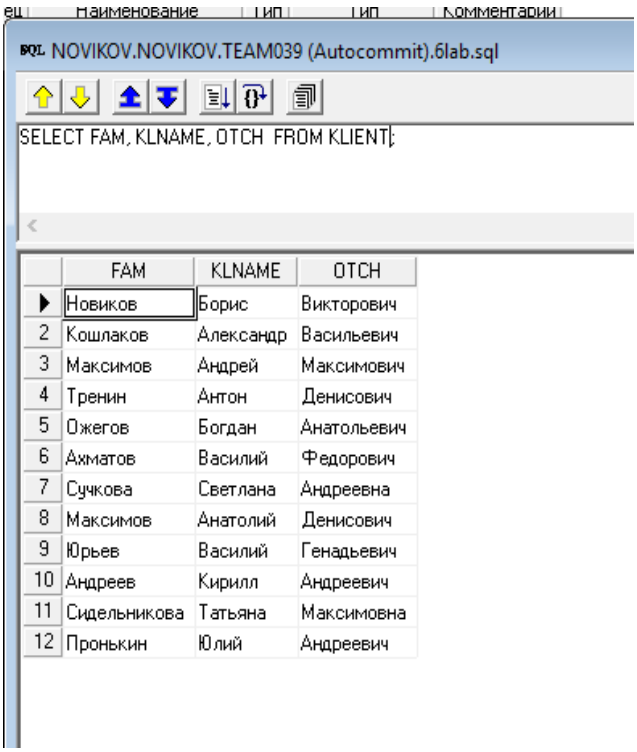
В верхнем окне редактора введен текст запроса, выводящего все строки таблицы KLIENT, и нажата кнопка «Выполнить текущий» (рисунок 9.2).



	ID_KL	KLNAME	FAM	OTCH	KLADRES	FIO
▶	1	Борис	Новиков	Викторович	Мира 29	Новиков Б.В.
2	2	Александр	Кошлаков	Васильевич	Розы Люксембург	Кошлаков А.В.
3	3	Андрей	Максимов	Максимович	Чехова 27	Максимов А.М.
4	4	Антон	Тренин	Денисович	Розы Люксембург	Тренин А.Д.
5	5	Богдан	Ожегов	Анатольевич	Чкалова 15	Ожегов Б.А.
6	6	Василий	Ахматов	Федорович	Чаадаева 1	Ахматов В.Ф.
7	7	Светлана	Сучкова	Андреевна	Мира 15	Сучкова С.А.
8	8	Анатолий	Максимов	Денисович	Проезд чкалова 15	Максимов А.Д.
9	9	Василий	Юрьев	Генадьевич	Антонова 12	Юрьев В.Г.
10	10	Кирилл	Андреев	Андреевич	Антонова 10	Андреев К.А.
11	11	Татьяна	Сидельникова	Максимовна	Ворошилова 29	Сидельникова Т.М.
12	12	Юлий	Пронькин	Андреевич	Земкино	Пронькин Ю.А.

Рисунок 9.2 — Вывод всех записей таблицы KLIENT

В верхнем окне редактора введен новый текст запроса, задающий порядок столбцов, отличный от исходного, и нажата кнопка «Выполнить текущий» (рисунок 9.3).



	FAM	KLNAME	OTCH
▶	Новиков	Борис	Викторович
2	Кошлаков	Александр	Васильевич
3	Максимов	Андрей	Максимович
4	Тренин	Антон	Денисович
5	Ожегов	Богдан	Анатольевич
6	Ахматов	Василий	Федорович
7	Сучкова	Светлана	Андреевна
8	Максимов	Анатолий	Денисович
9	Юрьев	Василий	Генадьевич
10	Андреев	Кирилл	Андреевич
11	Сидельникова	Татьяна	Максимовна
12	Пронькин	Юлий	Андреевич

Рисунок 9.3 — Столбцы таблицы KLIENT с измененным порядком

В верхнем окне редактора введен новый текст запроса, демонстрирующего действие модификатора DISTINCT, и нажата кнопка «Выполнить текущий» (рисунок 9.4).

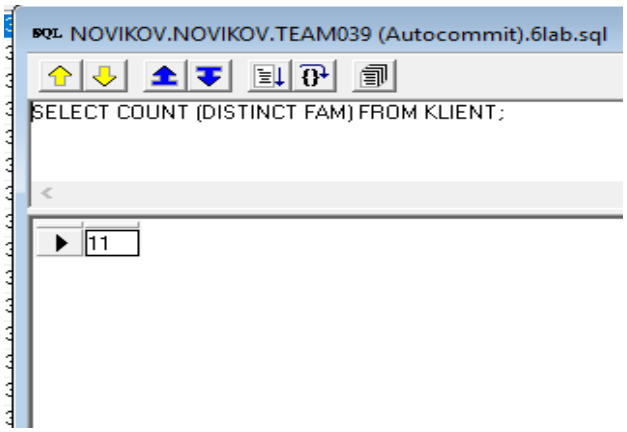


Рисунок 9.4 — Модификатор DISTINCT

В верхнем окне редактора введен новый текст запроса, демонстрирующего использование WHERE с простым условием, и нажата кнопка «Выполнить текущий» (рисунок 9.5).

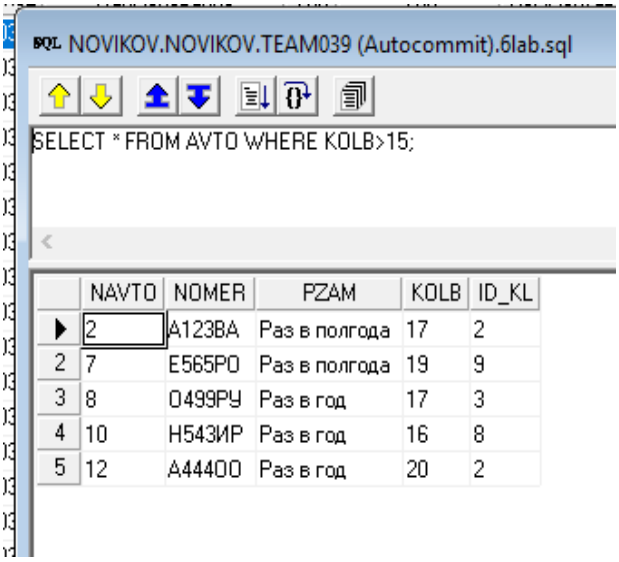


Рисунок 9.5 — WHERE с простым условием

В верхнем окне редактора введен новый текст запроса, демонстрирующего использование WHERE с составным условием , и нажата кнопка «Выполнить текущий»(рисунок 9.6).

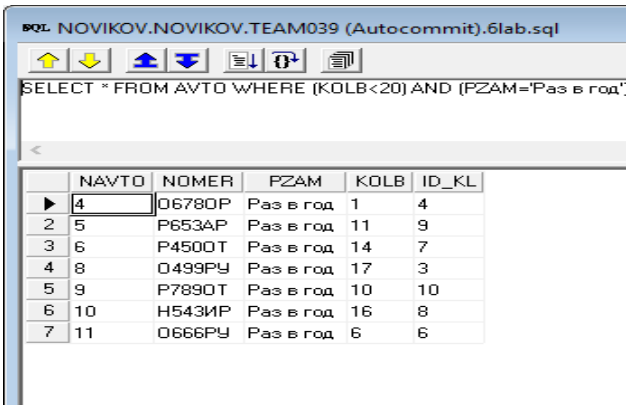


Рисунок 9.6 — WHERE с составным условием

В верхнем окне редактора последовательно введены тексты запросов, демонстрирующих действие специальных функций IN, BETWEEN, LIKE и IS NULL в условии, и нажата кнопка «Выполнить текущий» (рисунки 9.7-9.10).

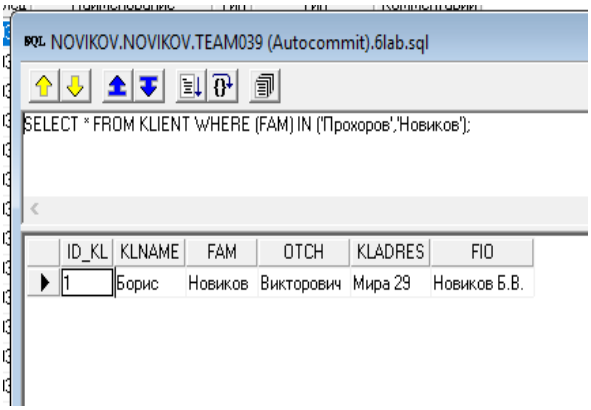


Рисунок 9.7 — Специальная функция IN в условии

–

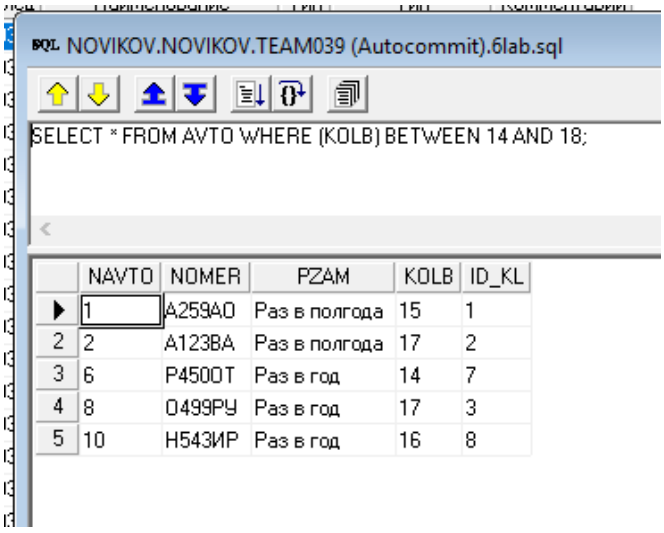


Рисунок 9.8 — Специальная функция BETWEEN в условии

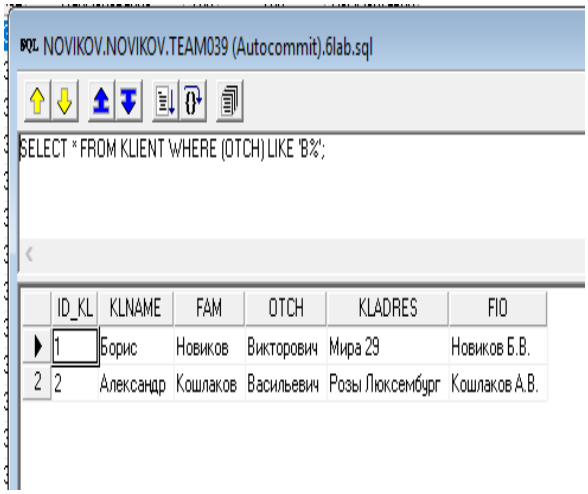


Рисунок 9.9 — Специальная функция LIKE в условии

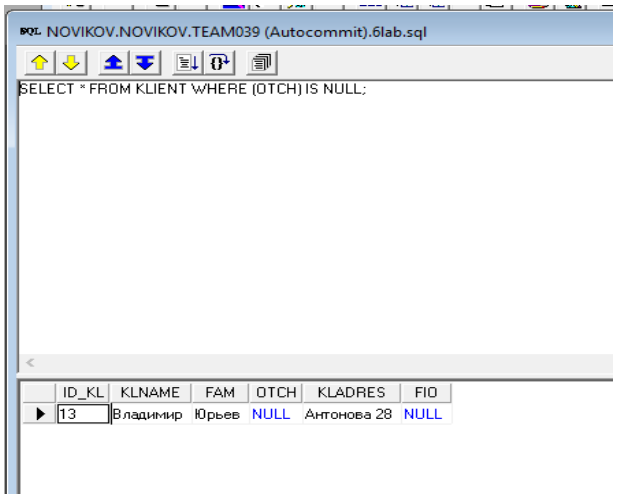
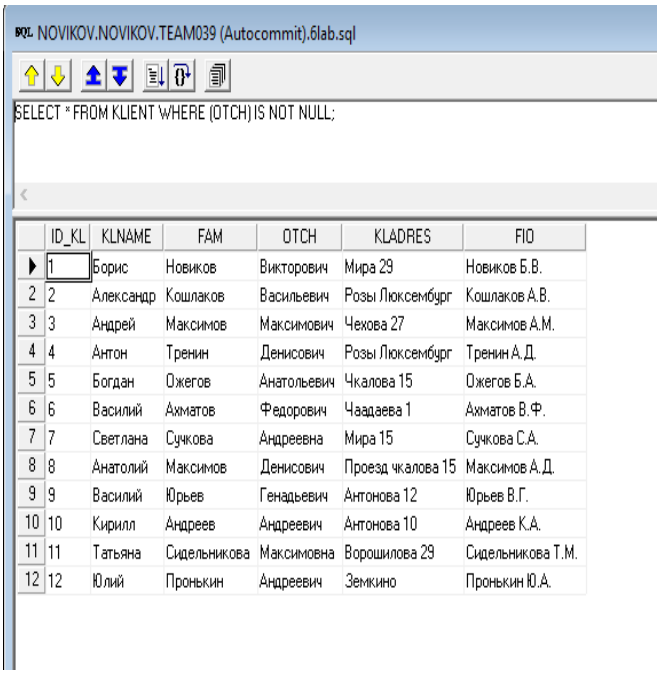


Рисунок 9.10 — Специальная функция IS NULL в условии

В верхнем окне редактора введен новый текст запроса, демонстрирующего использование работу специальной функций is null с условием NOT, и нажата кнопка «Выполнить текущий» (рисунок 9.11).



ID_KL	KLNAME	FAM	OTCH	KLADRES	FIO
1	Борис	Новиков	Викторович	Мира 29	Новиков Б.В.
2	Александр	Кошлаков	Васильевич	Розы Люксембург	Кошлаков А.В.
3	Андрей	Максимов	Максимович	Чехова 27	Максимов А.М.
4	Антон	Тренин	Денисович	Розы Люксембург	Тренин А.Д.
5	Богдан	Ожегов	Анатольевич	Чкалова 15	Ожегов Б.А.
6	Василий	Ахматов	Федорович	Чаадаева 1	Ахматов В.Ф.
7	Светлана	Сучкова	Андреевна	Мира 15	Сучкова С.А.
8	Анатолий	Максимов	Денисович	Проезд чкалова 15	Максимов А.Д.
9	Василий	Юрьев	Генадьевич	Антонова 12	Юрьев В.Г.
10	Кирилл	Андреев	Андреевич	Антонова 10	Андреев К.А.
11	Татьяна	Сидельникова	Максимовна	Ворошилова 29	Сидельникова Т.М.
12	Юлий	Пронькин	Андреевич	Земкино	Пронькин Ю.А.

Рисунок 9.11 - Специальная функция is null с условием NOT

В верхнем окне редактора введен новый текст запроса, демонстрирующего использование работу агрегатных функций COUNT, SUM, AVG, MAX, MIN, и нажата кнопка «Выполнить текущий» (рисунки 9.12-9.16).

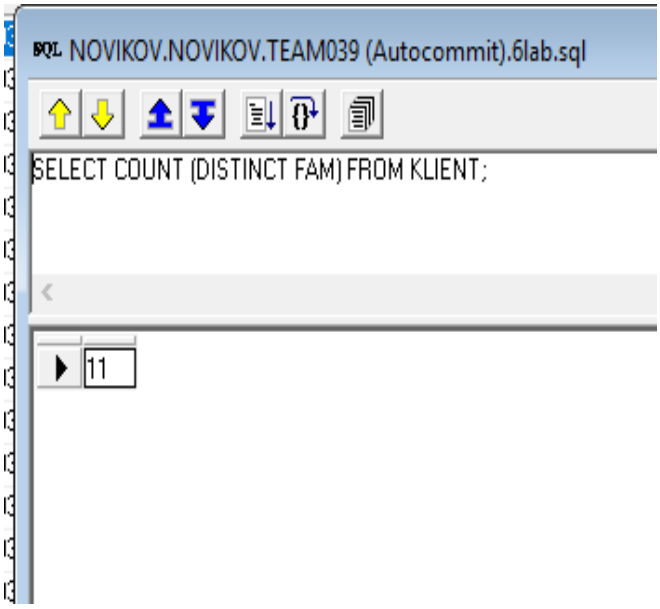


Рисунок 9.12 - Агрегатная функция COUNT

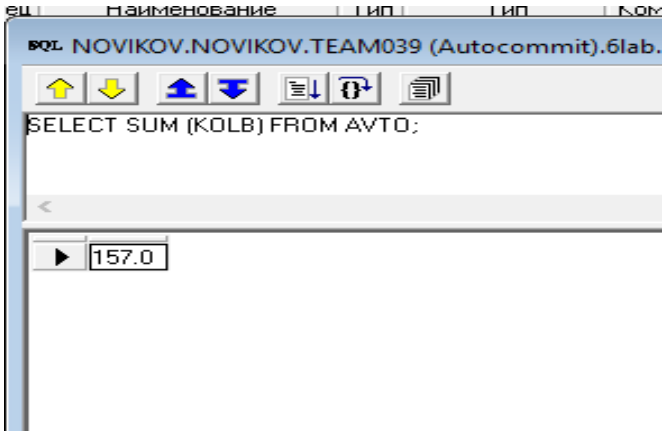


Рисунок 9.13 — Агрегатная функция SUM

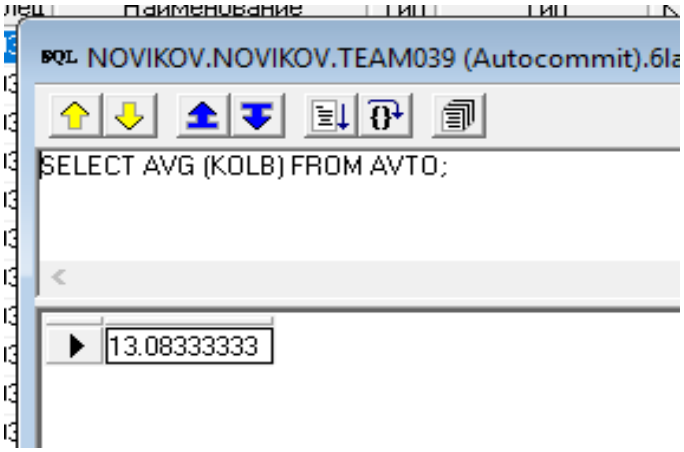


Рисунок 9.14 — Агрегатная функция AVG

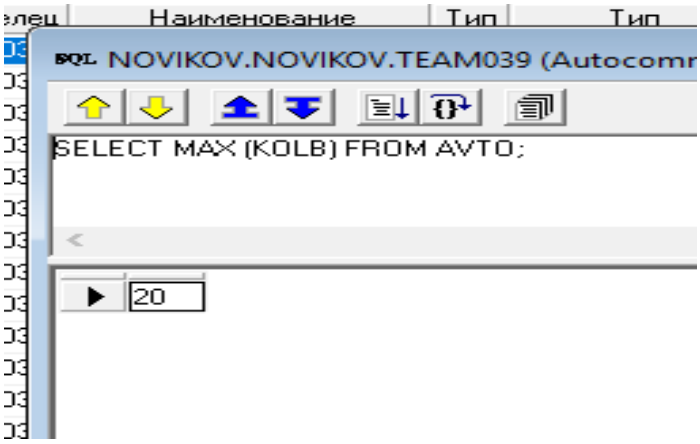


Рисунок 9.15 — Агрегатная функция MAX

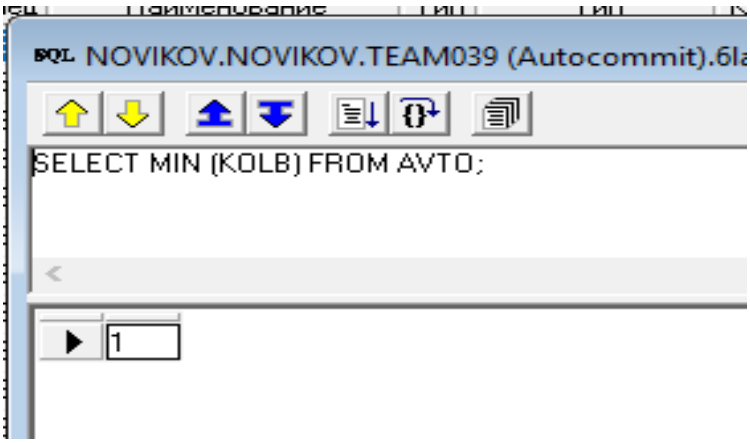


Рисунок 9.16 — Агрегатная функция MIN

В верхнем окне редактора введен новый текст запроса, демонстрирующего использование функции COUNT(*), и нажата кнопка «Выполнить текущий» (рисунок 9.17).

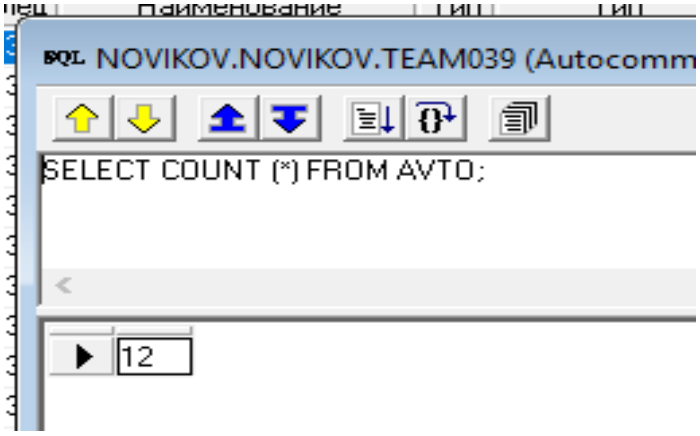


Рисунок 9.17 — Функция COUNT(*)

В верхнем окне редактора введен новый текст запроса, демонстрирующего выполнение простых вычислений в запросе, и нажата кнопка «Выполнить текущий» (рисунок 9.18).

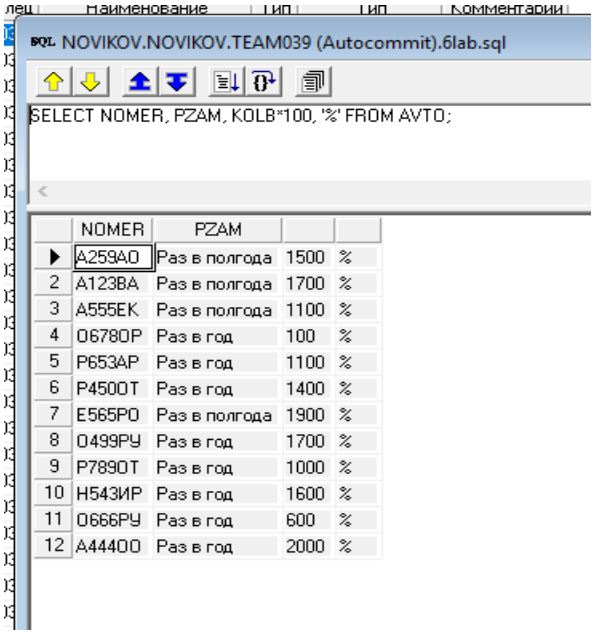


Рисунок 9.18 — Простые вычисления в запросе

В верхнем окне редактора введен новый текст запроса, демонстрирующего использование простого вычисления как параметр агрегатной функции, и нажата кнопка «Выполнить текущий» (рисунок 10.19).

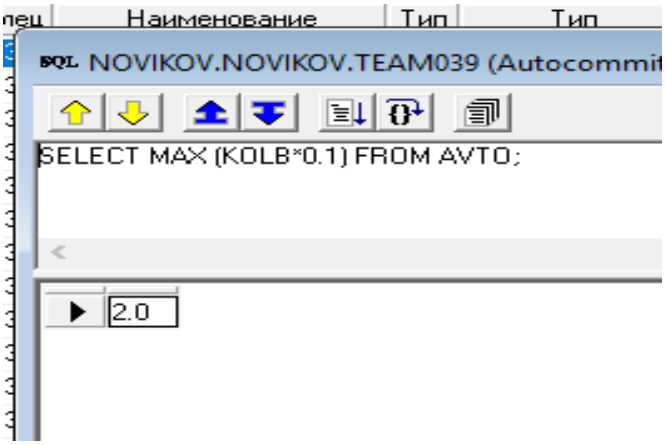


Рисунок 9.19— Простые вычисления, как параметр агрегатной функции

В верхнем окне редактора введен новый текст запроса, демонстрирующего работу предложения GROUP BY, и нажата кнопка «Выполнить текущий» (рисунок 9.20).

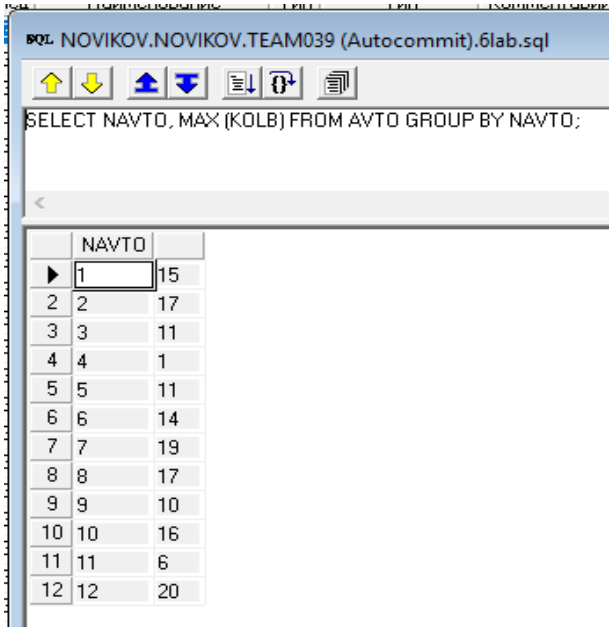


Рисунок 9.20— Предложения GROUP BY

В верхнем окне редактора введен новый текст запроса, демонстрирующего работу предложения HAVING, и нажата кнопка «Выполнить текущий» (рисунок 9.21).

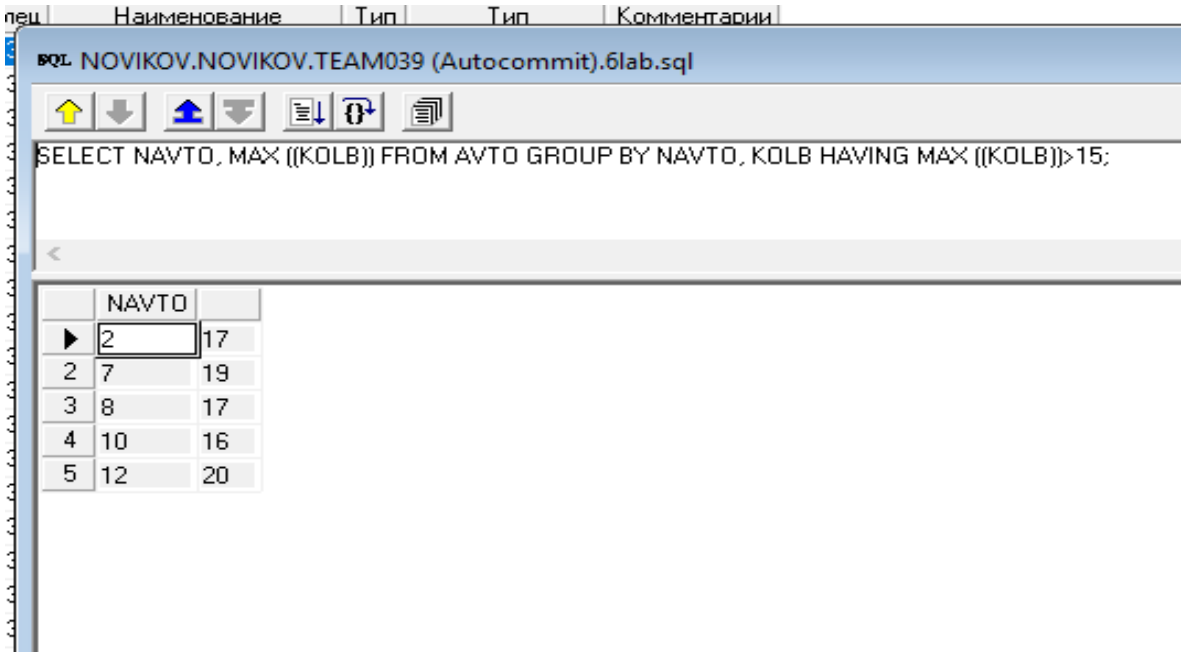


Рисунок 9.21 — Предложения HAVING

Таким образом, были выполнены ращличные SQL-забросы к базе данных Novikov.

10 Выборка данных из связанных таблиц

Для создания дизайнерского запроса была выполнена команда Запрос > Создать (рисунок 10.1), после чего была выделена строка «Дизайнер запросов» и нажата кнопка ОК. После чего открылось диалоговое окно, представленное на рисунке 10.2.

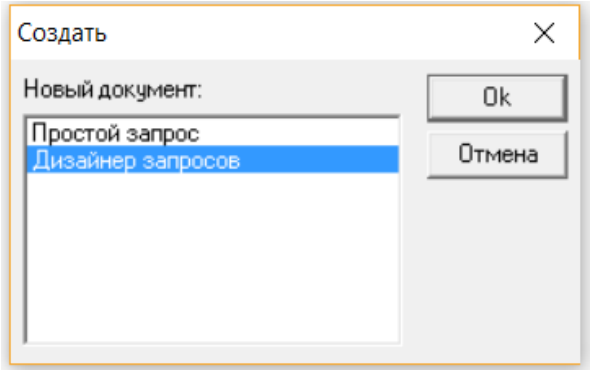


Рисунок 10.1 – Создание запроса

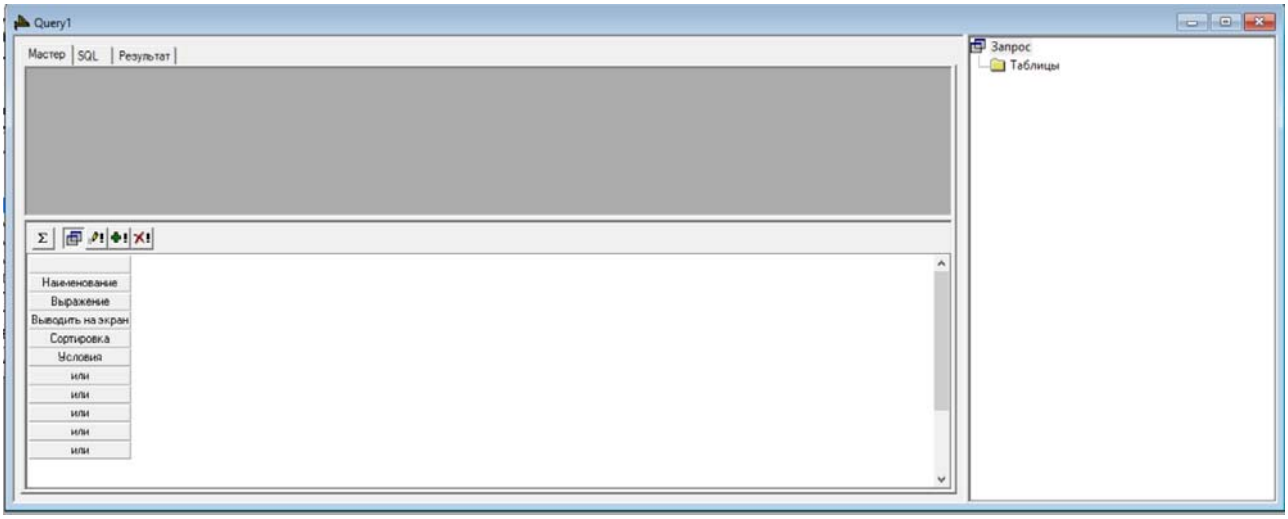


Рисунок 10.2 - Диалоговое окно дизайнера запросов

В правой части диалогового окна дизайнера запросов (рисунок 10.2) был выделен пункт «Таблица» и нажато по нему левой кнопкой мыши (рисунок 10.3). В выпадающем меню выбран пункт «Добавить новую таблицу», что позволило добавить все поля выбранной таблицы базы данных в запрос.

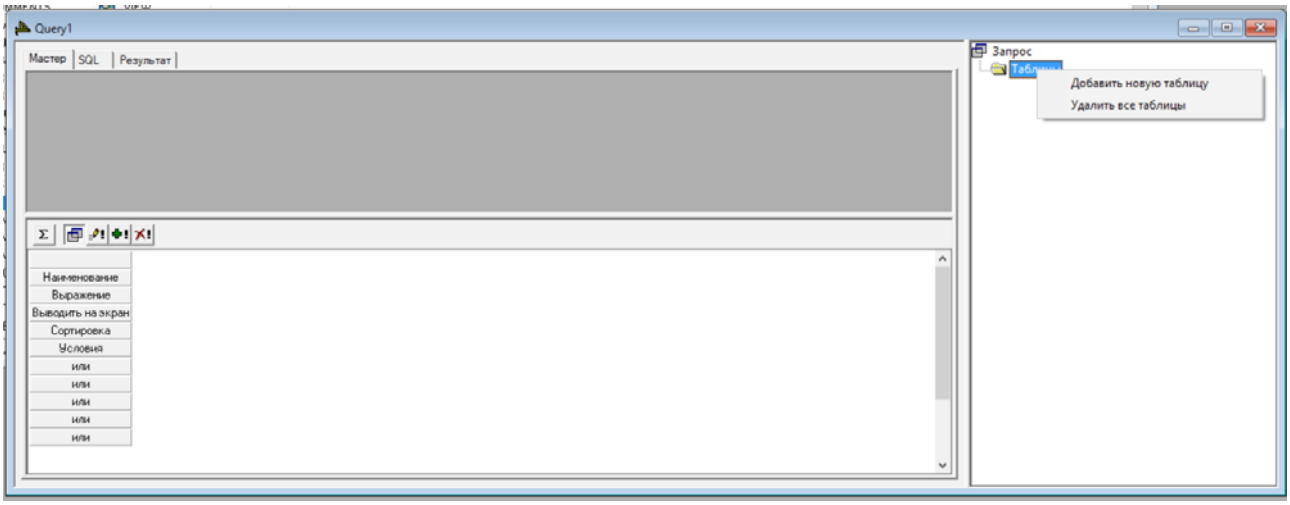


Рисунок 10.3 - Добавление новой таблицы в дизайнере запросов

В открывшемся окне «Добавить таблицу» (рисунок 10.4) выбрана родительская таблица (рисунке 10.5). Результат добавления представлен на рисунке 10.6.

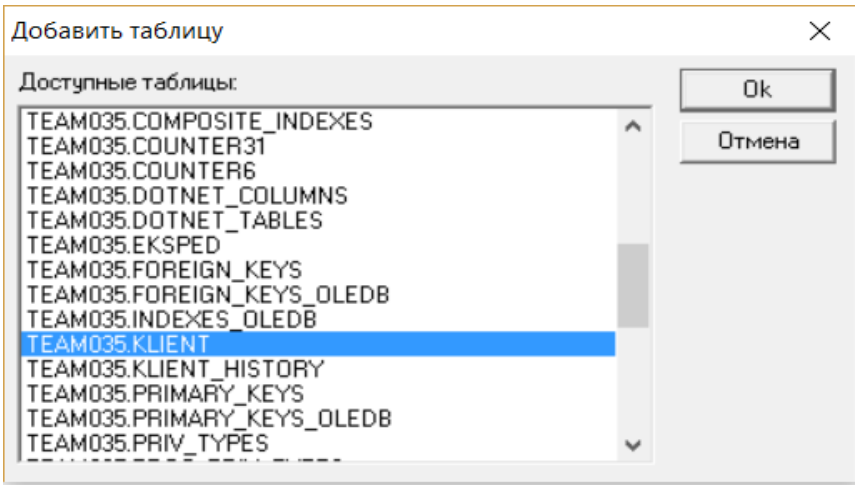


Рисунок 10.4 - Диалоговое окно «Добавить таблицу»

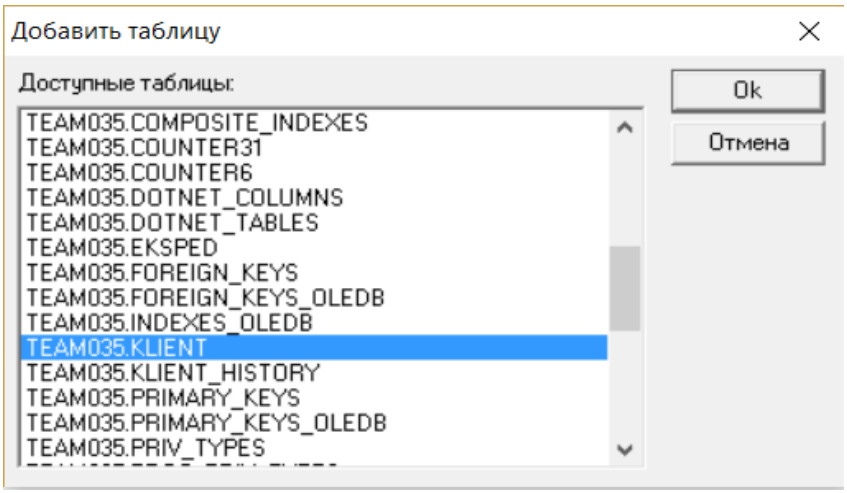


Рисунок 10.5 - Добавление родительской таблицы

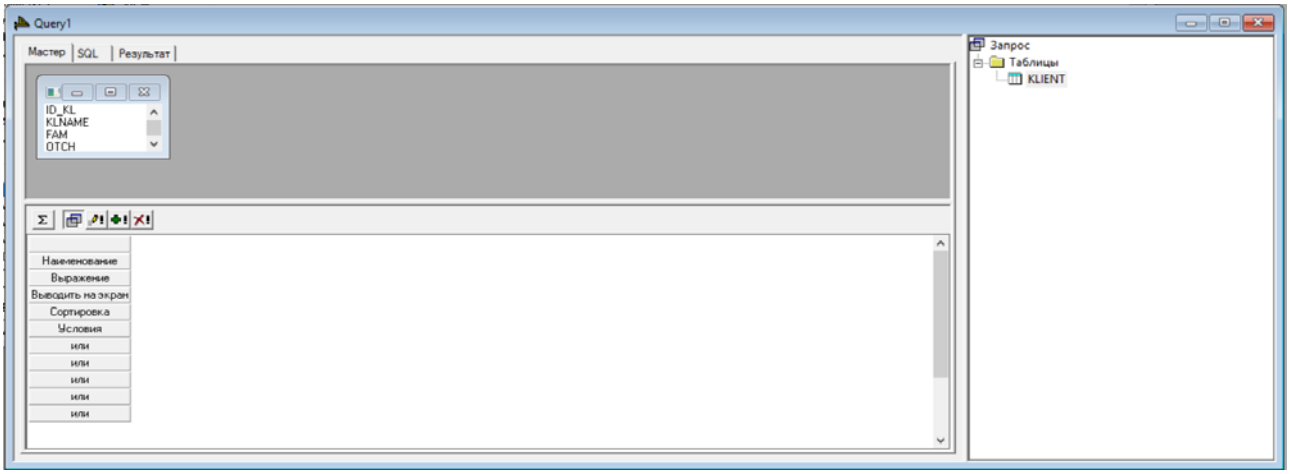


Рисунок 10.6 - Результат добавления родительской таблицы

После выделения в правом окне родительской таблицы и, нажатия правой кнопки мыши, в выпадающем списке был выбран пункт «Включить все столбцы» (рисунок 10.7). Результат работы представлен на рисунке 10.8.

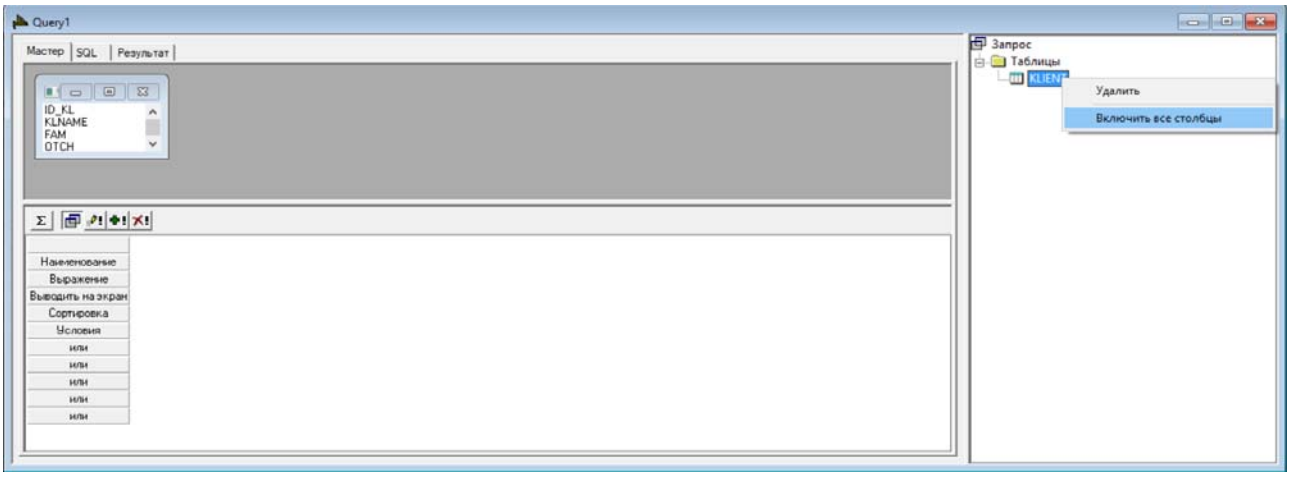


Рисунок 10.7 - Включение всех столбцов родительской таблицы

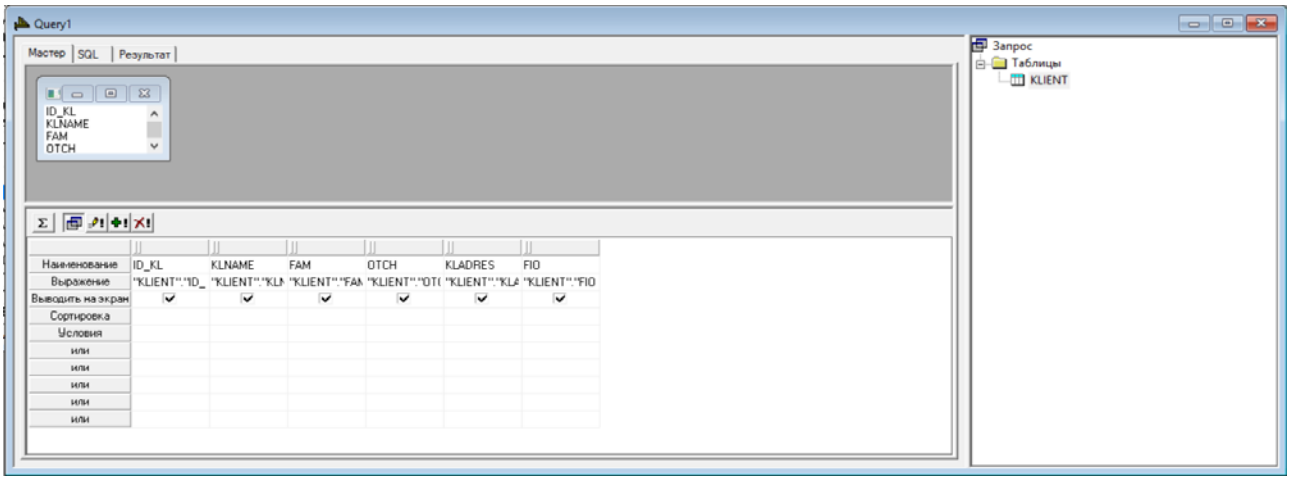


Рисунок 10.8 - Результат добавления родительской таблицы

Предыдущие действия были повторены для дочерней таблицы связанных таблиц базы данных (рисунки 10.9-10.12).

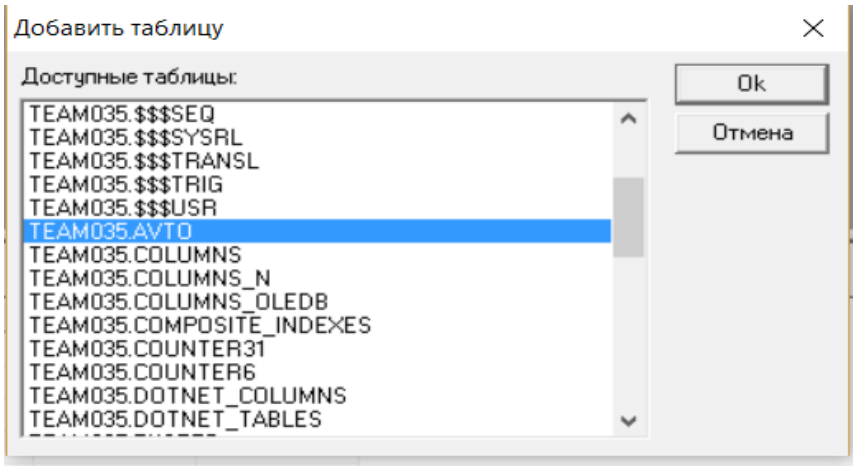


Рисунок 10.9 - Добавление дочерней таблицы

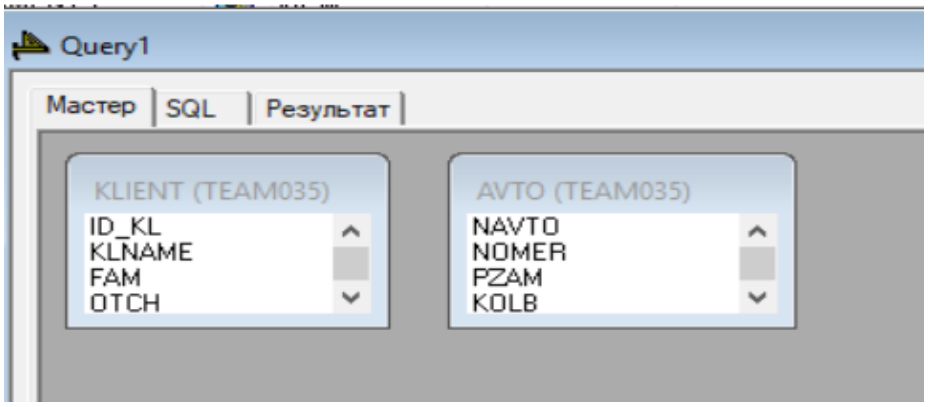


Рисунок 10.10 - Результат добавления дочерней таблицы в «Мастер»

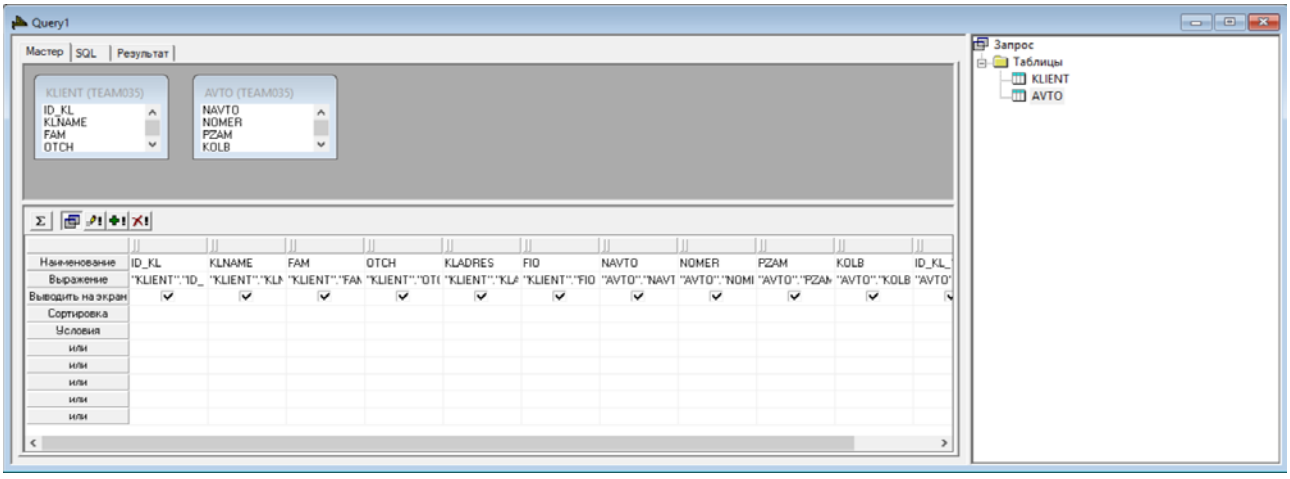


Рисунок 10.11 - Включение всех столбцов дочерней таблицы

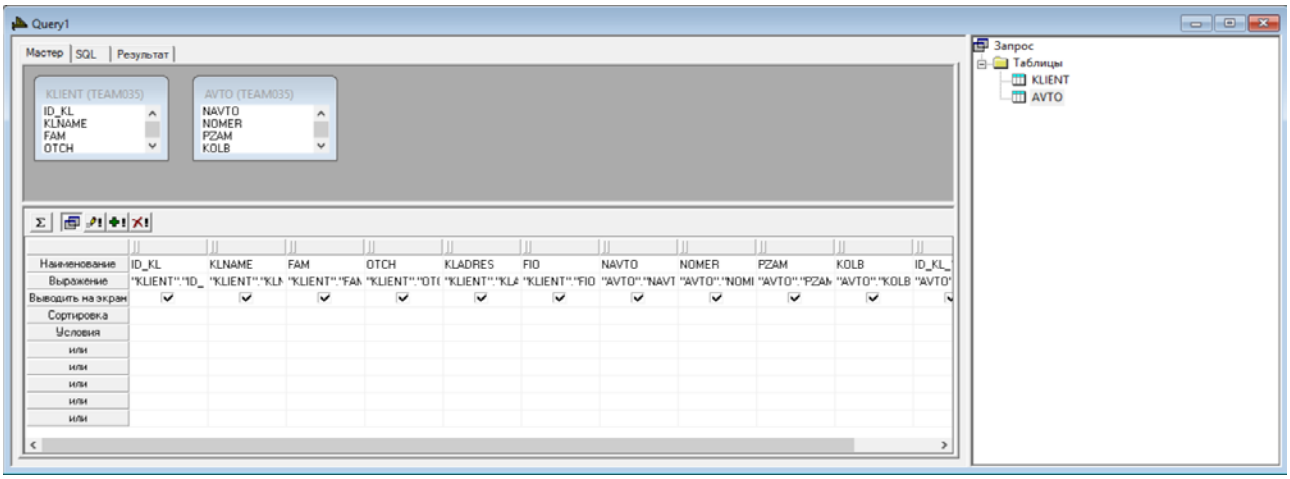


Рисунок 10.12- Результат добавления дочерней таблицы

В ячейке внешнего ключа, входящего в состав первичного ключа дочерней таблицы, строки «Условия» установлено значение «=KLIENT.ID_KL», где KLIENT - имя родительской таблицы, ID_KL -имя первичного ключа родительской таблицы.(рисунок 10.13).

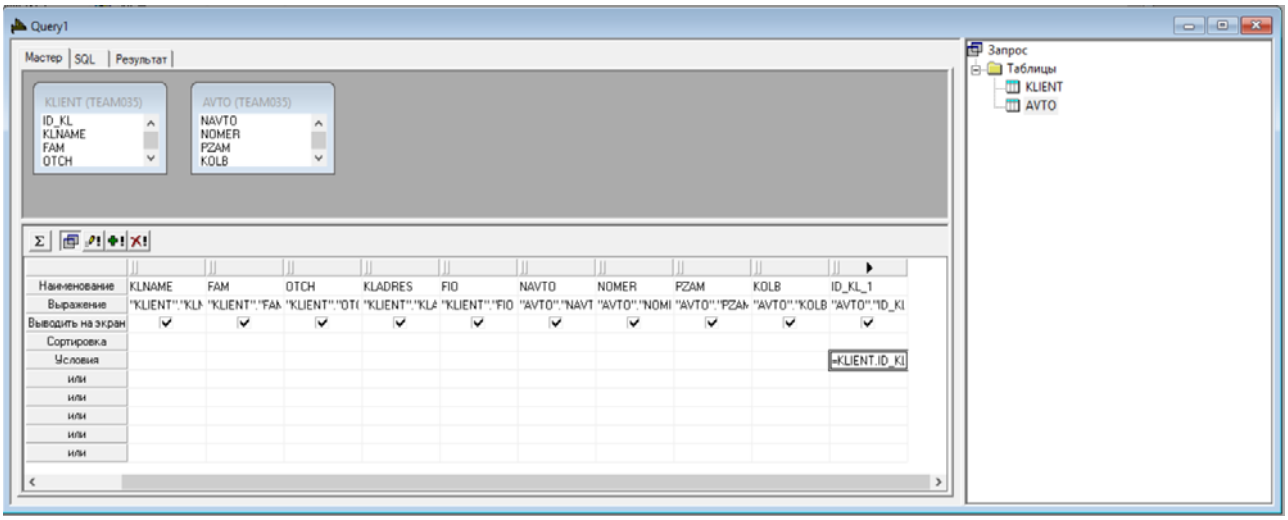


Рисунок 10.13 - Добавления условия для выполнения выборки данных из связанных таблиц

После перехода на вкладку «SQL» произойдет генерация SQL – запроса выборки данных из связанных таблиц (рисунок 10.14).

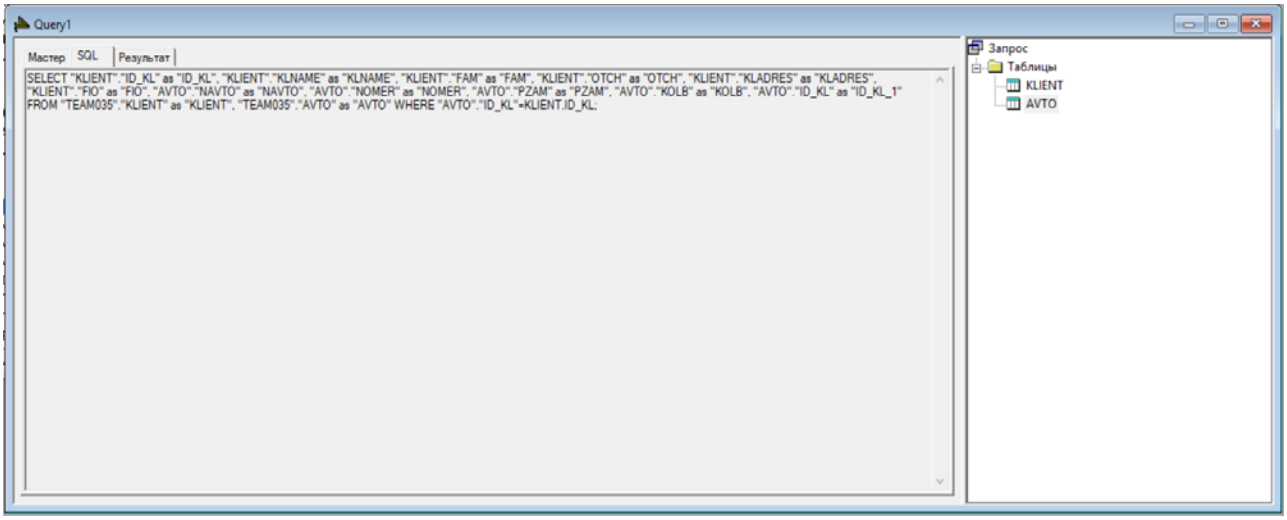
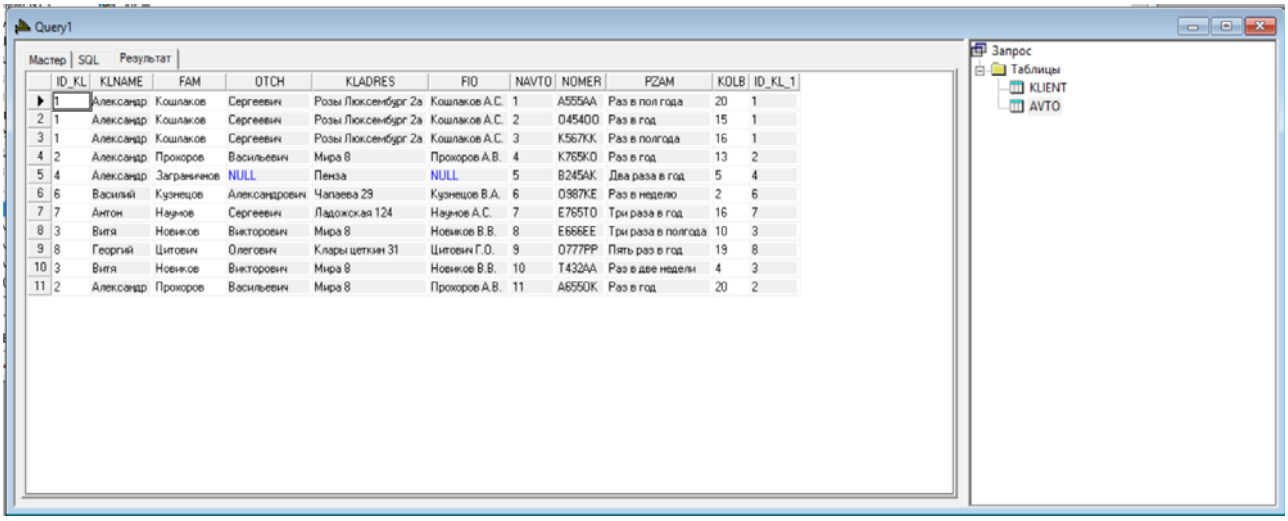


Рисунок 10.14 - SQL- запрос выборки данных из связанных таблиц

После перехода на вкладку «Результат» произведена выборка данных из связанных таблиц (рисунок 10.15).



ID_KL	KLNAME	FAM	OTCH	KLADRES	FIO	NAVTO	NOMER	PZAM	KOLB	ID_KL_1
1	Александр	Кошлаков	Сергеевич	Розы Люксембург 2а	Кошлаков А.С.	1	A555AA	Раз в пол года	20	1
1	Александр	Кошлаков	Сергеевич	Розы Люксембург 2а	Кошлаков А.С.	2	045400	Раз в год	15	1
1	Александр	Кошлаков	Сергеевич	Розы Люксембург 2а	Кошлаков А.С.	3	K567KK	Раз в полгода	16	1
2	Александр	Прокоров	Васильевич	Мира 8	Прокоров А.В.	4	K765KO	Раз в год	13	2
4	Александр	Заграничнов	NULL	Пенза	NULL	5	B245AK	Два раза в год	5	4
6	Василий	Кузнецов	Александрович	Чапаева 29	Кузнецов В.А.	6	O987KE	Раз в неделю	2	6
7	Антон	Наумов	Сергеевич	Ладожская 124	Наумов А.С.	7	E765TO	Три раза в год	16	7
3	Витя	Новиков	Викторович	Мира 8	Новиков В.В.	8	E666EE	Три раза в полгода	10	3
8	Георгий	Цитович	Олегович	Клары цеткин 31	Цитович Г.О.	9	O777PP	Пять раз в год	19	8
3	Витя	Новиков	Викторович	Мира 8	Новиков В.В.	10	T432AA	Раз в две недели	4	3
2	Александр	Прокоров	Васильевич	Мира 8	Прокоров А.В.	11	A655OK	Раз в год	20	2

Рисунок 10.15 - Результат выборки данных из связанных таблиц

В строке «Наименования» (рисунок 10.16) «Мастера запросов» названия всех столбцов были переведены на русский язык.

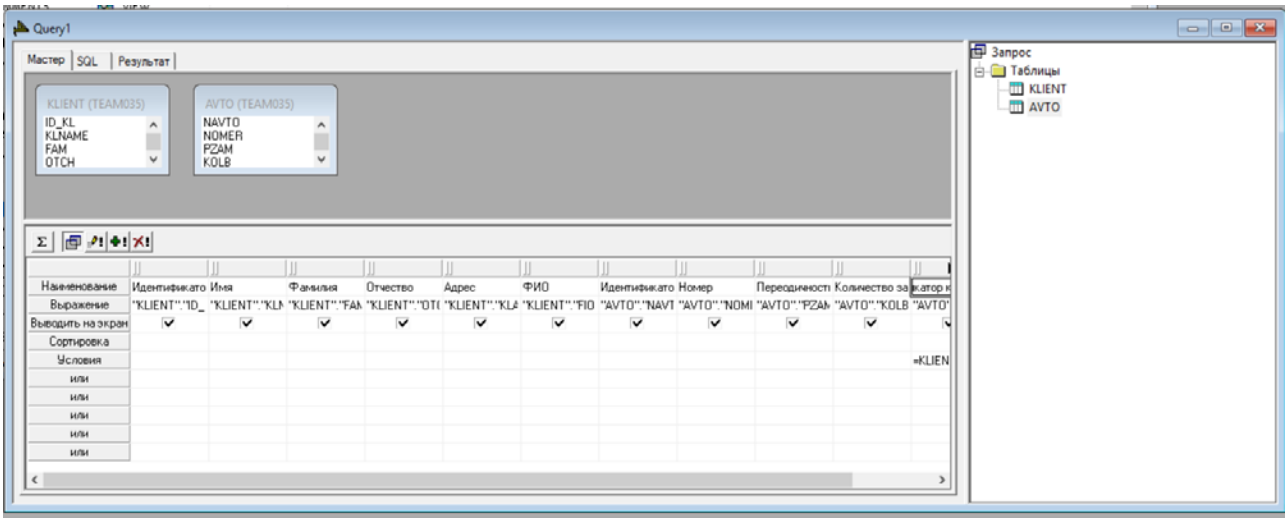


Рисунок 10.16 - Переименование названий столбцов

Далее перешли на вкладку SQL, чтобы просмотреть текст SQL-скрипта запроса (рисунок 10.17).

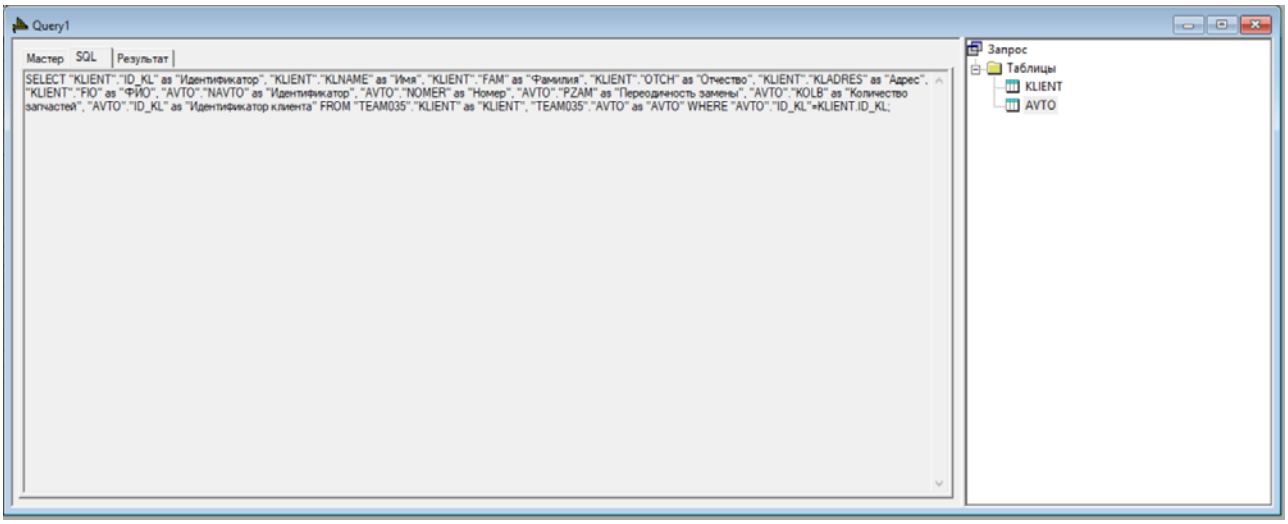


Рисунок 10.17 - Текст SQL-скрипта

Затем перешли на вкладку «Результат», чтобы увидеть результат выполнения запроса (рисунок 10.18).

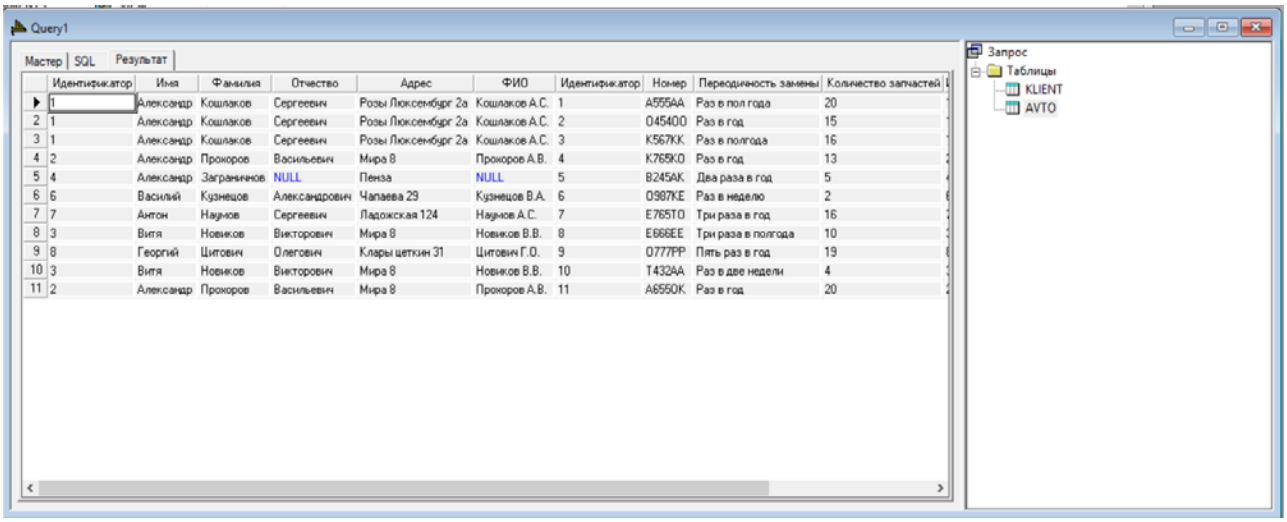


Рисунок 10.18 - Результат выборки данных из связанных таблиц с переименованными столбцами

«Мастер запросов» позволяет удалить ненужные столбцы из таблицы с результатами выполнения запроса. Из результата запроса был удален столбец «Идентификатор клиента». Текст запроса и результат его выполнения представлены на рисунках 10.19 — 10.22.

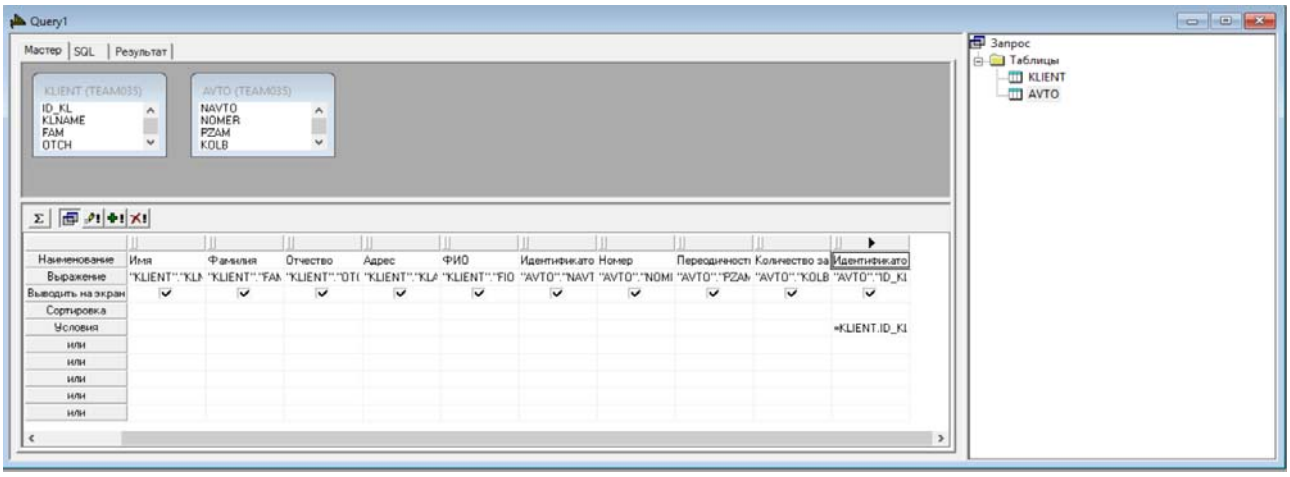
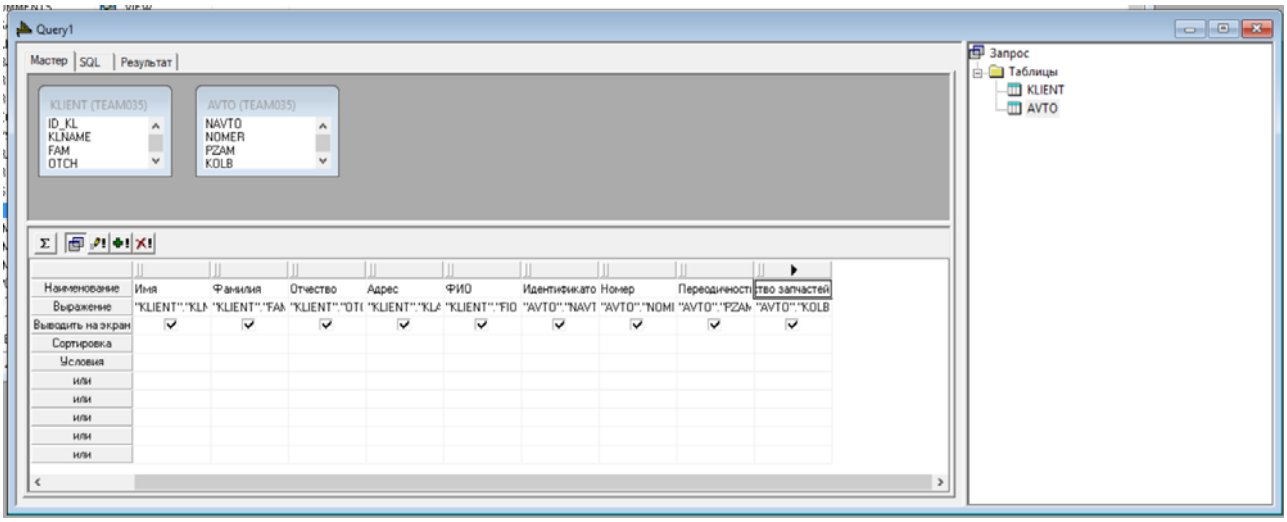


Рисунок 10.19 - Удаление столбца



Риунок 10.20 - Результат удаления столбца

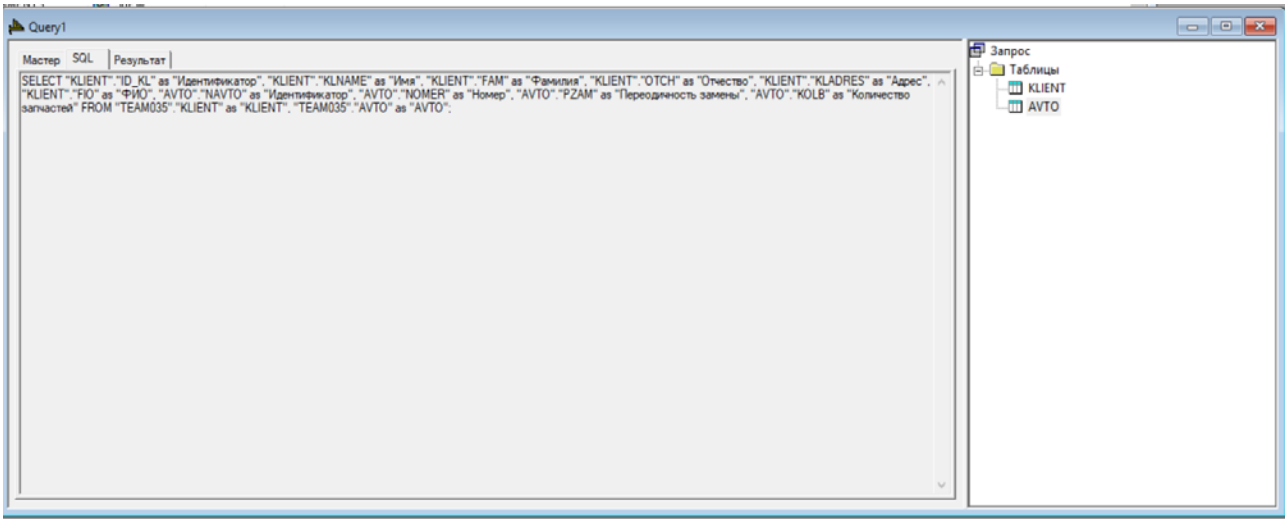


Рисунок 10.21 - Текст SQL-запроса

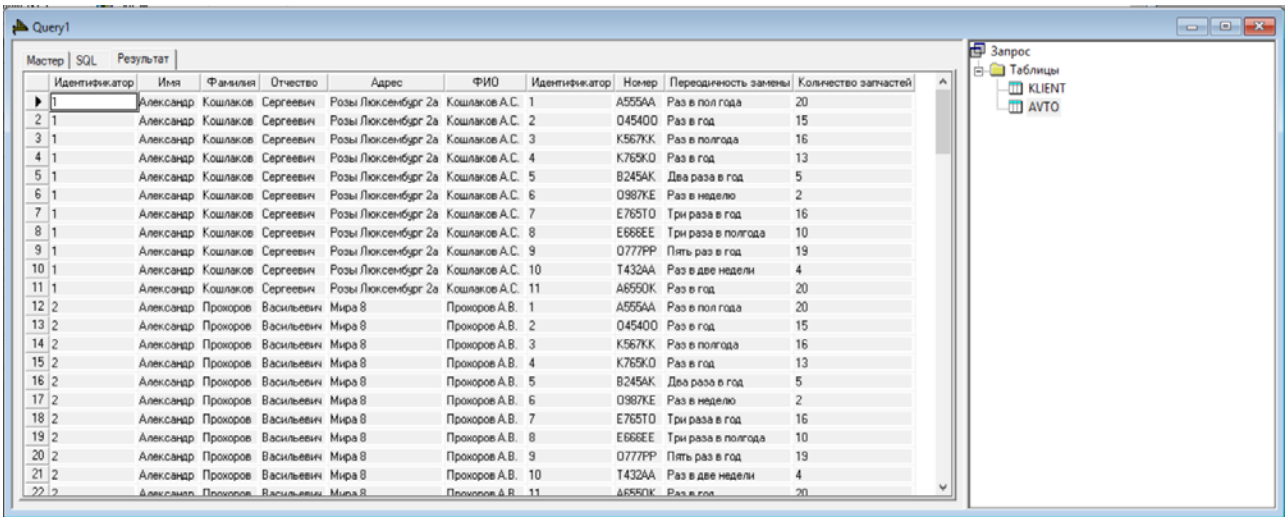


Рисунок 10.22 - Результат выполнения запроса

После закрытия «Мастера запросов» появилось сообщение о сохранении скрипта запроса (рисунок 10.23). Было выполнено сохранение в папку "LR78" (рисунок 10.24).

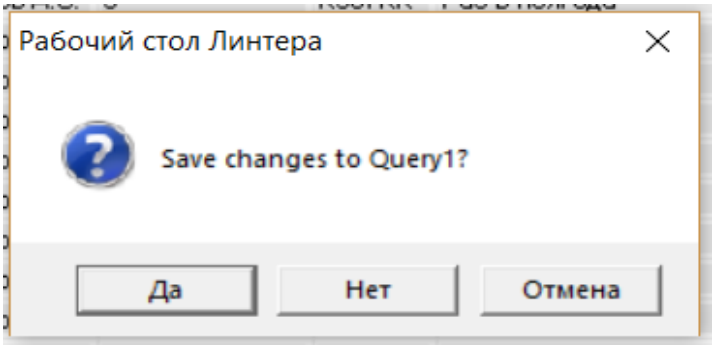


Рисунок 10.23 - Запрос на сохранение скрипта

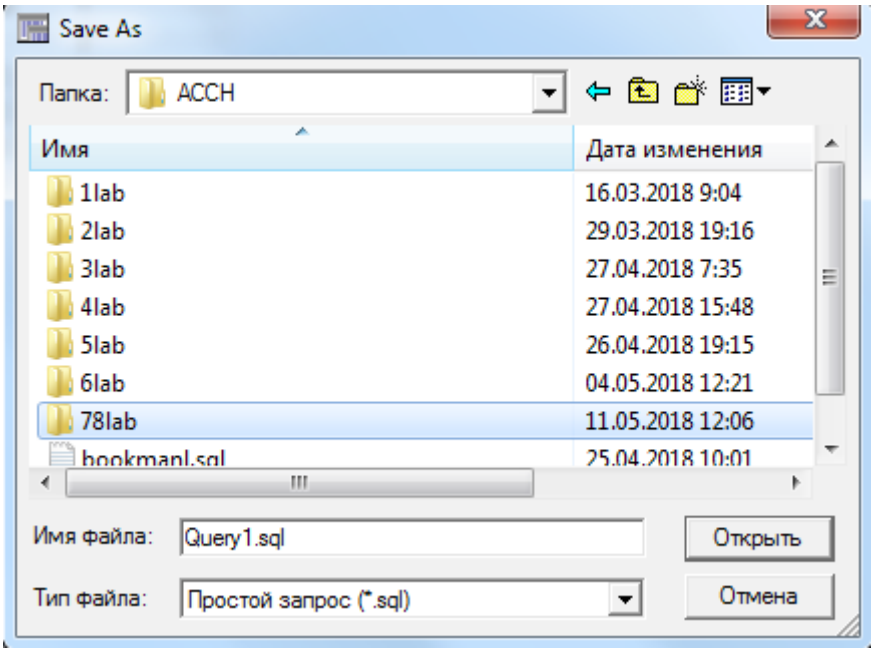


Рисунок 10.24 - Сохранение скрипта в папке «ЛР7»

После этого была выполнена выборка данных из связанных таблиц без использования дизайнера запросов в соответствии с рисунками 10.25 — 10.26.

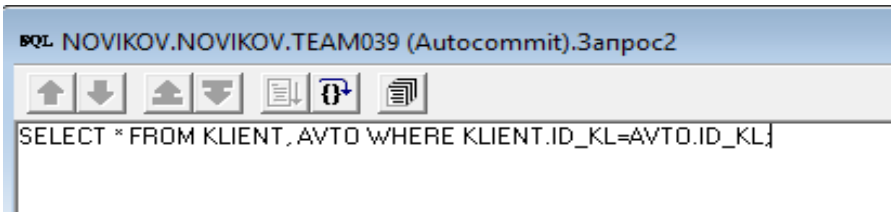
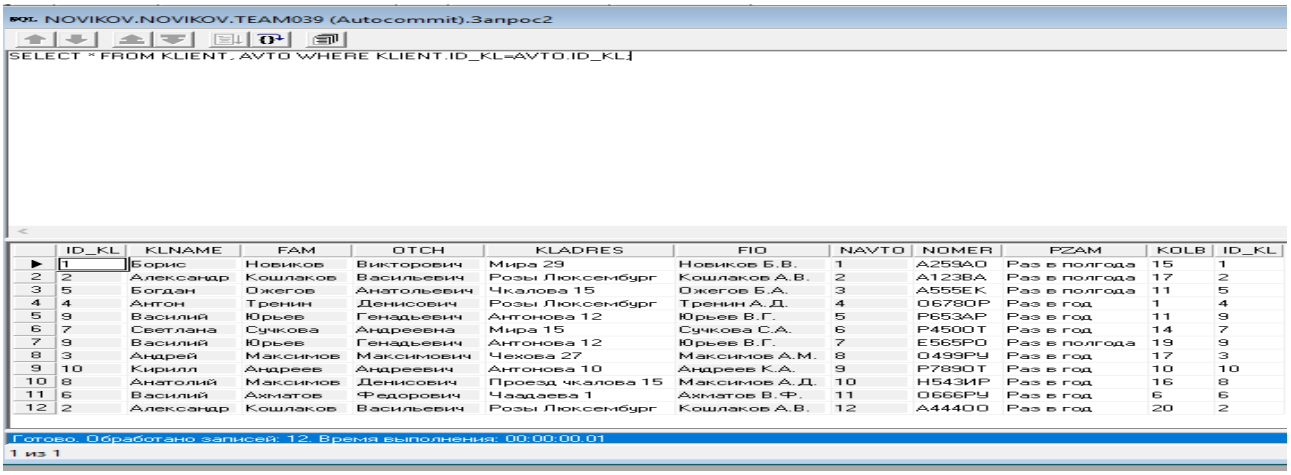


Рисунок 10.25 - Создание запроса



	ID_KL	KLNAME	FAM	OTCH	KLADRES	FIO	NAVTO	NOMER	PZAM	KOLB	ID_KL
▶	1	Борис	Новиков	Викторович	Мира 29	Новиков Б.В.	1	A259AO	Раз в полгода	15	1
2	2	Александр	Кошлаков	Васильевич	Розы Люксембург	Кошлаков А.В.	2	A123BA	Раз в полгода	17	2
3	5	Богдан	Ожегов	Анатольевич	Чкалова 15	Ожегов Б.А.	3	A555EK	Раз в полгода	11	5
4	4	Антон	Тренин	Денисович	Розы Люксембург	Тренин А.Д.	4	O678OP	Раз в год	1	4
5	9	Василий	Юрьев	Генадьевич	Антонова 12	Юрьев В.Г.	5	P653AP	Раз в год	11	9
6	7	Светлана	Сучкова	Андреевна	Мира 15	Сучкова С.А.	6	P450OT	Раз в год	14	7
7	9	Василий	Юрьев	Генадьевич	Антонова 12	Юрьев В.Г.	7	E565PO	Раз в полгода	19	9
8	3	Андрей	Максимов	Максимович	Чехова 27	Максимов А.М.	8	O499PY	Раз в год	17	3
9	10	Кирилл	Андреев	Андреевич	Антонова 10	Андреев К.А.	9	P789OT	Раз в год	10	10
10	8	Анатолий	Максимов	Денисович	Проезд чкалова 15	Максимов А.Д.	10	H543IP	Раз в год	16	8
11	6	Василий	Ахматов	Федорович	Чаадаева 1	Ахматов В.Ф.	11	O666PY	Раз в год	6	6
12	2	Александр	Кошлаков	Васильевич	Розы Люксембург	Кошлаков А.В.	12	A444OO	Раз в год	20	2

Рисунок 10.26 - Результат выполнения запроса

Далее был создан и выполнен параметрический запрос без использования «Мастера запросов» (рисунки 10.27-10.28). Для выполнения параметрического запроса было введено значение параметра kolb равное 5(рисунок 10.29).

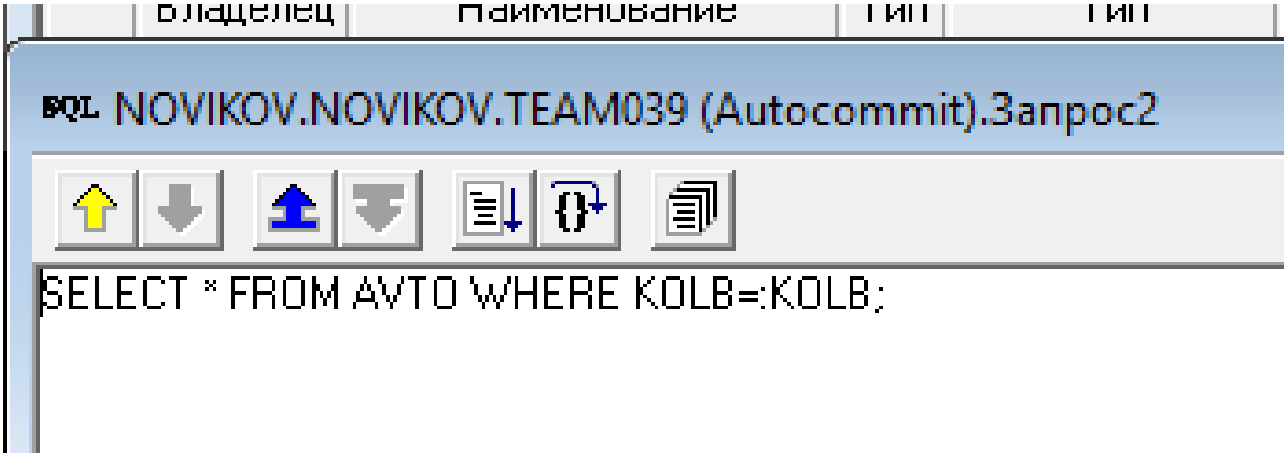


Рисунок 10.27 - Текст параметрического запроса

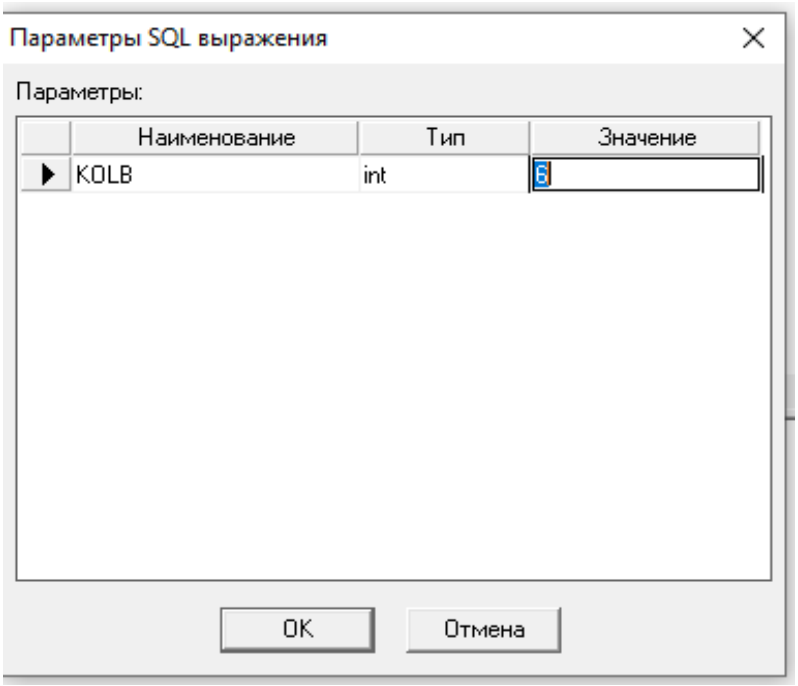


Рисунок 10.28 - Ввод параметра запроса

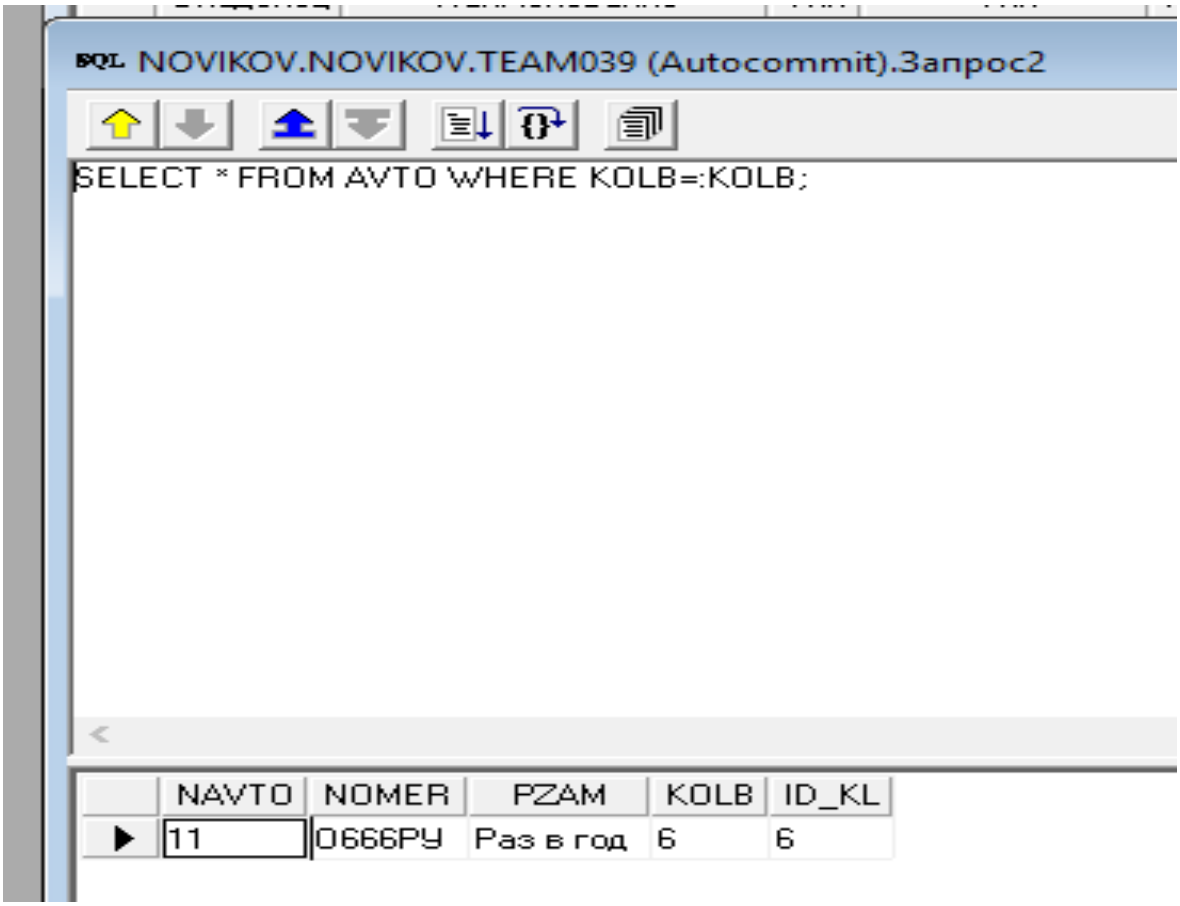


Рисунок 10.29 - Результат выполнения параметрического запроса

Таким образом, были созданы параметрический и дизайнерский запрос к базе данных Novikov.

11 Резервное копирование и восстановление базы данных

Для создания резервной копии базы данных было использовано приложение "Резервное копирование" (рисунок 11.1), которое было запущено на выполнение командой меню «Пуск > Все программы > СУБД ЛИНТЕР > Расширенные средства администрирования > Резервное копирование».

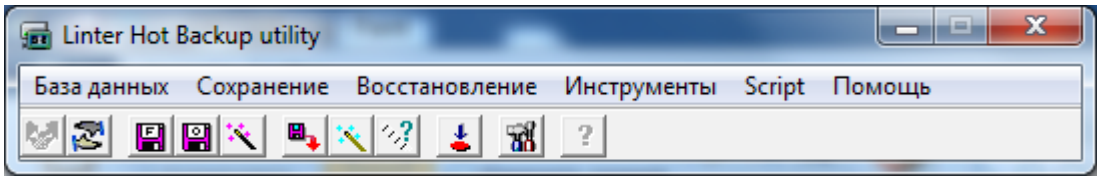


Рисунок 11.1 – Приложение "Резервное копирование"

В открывшемся диалоговом окне "Соединение с базой данных" (рисунок 11.2) в поле Имя пользователя было введено имя пользователя TEAM039 и в следующем за ним поле — пароль slave039.

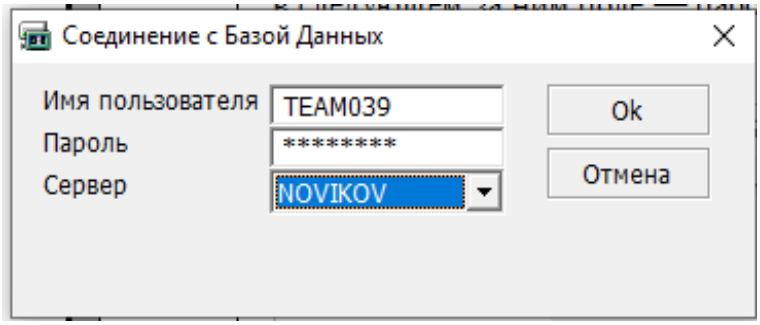


Рисунок 11.2 – Соединение с БД

Затем в поле Сервер из выпадающего меню выбран сервер Novikov. Далее в окне "Linter Hot Backup utility" (рисунок 11.3) выполнена команда меню «Сохранение > Мастер сохранения». Было открыто окно "Мастер сохранения базы данных. Шаг 1 — Выбор метода сохранения" (рисунок 11.4).

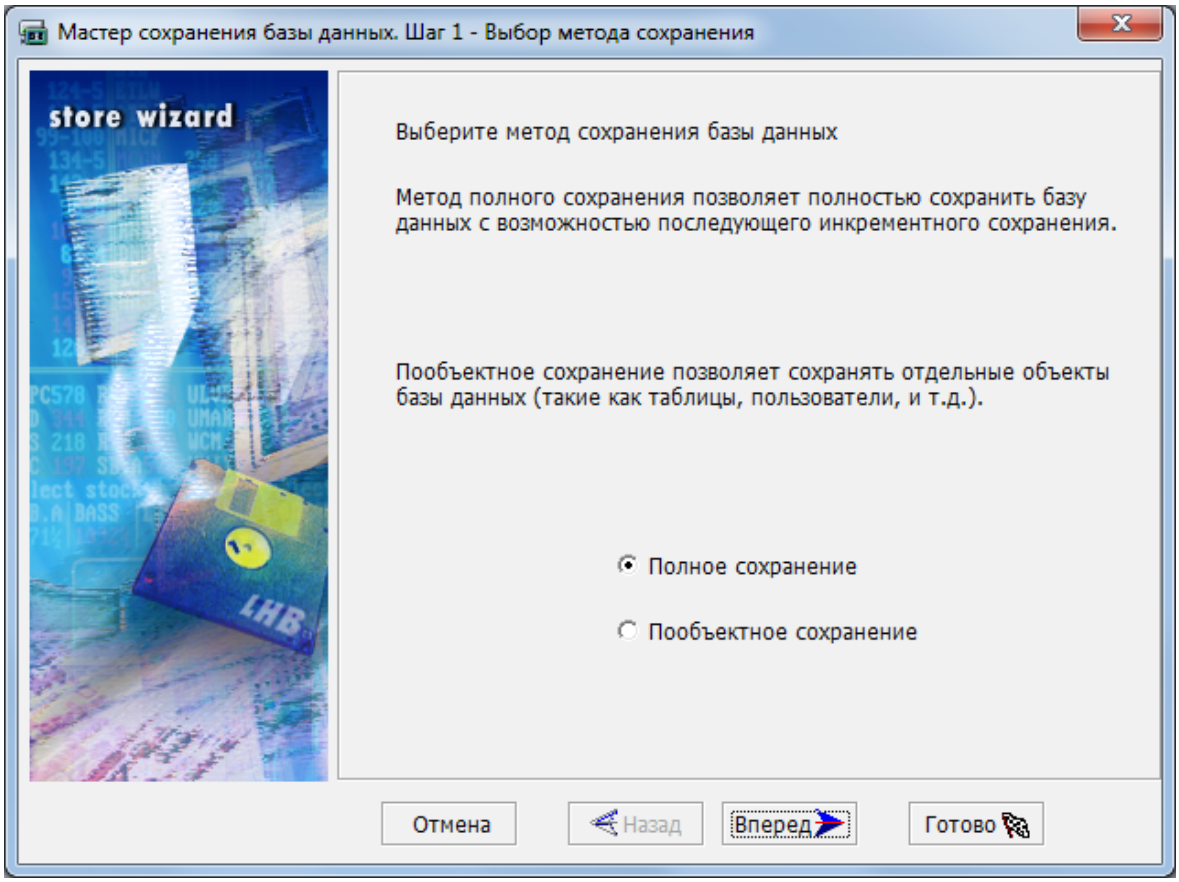


Рисунок 11.3 – Мастер сохранения

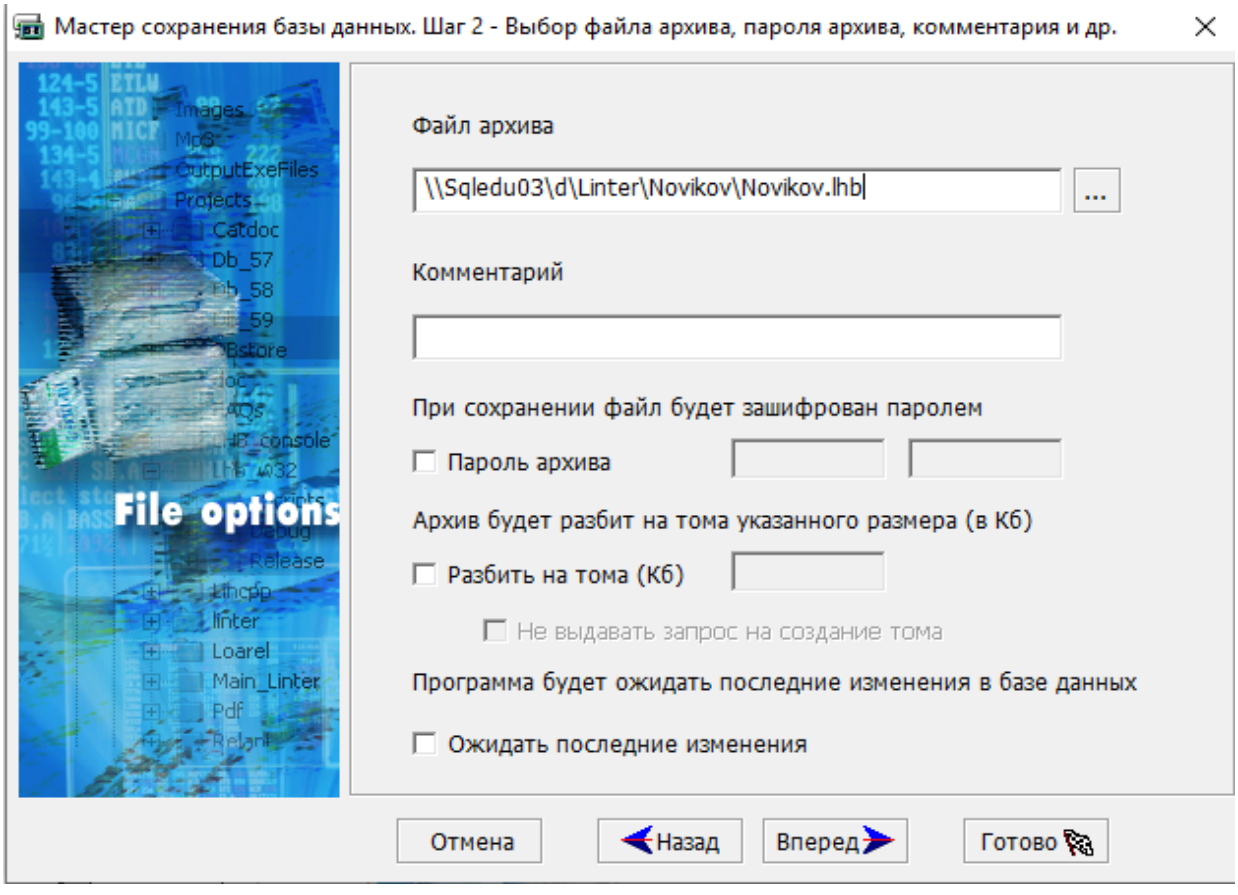


Рисунок 11.4 – Выбор файла

В этом окне выбран переключатель Полное сохранение. В следующем окне мастера была нажата кнопка «Готово» (рисунок 11.5). «Окно вывода» (рисунок 11.6) необходимо закрыть.

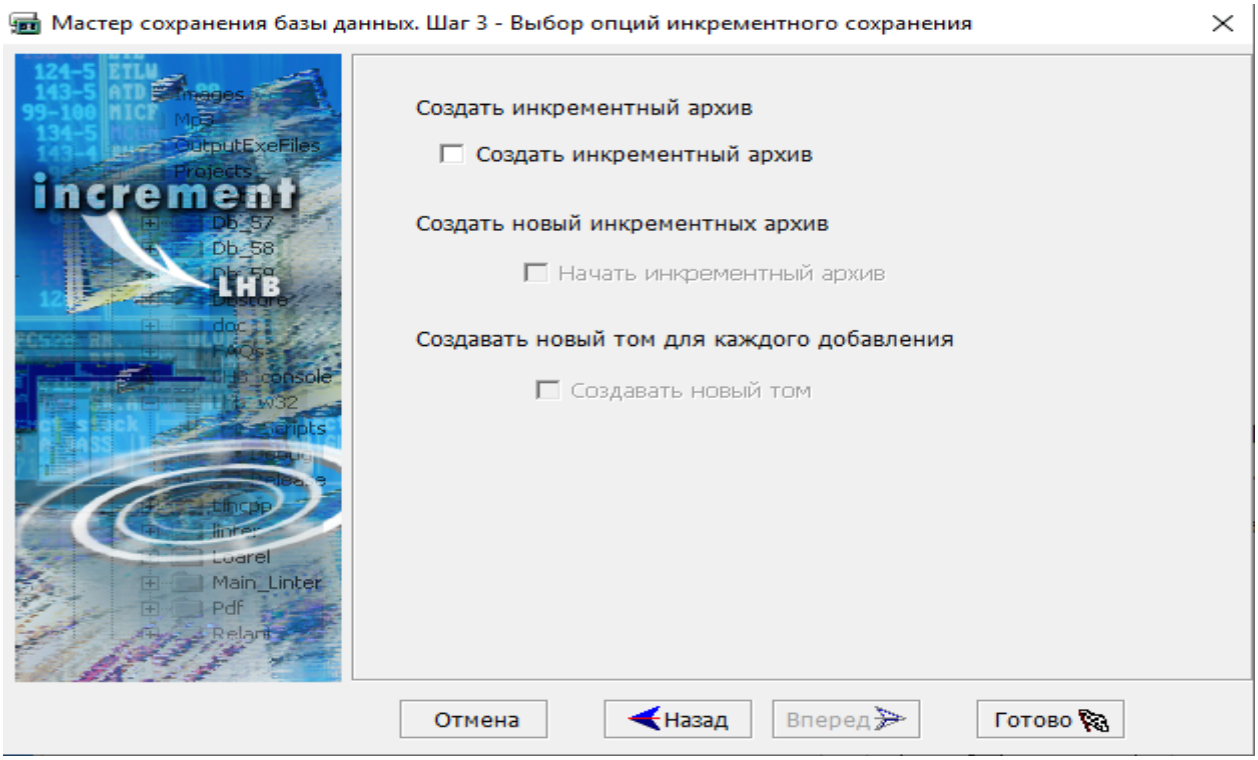


Рисунок 11.5 – Завершение работы мастера сохранения

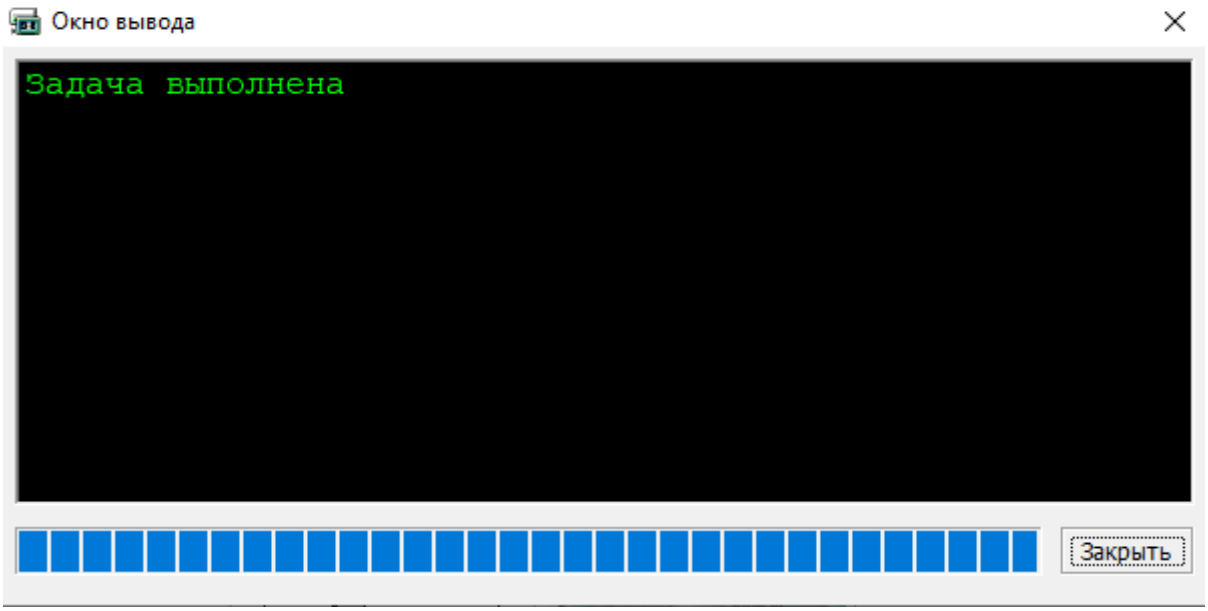


Рисунок 11.6 – Окно вывода

Для восстановления базы данных из ее архива была выполнена команда «Восстановление > Мастер восстановления». Далее открылось окно «Мастер восстановления базы данных». Шаг 1 — «Выбор файла архива» (рисунок 11.7).

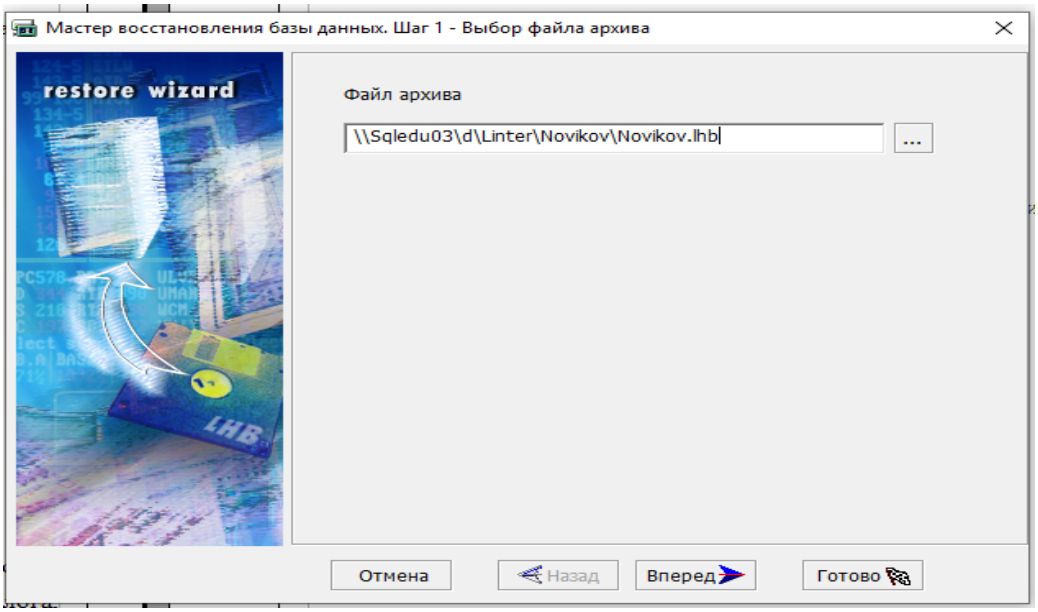


Рисунок 11.7 – Мастер восстановления

В этом окне выбран файл архива из выпадающего списка. Следующее окно мастера "Мастер восстановления базы данных. Шаг 2 — Выбор каталога, куда будут помещены файлы БД" (рисунок 11.8), в котором с помощью выпадающего меню был введен путь к папке Novikov, расположенной на сервере sqledu03.

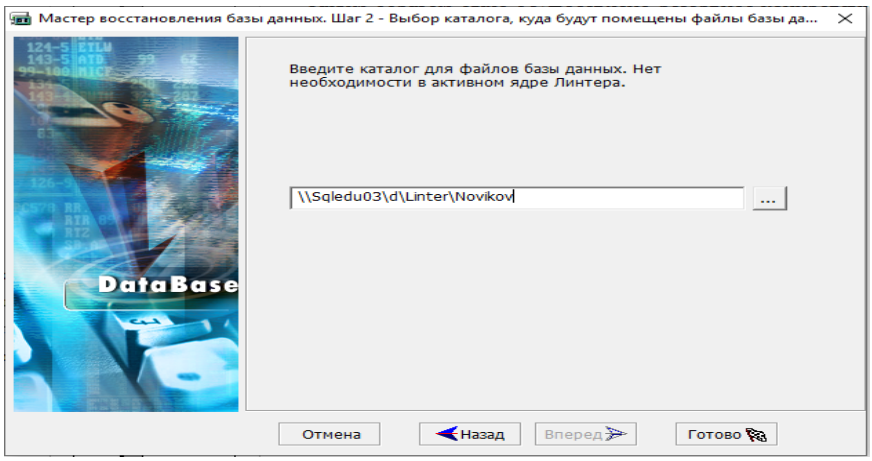


Рисунок 11.8 – Выбор каталога

В окне "Окно вывода" появилась информация о выполненном восстановлении базы данных (рисунок 11.9).

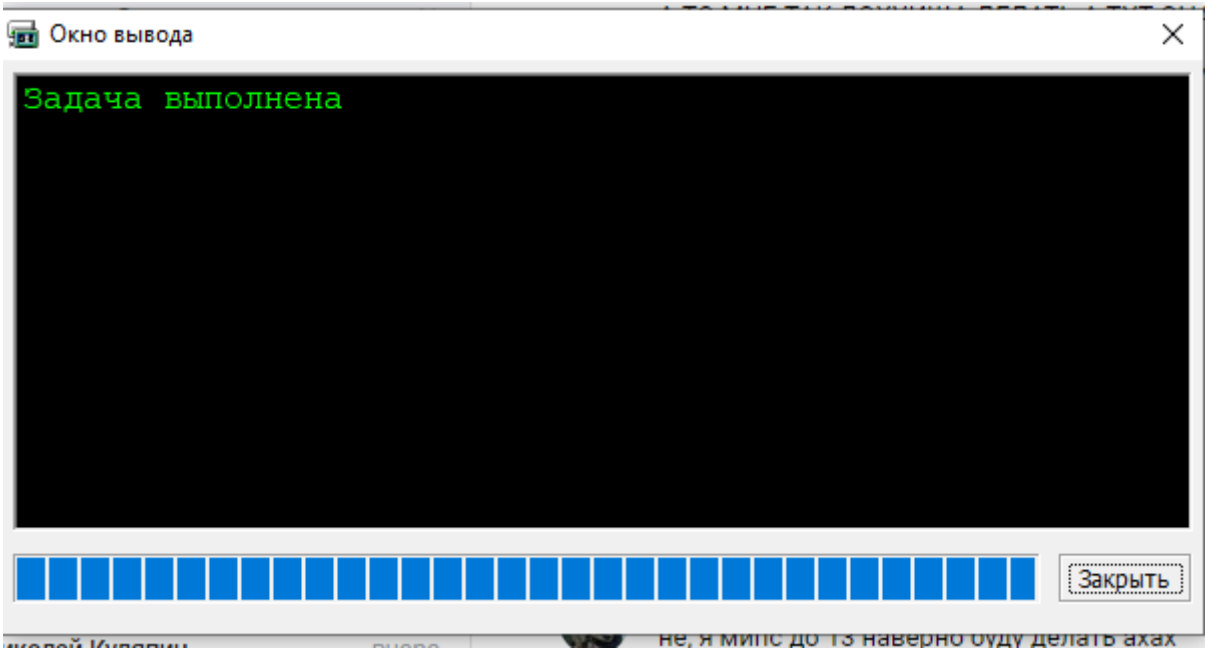


Рисунок 11.19 – Окно вывода

Таким образом было осуществлено резервное копирование и восстановление базы данных Novikov.

12 Тестирование базы данных

Для проверки доступа к созданной базе данных в дереве серверов приложения «Администратор СУБД ЛИНТЕР» была выделена база данных Novikov, которая была ранее запущена, затем выполнена команда меню «Сеть –> Таблица серверов». В появившемся окне «Редактирование таблицы серверов» была выделена база данных Novikov и нажата кнопка «Проверить» (рисунок 12.1). Далее была нажата кнопка «ОК», а затем – кнопка «Сохранить».

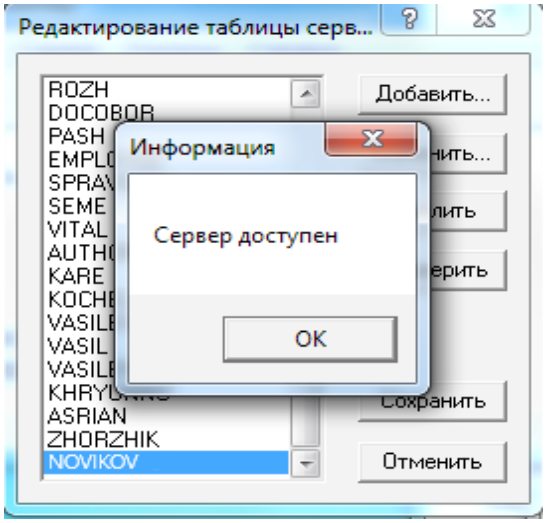


Рисунок 12.1 – Окно «Редактирование таблицы серверов»

Для проверки работы генератора GENERATEPK и таблицы SEQUENCES, а также созданных для её вызова триггеров INS_KLIENT, INS_AVTO, INS_ZAMENA и INS_EKSPED в таблицах KLIENT, AVTO, ZAMENA и EKSPED были добавлены записи (рисунки 12.7-13.2). Результаты изменения в таблице SEQUENCES представлены на рисунке 12.2.

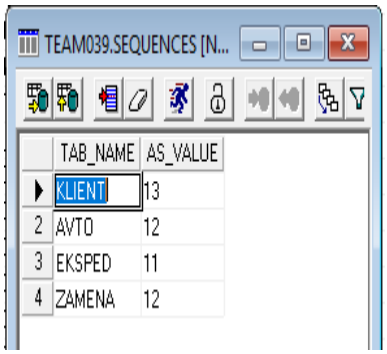


Рисунок 12.2 – Таблица SEQUENCES

Как видно из рисунка 12.2 значения колонка TAB_NAME хранит имена таблиц KLIENT, AVTO, ZAMENA и EKSPED, для которых генерируются уникальные значения, а колонка CUR_VALUE – текущее значение для генерации.

Для проверки работы таблицы KLIENT_HISTORY, а также созданного для их вызова триггера KLIENT_HISTORY_HIST в таблицу KLIENT были добавлены записи (рисунок 12.7). Результаты изменения в таблице KLIENT_HISTORY представлены на рисунке 12.3.

Рисунок 12.3 – Таблица KLIENT_HISTORY

Из рисунка 12.3 видно, что в поля таблицы заносится информация о фамилии клиента, дате добавления клиента.

Для проверки работы ограничения NOT NULL в поле FAM таблицы KLIENT было добавлено пустое значение. Результат работы ограничения представлен на рисунке 12.4.

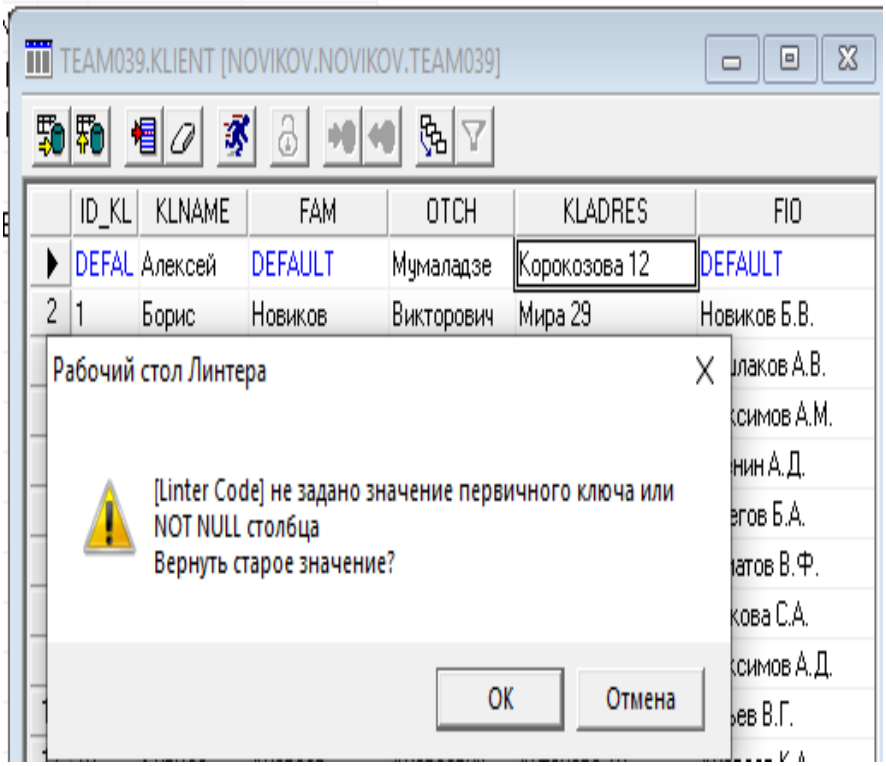


Рисунок 12.4 – Ограничение NOT NULL

Для проверки работы ограничения UNIQUE в поле NOMER таблицы AVTO было добавлено повторяющееся значение «A000AA». Результат работы ограничения представлен на рисунке 12.5.

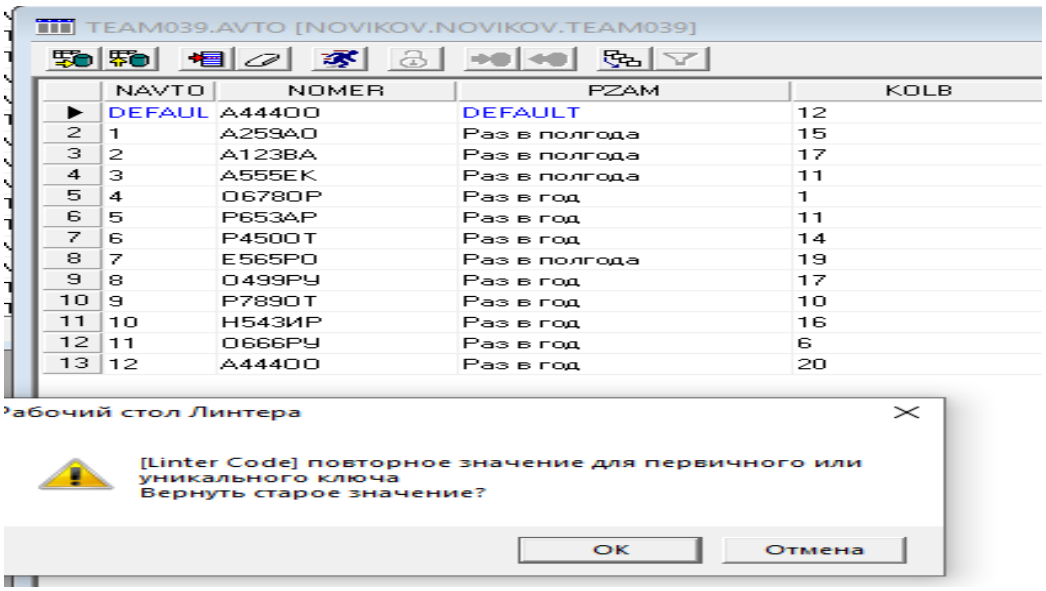


Рисунок 12.5 – Ограничение UNIQUE

Для проверки работы ограничения CHECK в поле KOLZB таблицы ZAMENA было добавлено значение «22». Результат работы ограничения представлен на рисунке 12.6.

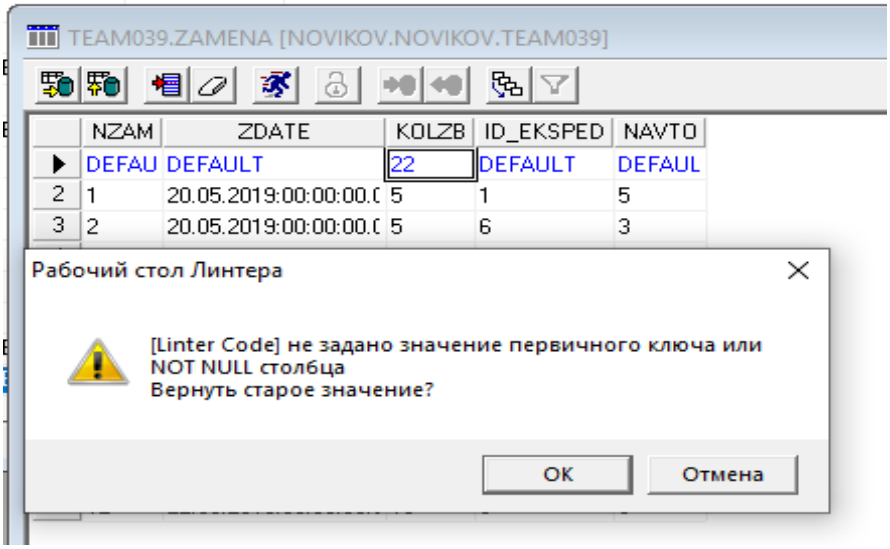
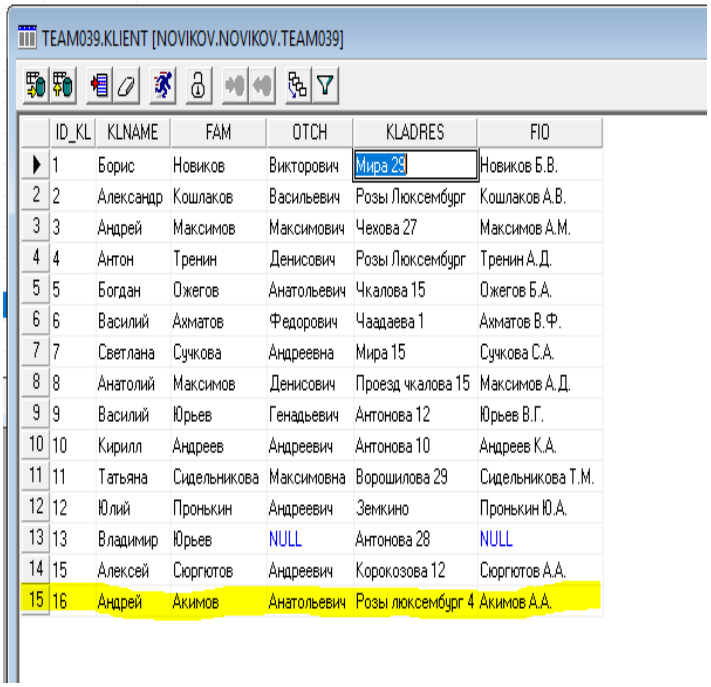


Рисунок 12.6 – Ограничение CHECK

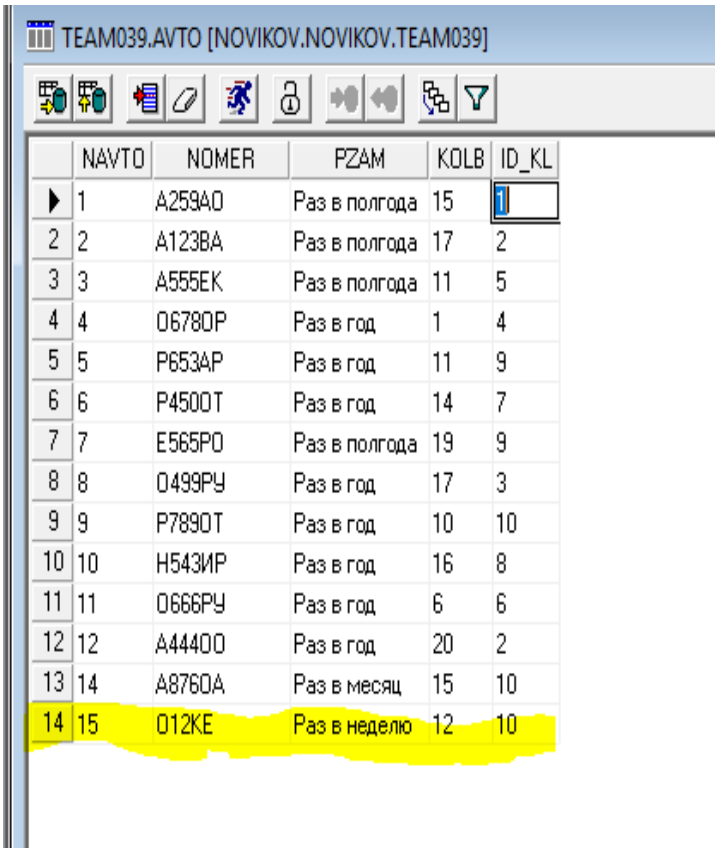
Для проверки правильности добавления записей в таблицу «KLIENT» в поле «KLNAME» было введено значение «Андрей»,в поле «FAM» было введено значение «Акимов», в поле «OTCH» было введено значение «Анатольевич», в поле «KLADRES» было введено значение «Розы люксембург 4». После чего была нажата кнопка «Сохранить». Результат работы добавления данных в таблицу «KLIENT» представлен на рисунке 12.7.



	ID_KL	KLNAME	FAM	OTCH	KLADRES	FIO
▶	1	Борис	Новиков	Викторович	Мира 29	Новиков Б.В.
2	2	Александр	Кошлаков	Васильевич	Розы Люксембург	Кошлаков А.В.
3	3	Андрей	Максимов	Максимович	Чехова 27	Максимов А.М.
4	4	Антон	Тренин	Денисович	Розы Люксембург	Тренин А.Д.
5	5	Богдан	Ожегов	Анатольевич	Чкалова 15	Ожегов Б.А.
6	6	Василий	Ахматов	Федорович	Чаадаева 1	Ахматов В.Ф.
7	7	Светлана	Сучкова	Андреевна	Мира 15	Сучкова С.А.
8	8	Анатолий	Максимов	Денисович	Проезд чкалова 15	Максимов А.Д.
9	9	Василий	Юрьев	Генадьевич	Антонова 12	Юрьев В.Г.
10	10	Кирилл	Андреев	Андреевич	Антонова 10	Андреев К.А.
11	11	Татьяна	Сидельникова	Максимовна	Ворошилова 29	Сидельникова Т.М.
12	12	Юлий	Пронькин	Андреевич	Земкино	Пронькин Ю.А.
13	13	Владимир	Юрьев	NULL	Антонова 28	NULL
14	15	Алексей	Сюргютов	Андреевич	Корокозова 12	Сюргютов А.А.
15	16	Андрей	Акимов	Анатольевич	Розы люксембург 4	Акимов А.А.

Рисунок 12.7 – Добавление данных в таблицу «KLIENT»

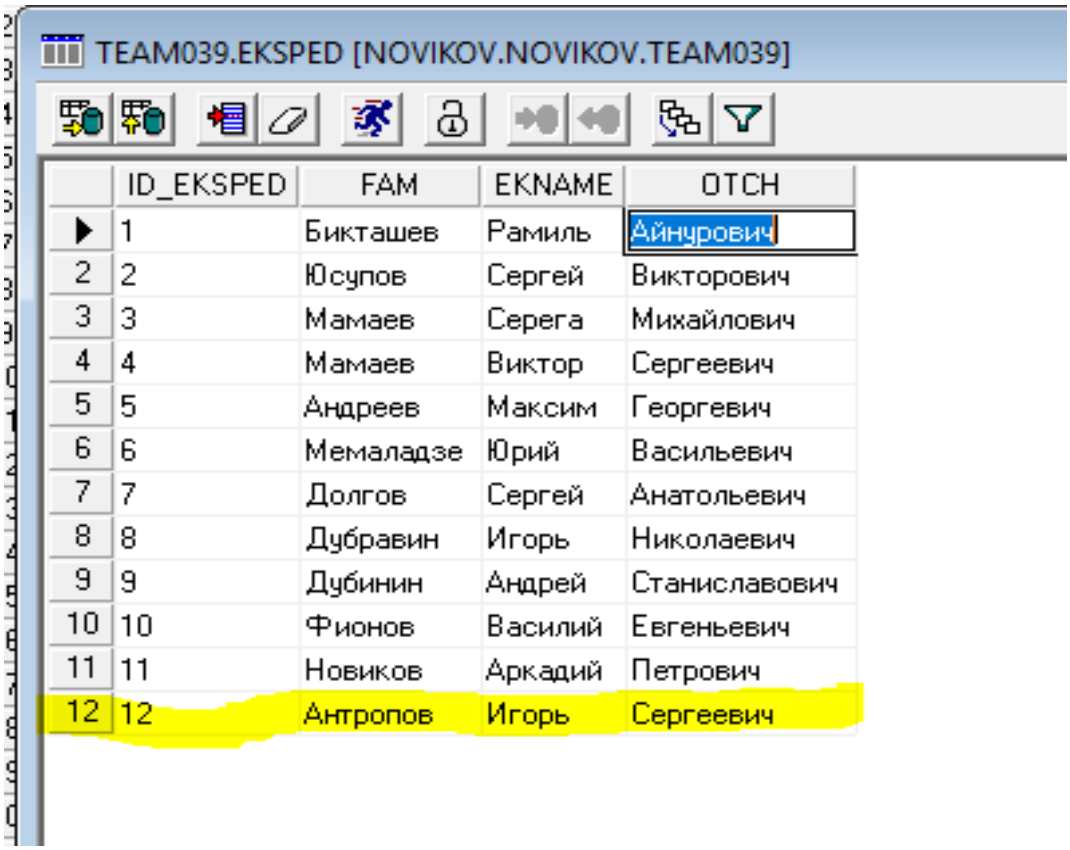
Для проверки правильности добавления записей в таблицу «AVTO» в поле «NOMER» было введено значение «O123KE», в поле «PZAM» было введено значение «Раз в неделю», в поле «KOLB» было введено значение «12», в поле «ID_KL» было введено значение «10». После чего была нажата кнопка «Сохранить». Результат работы добавления данных в таблицу «AVTO» представлен на рисунке 12.8.



	NAVTO	NOMER	PZAM	KOLB	ID_KL
▶ 1	1	A259AO	Раз в полгода	15	1
2	2	A123BA	Раз в полгода	17	2
3	3	A555EK	Раз в полгода	11	5
4	4	O678OP	Раз в год	1	4
5	5	P653AP	Раз в год	11	9
6	6	P450OT	Раз в год	14	7
7	7	E565PO	Раз в полгода	19	9
8	8	O499PУ	Раз в год	17	3
9	9	P789OT	Раз в год	10	10
10	10	H543ИP	Раз в год	16	8
11	11	O666PУ	Раз в год	6	6
12	12	A444OO	Раз в год	20	2
13	14	A876OA	Раз в месяц	15	10
14	15	O12KE	Раз в неделю	12	10

Риунок 12.8 – Добавление данных в таблицу «AVTO»

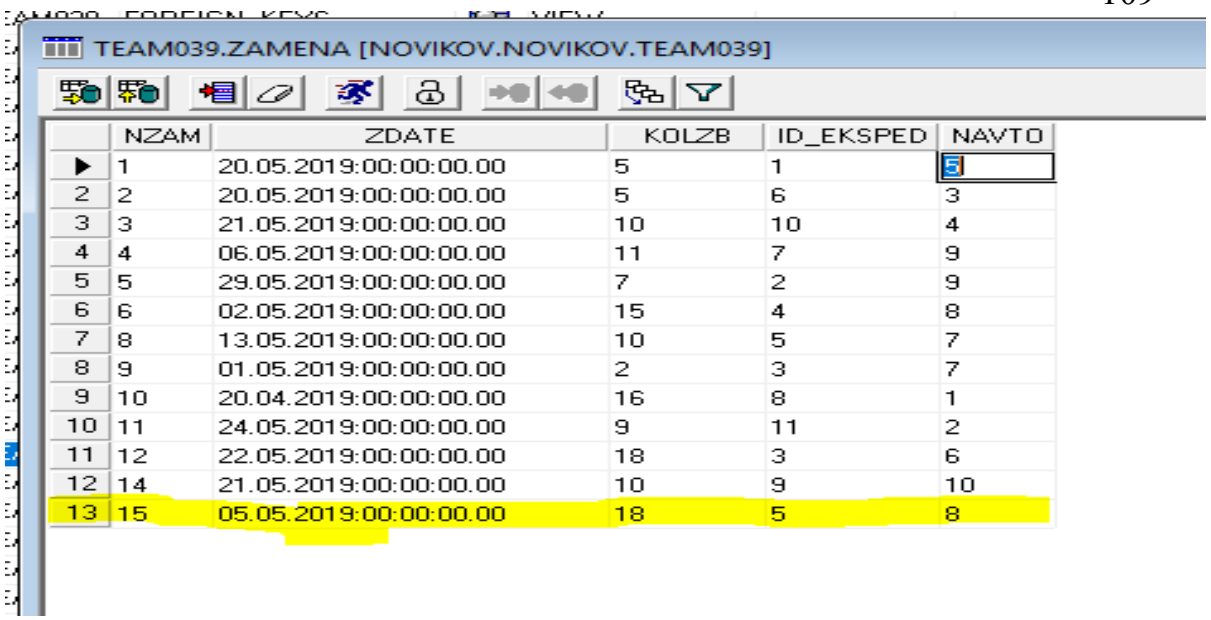
Для проверки правильности добавления записей в таблицу «EKSPED» в поле «EKNAME» было введено значение «Антропов», в поле «FAM» - «Игорь»,в поле «OTCH» - «Сергеевич». После чего была нажата кнопка «Сохранить». Результат работы добавления данных в таблицу «EKSPED» представлен на рисунке 12.9.



	ID_EKSPED	FAM	EKNAME	OTCH
▶	1	Бикташев	Рамиль	Айнурович
2	2	Юсупов	Сергей	Викторович
3	3	Мамаев	Серега	Михайлович
4	4	Мамаев	Виктор	Сергеевич
5	5	Андреев	Максим	Георгевич
6	6	Мемаладзе	Юрий	Васильевич
7	7	Долгов	Сергей	Анатольевич
8	8	Дубравин	Игорь	Николаевич
9	9	Дубинин	Андрей	Станиславович
10	10	Фионов	Василий	Евгеньевич
11	11	Новиков	Аркадий	Петрович
12	12	Антропов	Игорь	Сергеевич

Рисунок 12.9 – Добавление данных в таблицу «EKSPED»

Для проверки правильности добавления записей в таблицу «ZAMENA» в поле «ZDATE» было введено значение «05.05.2019», в поле «KOLZB» - «18»,в поле «ID_EKSPED» - «5», в поле «NAVTO» - «8». После чего была нажата кнопка «Сохранить». Результат работы добавления данных в таблицу «ZAMENA» представлен на рисунке 12.10.



TEAM039.ZAMENA [NOVIKOV.NOVIKOV.TEAM039]

	NZAM	ZDATE	KOLZB	ID_EKSPED	NAVTO
▶	1	20.05.2019:00:00:00.00	5	1	5
2	2	20.05.2019:00:00:00.00	5	6	3
3	3	21.05.2019:00:00:00.00	10	10	4
4	4	06.05.2019:00:00:00.00	11	7	9
5	5	29.05.2019:00:00:00.00	7	2	9
6	6	02.05.2019:00:00:00.00	15	4	8
7	8	13.05.2019:00:00:00.00	10	5	7
8	9	01.05.2019:00:00:00.00	2	3	7
9	10	20.04.2019:00:00:00.00	16	8	1
10	11	24.05.2019:00:00:00.00	9	11	2
11	12	22.05.2019:00:00:00.00	18	3	6
12	14	21.05.2019:00:00:00.00	10	9	10
13	15	05.05.2019:00:00:00.00	18	5	8

Рисунок 12.10 – Добавление данных в таблицу «ZAMENA»

Далее была проверена правильность работы хранимой процедуры «DETALI» и «PERZAM» . Для этого были вызваны хранимые процедуры «DETALI» и «PERZAM» с помощью команды call . Результат выполнения хранимых процедур представлен на рисунке 8.16 и рисунке 8.18.

Полученные результаты тестирования базы данных «Novikov» показали, что все требования технического задания выполнены в полном объеме, что свидетельствует о правильности работы базы данных «Novikov».

Заключение

В ходе выполнения курсового проекта была разработана база данных для автоматизированной системы «Сведения об учёте замены газовых баллонов». База данных была создана для СУБД Линтер с именем Novikov для пользователя с именем TEAM039 и паролем slave039.

Проведено концептуальное, логическое и физическое проектирование базы данных. Построены концептуальная, логическая и физическая модели базы данных.

Проектирование базы данных проведено с использованием CASE-средства Open ModelSphere 3.2 и инструментального средства СУБД Линтер «Рабочий стол Линтера».

Получены навыки использования приложений «Администратор СУБД Линтер», «Рабочий стол Линтера», «Резервное копирование» для создания, удаления, регистрации, подключения, извлечения метаданных, резервного копирования и восстановления базы данных СУБД ЛИНТЕР.

Были созданы следующие таблицы: KLIENT – для хранения данных о клиентах; AVTO — для хранения данных об автомобилях; EKSPED – для хранения данных об экспедиторах; ZAMENA – для хранения данных о заменах; SEQUENCES - для хранения значений первичных ключей таблиц; KLIENT_HIST – для хранения изменений в таблице KLIENT. Были созданы ограничения целостности, генераторы и триггеры для таблиц KLIENT, AVTO, EKSPED и ZAMENA. Произведена выборка из двух связанных таблиц «KLIENT» и «AVTO».

Проведено тестирование базы данных. Полученные результаты показали, что что все требования технического задания выполнены в полном объеме, что свидетельствует о правильности работы базы данных разработанной для автоматизированной системы «Сведения об учёте замены запчастей».

Список использованных источников

1.Осипов, Д.Л. InterBase и Delphi. Клиент-серверные базы данных [Электронный ресурс] — Электрон. дан. — Москва : ДМК Пресс, 2015. — 536 с.

2.Макарова Н.В. Информатика. Базовый курс [Текст]: учеб. пособие для бакалавров / Н. В. Макарова, В. Б. Волков. - СПб.: Питер, 2012. – 576 с. (49 экз.)

3.СУБД ЛИНТЕР. Официальный сайт. Документация. — Режим доступа: http://www.linter.ru/ru/documentation/

4.Руководящий документ. Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации». – Режим доступа: http:// fstec.ru /component / content /article/ 114 -deyatelnost/ tekushchaya /tekhnicheskaya -zashchita - in formatsii/normativnye – I – metodicheskie - dokumenty /spetsialnye-normativnye-dokumenty /384-rukovodyashchij- dokument -reshenie- predsedatelya - gosekhkomissii -rossii-ot-30-marta- 1992-g

Приложение А

(обязательное)

SQL-скрипт базы данных

```
CREATE OR REPLACE TABLE AVTO(
NAVTO INT PRIMARY KEY AUTOINC,
NOMER CHAR(10),
PZAM CHAR(30),
KOLB INTEGER,
ID_KL INT, FOREIGN KEY(ID_KL) REFERENCES KLIENT(ID_KL)
ON DELETE SET NULL
ON UPDATE CASCADE
);
CREATE OR REPLACE TABLE ZAMENA(
NZAM INT PRIMARY KEY AUTOINC,
ZDATE DATE,
KOLZB INTEGER,
ID_EKSPED INT, FOREIGN KEY(ID_EKSPED) REFERENCES EKSPED(ID_EKSPED),
NAVTO INT, FOREIGN KEY(NAVTO) REFERENCES AVTO(NAVTO)
ON DELETE SET NULL
ON UPDATE CASCADE
);
CREATE OR REPLACE TABLE EKSPED(
ID_EKSPED INT PRIMARY KEY AUTOINC,
FAM CHAR(30),
EKNAME CHAR(30),
OTCH CHAR(30)
);
CREATE OR REPLACE TABLE KLIENT(
ID_KL INT PRIMARY KEY AUTOINC,
KLNAME CHAR(30),
FAM CHAR(30),
OTCH CHAR(30),
KLADRES CHAR(40),
FIO CHAR(30) GENERATED BY DEFAULT AS (
FAM ||' '||
SUBSTRING(KLNAME FROM 1 FOR 1) ||'.'||
SUBSTRING(OTCH FROM 1 FOR 1) ||'.')
);
create or replace procedure detali (in det int) result int for debug
declare
var new_det int;
code
if det<=20 then
new_det:=det;
elseif det>20 then
new_det:=20;
```

```
endif
return new_det;
end;
create or replace procedure perzam (in pz char(30)) result char(30) for debug
declare
var new_pz char(30);
code
if pz=null then
new_pz:="Раз в год";
else new_pz:=pz;
endif
return new_pz;
end;
CREATE OR REPLACE TABLE SEQUENCES(
TAB_NAME VARCHAR(64) PRIMARY KEY NOT NULL,
AS_VALUE INT NOT NULL
);
create or replace procedure GeneratePK(in tabname char(64)) result int for debug
declare
var v cursor(value int);
code
execute direct "lock table SEQUENCES wait;";
execute direct "update SEQUENCES set AS_VALUE = AS_VALUE + 1 where TAB_NAME = '"+tabname+"';";
if rowcount() > 0 then
open v for direct "select AS_VALUE from SEQUENCES where TAB_NAME = '"+tabname+"';";
else
execute direct "insert into SEQUENCES values ('"+tabname+"',1);";
v.value := 1;
endif
execute direct "unlock table SEQUENCES;";
commit;
return v.value;
end;
create or replace trigger INS_AVTO before insert on AVTO
for each row execute for debug
code
call GeneratePK("AVTO") into new.NAVTO; //
end;
create or replace trigger INS_ZAMENA before insert on ZAMENA
for each row execute for debug
code
call GeneratePK("ZAMENA") into new.NZAM; //
end;
```

```
create or replace trigger INS_EKSPED before insert on EKSPED
for each row execute for debug
code
call GeneratePK("EKSPED") into new.ID_EKSPED; //
end;
create or replace trigger INS_KLIENT before insert on KLIENT
for each row execute for debug
code
call GeneratePK("KLIENT") into new.id_kl; //
end;
CREATE OR REPLACE TABLE KLIENT_HISTORY (id int, dt date, value varchar(40), status char(1));
ALTER TABLE KLIENT_HISTORY add primary key(id, dt);
CREATE OR REPLACE TRIGGER INS_KLIENT_HIST after insert on KLIENT
for each row execute for debug
code
execute direct
"insert into KLIENT_HISTORY(id, dt, value, status) values(" +
itoa(new.id_kl) + ", sysdate, '" + new.fam + "', 'I');";
end;
create or replace trigger INSS_AVTO before insert on AVTO
for each row execute for debug
code
call perzam(new.pzam) into new.pzam; //
end;
create or replace trigger INSSS_AVTO before insert on AVTO
for each row execute for debug
code
call detali(new.kolb) into new.kolb; //
end;
```