ГУО «БГУИР»

Инженерно-экономический факультет

Кафедра экономической информатики

Отчет по

Лабораторной работе №3

Выполнение запросов к базе данных

Подготовил:

Студент гр.914302

Понкратов А.М.

Проверила:

Лукашевич А.Э.

Минск 2022

**Вариант 18**

**Цель практической работы:** изучить создание таблиц с помощью операторов языка T-SQL, создание запросов на выборку, многотабличных, запросов на соединение, запросов на сортировку, группировку и обновление данных, научиться использовать встроенные функции

**Часть 1**

1. **Создать файл базы данных, согласно номеру варианта, выданного в лабораторной работе №1 с помощью sql-команды.**

SQL-команда:

CREATE DATABASE ATE

ON PRIMARY

( NAME = ATE\_data,

FILENAME = 'E:\SIMUBD\DB\ate\_data.mdf',

SIZE = 10MB,

MAXSIZE = 75MB,

FILEGROWTH = 3MB ),

FILEGROUP Secondary

( NAME = ATE2\_data,

FILENAME = 'E:\SIMUBD\DB\ate\_data2.ndf',

SIZE = 3MB,

MAXSIZE = 50MB,

FILEGROWTH = 15% ),

( NAME = ATE3\_data,

FILENAME = 'E:\SIMUBD\DB\ate\_data3.ndf',

SIZE = 4MB,

FILEGROWTH = 4MB )

LOG ON

( NAME = ATE\_log,

FILENAME = 'E:\SIMUBD\DB\ate\_log.ldf',

SIZE = 1MB,

MAXSIZE = 10MB,

FILEGROWTH = 20% ),

( NAME = ATE2\_log,

FILENAME = 'E:\SIMUBD\DB\ate\_log2.ldf',

SIZE = 512KB,

MAXSIZE = 15MB,

FILEGROWTH = 10% )

GO

1. **Создать все таблицы базы данных, ключи, ограничения и связи.**

CREATE TABLE ate (

id\_ate INT PRIMARY KEY,

name\_ate VARCHAR(100) NOT NULL,

type\_ate VARCHAR(13) NOT NULL,

[address] VARCHAR(100),

city VARCHAR(50),

qty\_abonents INT

)

CREATE TABLE abonent (

id\_abonent INT PRIMARY KEY,

last\_name VARCHAR(100) NOT NULL,

first\_name VARCHAR(100) NOT NULL,

patronymic VARCHAR(100),

type\_phone VARCHAR(12) NOT NULL,

phone\_number VARCHAR(8) NOT NULL,

is\_intercity BIT NOT NULL,

benefit BIT NOT NULL,

[address] VARCHAR(100)

)

CREATE TABLE call\_sheet (

id\_call INT PRIMARY KEY,

id\_abonent INT,

time\_start DATETIME NOT NULL,

time\_end DATETIME NOT NULL,

is\_intercity BIT NOT NULL,

CONSTRAINT FK\_CallSheet\_Abonent FOREIGN KEY (id\_abonent) REFERENCES abonent ON UPDATE CASCADE ON DELETE SET NULL

)

CREATE TABLE ate\_price (

id\_ate INT PRIMARY KEY,

price\_city FLOAT,

price\_intercity FLOAT,

CONSTRAINT FK\_AtePrice\_Ate FOREIGN KEY (id\_ate) REFERENCES ate ON UPDATE CASCADE ON DELETE CASCADE

)

CREATE TABLE subscription\_statement\_sheet (

id\_record INT PRIMARY KEY,

id\_ate INT NOT NULL,

id\_abonent INT NOT NULL,

[month] INT NOT NULL,

[year] INT NOT NULL,

minutes\_city INT,

minutes\_intercity INT,

price FLOAT NOT NULL,

benefits\_total FLOAT NOT NULL,

price\_total FLOAT NOT NULL,

CONSTRAINT FK\_SubscriptionStatementSheet\_Abonent FOREIGN KEY (id\_abonent) REFERENCES abonent ON UPDATE CASCADE ON DELETE NO ACTION,

CONSTRAINT FK\_SubscriptionStatementSheet\_Ate FOREIGN KEY (id\_ate) REFERENCES ate ON UPDATE CASCADE ON DELETE NO ACTION

)

1. **Заполните таблицы данными не менее 5 записей в каждой.**

SQL-команды:

INSERT INTO ate VALUES

(‘Советский’, ‘Городская’, ‘Якуба Коласа 35’, ‘Минск’, 200000),

(‘Центральный’, ‘Городская’, ‘Бобруйская 3’, ‘Минск’, 605000),

(‘Фрунзенский’, ‘Городская’, ‘Лобанка 17’, ‘Минск’, 270651),

(‘Первомайский’, ‘Городская’, ‘Восточная 57’, ‘Минск’, 402873),

(‘Заводской’, ‘Городская’, ‘Яна Райниса 63’, ‘Минск’, 368742)

INSERT INTO abonent VALUES

(‘Понкратов’, ‘Алексей’, ‘Михайлович’, ‘Основной’, ‘3708567’, 1, 0, ‘Якуба Коласа 28-14’),

(‘Сморгун’, ‘Евгений’, ‘Святославович’, ‘Спаренный’, ‘5675217’, 1, 0, ‘Яна Райниса 56-78’),

(‘Борисенко’, ‘Александр’, ‘Игоревич’, ‘Параллельный’, ‘8885522’, 1, 1, ‘Гикало 18-90’),

(‘Воробей’, ‘Дарья’, ‘Антоновна’, ‘Основной’, ‘3708567’, 1, 0, ‘пр.Дзержинского 90-67’),

(‘Тимошенко’, ‘Илья’, ‘Николаевич’, ‘Основной’, ‘3708567’, 1, 0, ‘Якуба Коласа 28-45’)

INSERT INTO call\_sheet VALUES

(1, ‘2022-09-26 10:06:19’, ‘2022-09-26 10:08:58’, 0),

(2, ‘2022-09-26 11:45:02’, ‘2022-09-26 11:59:28’, 1),

(3, ‘2022-09-26 14:24:07’, ‘2022-09-26 14:24:40’, 0),

(4, ‘2022-09-26 16:55:41’, ‘2022-09-26 17:02:10’, 0),

(5, ‘2022-09-26 19:30:00’, ‘2022-09-26 19:36:35’, 1)

INSERT INTO ate\_price VALUES

(1, 0.01, 0.02),

(2, 0.02, 0.015),

(3, 0.008, 0.028),

(4, 0.009, 0.018),

(5, 0.011, 0.021)

INSERT INTO subscription\_statement\_sheet VALUES

(1, 1, 8, 2022, 105, 43, 1.91, 0, 1.91),

(2, 2, 8, 2022, 25, 10, 0.65, 0, 0.65),

(3, 3, 8, 2022, 84, 39, 1.764, 0.882, 0.882),

(4, 4, 8, 2022, 12, 156, 2.916, 0, 2.916),

(5, 5, 8, 2022, 0, 10, 0.21, 0, 0.21)

Результаты заполнения таблиц:

Таблица “ate”

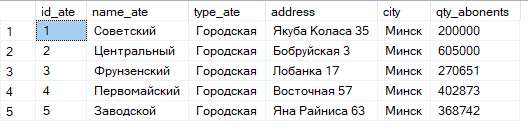


Таблица “abonent”

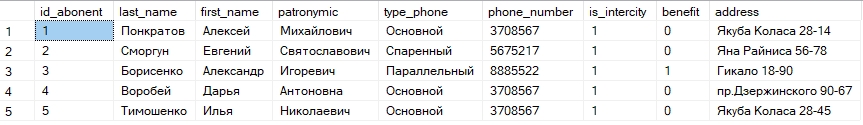


Таблица “call\_sheet”

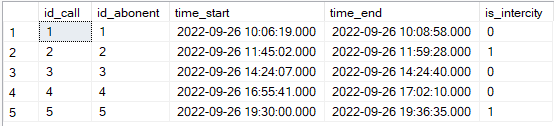
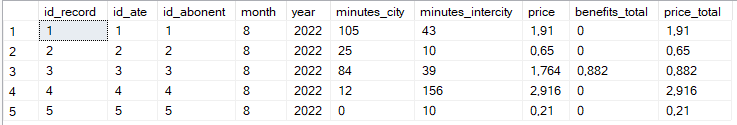


Таблица “ate\_price”

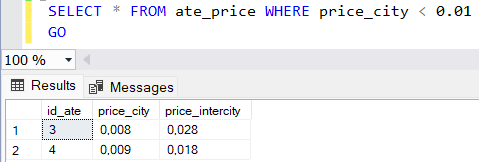


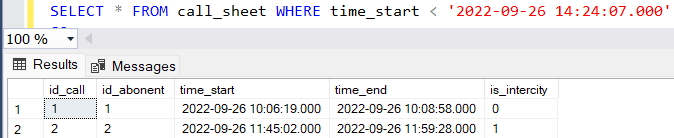
Таблица “subscription\_statement\_sheet”

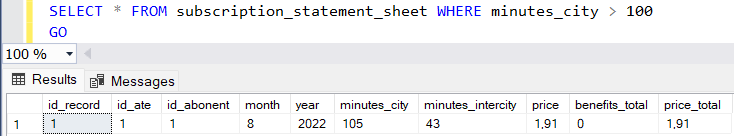


**Часть 2**

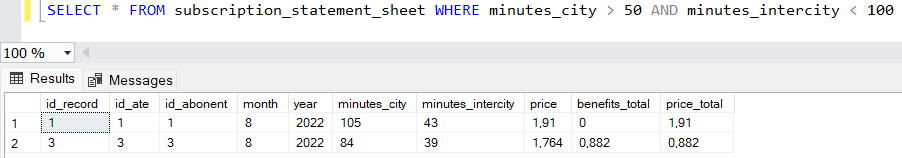
1. 3 простейших запроса с оператором сравнения

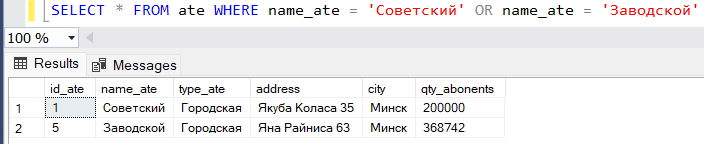


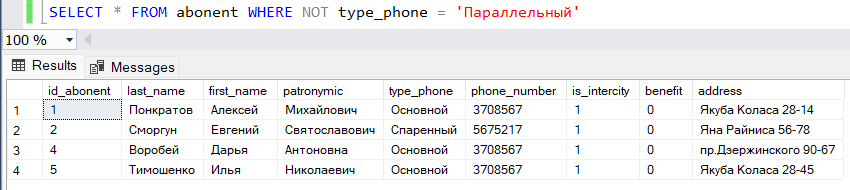




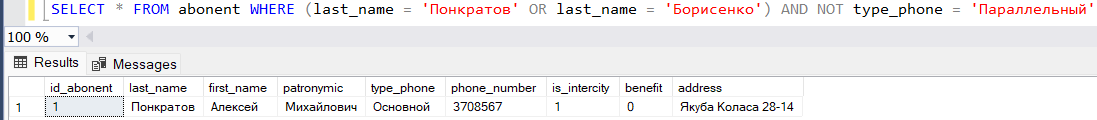
1. -3 запроса с использованием логических операторов AND, OR и NOT

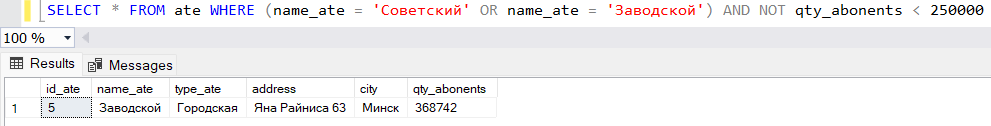




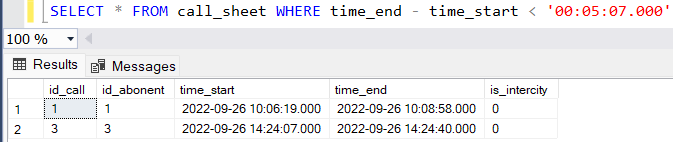


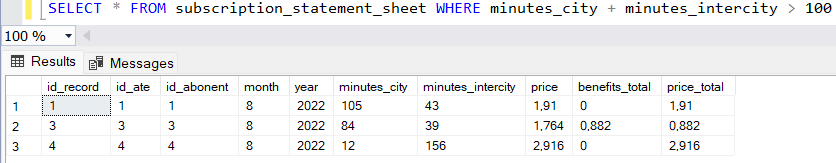
1. 2 запроса на использование комбинации логических операторов



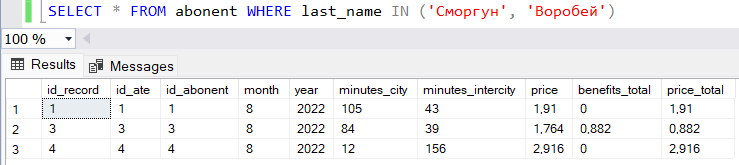


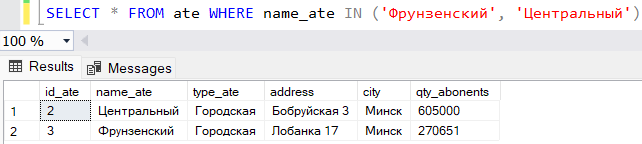
1. 2 запроса на использование выражений над столбцами



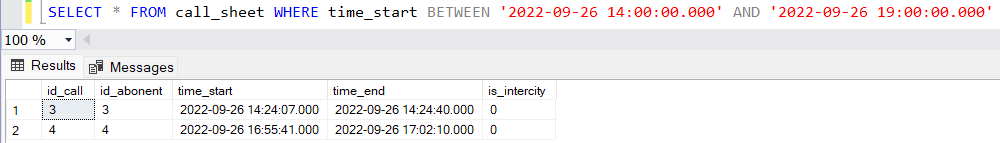


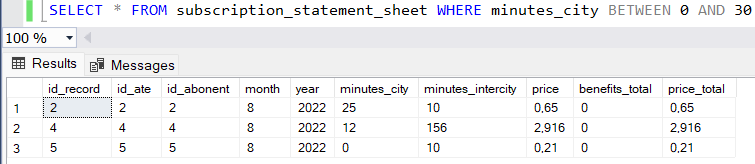
1. 2 запроса с проверкой на принадлежность множеству



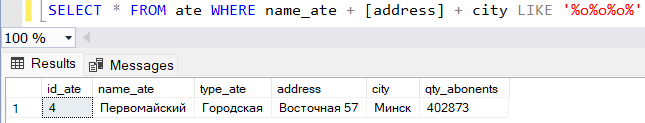


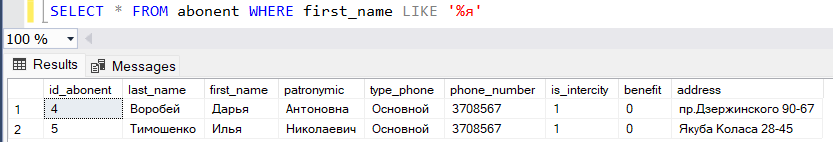
1. 2 запроса с проверкой на принадлежность диапазону значений



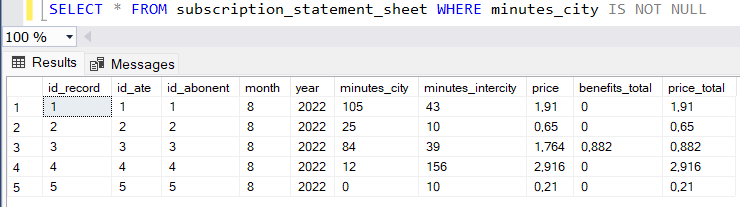


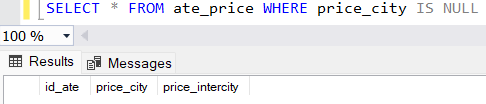
1. 2 запроса с проверкой на соответствие шаблону





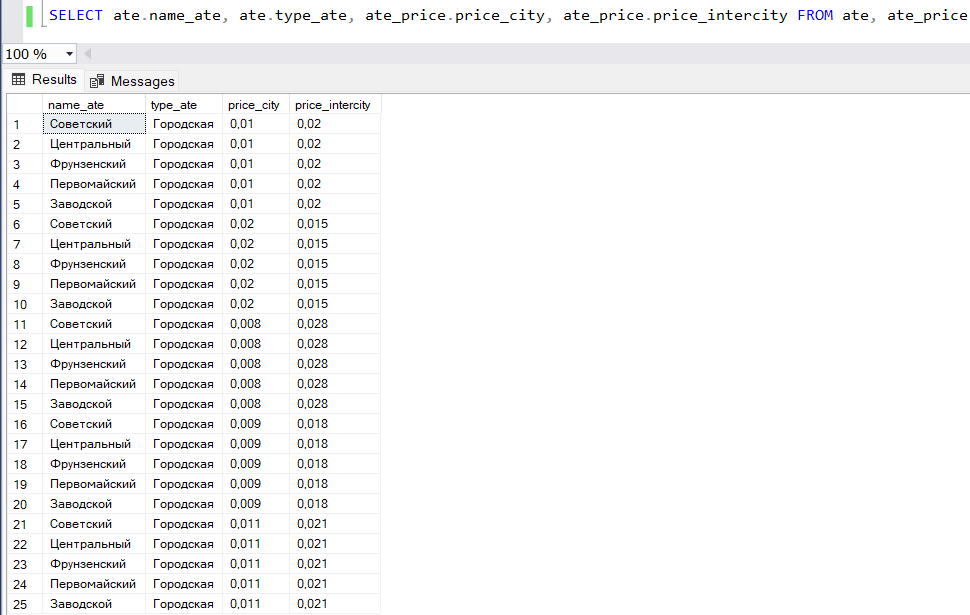
1. 2 запроса с проверкой на неопределенное значение

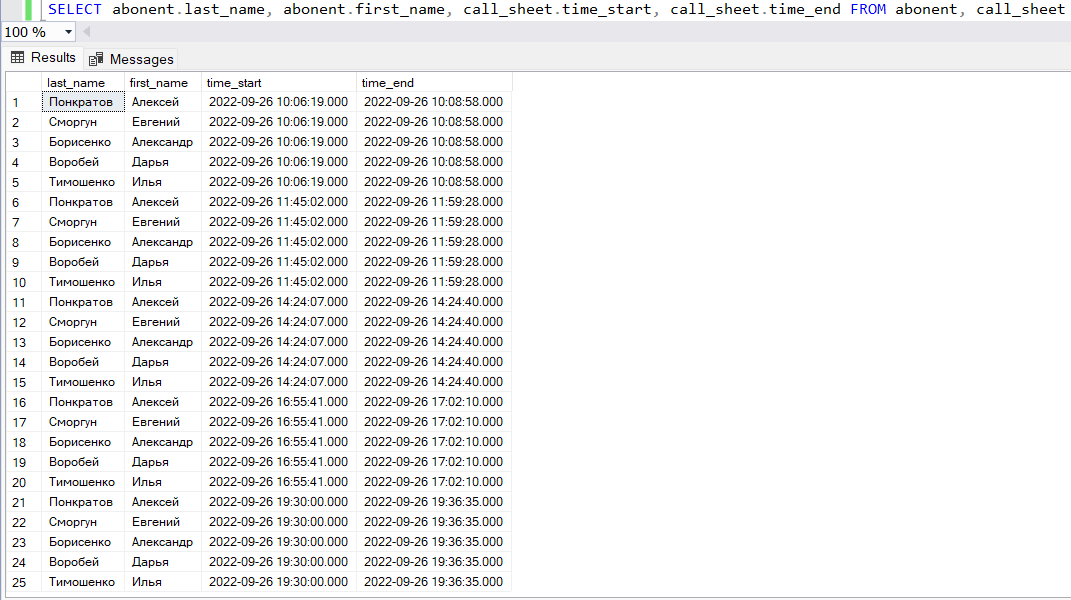




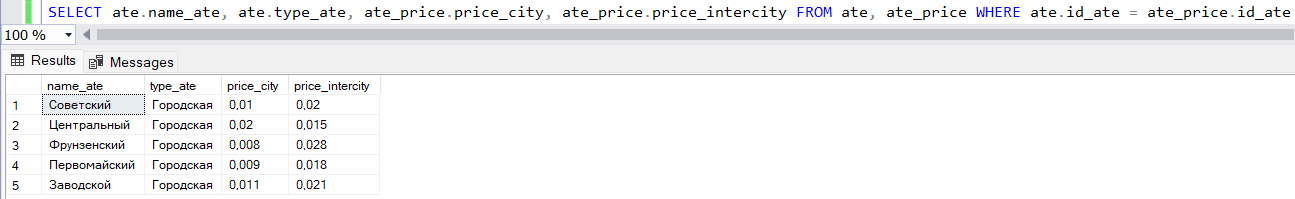
**Часть 3**

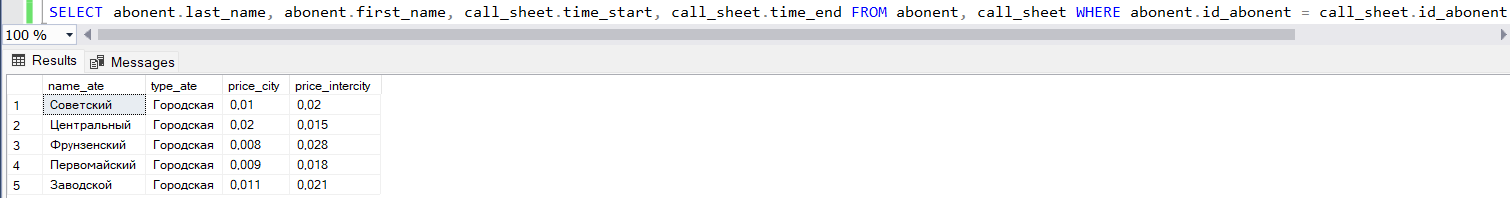
1. 2 запроса с использованием декартового произведения двух таблиц

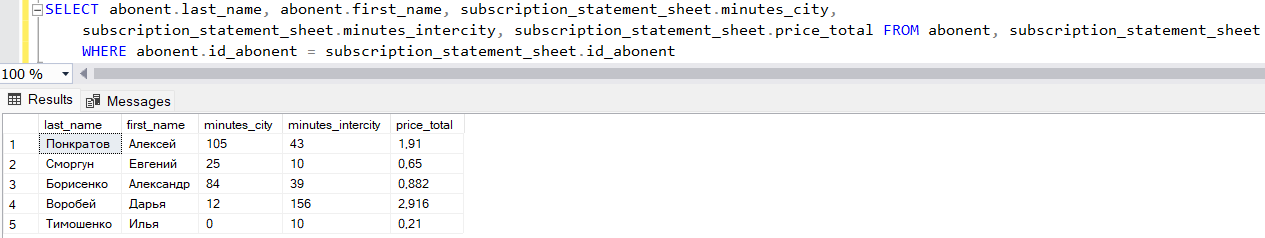




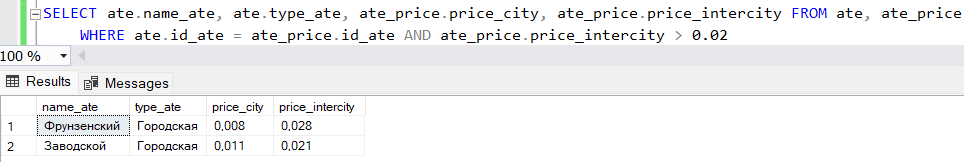
1. 3 запроса с использованием соединения двух таблиц по равенству

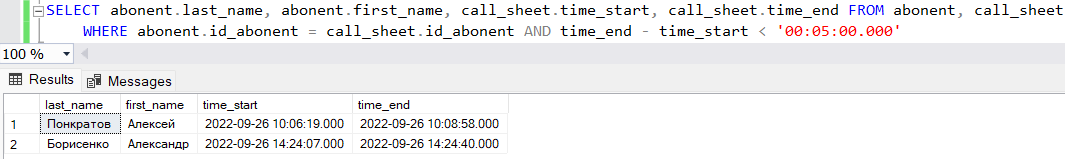




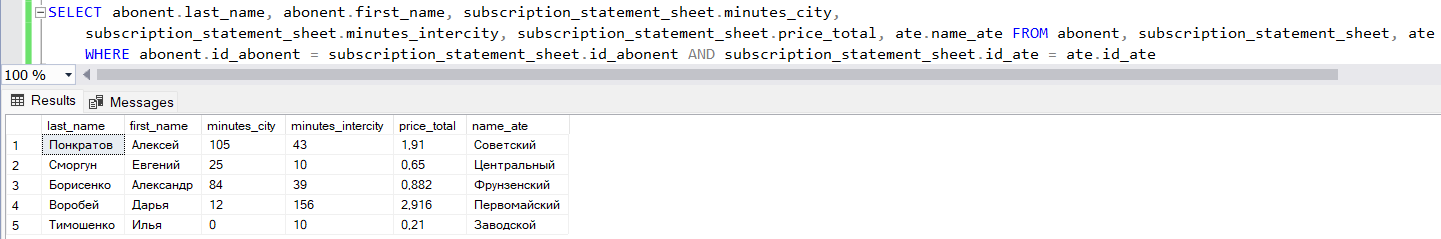


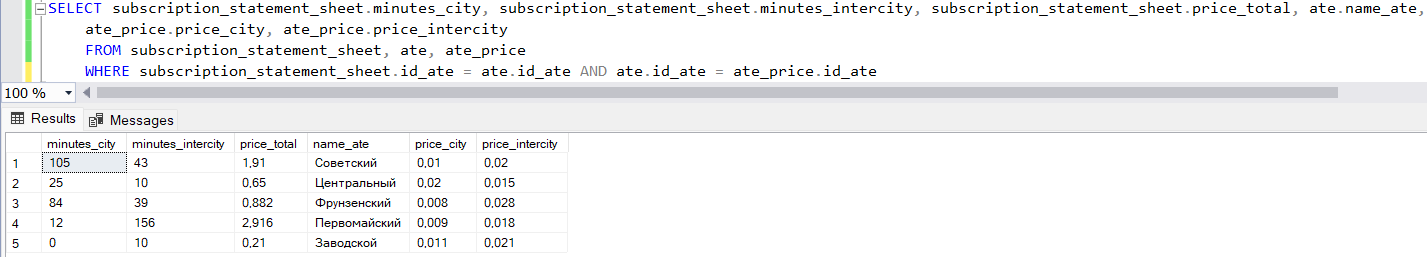
1. 2 запроса с использованием соединения двух таблиц по равенству и условием отбора



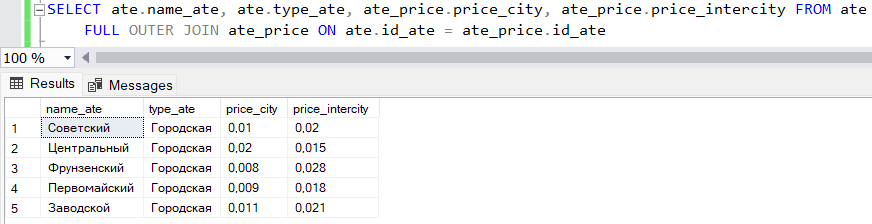


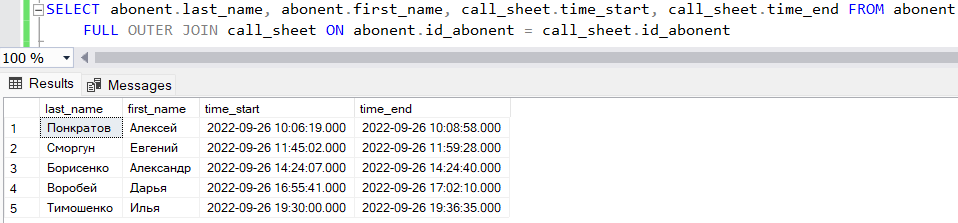
1. 2 запроса с использованием соединения по трем таблицам

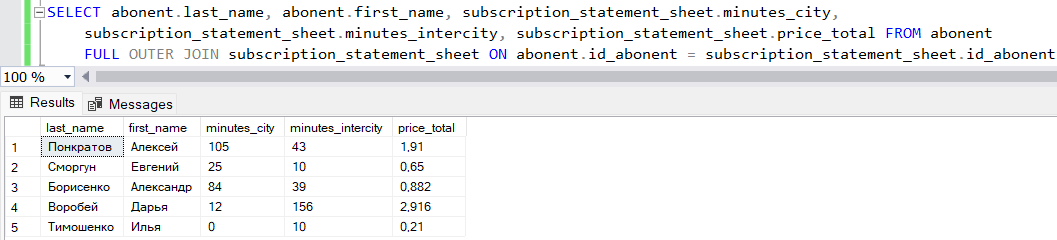




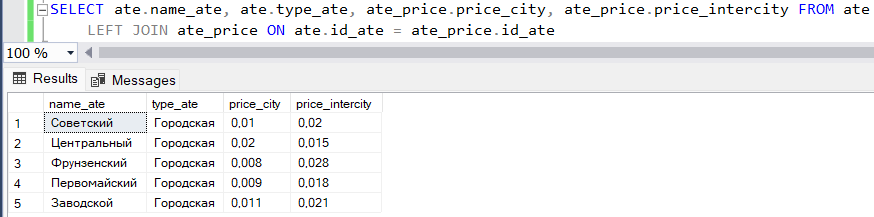
1. Создать копии ранее созданных запросов на соединение по равенству на запросы с использованием внешнего полного соединения таблиц

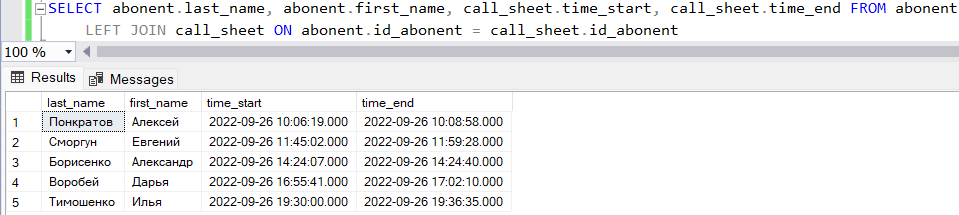




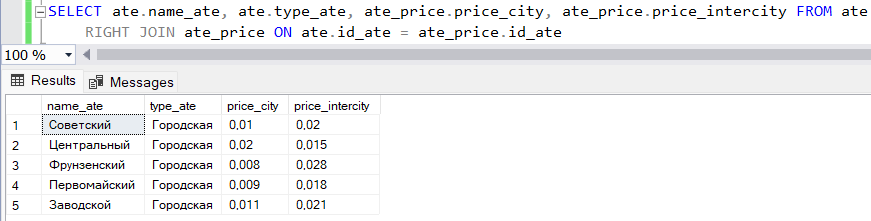


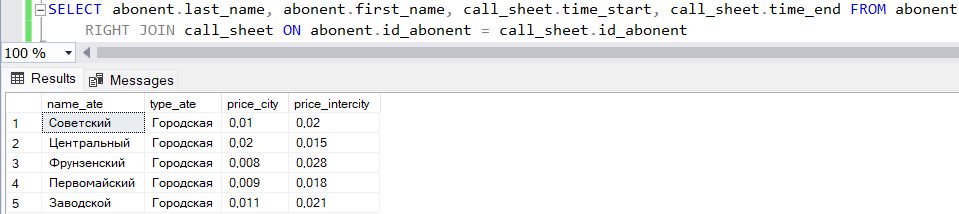
1. 2 запроса с использованием левого внешнего соединения



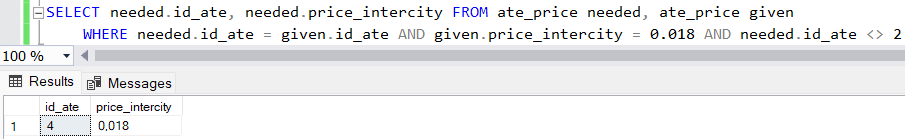
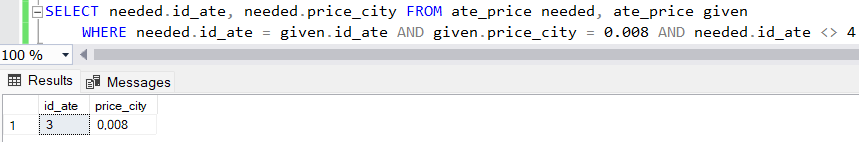


1. 2 запроса на использование правого внешнего соединения



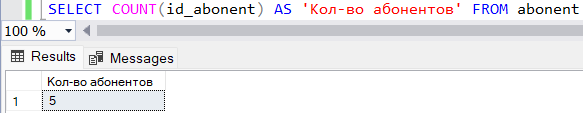


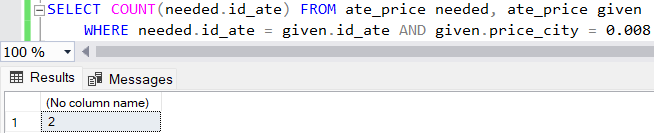
1. 2 запроса с использованием симметричного соединения и удаление избыточности



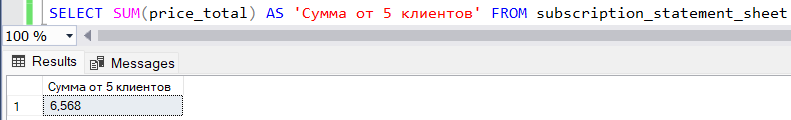
**Часть 4**

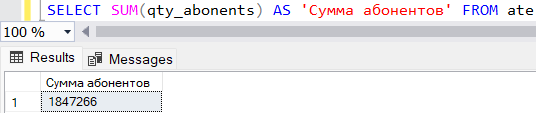
1. 2 запроса с использованием функции COUNT



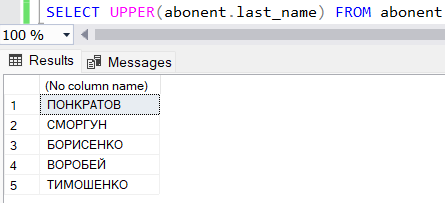


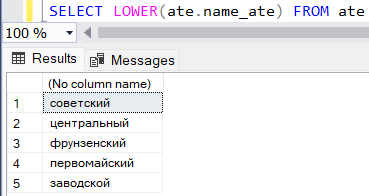
1. 2 запроса с использованием функции SUM



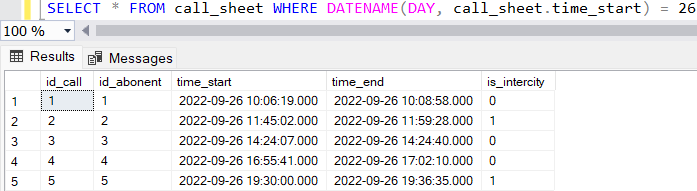


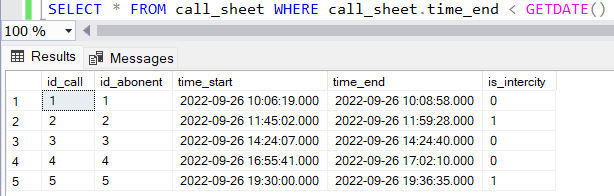
1. 2 запроса с использованием функций UPPER, LOWER



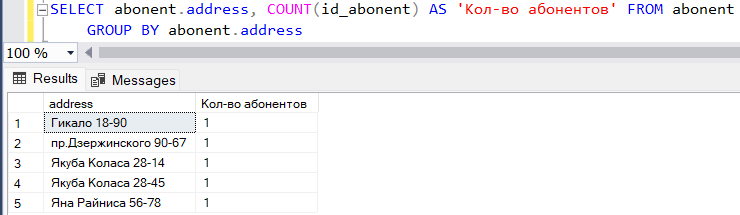


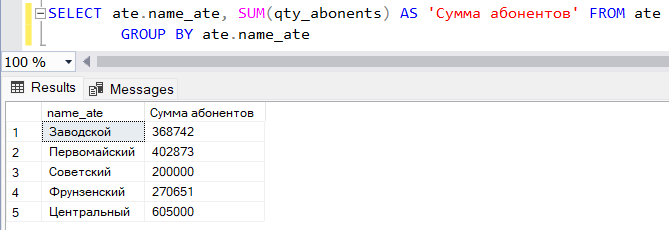
1. 2 запроса с использованием временных функций



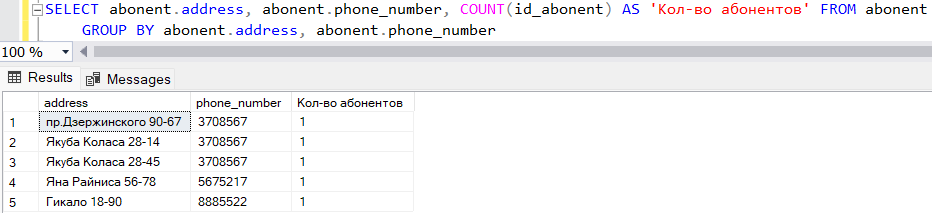


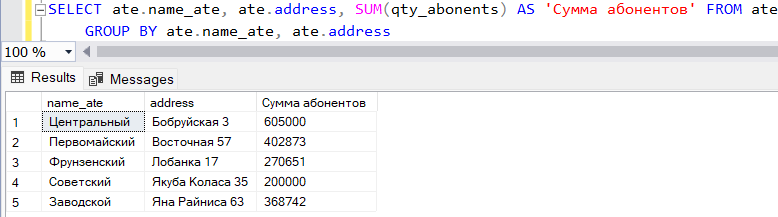
1. 2 запроса с использованием группировки по одному столбцу



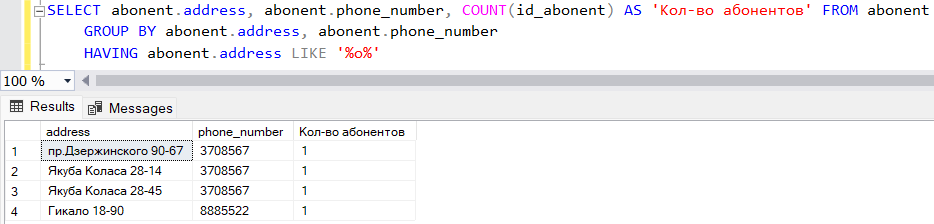


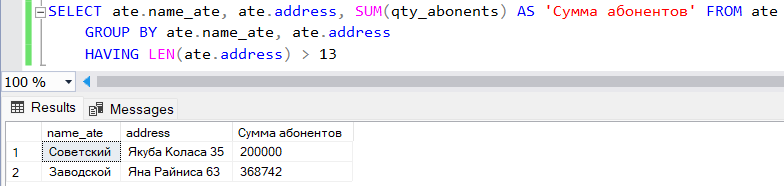
1. 2 запроса на использование группировки по нескольким столбцам



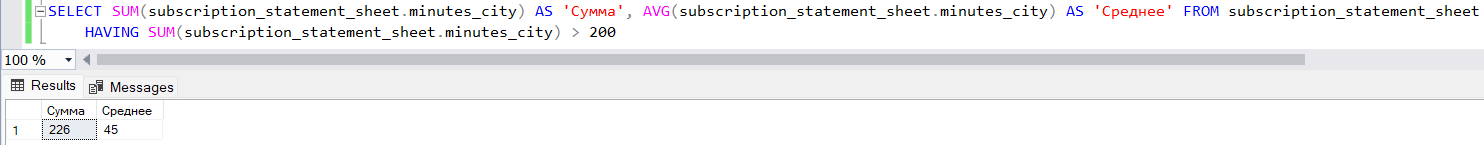


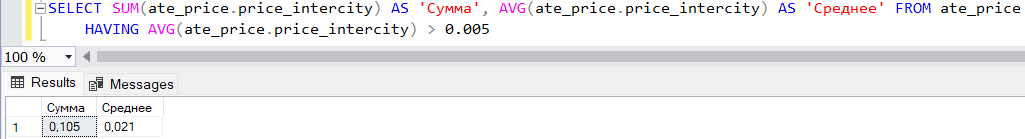
1. 2 запроса с использованием условия отбора групп HAVING



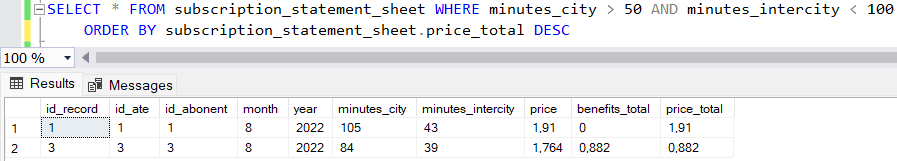


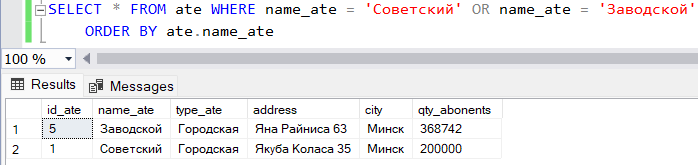
1. 2 запроса с использованием фразы HAVING без фразы GROUP BY



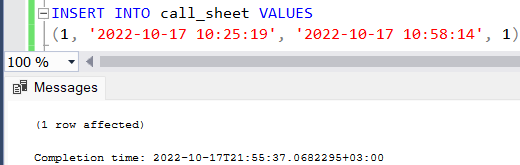


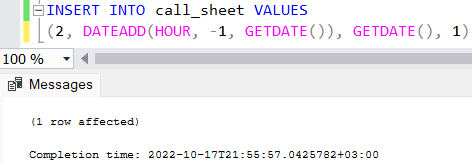
1. 2 запроса с использованием сортировки по столбцу



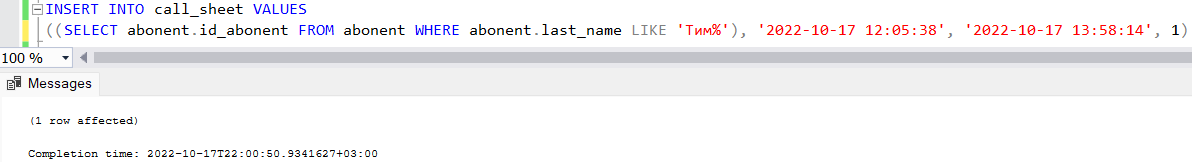


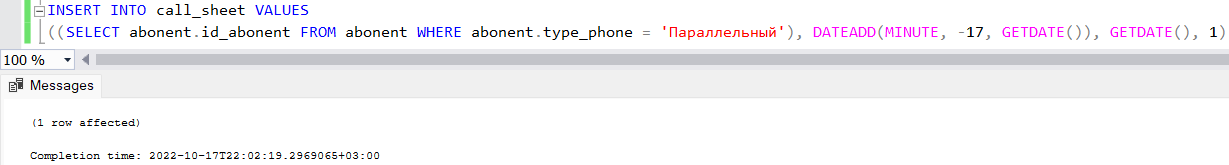
1. 2 запроса на добавление новых данных в таблицу



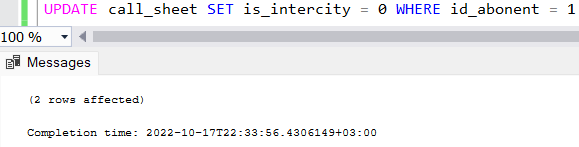


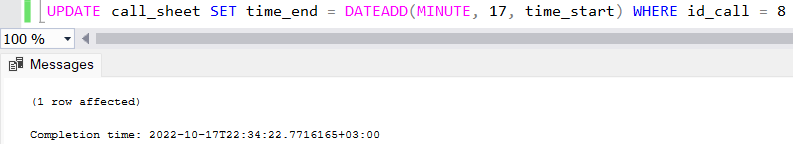
1. 2 запроса на добавление новых данных по результатам запроса в качестве вставляемого значения



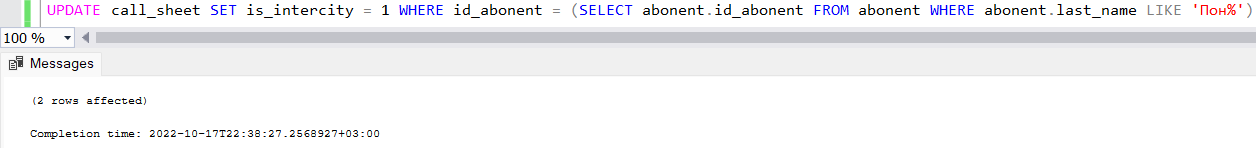


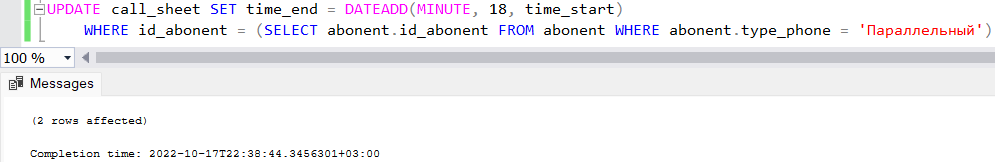
1. 2 запроса на обновление существующих данных в таблице



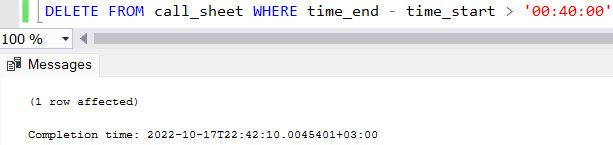


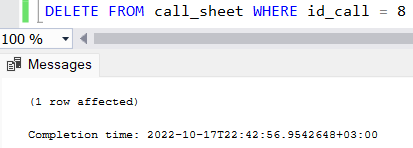
1. 2 запроса на обновление существующих данных по результатам подзапроса во фразе WHERE





1. 2 запроса на удаление существующих данных





**Вывод:** в ходе данной лабораторной работы был изучен процесс создания таблиц с помощью операторов языка T-SQL, создания запросов на выборку, многотабличных, запросов на соединение, запросов на сортировку, группировку и обновление данных, научиться использовать встроенные функции.