ГУО «БГУИР»

Инженерно-экономический факультет

Кафедра экономической информатики

Отчет по

Лабораторной работе №2

Проектирование базы данных

Подготовил:

Студент гр.914302

Понкратов А.М.

Проверила:

Лукашевич А.Э.

Минск 2022

**Вариант 18**

**Цель практической работы:** познакомиться с основными принципами создания базы данных в MS SQL Server. Изучить операции, проводимые с базами данных в целом. Получить навыки использования программы "SQL Server Management Studio" для создания, удаления, регистрации, подключения, извлечения метаданных, резервного копирования и восстановления базы данных. Изучить SQL-операторы для создания, подключения и удаления базы данных. Познакомиться с основными принципами управления учетными записями и ролями. Изучить способы создания, изменения и удаления таблиц. Получить навыки использования приложения " SQL Server Management Studio " для создания, удаления и изменения структуры таблиц. Изучить SQL-операторы для работы с таблицами и индексами. Изучить используемые в SQL Server типы ограничений. Получить навыки использования программы "SQL Server Management Studio" для создания, изменения и удаления ограничений. Изучить SQL-операторы для работы с ограничениями.

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',

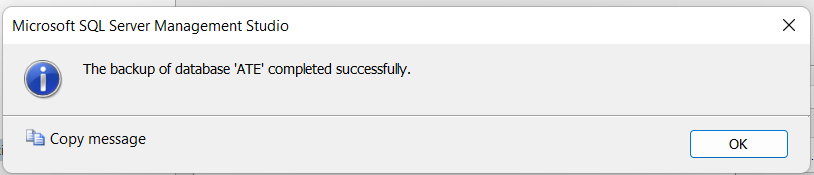
SIZE = 512KB,

MAXSIZE = 15MB,

FILEGROWTH = 10% )

GO

1. **Создать резервную копию базы данных**



1. **Определить 2-3 должностных лица, которые смогут работать с таблицами БД. Для каждого должностного лица определить набор привилегий, которыми он может пользоваться.**

|  |  |  |
| --- | --- | --- |
| Пользователь | Роль | Привилегии |
| DESKTOP-V8TJ2DQ\super | dbcreator | Создание БД, выдача прав, просмотр, создание, удаление, редактирование таблиц и данных в них |
| DESKTOP-V8TJ2DQ\user1 | ate\_master | Просмотр, добавление, удаление, редактирование данных таблицы “ate” |
| DESKTOP-V8TJ2DQ\user2 | ate\_operator | Просмотр, добавление, удаление, редактирование данных таблиц “abonent”, “ate\_price”, “call\_sheet”, “subscription\_statement\_sheet” |
| DESKTOP-V8TJ2DQ\user3 | ate\_support | Просмотр, добавление, редактирование данных таблиц “abonent”, “ate\_price”, “call\_sheet”, “subscription\_statement\_sheet” |

SQL-команды:

-- доступ на соединение с бд

EXEC sp\_grantlogin 'DESKTOP-V8TJ2DQ\super'

EXEC sp\_grantlogin 'DESKTOP-V8TJ2DQ\user1'

EXEC sp\_grantlogin 'DESKTOP-V8TJ2DQ\user2'

EXEC sp\_grantlogin 'DESKTOP-V8TJ2DQ\user3'

GO

-- присвоение учётной записи dbcreator пользователю super

EXEC sp\_addsrvrolemember 'DESKTOP-V8TJ2DQ\super', 'dbcreator'

GO

-- доступ на вход в бд и задание внутреннего имени

EXEC sp\_grantdbaccess 'DESKTOP-V8TJ2DQ\user1', 'sql1'

EXEC sp\_grantdbaccess 'DESKTOP-V8TJ2DQ\user2', 'sql2'

EXEC sp\_grantdbaccess 'DESKTOP-V8TJ2DQ\user3', 'sql3'

GO

1. **Создать под каждое должностное лицо соответствующую роль, наделить эту роль определенными привилегиями. Далее создать по одному пользователю на каждую должность и присвоить им соответствующие роли.**

SQL-команды:

EXEC sp\_addrole 'ate\_master'

EXEC sp\_addrole 'ate\_operator'

EXEC sp\_addrole 'ate\_support'

GO

EXEC sp\_addrolemember 'db\_accessadmin', 'sql0'

EXEC sp\_addrolemember 'ate\_master', 'sql1'

EXEC sp\_addrolemember 'ate\_operator', 'sql2'

EXEC sp\_addrolemember 'ate\_support', 'sql3'

GO

Ниже приведена часть кода, отвечающая за запрет/разрешение участникам ролей на работу с таблицами:

GRANT SELECT, INSERT, UPDATE, DELETE

ON ate TO ate\_master

GRANT SELECT, INSERT, UPDATE

ON abonent TO ate\_master WITH GRANT OPTION

GRANT SELECT, INSERT, UPDATE

ON ate\_price TO ate\_operator WITH GRANT OPTION

GRANT SELECT, INSERT, UPDATE

ON call\_sheet TO ate\_operator WITH GRANT OPTION

DENY INSERT, UPDATE, DELETE

ON ate\_price TO ate\_support

DENY INSERT, UPDATE, DELETE

ON subscription\_statement\_sheet TO ate\_support

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,

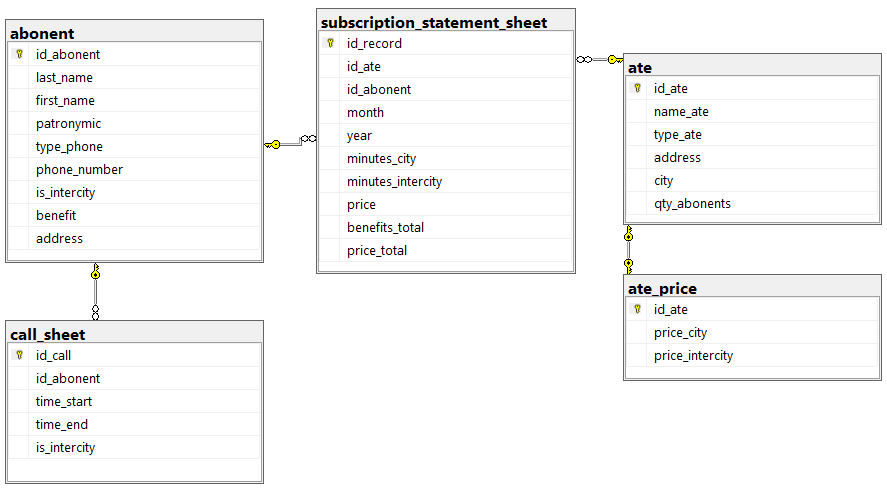
price\_total FLOAT NOT NULL,

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. **Создать диаграмму базы данных**

****

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”

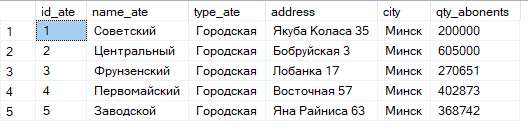


Таблица “abonent”

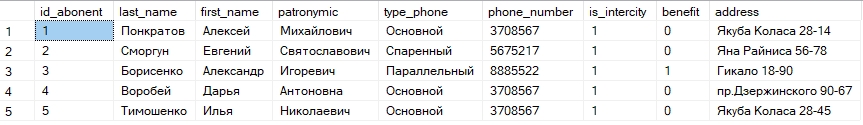


Таблица “call\_sheet”

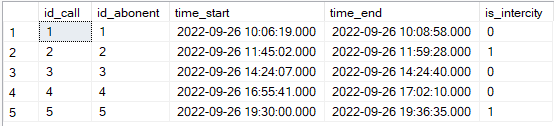
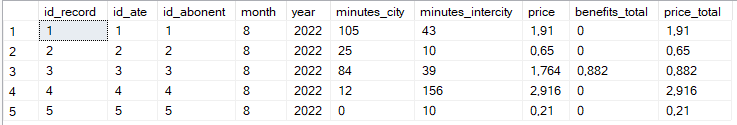


Таблица “ate\_price”



Таблица “subscription\_statement\_sheet”



**Вывод:** в ходе данной лабораторной работы был изучен процесс создания и выдачи ролей пользователям, создание таблиц и их заполнение.