

Лабораторная работа 13

Задание 1. Подключитесь к серверу *Sqlserver* с помощью утилиты *Management Studio*.

Указания к выполнению:

1. Запустите *SQL Server Management Studio* через меню **Пуск – Программы – Microsoft SQL Server 2008**.

2. Выберите тип аутентификации: *SQL Server Authentication*. Укажите *User name: sa*, и *Password: - установленный Вами пароль администратора сервера* и нажмите кнопку **Connect**.

Задание 2. Определите список ролей сервера.

Указания к выполнению:

1. Создайте новый запрос или через команду меню **File – New – Query with Current Connection** или при помощи кнопки **New Query** на панели инструментов.

2. Во вкладке *SQLQuery1.sql* выполните команду **sp_helpsrvrole** (см. рис. 6.1).

```
SELECT
  r.rolname AS "Role",
  COALESCE(r.rolpassword, '') AS "Password",
  CASE
    WHEN r.rolcanlogin THEN 'yes'
    ELSE 'no'
  END AS "Can login",
  CASE
    WHEN r.rolsuper THEN 'yes'
    ELSE 'no'
  END AS "Superuser",
  ARRAY_AGG(m.rolname) AS "Member of"
FROM
  pg_roles r
  LEFT JOIN pg_auth_members am ON r.oid = am.roleid
  LEFT JOIN pg_roles m ON am.member = m.oid
GROUP BY
  r.oid, r.rolname, r.rolpassword, r.rolcanlogin, r.rolsuper
ORDER BY
  "Role";
```

Data Output		Messages		Notifications	
<div><div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div></div></div>					
	Role name	Password text	Can login text	Superuser text	Member of name[]
1	pg_checkpoint	*****	no	no	{NULL}
2	pg_database_owner	*****	no	no	{NULL}
3	pg_execute_server_program	*****	no	no	{NULL}
4	pg_monitor	*****	no	no	{NULL}
5	pg_read_all_data	*****	no	no	{NULL}
6	pg_read_all_settings	*****	no	no	{pg_monitor}
7	pg_read_all_stats	*****	no	no	{pg_monitor}
8	pg_read_server_files	*****	no	no	{NULL}
9	pg_signal_backend	*****	no	no	{NULL}
10	pg_stat_scan_tables	*****	no	no	{pg_monitor}
11	pg_write_all_data	*****	no	no	{NULL}
12	pg_write_server_files	*****	no	no	{NULL}
13	postgres	*****	yes	yes	{NULL}

Задание 3. Создайте и настройте новую учетную запись *TempUser_N* для входа в SQL Server. *N-номер студента по журналу*

Указания к выполнению:

1. Для добавления учетной записи используйте хранимую процедуру **sp_addlogin**:
`sp_addlogin 'TempUser_N', 'Password!'`

```
CREATE USER "TempUser_2" WITH PASSWORD 'Password!';
```

Data Output **Messages** Notifications

CREATE ROLE

Query returned successfully in 80 msec.

2. Убедитесь, что учетная запись была добавлена при помощи хранимой процедуры **sp_helplogins** (см. рис. 6.2).

```
SELECT
  r.rolname AS "Role",
  COALESCE(r.rolpassword, '') AS
"Password",
  CASE
    WHEN r.rolcanlogin THEN 'yes'
    ELSE 'no'
  END AS "Can login",
  CASE
    WHEN r.rolsuper THEN 'yes'
    ELSE 'no'
  END AS "Superuser",
  ARRAY_AGG(m.rolname) AS "Member of"
FROM
  pg_roles r
  LEFT JOIN pg_auth_members am ON r.oid
= am.roleid
  LEFT JOIN pg_roles m ON am.member =
m.oid
GROUP BY
  r.oid, r.rolname, r.rolpassword, r.rolcanlogin, r.rolsuper
ORDER BY
  "Role";
```

	Role name	Password text	Can login text	Superuser text	Member of name[]
1	TempUser_N	*****	yes	no	{NULL}
2	pg_checkpoint	*****	no	no	{NULL}
3	pg_database_owner	*****	no	no	{NULL}
4	pg_execute_server_program	*****	no	no	{NULL}
5	pg_monitor	*****	no	no	{NULL}
6	pg_read_all_data	*****	no	no	{NULL}
7	pg_read_all_settings	*****	no	no	{pg_monitor}
8	pg_read_all_stats	*****	no	no	{pg_monitor}
9	pg_read_server_files	*****	no	no	{NULL}
10	pg_signal_backend	*****	no	no	{NULL}
11	pg_stat_scan_tables	*****	no	no	{pg_monitor}
12	pg_write_all_data	*****	no	no	{NULL}
13	pg_write_server_files	*****	no	no	{NULL}
14	postgres	*****	yes	yes	{NULL}

3. Попробуйте войти на сервер под созданной учетной записью.
4. Зайдите снова под учетной записью **sa**, т.к. для дальнейших действий снова потребуются права администратора.
5. Для присвоения учетной записи для входа встроенной серверной роли используется процедура:

```
sp_addsrvrolemember 'TempUser_N', 'securityadmin'
```

```
ALTER USER "TempUser_N" WITH SUPERUSER;
```

Data Output Messages Notifications

ALTER ROLE

Query returned successfully in 53 msec.

Задание 4. Определите список ролей базы данных, созданной в ЛР-4 и членов роли *db_owner*. (Вместо фамилии *Иванов* используйте БД со своей фамилией)

Указания к выполнению:

1. Выполните хранимую процедуру **sp_helprole** для получения списка как встроенных, так и определенных пользователем ролей базы данных.

```
SELECT
  r.rolname AS "Role",
  COALESCE(r.rolpassword, '') AS
"Password",
  CASE
    WHEN r.rolcanlogin THEN 'yes'
    ELSE 'no'
  END AS "Can login",
  CASE
    WHEN r.rolsuper THEN 'yes'
    ELSE 'no'
  END AS "Superuser",
  ARRAY_AGG(m.rolname) AS "Member of"
FROM
  pg_roles r
  LEFT JOIN pg_auth_members am ON r.oid =
am.roleid
  LEFT JOIN pg_roles m ON am.member =
m.oid
GROUP BY
  r.oid, r.rolname, r.rolpassword, r.rolcanlogin, r.rolsuper
ORDER BY
  "Role";
```

	Role name	Password text	Can login text	Superuser text	Member of name[]
1	TempUser_2	*****	yes	no	{NULL}
2	pg_checkpoint	*****	no	no	{NULL}
3	pg_database_owner	*****	no	no	{NULL}
4	pg_execute_server_program	*****	no	no	{NULL}
5	pg_monitor	*****	no	no	{NULL}
6	pg_read_all_data	*****	no	no	{NULL}
7	pg_read_all_settings	*****	no	no	{pg_monitor}
8	pg_read_all_stats	*****	no	no	{pg_monitor}
9	pg_read_server_files	*****	no	no	{NULL}
10	pg_signal_backend	*****	no	no	{NULL}
11	pg_stat_scan_tables	*****	no	no	{pg_monitor}
12	pg_write_all_data	*****	no	no	{NULL}
13	pg_write_server_files	*****	no	no	{NULL}
14	postgres	*****	yes	yes	{NULL}

2. При помощи команды **sp_helprolemember** 'db_owner' определите членов роли *db_owner* (см. рис. 6.3).

```
SELECT
    r.rolname AS role_name,
    m.rolname AS member_name
FROM
    pg_catalog.pg_auth_members am
    JOIN pg_catalog.pg_roles r ON r.oid =
am.roleid
JOIN pg_catalog.pg_roles m ON m.oid =
am.member;
```

Data Output			Messages	Notifications
	role_name name		member_name name	
1	pg_read_all_settings		pg_monitor	
2	pg_read_all_stats		pg_monitor	
3	pg_stat_scan_tables		pg_monitor	

Задание 5. Создайте нового пользователя базы данных для логина *TempUser_N*.

Указания к выполнению:

1. При помощи хранимой процедуры добавьте пользователя:

```
sp_adduser 'TempUser_N', 'MyFirstUser'
```

```
CREATE USER "TempUser_N" WITH PASSWORD 'MyFirstUser';
```

Data Output	Messages	Notifications
CREATE ROLE		
Query returned successfully in 56 msec.		

2. При помощи процедуры **sp_helpuser** убедитесь, что пользователь был добавлен. Какая роль ему была присвоена?

```
CREATE USER "TempUser_N" WITH PASSWORD 'MyFirstUser';
```

Data Output						Messages	Notifications
	Role name	Password text	Can login text	Superuser text	Member of name[]		
1	TempUser_N	*****	yes	no	(NULL)		
2	pg_checkpoint	*****	no	no	(NULL)		
3	pg_database_owner	*****	no	no	(NULL)		
4	pg_execute_server_program	*****	no	no	(NULL)		
5	pg_monitor	*****	no	no	(NULL)		
6	pg_read_all_data	*****	no	no	(NULL)		
7	pg_read_all_settings	*****	no	no	(pg_monitor)		
8	pg_read_all_stats	*****	no	no	(pg_monitor)		
9	pg_read_server_files	*****	no	no	(NULL)		
10	pg_signal_backend	*****	no	no	(NULL)		
11	pg_stat_scan_tables	*****	no	no	(pg_monitor)		
12	pg_write_all_data	*****	no	no	(NULL)		
13	pg_write_server_files	*****	no	no	(NULL)		
14	postgres	*****	yes	yes	(NULL)		

3. Добавьте пользователю роль *db_datareader*:

```
sp_addrolemember 'db_datareader, 'MyFirstUser'
```

```
GRANT SELECT ON ALL TABLES IN SCHEMA public TO "TempUser_N";
```

Data Output	Messages	Notifications
-------------	----------	---------------

GRANT

Query returned successfully in 57 msec.

Задание 6. Настройте права доступа пользователю *Andy*: предоставьте явным образом право только для выборки из таблицы *STUDENTS* и обновления только полей *SFAM* и *STIP* этой таблицы.

Указания к выполнению:

1. С помощью следующей команды пользователю *TestUser* базы данных *из ЛР-4* предоставляются права выборки и изменения данных таблицы *STUDENTS* этой базы данных:

```
CREATE USER "Andy" WITH PASSWORD 'password';  
GRANT SELECT ON TABLE students TO "Andy";
```

Data Output	Messages	Notifications
-------------	----------	---------------

GRANT

Query returned successfully in 74 msec.

2. Следующая команда предоставляет пользователю *Andy* права только выборки данных полей *SFAM* и *STIP* таблицы *STUDENTS* базы данных *из ЛР-4*:

```
GRANT UPDATE (sfam, stip) ON TABLE students TO "Andy";
```

Data Output	Messages	Notifications
-------------	----------	---------------

GRANT

Query returned successfully in 55 msec.

Задание 7. Изучите выполнение вышеупомянутых функций при помощи графического интерфейса утилиты *Management Studio*.

> postgres

Login/Group Roles (13)

- Create > Login/Group Role...
- Refresh
- pg_execute_server_program
- pg_monitor
- pg_read_all_data
- pg_read_all_settings
- pg_read_all_stats

25

28 SELECT

29

30

31 FROM

32

33

Group Role - Login/Group Roles

General Definition Privileges Membership Parameters Security SQL

Can login? ☒

Superuser? ☐

Create roles? ☒

Create databases? ☒

Inherit rights from the parent roles? ☒

Can initiate streaming replication and backups? ☐

Member of

User/Role	WITH ADMIN
-----------	------------

Members

User/Role	WITH ADMIN
-----------	------------

Group Role - Login/Group Roles

General Definition Privileges Membership Parameters Security SQL

Name user

Comments

Group Role - Login/Group Roles

General Definition Privileges Membership Parameters Security SQL

```
1 CREATE ROLE "user" WITH
2 LOGIN
3 NOSUPERUSER
4 CREATEDB
5 CREATEROLE
6 INHERIT
7 NOREPLICATION
8 CONNECTION LIMIT -1
9 PASSWORD 'xxxxxx';
```

Login/Group Roles (14)

- pg_checkpoint
- pg_database_owner
- pg_execute_server_program
- pg_monitor
- pg_read_all_data
- pg_read_all_settings
- pg_read_all_stats
- pg_read_server_files
- pg_signal_backend
- pg_stat_scan_tables
- pg_write_all_data
- pg_write_server_files
- postgres
- user

Задание 8. Отмените присвоение роли учетной записи и удалите учетную запись *TempUser*.

Указания к выполнению:

1. Отмена присвоенной пользователю роли может быть выполнена с помощью процедуры:

```
REVOKE pg_monitor FROM "user";
```

Data Output	Messages	Notifications
-------------	----------	---------------

REVOKE ROLE

Query returned successfully in 78 msec.

2. Для удаления пользователя БД используются процедуры:

```
sp_dropuser 'MyFirstUser'
```

```
DROP USER "user";
```

Data Output	Messages	Notifications
-------------	----------	---------------

DROP ROLE

Query returned successfully in 56 msec.