

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Учреждение образования
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
КАФЕДРА ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

ЛАБОРАТОРНАЯ РАБОТА

ПО ДИСЦИПЛИНЕ

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ АВТОМАТИЗИРОВАННОГО РАБОЧЕГО МЕСТА

Система безопасности MS SQL Server 7.0

Знакомство с языком DDL. Создание объектов сервера

для студентов специальности

АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ

УДК 004.658

Методические указания к лабораторной работе предназначены для использования студентами в ходе выполнения лабораторной работы. Содержат краткие теоретические сведения по теме ВЫБОРКА ДАННЫХ, а так же задания к выполнению.

Составители: Швецова Е.В., ст. преподаватель

Шуть В.Н., доцент кафедры ИИТ

Рецензенты: К.ф.м.н., доцент, зав.каф "Прикладной математики и информатики" БрГУ Мадорский В.М.

Лабораторная работа

Тема. Система безопасности MS SQL Server 7.0

Цель работы: изучить систему безопасности MS SQL Server 7.0

Краткие теоретические сведения:

Архитектура СУБД MS SQL Server 7.0

SQL Server 7.0 состоит из четырех служб:

- MSSQLServer
- SQLServerAgent
- MSDTC
- MSSearch

В стандартный пакет MS SQL Server 7.0 входит целый ряд утилит администрирования.

Среди прочих присутствуют такие утилиты:

- SQL Service Manager – утилита управления службами SQL Server.
- SQL Server Enterprise Manager – утилита управления объектами SQL Server.
- Query Analyzer – инструментальная оболочка работы с SQL кодом.
- BooksOnline – документация по SQL Server

Каждая из служб выполняет свои специфичные функции. Ядром СУБД является MSSQLServer. Т.е. для работы с СУБД необходимо запустить эту службу. Это можно сделать с помощью утилиты MS SQL Service Manager (рис.1) либо с помощью утилиты Enterprise Manager (рис.2).

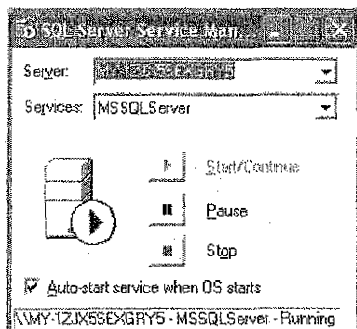


рис. 1

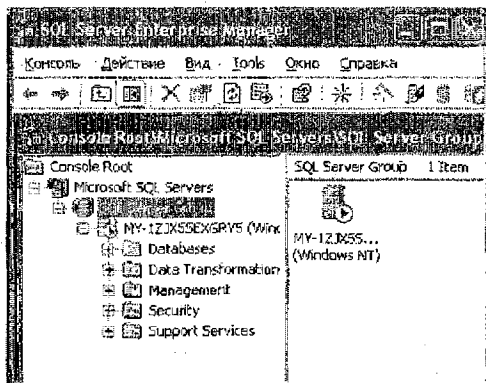


рис. 2

Система безопасности MS SQL Server 7.0

Система безопасности MS SQL Server базируется на пользователях и ролях. Пользователи проходят два этапа проверки системы безопасности: аутентификацию и проверку прав доступа.

На этапе аутентификации пользователь идентифицируется с помощью учетной записи *login* и пароля *password* (рис.3).

Как учетная запись, так и пароль должны быть к моменту аутентификации зарегистрированы на сервере (рис.4). Обычно эту операцию выполняют системный администратор (*system administrator*) либо администратор безопасности (*security administrator*).

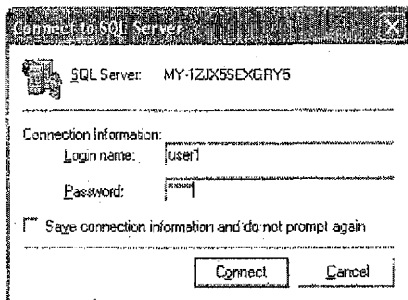


рис.3

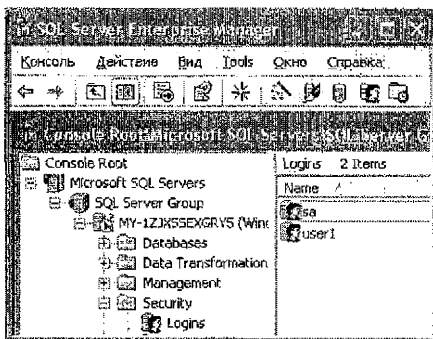


рис.4

После успешной аутентификации в зависимости от назначенных прав пользователь получает либо не получает одну из ролей сервера (Server Role) (рис.5).

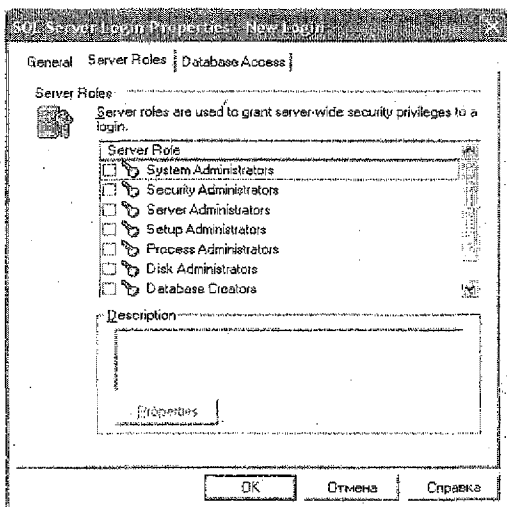


рис.5

Успешная аутентификация не дает автоматического доступа к базам данных. Для каждой базы данных учетная запись должна отображаться в пользователя базы данных *user*.

Далее пользователю базы данных назначается та или иная роль базы данных (рис.6). В каждой базе данных имеется специальная роль *public*. Каждый

пользователь базы данных – член этой роли. Она не может быть удалена либо присвоена вручную. Все разрешения на действия, определенные в этой роли автоматически, присваиваются каждому пользователю базы данных.

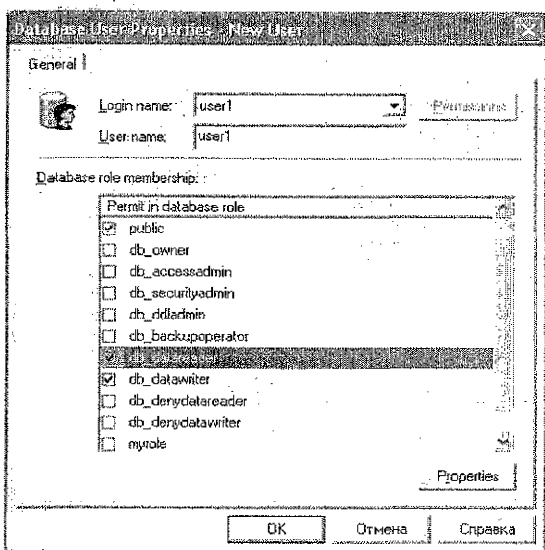


рис.6

Если учетная запись не связывается явно с пользователем базы данных, Пользователю присваивается *невный доступ* с использованием имени *guest*, т.е. гость. Обычно гостю присваиваются минимальные возможности внутри базы данных, например "только чтение". Если удалить гостя из пользователей базы данных, то учетные записи, не имеющие явного отображения в пользователя базы данных, не смогут получить доступ к ресурсам базы данных.

Владелец базы данных (Data Base Owner-DBO)-специальный пользователь, обладающий максимальными правами внутри базы данных. Владелец базы данных становится учетная запись, от имени которой была создана база данных: Также в пользователя DBO отображается любой член роли системных администраторов.

При этом, если таблица была создана членом роли системных администраторов, то полное имя таблицы будет *dbo.table_name*. В то же время, если же эта таблица была создана владельцем базы данных, не являющимся системным администратором, то полное имя таблицы будет *name_of_databaseowner.table_name*, где *name_of_databaseowner* – пользовательское имя владельца базы данных. Чтобы пользователь мог создавать объекты в базе данных, владелец базы данных должен предоставить ему соответствующие права (рис.7).

Внутри базы данных пользователями, наделенными соответствующими правами, могут создаваться пользовательские роли, которые действуют только в этой базе данных (рис.8). Что дает возможность реализовать универсальную модель системы безопасности.

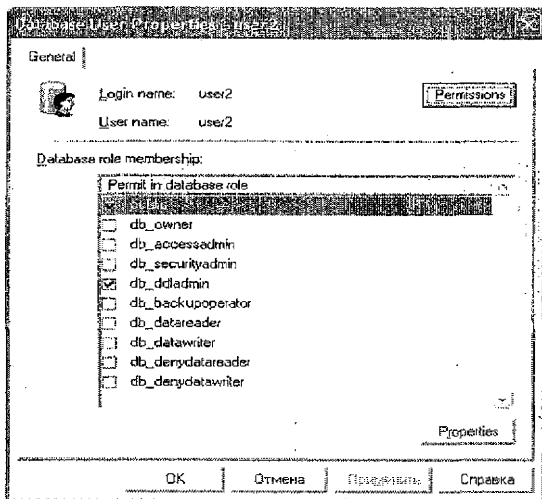


рис.7

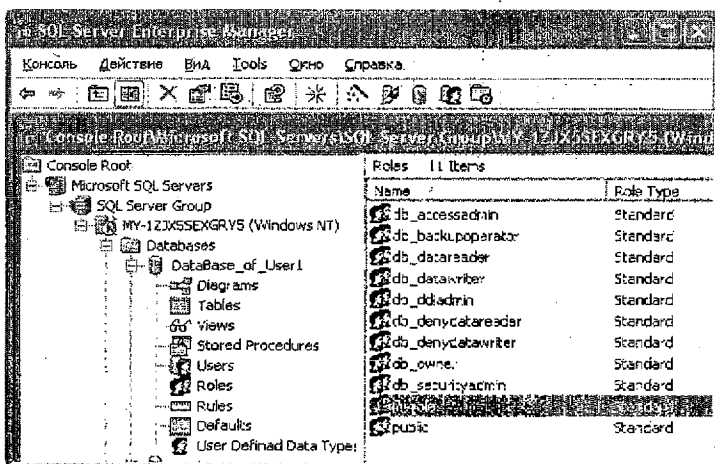


рис.8

Изменение владельца базы данных

Изменение владельца базы данных осуществляется с помощью системной хранимой процедуры, которая должна выполняться в контексте базы данных, владельца которой необходимо изменить. Это делается либо соответствующей установкой в *Query Analyzer*, либо следующим образом:

```
Use Renamed_DataBase
Exec sp_changedbowner 'new_login_owner'
```

Переименование базы данных

Для изменения имени базы данных используется системная хранимая процедура:

```
Sp_renamedb 'old_database_name','ne_database_name'
```

Право переименовывать базы данных имеют только системные администраторы. Для того чтобы переименовать базу данных, прежде всего ее нужно перевести в однопользовательский режим. После переименования ее необходимо вернуть в многопользовательский режим. Это можно сделать с помощью системной хранимой процедуры

```
Sp_dboption 'data_base_name','option','value'
```

Порядок выполнения работы

- Установить на сервере режим обязательной аутентификации.
- Скрыть системные БД при помощи опции «меню».
- Создать учетную запись, которая не является членом роли системных администраторов, и в то же время имеет право конфигурировать систему безопасности сервера.
- Создать учетную запись, которая не является членом роли системных администраторов, и в то же время имеет право создавать базы данных на сервере.
- Создать учетную запись, которая в контексте сервера не имеет никаких полномочий.
- От имени учетной записи, не являющейся членом роли системных администраторов, создать базу данных.
- В созданной базе данных создать таблицу (5-6 полей, 5-6 записей).
- Создать роль базы данных, которая позволяет участнику этой роли только просматривать данные созданной таблицы.
- Создать роль базы данных, которая позволяет участнику этой роли только вставлять данные в созданную таблицу.
- Создать роль базы данных, которая позволяет участнику этой роли только удалять данные из созданной таблицы.
- В созданной базе данных осуществить смену владельца.
- Осуществить переименование базы данных.
- Сделать базу данных недоступной для обращения.

Содержание отчета

- Цель работы
- Описание выполнения задания с подтверждением результатов скриншотами и распечаткой результатов выполнения запросов.

Контрольные вопросы

1. Перечислите все службы MS SQL Server 7.0, описав назначение каждой из них.
2. На каких объектах базируется система безопасности MS SQL Server 7.0?
3. Объясните понятия: *login*, *role*, *user*, *guest*, *public*, *dbo*.
4. Назовите основные виды ролей, используемые в системе безопасности MS SQL Server 7.0.

Лабораторная работа

Тема. Знакомство с языком DDL. Создание объектов сервера

Цель работы: приобрести навыки управления объектами СУБД MS SQL Server через ознакомление с командами одного из разделов SQL-DDL (Data Definition Language).

Необходимые теоретические сведения:

DDL(Data Definition Language)- язык определения данных дает возможность создания, удаления и изменения различных объектов СУБД (таблиц, индексов, пользователей и т.д.).

Учетная запись (login) – объект системы безопасности SQL Server на ряду с пользователями (user), ролями (role), группами (group). Пользователь, подключающийся к SQL Server, должен определить себя в системе СУБД, используя учетную запись. Учетная запись может отображаться в пользователя базы данных. Также становится членом определенной роли.

Неполный синтаксис команды добавления учетной записи :

```
sp_addlogin 'login', ['password'], ['database'],
```

login – имя учетной записи.

password – пароль, ассоциируемый с учетной записью.

database- база данных по умолчанию. К этой базе данных пользователь получит доступ после подключения к SQL Server.

Пример

```
sp_addlogin 'Victoria'
```

```
sp_addlogin 'Albert', 'food', 'corporate'
```

Синтаксис команды удаления учетной записи:

```
sp_droplogin 'login',
```

login – имя учетной записи.

Пример

```
sp_droplogin 'Victoria'
```

Для изменения пароля, ассоциированного с учётной записью, используется следующая команда:

```
sp_password 'old_password', 'new_password', 'login'
```


old_password- старый пароль, ассоциируемый с учётной записью.
new_password- новый пароль, ассоциируемый с учётной записью.
login – имя учетной записи.

Пример.

```
sp_password NULL, 'ok', 'Victoria'
```

С помощью команды **sp_helplogins** можно получить список учетных записей, зарегистрированных на сервере.

База данных – это объект SQL Server, содержащий в свою очередь набор других объектов, называемых объектами базы данных (таблицы, хранимые процедуры, пользователи и т.д.). База данных SQLServer состоит как минимум из двух файлов: файла, содержащего данные (расширение *.mdf), и файла, содержащего информацию о транзакциях (расширение *.ldf). Эти файлы лучше всего хранить в разных местах на диске или вообще на разных дисках для повышения отказоустойчивости системы. Неполный синтаксис команды создания базы данных выглядит следующим образом:

```
CREATE DATABASE databasename  
ON ( NAME = logical_file_name,  
     FILENAME = 'os_file_name',  
     SIZE = size,  
     MAXSIZE = { max_size | UNLIMITED },  
     FILEGROWTH = growth_increment )  
LOG ON  
  ( NAME = logical_file_name,  
    FILENAME = 'os_file_name',  
    SIZE = size,  
    MAXSIZE = { max_size | UNLIMITED },  
    FILEGROWTH = growth_increment )
```

databasename - имя создаваемой базы данных. Должно быть уникальным в пределах сервера.

NAME = logical_file_name – задается имя базы данных, которое будет использоваться в командах Transact-SQL после создания базы данных.

FILENAME = 'os_file_name' – содержит путь к файлу и имя самого файла, которое будет использовать операционная система для обращения к физически существующему на диске файлу, содержащему данные

SIZE = size – размер создаваемого файла базы данных в Кб или Мб.

MAXSIZE = { max_size | UNLIMITED } – максимальный размер файла базы данных, который он может принимать.

FILEGROWTH = growth_increment- задает размер приращения файла базы данных в Мб, Кб либо в процентах от исходного размера базы данных.

Пример

```
CREATE DATABASE Sales
ON
( NAME = Sales_dat,
FILENAME = 'c:\mssql7\data\saledat.mdf,
SIZE = 10,
MAXSIZE = 50,
FILEGROWTH = 5 )
LOG ON
( NAME = 'Sales_log',
FILENAME = 'c:\mssql7\data\salelog.ldf',
SIZE = 5MB,
MAXSIZE = 25MB,
FILEGROWTH = 5MB )
```

Пользователь DBO(database owner) может добавлять либо удалять в свою базу данных других пользователей. Это можно сделать с помощью следующих команд:

sp_adduser 'login','user' – команда добавления пользователя в базу данных

sp_dropuser 'user' – команда удаления пользователя из базы данных

login – учетная запись, отображающаяся в создаваемого пользователя базы данных.

User -создаваемый пользователь базы данных

Для того, чтобы было можно перенести базу данных на другой компьютер или сервер, необходимо отсоединить базу данных от исходного сервера и в дальнейшем её присоединить к другому серверу. Эту процедуру можно осуществить с помощью следующих команд:

sp_detach_db 'databasename' – отсоединяет базу данных от сервера.

После отсоединения файлы базы данных можно скопировать на носитель и перенести на другой компьютер. В дальнейшем файлы перенесенной базы данных необходимо присоединить к серверу. Эта процедура осуществляется с помощью команды:

sp_attach_db 'databasename' – присоединяет базу данных к серверу.

В обеих командах параметр *databasename* обозначает название базы данных, которую отсоединяют/присоединяют к серверу.

Удалить базу данных можно с помощью команды:

```
DROP DATABASE database_name
database_name – имя удаляемой базы данных.
```

Ход работы

1. Пользуясь SQL – командами:
 - Добавить учетную запись на сервер.
 - Удалить учетную запись.
 - Изменить пароль учетной записи.
 - = Осуществить просмотр имен пользователей.
 - Создать базу данных на сервере, так чтобы ее файл с данными и файл с информацией о транзакциях хранились в разных каталогах на диске или на разных дисках.
 - Создать пользователя базы данных.
 - Удалить пользователя базы данных.
2. В соответствии с условием ранее выданной вам задачи создать таблицы базы данных, подобрав подходящие имена таблицам и их полям.
3. В таблицах определить следующие ограничения целостности :
 - ⌘ Первичные ключи.
 - ⌘ Внешние ключи.
 - ⌘ Умолчания.
 - ⌘ Ограничения на значения колонок.
4. Структуру взаимосвязей созданных таблиц базы данных отобразить с помощью диаграмм.
5. Скопировать базу данных к себе на диск.
6. С помощью SQL-команды удалить базу данных.

Литература

1. Мартин Грабер Введение в SQL. - Москва: "Лори",1994. - 348 с.
2. Мамаев Е., Вишневецкий А. - СПб.: Питер,2000. - 894с.

УЧЕБНОЕ ИЗДАНИЕ

Составители: Швецова Елена Владимировна
Шуть Василий Николаевич

ЛАБОРАТОРНАЯ РАБОТА

ПО ДИСЦИПЛИНЕ

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ АВТОМАТИЗИРОВАННОГО РАБОЧЕГО МЕСТА

Система безопасности MS SQL Server 7.0

Знакомство с языком DDL. Создание объектов сервера

для студентов специальности

АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ

Ответственный за выпуск: *Швецова Е.В.*

Редактор: *Строкач Т.В.*

Компьютерная вёрстка: *Кармаш Е.Л.*

Корректор: *Никитчик Е.В.*

Подписано в печать 26.03.2008 г. Формат 60x84¹/₁₆. Бумага писч. Усл.п.л. 0,69.
Усл.изд.л. 0,75. Тираж 80 экз. Заказ № 392. Отпечатано на ризографе УО
"Брестский государственный технический университет".
224017, Брест, ул. Московская, 267