**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра вычислительной техники**

**КУРСОВАЯ РАБОТА**

**по индивидуальному домашнему заданию**

**По дисциплине «Базы данных»**

**Тема: «Станция техобслуживания»**

|  |  |
| --- | --- |
| Студент гр. 1306 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Тималёв Н.С. |
| Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Новакова Н.Е. |

Санкт-Петербург

2023

**АННОТАЦИЯ**

В процессе выполнения индивидуального домашнего задания будет спроектирована база данных для станции техобслуживания автомобилей. Будет описана предметная область, построена концептуальная логическая и физическая модели данных, создана база данных с дальнейшим заполнением таблиц. Будут созданы объекты промежуточного слоя: представления, UDF-ы и хранимые процедуры. Также будет создана стратегия резервного копирования.

**ANNOTATION**

In the process of completing the individual homework, a database for the car service station will be designed. The subject area will be described, a conceptual logical and physical data model will be built, a database will be created with further filling of tables. Middleware objects will be created: views, UDFs, and stored procedures. A backup strategy will also be created.

**СОДЕРЖАНИЕ**

Введение .............................................................................................................. 4

1. Описание предметной области....................................................................... 5

2. Проектирование базы данных (БД)................................................................ 5

2.1 Концептуальная модель данных............................................................... 5

2.2 Физическая модель данных....................................................................... 7

3. Создание БД.................................................................................................... 10

4. Создание таблиц............................................................................................. 11

5. Заполнение таблиц ........................................................................................ 13

6. Разработка объектов промежуточного слоя............................................... 16

7. Разработка стратегии резервного копирования ......................................... 22   
Заключение ....................................................................................................... 24 Список используемых источников.................................................................. 25

**ВВЕДЕНИЕ**

Целью проекта является укрепление теоретических знаний и получение практических навыков в проектировании баз данных.

Задачи:

– Краткое описание предметной области;

– Проектирование базы данных (БД);

– Создание БД;

– Создание таблицы и ограничений целостности к ним;

– Заполнение таблицы данными;

– Разработка объекта промежуточного слоя (представления, хранимые процедуры, UDF);

– Разработка стратегии резервного копирования.

*Формулировка задания:*

Спроектировать базу данных (БД) для работников станции технического обслуживания автомобилей, используя метод ER-диаграмм.

В БД должны храниться сведения о водителях, принадлежащих им автомобилях и данные о поломках; также должны храниться данные о работниках станции и время устранения неисправностей.

Сведения о водителях должны включать: ФИО, адрес, марку автомобиля, номер госрегистрации. Сведения об автомобилях должны включать: год выпуска, фирму-изготовитель, перечень устранённых неисправностей, ФИО работника, время устранения каждой неисправности. Предполагается, что каждый водитель обязательно имеет один или несколько автомобилей, а также может купить новый или продать старый.

**1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ**

База данных «СТАНЦИЯ ТЕХОБСЛУЖИВАНИЯ» предназначена для работников станции технического обслуживания автомобилей и обеспечивает в режиме диалога доступ к информации об автомобилях: номер госрегистрации, марка, тип кузова, VIN; о водителях: фамилия, имя-отчество, номер ВУ; о неисправностях: шифр, название, описание, время ремонта, цена. Предусмотрена возможность, как внесения изменений, так и получения справок с выводом информации на печать.

**2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ**

**2.1 Концептуальная модель данных**

После проведения анализа предметной области выделим следующие сущности:

1. ВОДИТЕЛЬ

Данная сущность хранит в себе информацию о водителе, сдавшем авто в ремонт и содержит такие атрибуты, как ФИО водителя и номер его ВУ. Потенциальный ключ – номер ВУ

2. АВТОМОБИЛЬ

Сущность хранит в себе информацию о транспортном средстве - номер госрегистрации, марка, тип кузова, VIN. Потенциальный ключ – VIN-код.

3. НЕИСПРАВНОСТЬ

Сущность хранит в себе информацию о разных неисправностях, такую как шифр, название, описание, время ремонта, цена. Потенциальный ключ - шифр

4. РАБОТНИК

С помощью данной сущности можно идентифицировать, кто занимался ремонтом той или иной неисправности. Хранит в себе табельный номер, ФИО, статус (в отпуске/работает/на больничном). Потенциальный ключ - Табельный номер

5. ЧЕК

Сущность хранит в себе набор информации о ремонте конкретной неисправности конкретного автомобиля - VIN автомобиля, шифр неисправности, дата начала ремонта, дата выдачи чека, табельный номер работника. Потенциальный ключ - дата выдачи

На основе представленных данных построим ER-диаграмму сущностей и отношений между ними

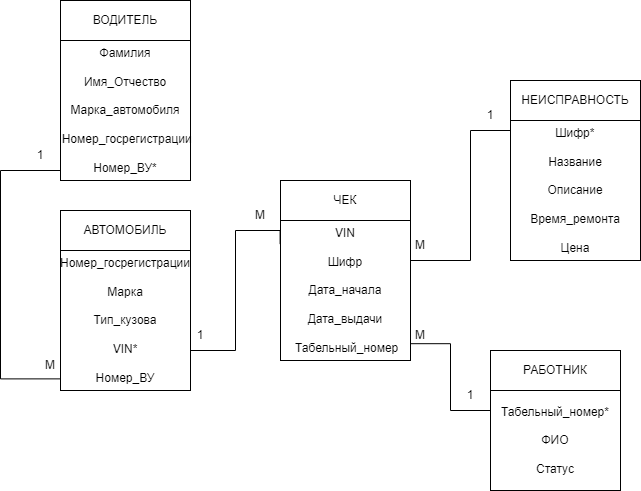
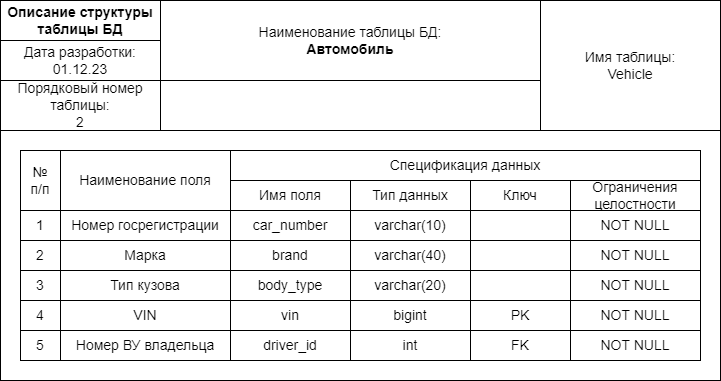
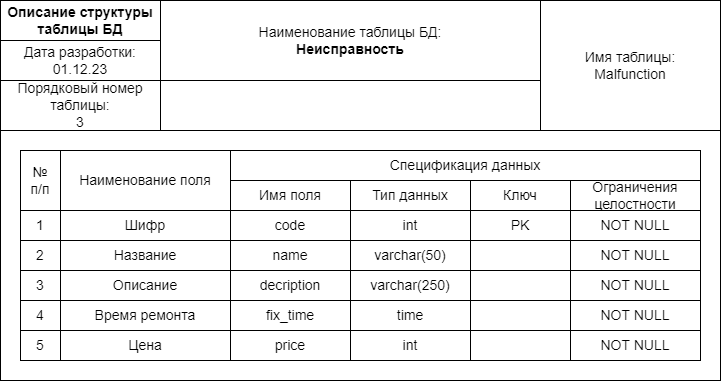
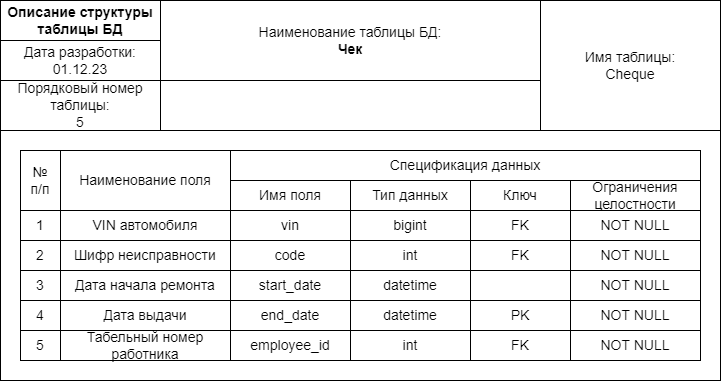
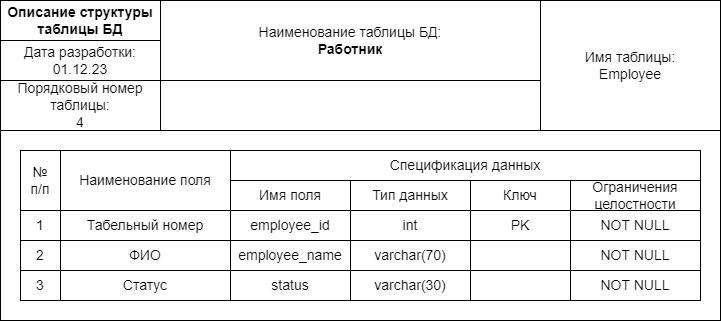


Рисунок 1

**2.2 Физическая модель данных**

Опишем данные в терминах T-SQL:  
  
  
  


**3. СОЗДАНИЕ БАЗЫ ДАННЫХ**

Создадим БД ServiceStation. Для этого создадим новый запрос, в котором укажем имя файла, размер файла, максимальный размер файла. Результат выполнения запроса представлен на рисунке 2.

CREATE DATABASE ServiceStation

ON (

name = ServiceStationDB,

filename = 'C:\Users\5286\Desktop\Study\БД\ServiceStation.mdf',

size = 8MB,

maxsize = 40MB,

filegrowth = 4MB

)

LOG ON (

name = ServiceStationDB\_log,

filename = 'C:\Users\5286\Desktop\Study\БД\ServiceStation\_log.ldf',

size = 8MB,

maxsize = 40MB,

filegrowth = 4MB

);

GO

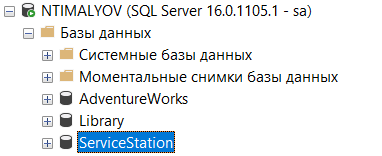


Рисунок 2

**4. СОЗДАНИЕ ТАБЛИЦ**

Создадим все таблицы и ограничения целостности, которые были определены в физической модели данных. Результат выполнения запроса представлен на рисунке 3.

CREATE SCHEMA Info

CREATE TABLE [Driver](

---defining columns

last\_name varchar(20) NOT NULL,

first\_name varchar(40) NOT NULL,

driver\_id bigint NOT NULL,

---restrictions

CONSTRAINT PK\_Driver PRIMARY KEY (driver\_id),

)

CREATE TABLE [Vehicle](

---defining columns

car\_number varchar(10) NOT NULL,

brand varchar(40) NOT NULL,

body\_type varchar(20) NOT NULL,

vin bigint NOT NULL,

driver\_id bigint NOT NULL,

---restrictions

CONSTRAINT PK\_Vehicle PRIMARY KEY (vin),

CONSTRAINT FK\_Vehicle FOREIGN KEY (driver\_id) REFERENCES Driver (driver\_id)

)

CREATE TABLE [Malfunction](

---defining columns

code int NOT NULL,

name varchar(50) NOT NULL,

description varchar(250) NOT NULL,

fix\_time time NOT NULL,

price int NOT NULL

---restrictions

CONSTRAINT PK\_Malfunction PRIMARY KEY (code)

)

CREATE TABLE [Employee](

---defining columns

employee\_id int NOT NULL,

employee\_name varchar(70) NOT NULL,

status varchar(30) NOT NULL,

---restrictions

CONSTRAINT PK\_Employee PRIMARY KEY (employee\_id)

)

CREATE TABLE [Cheque](

---defining columns

vin bigint NOT NULL,

code int NOT NULL,

start\_date datetime NOT NULL,

end\_date datetime NOT NULL,

employee\_id int NOT NULL

---restrictions

CONSTRAINT PK\_Cheque PRIMARY KEY (end\_date),

CONSTRAINT FK\_Cheque\_V FOREIGN KEY (vin) REFERENCES Vehicle (vin),

CONSTRAINT FK\_Cheque\_M FOREIGN KEY (code) REFERENCES Malfunction (code),

CONSTRAINT FK\_Cheque\_E FOREIGN KEY (employee\_id) REFERENCES Employee (employee\_id),

)

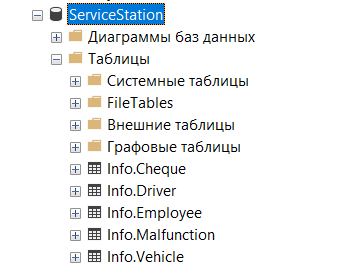


Рисунок 3

**5. ЗАПОЛНЕНИЕ ТАБЛИЦ**

Заполним все таблицы данными. Результаты выполнения запросов представлены на рисунках 4-13.

1. Driver

USE ServiceStation

INSERT INTO Info.Driver

VALUES

('Ivanov', 'Ilia Viktorovich', 708288277),

('Petrov', 'Viktor Sergeevich', 701397003),

('Vlasov', 'Aleksei Gennadevich', 508189924),

('Prokofev', 'Sergei Mikhailovich', 414292010),

('Zakharov', 'Nikolai Aleksandrovich', 907619042)

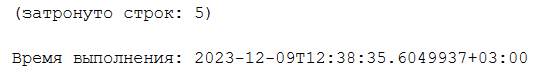


Рисунок 4

USE ServiceStation

Select \* From Info.Driver

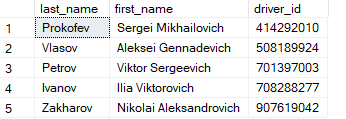


Рисунок 5

2. Vehicle

USE ServiceStation

INSERT INTO Info.Vehicle

VALUES

('A289PC', 'Opel Vectra', 'Wagon', 24453693697, 508189924),

('X006AE', 'Audi A6', 'Sedan', 776004128, 708288277),

('T988MK', 'Mercedes GLE', 'SUV', 3088656657, 414292010),

('K266MA', 'Volkswagen Touareg', 'SUV', 800050366148, 508189924)

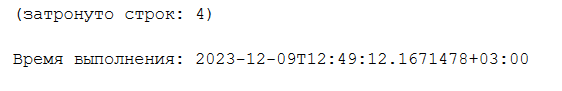


Рисунок 6

USE ServiceStation

Select \* From Info.Vehicle



Рисунок 7

3. Malfunction

USE ServiceStation

INSERT INTO Info.Malfunction

VALUES

(1357, 'Timing belt breaking', 'A belt rupture causes the camshaft to stop. As a result, the valves remain stationary, being in the position they were in at the time of the breakage. At the same time, the crankshaft continues to rotate, moving the pistons up and down.', '01:30:00', 12000),

(2425, 'Battery discharge', 'The car does not respond to attempts to unlock the alarm, then the engine does not start, the headlights do not turn on, and the electric consumers will also not function', '00:15:00', 1700),

(3742, 'Malfunction of the gas pump', 'Noise, knocking from the gas tank. The power unit does not start, but there are flashes in the cylinders. Candles stay dry.', '00:45:00', 2750),

(8891, 'Dent on the wing. 1 class', 'The dent is shallow, small in size, there are no chips on the damaged area, the coating layer is not cracked', '00:50:00', 6000),

(8892, 'Dent on the wing. 2 class', 'The dent is deep, big in size or there are chips on the damaged area, the coating layer is cracked', '02:10:00', 15000),

(2738, 'Contamination of the particulate filter', 'The speed of rotation of the turbocharger shaft decreases, which leads to under-inflation. There is also an overpressure in the turbine housing', '01:00:00', 7000)

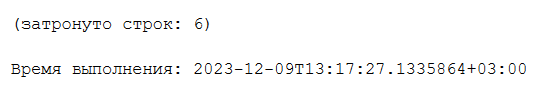


Рисунок 8

USE ServiceStation

Select \* From Info.Malfunction

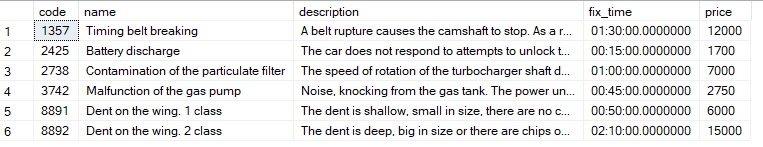


Рисунок 9

4. Employee

USE ServiceStation

INSERT INTO Info.Employee

VALUES

(10, 'IUsupov Marat Limarovich', 'On sick leave'),

(11, 'Karpov Sergei Olegovich', 'In working mode'),

(12, 'Dorofeev Viktor Alekseevich', 'On vacation'),

(15, 'Kovalyov Viktor Petrovich', 'In working mode')

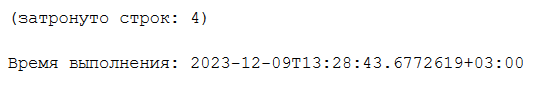


Рисунок 10

USE ServiceStation

Select \* From Info.Employee

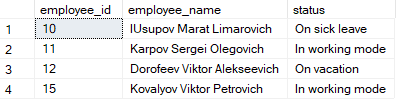


Рисунок 11

5. Cheque

USE ServiceStation

INSERT INTO Info.Cheque

VALUES

(24453693697, 1357, '2023-10-17T09:37:45', '2023-10-17T11:07:45', 11),

(776004128, 2425, '2023-10-23T14:18:02', '2023-10-23T14:33:02', 12),

(3088656657, 8891, '2023-10-22T12:21:21', '2023-10-22T13:11:21', 15),

(800050366148, 3742, '2023-10-25T17:46:38', '2023-10-25T18:31:38', 15),

(800050366148, 2738, '2023-10-25T18:31:38', '2023-10-25T19:31:38', 10)

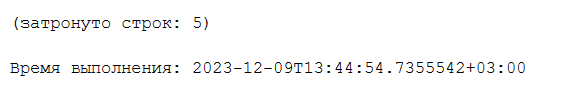


Рисунок 12

USE ServiceStation

Select \* From Info.Cheque

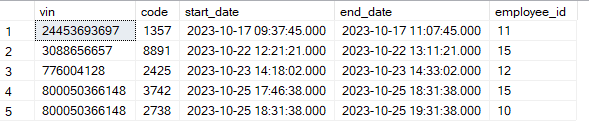


Рисунок 13

**6. РАЗРАБОТКА ОБЪЕКТОВ ПРОМЕЖУТОЧНОГО СЛОЯ**

1. UDF Info.DonePerDay

Функция предназначена для вывода работ, выполненных в вводимый пользователем день

CREATE FUNCTION Info.DonePerDay (@DateToCheck date)

RETURNS TABLE

AS

RETURN SELECT \* FROM Info.Cheque

WHERE @DateToCheck = CONVERT(date, Info.Cheque.start\_date)

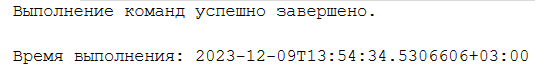


Рисунок 14

Проверим работу функции

USE ServiceStation

SELECT \* FROM Info.DonePerDay('2023-10-25')

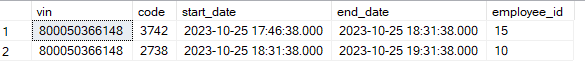


Рисунок 15

2. Представление Info.IsAvailable

Представление выводит информацию о работниках, которые не находятся на больничном или в отпуске

CREATE VIEW Info.IsAvailable

AS

SELECT \* FROM Info.Employee

WHERE Info.Employee.status LIKE 'In working mode'

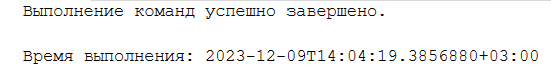


Рисунок 16

Проверим работу представления

USE ServiceStation

SELECT \* FROM Info.IsAvailable



Рисунок 17

3. UDF Info.SalesPerDay

Функция выводит информацию о работах и цене за вводимый пользователем день

CREATE FUNCTION Info.SalesPerDay (@DateToCheck date)

RETURNS TABLE

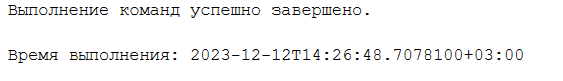
AS

RETURN SELECT Info.Malfunction.name, SUM(price) AS 'Сумма за день'

FROM Info.Cheque LEFT JOIN Info.Malfunction ON (Info.Cheque.code = Info.Malfunction.code)

WHERE @DateToCheck = CONVERT(date, Info.Cheque.end\_date)

GROUP BY Info.Cheque.end\_date, Info.Malfunction.name



SELECT \* FROM Info.SalesPerDay ('2023-10-25')



4. UDF Info.JobsCount

Функция выводит информацию о том, сколько чеков было выдано за месяц. На вход подаётся номер месяца

CREATE FUNCTION Info.JobsCount (@Month int)

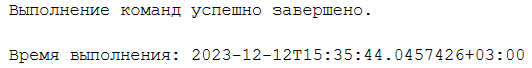
RETURNS TABLE

AS

RETURN SELECT COUNT(Info.Cheque.end\_date) AS 'Количество чеков'

FROM Info.Cheque

WHERE @Month = MONTH(Info.Cheque.end\_date)



Проверим правильность выполнения функции

USE ServiceStation

SELECT \* FROM Info.JobsCount(10)



6. UDF Info.EmployeesDone

Функция помогает определить, сколько поломок отремонтировал каждый работник за определённый месяц (на вход подаётся номер месяца)

CREATE FUNCTION Info.EmployeesDone (@Month int)

RETURNS TABLE

AS

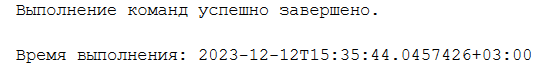
RETURN SELECT Info.Employee.employee\_name, COUNT(Info.Cheque.end\_date) AS 'Количество'

FROM

Info.Cheque LEFT JOIN Info.Employee ON (Info.Cheque.employee\_id = Info.Employee.employee\_id)

WHERE @Month = MONTH(Info.Cheque.end\_date)

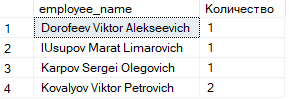
GROUP BY Info.Cheque.employee\_id, Info.Employee.employee\_name



Проверим правильность выполнения функции

USE ServiceStation

SELECT \* FROM Info.EmployeesDone(10)



7. Представление Info.MalfFrequency

Показывает статистику поломок авто (сколько поломок у каждой марки машины, поступившей в сервис)

CREATE VIEW Info.MalfFrequency

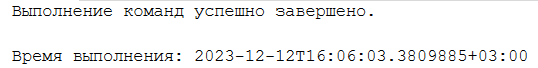
AS

SELECT Info.Vehicle.brand, COUNT(Info.Cheque.end\_date) AS 'Число поломок'

FROM

Info.Cheque LEFT JOIN Info.Vehicle ON (Info.Cheque.vin = Info.Vehicle.vin)

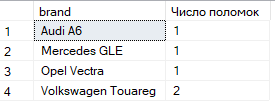
GROUP BY Info.Vehicle.brand



Проверим работу представления

USE ServiceStation

SELECT \* FROM Info.MalfFrequency



8. UDF Info.VisitsPerYear

UDF показывает, сколько раз водитель обращался в сервис за последние 12 месяцев. На вход подаётся номер ВУ водителя

CREATE FUNCTION Info.VisitsPerYear (@DL int)

RETURNS TABLE

AS

RETURN SELECT COUNT(Info.Cheque.vin) AS 'Количество обращений'

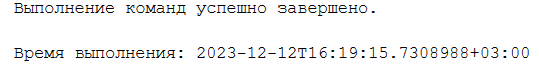
FROM

Info.Cheque

LEFT JOIN Info.Vehicle ON (Info.Cheque.vin = Info.Vehicle.vin)

LEFT JOIN Info.Driver ON (Info.Vehicle.driver\_id = Info.Driver.driver\_id)

WHERE Info.Driver.driver\_id = @DL



Проверим правильность работы функции

USE ServiceStation

SELECT \* FROM Info.VisitsPerYear(708288277)



9. UDF Info.FixedBy

Функция помогает определить, какой работник ремонтировал автомобиль определённого водителя. На вход подаётся фамилия водителя

CREATE FUNCTION Info.FixedBy (@Last\_name varchar(30))

RETURNS TABLE

AS

RETURN SELECT Info.Driver.last\_name AS 'Фамилия клиента', Info.Driver.first\_name AS 'ИО клиента', Info.Employee.employee\_name AS 'ФИО работника'

FROM

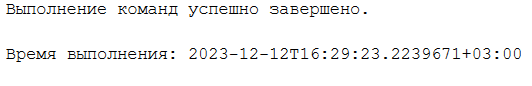
Info.Driver

JOIN Info.Vehicle ON (Info.Vehicle.driver\_id = Info.Driver.driver\_id)

JOIN Info.Cheque ON (Info.Cheque.vin = Info.Vehicle.vin)

JOIN Info.Employee ON (Info.Employee.employee\_id = Info.Cheque.employee\_id)

WHERE Info.Driver.last\_name = @Last\_name



Проверим правильность работы функции

USE ServiceStation

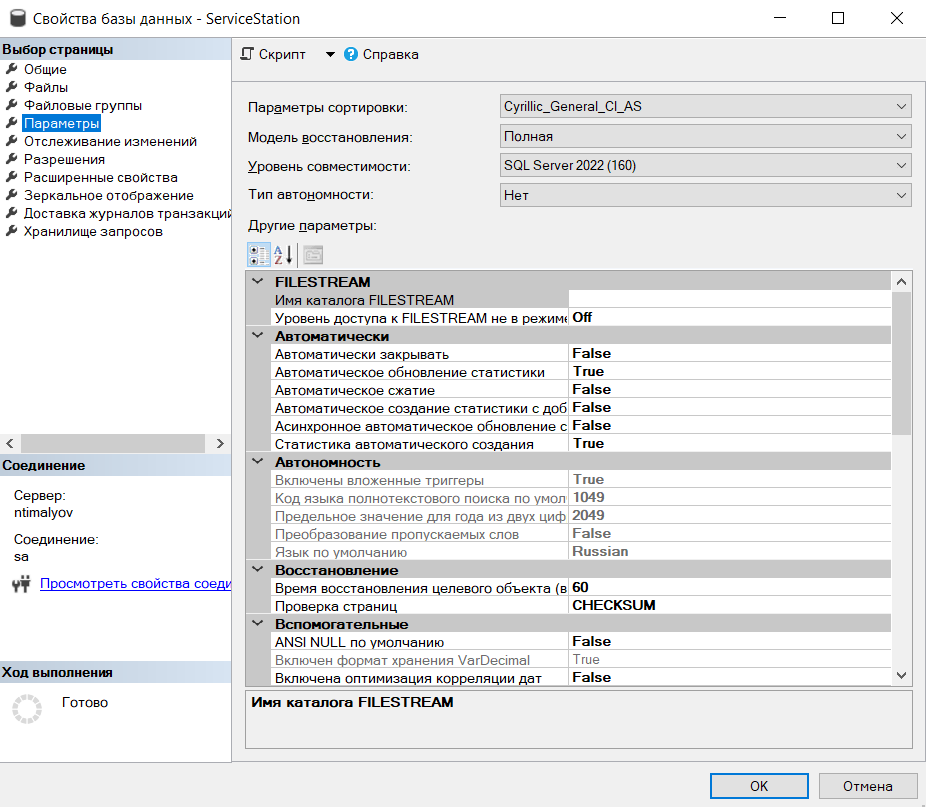
SELECT \* FROM Info.FixedBy('Prokofev')



**7. РАЗРАБОТКА СТРАТЕГИИ РЕЗЕРВНОГО КОПИРОВАНИЯ**

Перед началом изменим модель восстановления базы данных на

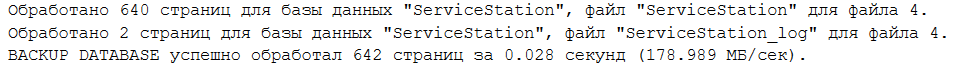
«Полную».



При небольшом количестве записей резервное копирование можно производить ежедневно в определённое время

BACKUP DATABASE ServiceStation

TO DISK = 'C:\Users\5286\Desktop\Study\БД\ServiceStation.bak'



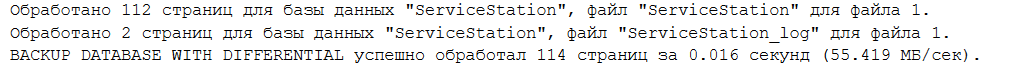
Если количество записей велико, полное резервное копирование можно производить раз в неделю, ежедневное копирование же заменить разностным резервным копированием

USE ServiceStation

BACKUP DATABASE ServiceStation

TO DISK = 'C:\Users\5286\Desktop\Study\БД\ServiceStation\_diff.bak'

WITH DIFFERENTIAL



Также производится резервное копирование журнала транзакций

BACKUP LOG ServiceStation

TO DISK = 'C:\Users\5286\Desktop\Study\БД\ServiceStation\_log.bak'



Для восстановления базы данных из резервной копии с помощью инструкции RESTORE необходимо указать параметр WITH NORECOVERY. При восстановлении журнала транзакций этот параметр не указывается

RESTORE DATABASE EmployeesOfCompany

FROM DISK = 'C:\Users\5286\Desktop\Study\БД\ServiceStation.bak'

WITH NORECOVERY

RESTORE DATABASE EmployeesOfCompany

FROM DISK = 'C:\Users\5286\Desktop\Study\БД\ServiceStation\_diff.bak'

WITH NORECOVERY

RESTORE LOG EmployeesOfCompany

FROM DISK = 'C:\Users\5286\Desktop\Study\БД\ServiceStation\_log.bak'

WITH RECOVERY

**ЗАКЛЮЧЕНИЕ**

В ходе выполнения индивидуального домашнего задания были закреплены теоретические знания и практические навыки, полученные в ходе изучения курса «Базы данных».

Была спроектирована база данных для станции технического обслуживания, в которой хранятся сведения об автомобилях, их владельцах, поломках, работниках и выполненных работах. В результате фактической реализации базы данных были созданы таблицы с ограничением целостности, разработаны объекты промежуточного слоя, также продемонстрирована их работоспособность. Разработана стратегия резервного копирования.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Распределенные базы данных: Методические указания к лабораторным работам / Сост.: А. В. Горячев, Н. Е. Новакова. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2008. 32 с
2. Справочник по Transact-SQL / Microsoft Learn [Электронный ресурс] Режим доступа: https://learn.microsoft.com/ru-ru/sql/t-sql/ (дата обращения 03.12.2022).