Система безопасности серверов БД

на примере MS SQL Server 2008

Принципы системы безопасности серверов БД

Системы безопасности большинства современных серверов

1) основана на принципах избирательного подхода

2) применении 2-х уровневой модели доступа

3) использовании

- **аутентификации**
- **авторизации**
- **шифрования**

Аутентификации – это установление соответствия лица названному им идентификатору

Авторизации – это предоставление возможностей в соответствие с положенными правами или проверка наличия прав при попытке выполнить какое-либо действие

Шифрование – это процесс кодирования информации

Принципы системы безопасности серверов БД

Система безопасности SQL Server 2008

1) основана на принципах избирательного подхода

2) реализуется 2-х уровневой моделью

3) использует

- учетную запись (login) или принципал сервера (server principal) для аутентификации;

- пользователь (user) или принципал базы данных (database principal)/схема (schema) для авторизации;

- роли (roles);

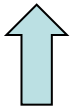
- группы (groups);

Избирательный подход

Суть: каждый пользователь обладает различными правами для работы с объектами БД

Обязательный подход

Суть: каждый пользователь обладает некоторым уровнем допуска, каждому объекту БД присваивается классификационный уровень и допуск к объекту получают только те пользователи, у которых есть соответствующий уровень.



Уровни безопасности сервера

1-й уровень – сервера

2-й уровень – базы данных

Понятия модели безопасности

Учетная запись или **принципал сервера** – это одна из моделей идентификации пользователя в системе, используя которую реализуется аутентификация

Аутентификация – это проверка подлинности подключаемых к серверу клиентов

Пользователь или **принципал базы данных** – это объект БД, с помощью которого определяются все **разрешения доступа** к объектам БД (таблицы, представления, ХП и т.д.)

Схема – это объект БД, с помощью которого определяются **владения** объектами БД (таблицы, представления, ХП и т.д.)

Схема группирует множество объектов БД

Роль – это поименованный набор полномочий (прав)

Группа – это поименованный набор пользователей с одинаковыми правами

Режимы аутентификации

- средствами Windows NT

- средствами MS SQL Server

Аутентификация Windows

LoginID сохраняется в SQL Server (системной БД master). Остальные параметры (имя пользователя, пароль и т.д) храниться в структурах Windows NT (БД системы безопасности домена)

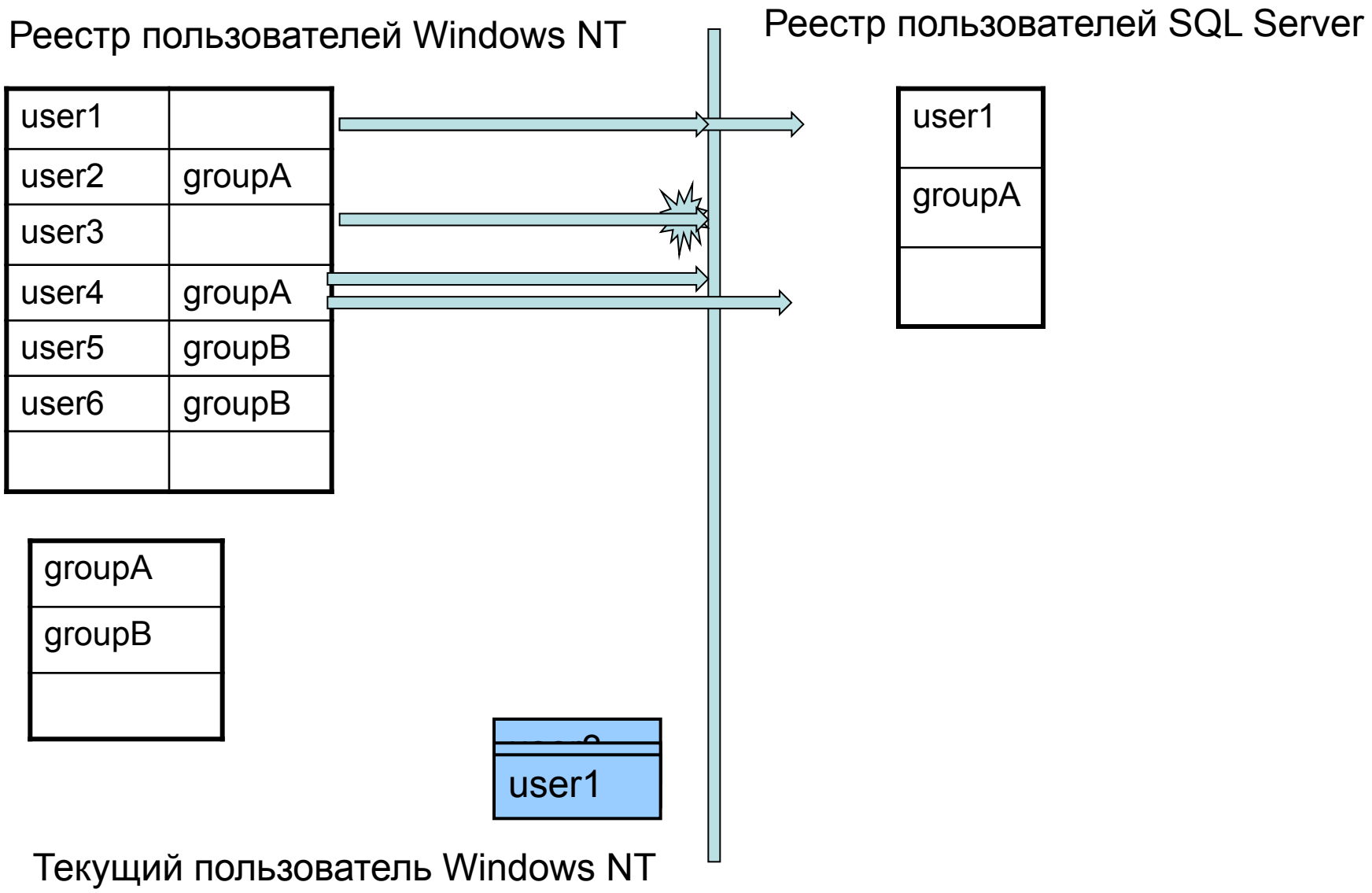
При подключении к SQL Server он выполняет считывание LoginID из БД системы безопасности домена. Проверка правильности ввода имени и пароля не производится, т.к. она выполнилась котроллером домена Windows NT.

SQL Server проверяет наличие LoginID пользователя Windows NT в своих структурах безопасности (системная таблица syslogins).

Если соответствие найдено, то доступ к серверу разрешается, если нет, то поиск продолжается для групп, к которым этот пользователь принадлежит, и если и там соответствие не найдено, то доступ к серверу отклоняется.

См.рис.

Аутентификация Windows



Аутентификация SQL Server

Доступ предоставляется на основании учетных записей SQL Server

При попытке получения доступа к SQL Server(у) он сам проверяет правильность имени пользователя и пароль, сравнивая их с данными в системная таблицах.

Учетные записи SQL Server

Учетные записи сервера (logins)

стандартные и пользовательские

Стандартные учетные записи создаются при установки сервера

- **BUILTIN\Administrators** — Учетная запись группы администраторов Windows NT, обеспечивающая доступ всем членам группы с полными правами

- **NT AUTHORITY\SYSTEM** — Учетная запись группы локальной учетной записи Windows NT

- **NT AUTHORITY\LOCAL SERVICE** — Учетная запись, используемая Windows для подключения к службам SQL Server (Reporting Services.)

-**sa** — **Учетная запись SQL Server для администратора сервера, обеспечивающая полный доступ. Она не может быть удалена.**

Пользовательские учетные записи создаются пользователями сервера, имеющие права на создание учетных записей!

Управление учетными записями

Учетных записи создаются командой T-SQL

CREATE LOGIN *loginName*

Имя учетной записи в SQL Server или полное имя учетной записи или группы Windows NT

{**WITH <option_list>** | **FROM <sources>** }

<option_list>::= для учетной записи SQL Server

PASSWORD = '*password*' [**MUST_CHANGE**] Пароль учетной записи

[, **SID** = sid | Идентификатор учетной записи

DEFAULT_DATABASE = *database* | БД по умолчанию

DEFAULT_LANGUAGE = *language* | Язык по умолчанию

CHECK_EXPIRATION = { ON | OFF} | Использование политики срока истечения пароля и политики силы пароля

CHECK_POLICY = { ON | OFF} [,...]

<sources> ::= для учетной записи Windows

WINDOWS [**WITH DEFAULT_DATABASE** = *database* |

DEFAULT_LANGUAGE = *language*[,...]] |

CERTIFICATE *certname* | **ASYMMETRIC KEY** *asym_key_name*

Управление учетными записями

Пример создания учетной записи SQL Server

```
CREATE LOGIN dev1
WITH PASSWORD=’12’,
DEFAULT_DATABASE=Заказы,
DEFAULT_LANGUAGE=[us_english],
CHECK_EXPIRATION=OFF,
CHECK_POLICY=OFF
```

Пример создания учетной записи Windows

```
CREATE LOGIN [IIT7\spfuser]
FROM WINDOWS WITH DEFAULT_DATABASE=Заказы,
DEFAULT_LANGUAGE=us_english
```

Для учетной записи домена Windows имя должен быть взято в квадратные скобки.

Управление учетными записями

Учетных записи создаются системной ХП

sp_addlogin [@loginame =] '*login*'

Имя учетной записи

[, [@passwd =] '*password*']

Пароль, ассоциируемый с учетной записью

[, [@defdb =] '*database*']

БД по умолчанию

[, [@deflanguage =] '*language*']

Язык по умолчанию

[, [@sid =] *sid*]

LoginID д.б. NULL

[, [@encryptopt =] '*encryption_option*']

Отмена режима шифрования пароля (skip)

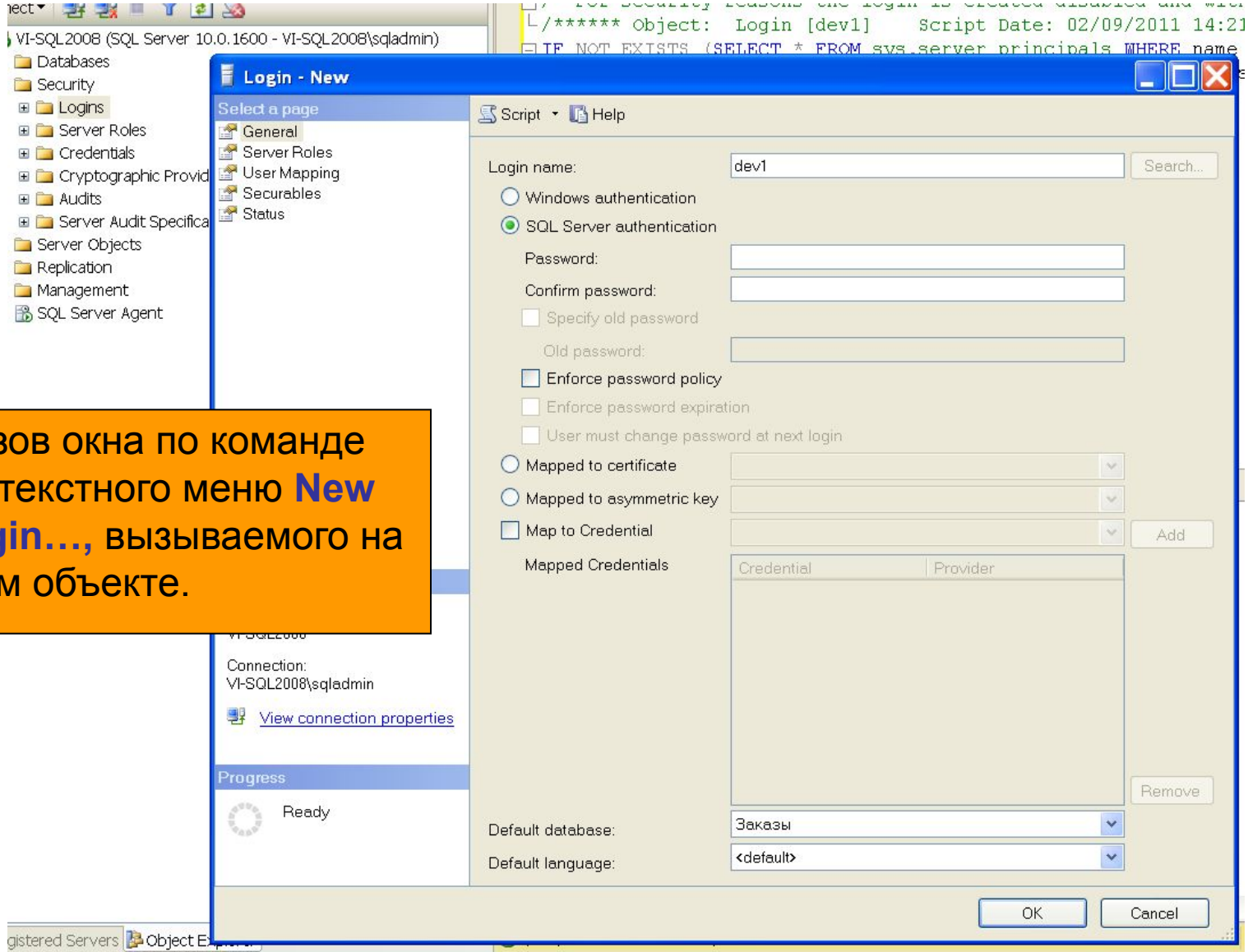
Разрешение доступа к серверу пользователям Windows NT выполняет системная ХП

sp_grantlogin [@loginame =] '*login*'

Полное имя учетной записи или группы Windows NT

Управление учетными записями

Создание учетных записей GUI в SSMS



Вызов окна по команде контекстного меню **New Login…**, вызываемого на этом объекте.

Пользователи БД

2-й уровень безопасности предусматривает получение доступа к БД сервера

Доступ к БД сервера получают пользователи БД

Пользователь – это объект БД, с помощью которого определяются все разрешения доступа к объектам БД (таблицы, представления, ХП, триггера и т.д.)

Для того, чтобы учетная запись (login) получила доступ к БД она должна быть “отображена” в пользователя этой БД (user)

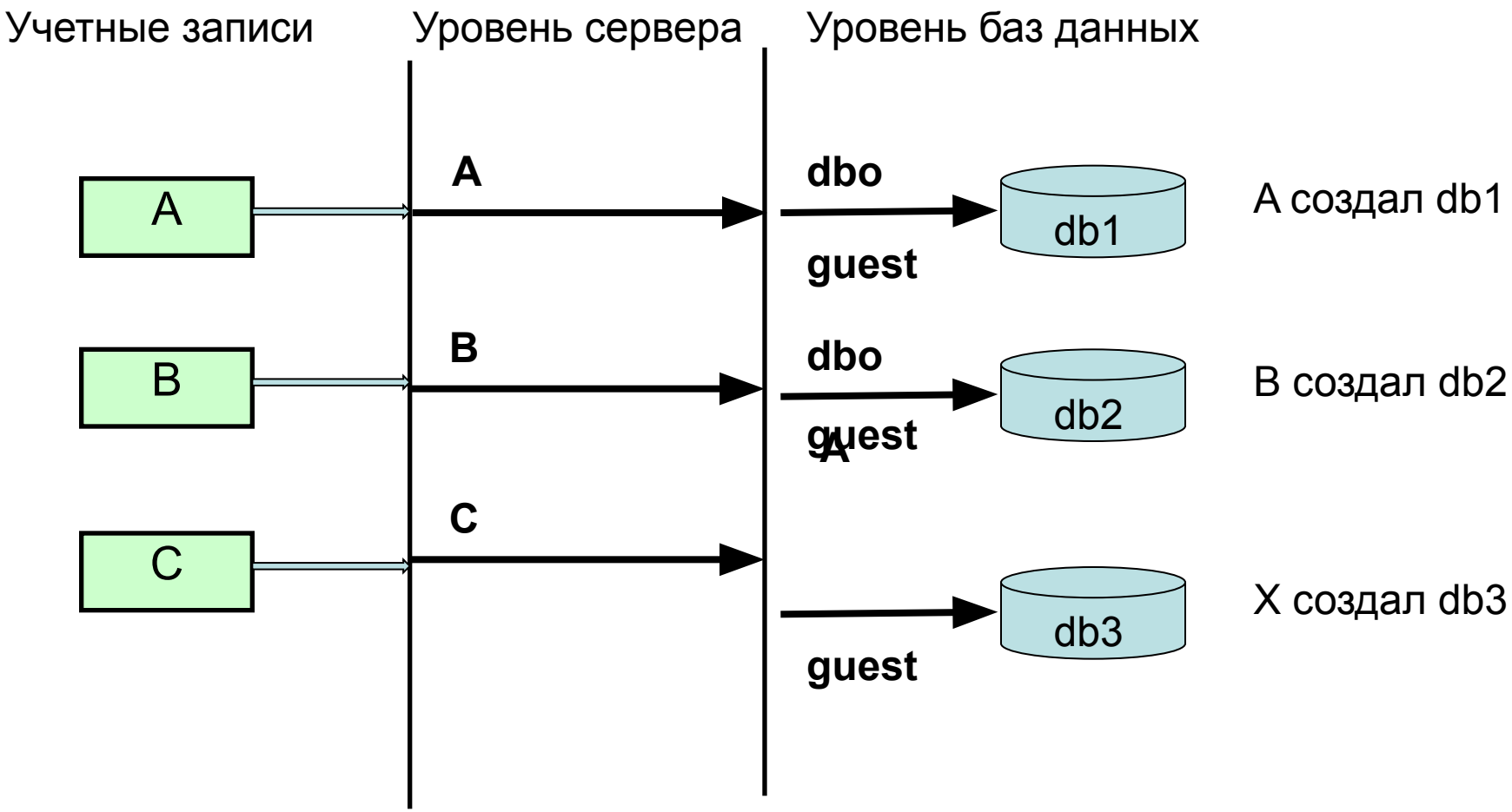
Пользователи БД

“Отображение” учетной записи в пользователя БД происходит:

- при создании БД	имя пользователя **dbo**	**права полные**
- явно	имя пользователя любое заданное	**права определенные**
- неявно	имя **guest**	права минимальные

см. рис.

Пользователи БД



Схемы БД

Раньше имя пользователя базы данных использовалось для идентификации принадлежности созданных им объектов

С версии SQL Server 2005 все объекты принадлежат схемам

Схема – это набор объектов в базе данных (таблицы, представления, ХП, триггера и т.д.), объединенных общим пространством имен.

Принадлежность объекта к схеме

Полный формат имени в SQL Server 2008

NameServer.NameDatabase.NameSchema.NameTable.NameColunm

Пользователю назначается схема по умолчанию. В эту схему SQL Server будет по умолчанию помещать объекты, которые создает этот пользователь.

Схемы БД

Применение схемы дает ряд дополнительных преимуществ по сравнению со старым подходом:

- нескольким пользователям можно назначить одну и ту же схему по умолчанию, что может быть удобно при разработке приложений;

- несколько пользователей (через группы Windows или роли баз данных) могут владеть одной и той же схемой. При этом один пользователь может являться владельцем сразу нескольких схем;

- при удалении пользователя из базы данных не придется переименовывать его объекты;

- упрощается предоставление разрешений для наборов объектов в базе данных.

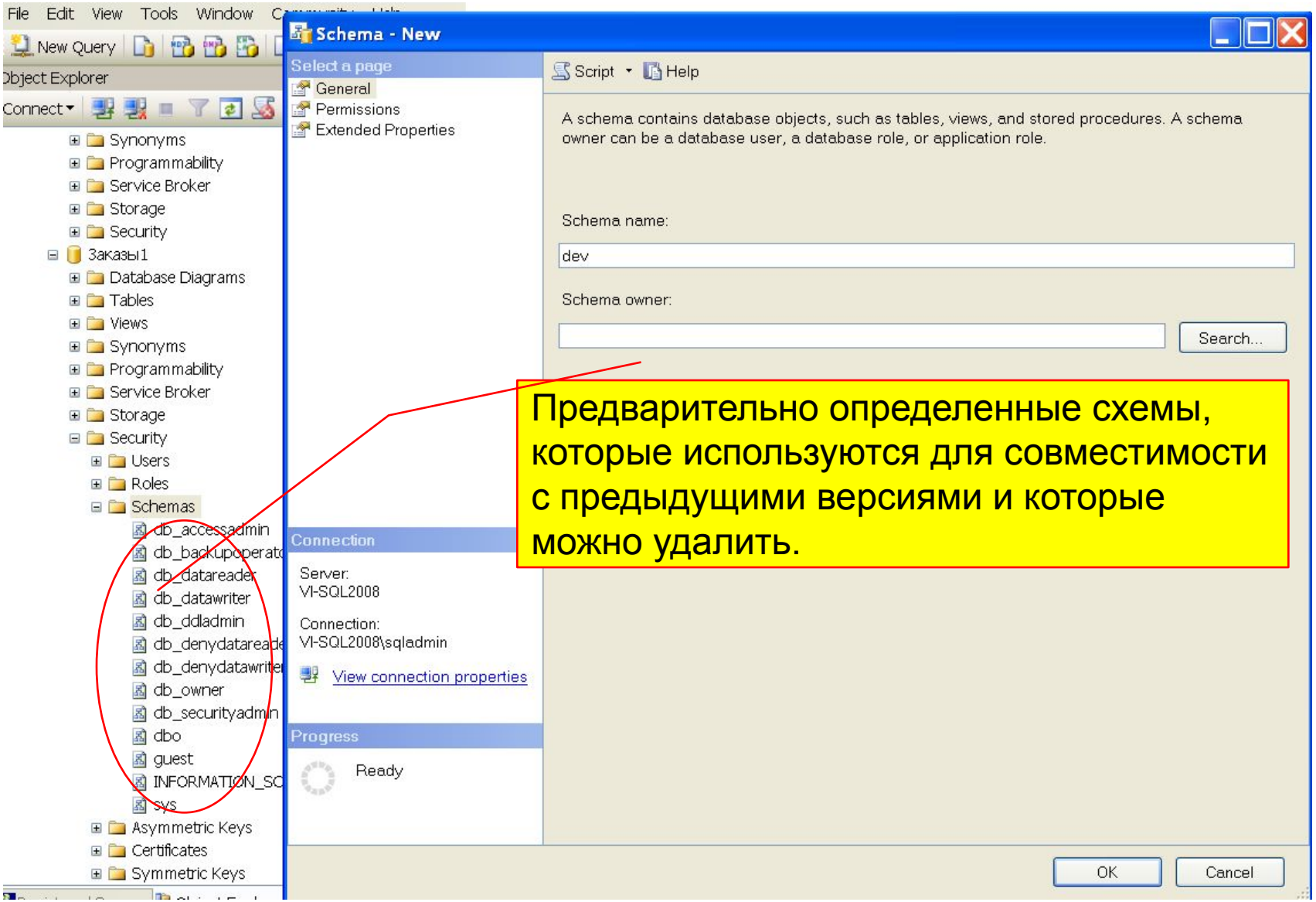
Создание схемы

Создание схемы БД выполняется:

- **GUI SSMS**
- **командой T-- командой T-SQL**

Создание схемы в SSMS

Диалоговое окно SSMS для создания схемы БД



Предварительно определенные схемы, которые используются для совместимости с предыдущими версиями и которые можно удалить.

Создание схемы в T-SQL

имя схемы в пределах базы данных

CREATE SCHEMA **schema_name** **AUTHORIZATION** ***owner_name***

Определяет имя пользователем базы данных, которому будет принадлежать схема. Этот пользователь может иметь другие схемы

Например,

CREATE SCHEMA dev AUTHORIZATION dbo

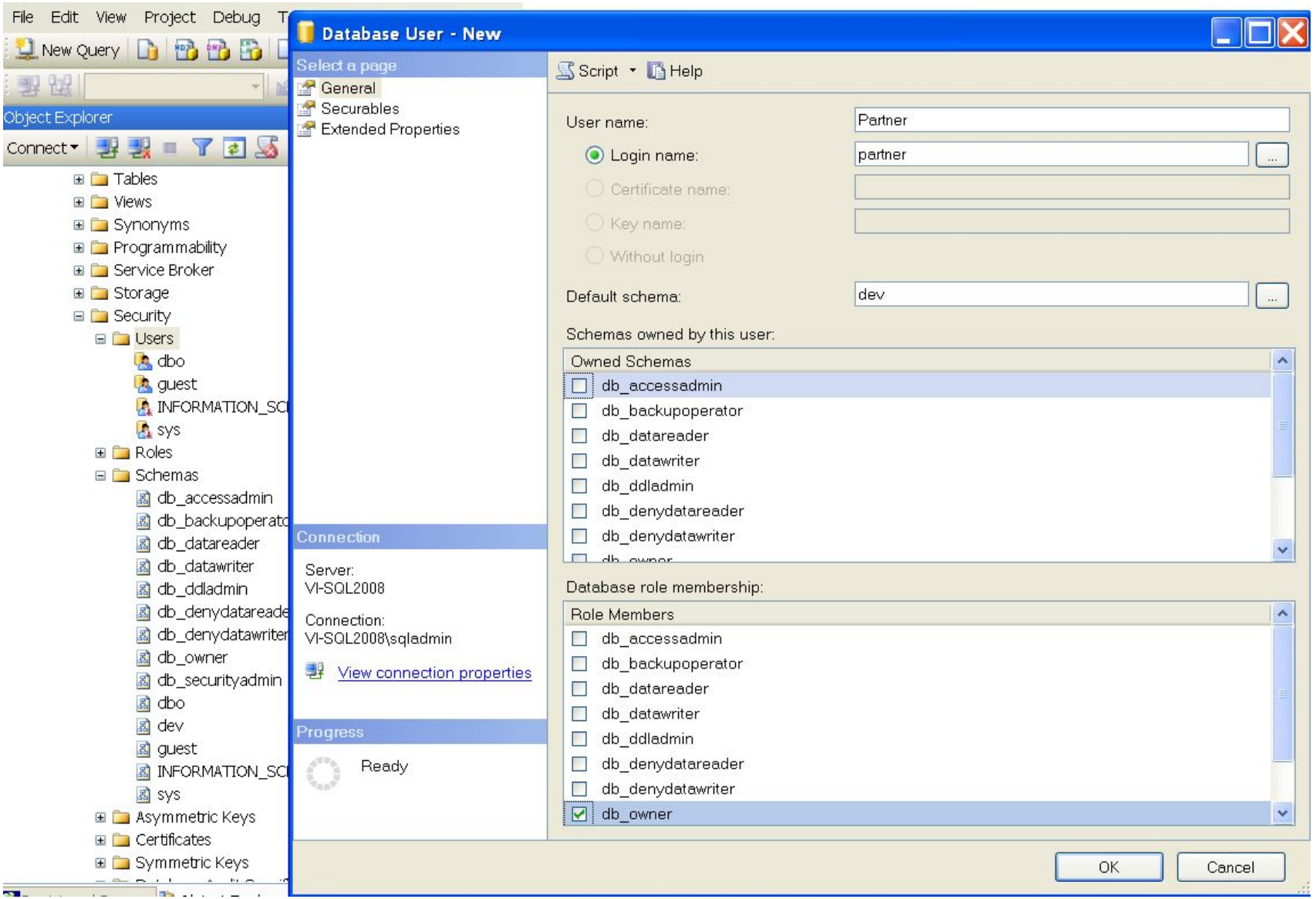
Создание пользователей БД

Создание пользователя БД выполняется:

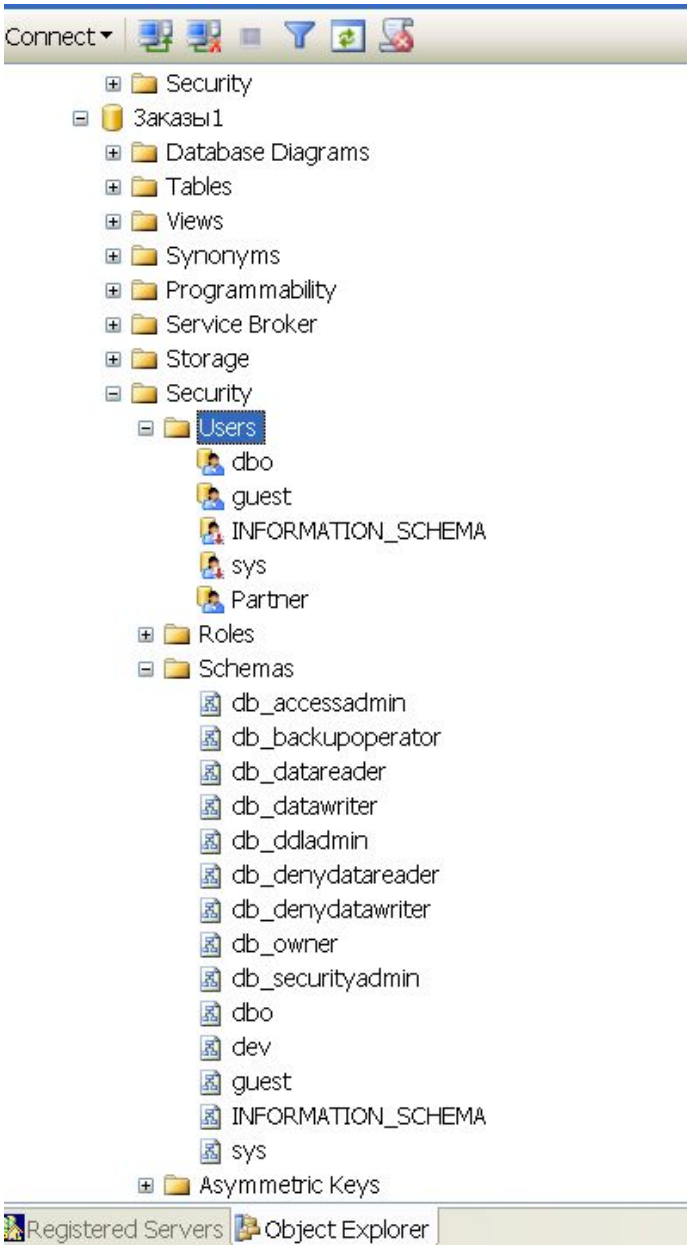
- **GUI SSMS**
- **командой T-- командой T-SQL**

Создание пользователей БД

Диалоговое окно SSMS для создания пользователя БД



Создание пользователей БД



Результат создания пользователя БД Заказ1

Создание пользователя в T-SQL

Имя пользователя базы данных

```
CREATE USER user_name FOR LOGIN login_name

[ WITH DEFAULT_SCHEMA = schema_name ]
```

Задает имя входа SQL Server, для которого создается пользователь базы данных.

Например,

```
CREATE USER Partner FOR LOGIN Partner
```

Создание пользователей БД

Создание пользователя БД хранимыми процедурами SQL Server

sp_adduser [**@loginame =**] '*login*'
[, [**@name_in_db =**] '*user*']
[, [**@grpname =**] '*group*']

имя уч.записи сервера

имя пользователя БД

роль пользователя в БД

sp_grantdbaccess [**@loginame =**] '*login*'
[,[**@name_in_db =**] '*name_in_db*']

имя уч.записи в Windows NT

имя пользователя БД с ролью public

Изменение владельца БД

sp_changedbowner [**@loginame =**] '*login*'
[, [**@map =**] *remap_alias_flag*]

имя уч.записи сервера нового владельца БД

определяет действия с уч. записью старого владельца БД

Роли и разрешения (права)

Роль – это именованный набор (комбинация) различных прав

Права – это разрешения на доступ и действия с объектами сервера или БД

В SQL Server имеется роли на уровне

- сервера

- базы данных

на уровне сервера только стандартные роли

на уровне БД роли

- фиксированные
- пользовательские
- неявные

Стандартные роли сервера

Стандартные роль сервера (fixed role server) определяют права учетной записи по администрированию сервера.

sysadmin	Можно выполнять любые действия на сервере
serveradmin	Можно выполнять конфигурирование и выключение сервера , но получать доступ к данным и изменять разрешения нельзя;
setupadmin	Можно управлять связанными серверами и процедурами, инсталлировать систему репликацией
processadmin	Можно управлять процессами, запускаемыми в SQL Server, т.е. закрытия пользовательских подключений к серверу (например, зависших)
diskadmin	Можно управлять файлами SQL Server
bulkadmin	Можно вставлять данные средствами массивного копирования, не имея непосредственного доступа к таблицам

Стандартные роли сервера

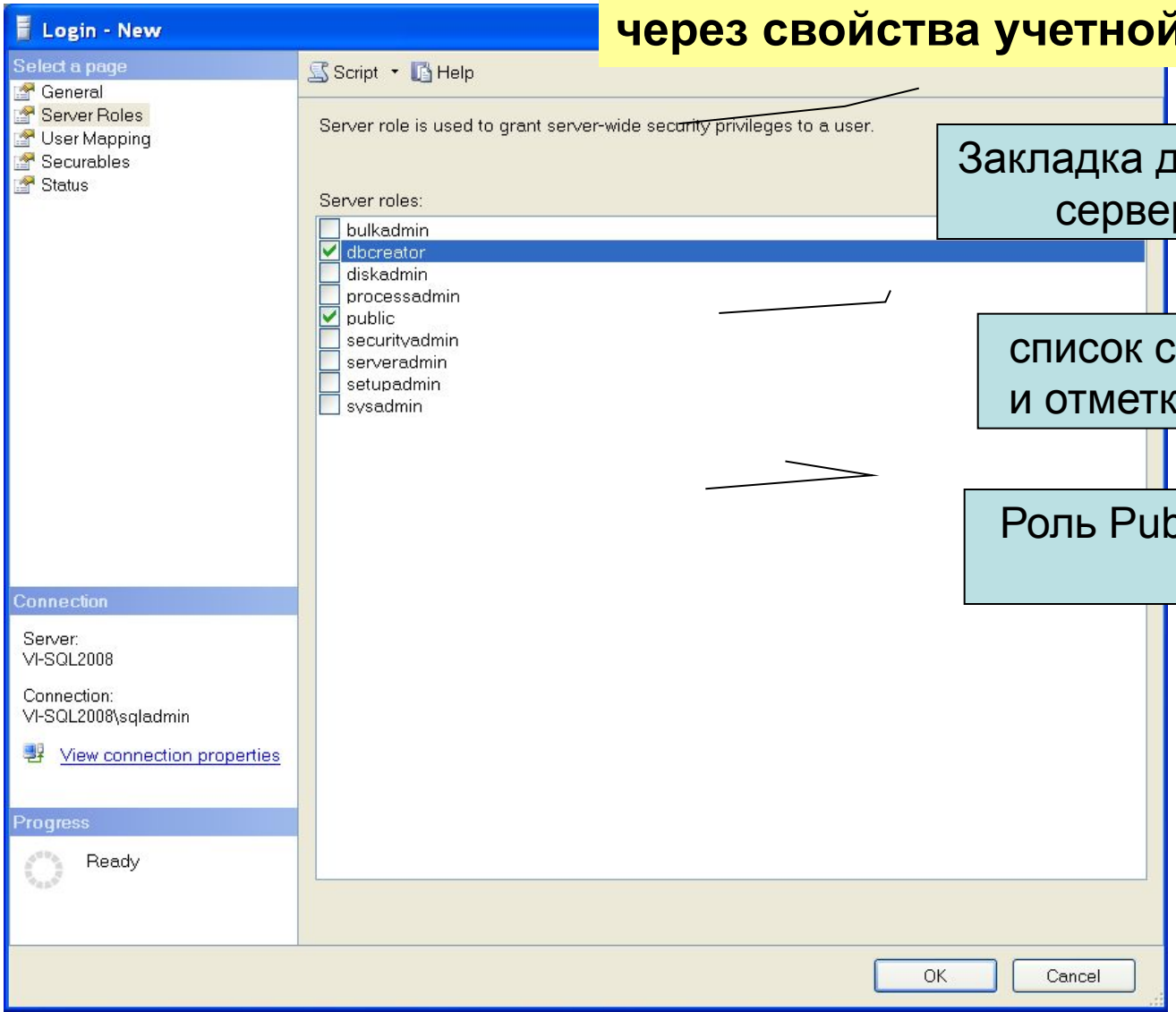
Стандартные роль сервера (fixed role server) определяют права учетной записи по администрированию сервера.

securityadmin	Можно управлять учетными записями и их свойствами, предоставлять, запрещать и отменять разрешения на уровне сервера, а также сбрасывать пароли для имен входа SQL Server.
dbcreator	Можно создавать, изменять, удалять и восстанавливать любые базы данных.
public	Можно только просматривать списки баз данных. Права этой роли автоматически получают все, кто подключился к SQL Server

Стандартные роли сервера

Серверные роли назначаются учетным записям в процессе их создания или позже.

Назначение серверных ролей в SSMS через свойства учетной записи



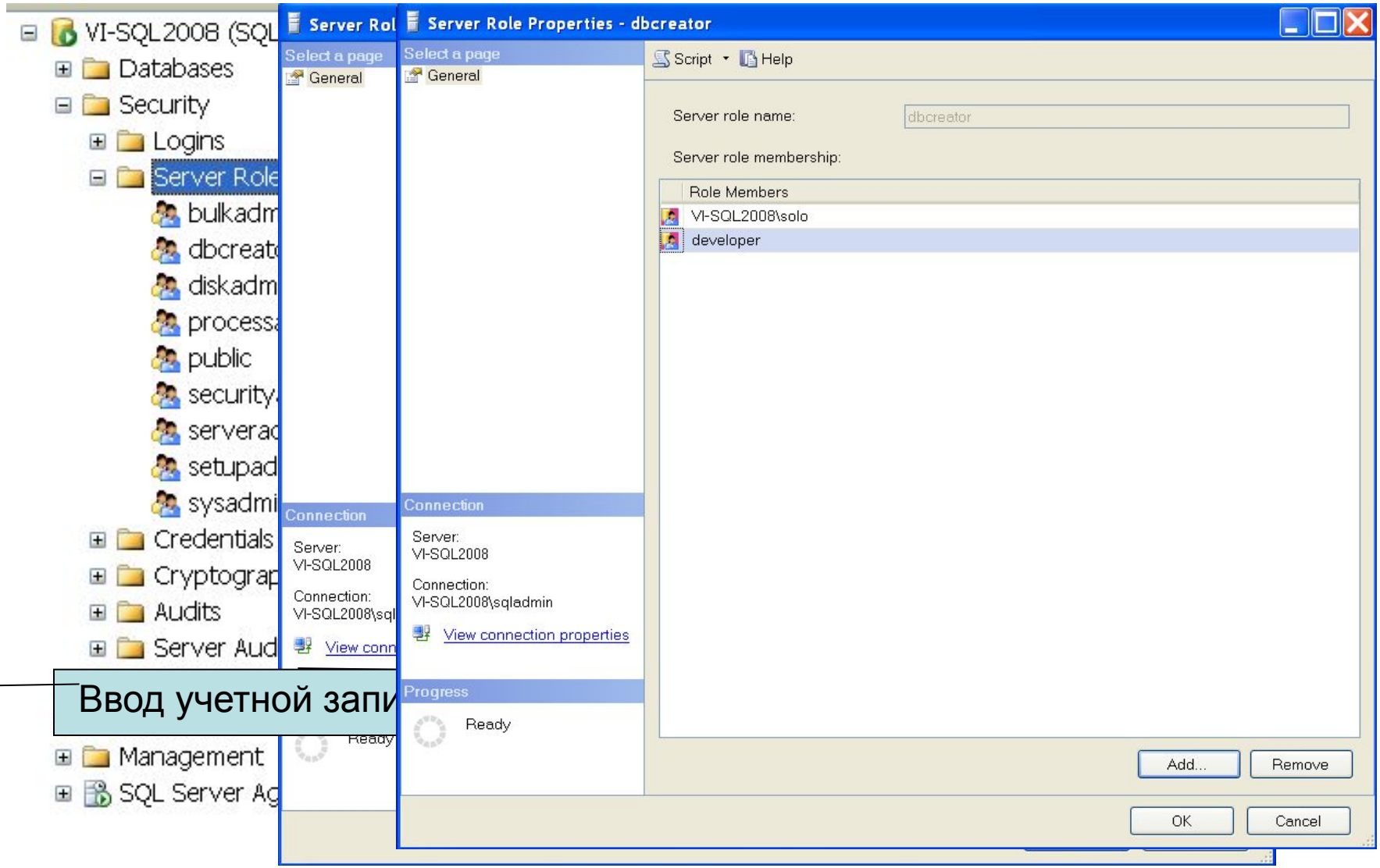
Закладка для назначения серверной роли

список серверных ролей и отметки их назначения

Роль Public назначается всегда

Стандартные роли сервера

Включение в серверную роль в GUI SSMS через свойства роли



Стандартные роли сервера

Включение в серверную роль, используя системную ХП

sp_addsrvrolemember [**@loginame =**] '*login*' имя уч.записи сервера
, [**@rolename =**] '*role*' имя серверной роли

Например,

sp_addsrvrolemember 'developer', 'dbcreator'

Исключение из серверной роли, используя системную ХП

sp_dropsrvrolemember [**@loginame =**] '*login*' имя уч.записи сервера
, [**@rolename =**] '*role*' имя серверной роли

Стандартные роли сервера

Включение в серверную роль, используя T-SQL

*Используется с версии 2012

```
ALTER SERVER ROLE name_role
ADD name_login
```

имя серверной роли

имя уч.записи сервера

Например,

```
ALTER SERVER ROLE dbcreator ADD developer
```

Исключение из серверной роли, используя T-SQL

```
ALTER SERVER ROLE name_role
DROP name_login
```

Роли уровня БД

на уровне БД роли

- фиксированные

- пользовательские

Фиксированные роли БД

Фиксированные роль БД (fixed roles database)

db_owner	Имеет все права БД
db_accessadmin	Можно создавать, изменять и удалять объекты пользователей баз данных, а также создавать схемы.
db_securityadmin	Можно управлять всеми разрешениями, объектами, ролями и членами ролей
db_ddladmin	Можно выполнять любые команды DDL
db_backupoperator	Можно выполнять резервное копирование БД
db_datareader	Можно выполнять выборку данных из таблиц и представлений БД
db_datarwriter	Можно выполнять любые команды DML
db_denydatareader	Запрещается выполнять выборку данных из таблиц и представлений БД
db_denydatawriter	Запрещается выполнять любые команды DML

Фиксированные роли БД

Роли БД в проводнике объектов в SSMS

Специальная роль

Все пользователи базы данных получают права этой роли автоматически.

Специально сделать пользователя членом этой роли или лишить его членства невозможно.

Фиксированные роли БД

Включение в роль БД, используя системную ХП

sp_addrolemember [**@rolename =**] '*role*', имя роли БД

[**@membername =**] '*user*' имя пользователя БД

Например,

sp_addrolemember 'db_datawriter ', 'dev2'

Исключение из роли БД, используя системную ХП

sp_droprolemember , [**@rolename =**] '*role*' , имя роли БД

[**@membername =**] '*user*' имя пользователя БД

Фиксированные роли БД

Включение в роль БД, используя T-SQL

*Используется с версии 2012

```
ALTER ROLE name_db_role
ADD MEMBER name_db_user
```

имя роли БД

имя пользователя БД

Например,

```
ALTER ROLE db_datawriter ADD MEMBER dev2
```

Исключение из роли БД, используя T-SQL

```
ALTER ROLE name_db_role
DROP MEMBER name_db_user
```

Например,

```
ALTER ROLE db_datawriter DROP MEMBER dev2
```

Пользовательские роли

Пользовательские роли

- стандартные

Позволяют логически сгруппировать пользователей в соответствии с предъявляемыми требованиями

- приложения

Для получения доступа к БД из приложения, запускаемого любым пользователем, даже не имеющим права работы с сервером, но имеющего право работать с приложением.

Роли приложения

Отличие роли приложения (***application role***) – разрешения предоставляются не пользователю, а приложению, которое подключается к базе данных.

Применение роли приложения:

В приложении от имени логина (которому соответствует пользователь в БД только с правом CONNECT и других прав не имеет) выполняется подключение к серверу с нужной БД с качестве текущей;

Затем выполняется ХП **sp_setapprole** для активизации указанной роли приложения, после чего приложение получает права этой роли (и теряет свои текущие права полученные при подключении);

Далее приложение может выполнять свои операции с БД, на которые в данной роли приложения должны быть предоставлены права …

После выполнения действий приложение может отключиться от сервера или переключиться на свою исходную учетную запись (и получить соответствующие ей права) можно, выполнив ХП **sp_unsetapprole**.

Разрешения (права)

Явные

Это права на доступ к объектам БД (конкретным таблицам, столбцам, представлениям, ХП) пользовательской БД

Неявные

Это права полученные при определенных обстоятельствах, например, вхождением в роль.

Права выдаются

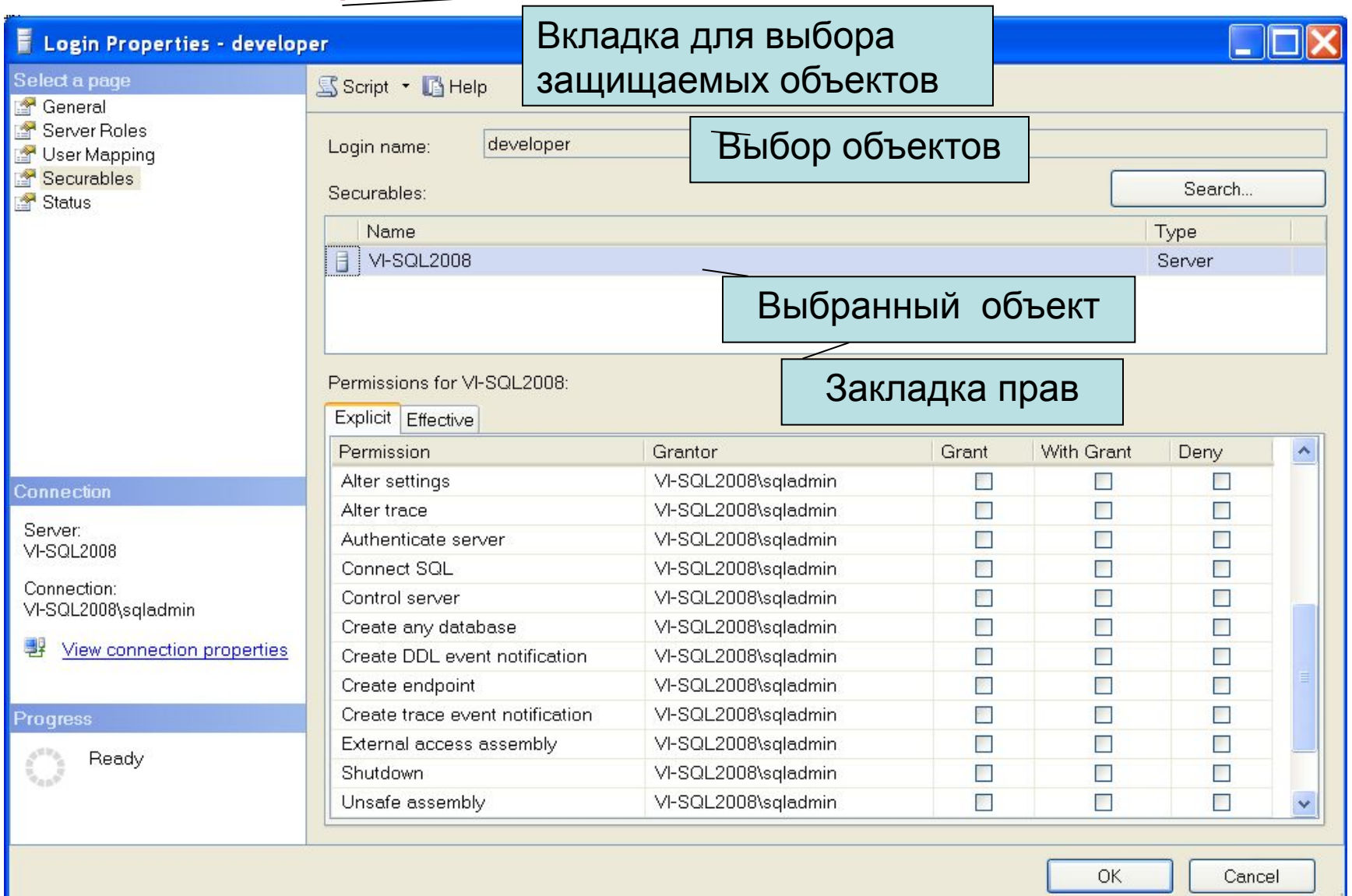
- администратором сервера

- владельцем БД

- владельцем объекта

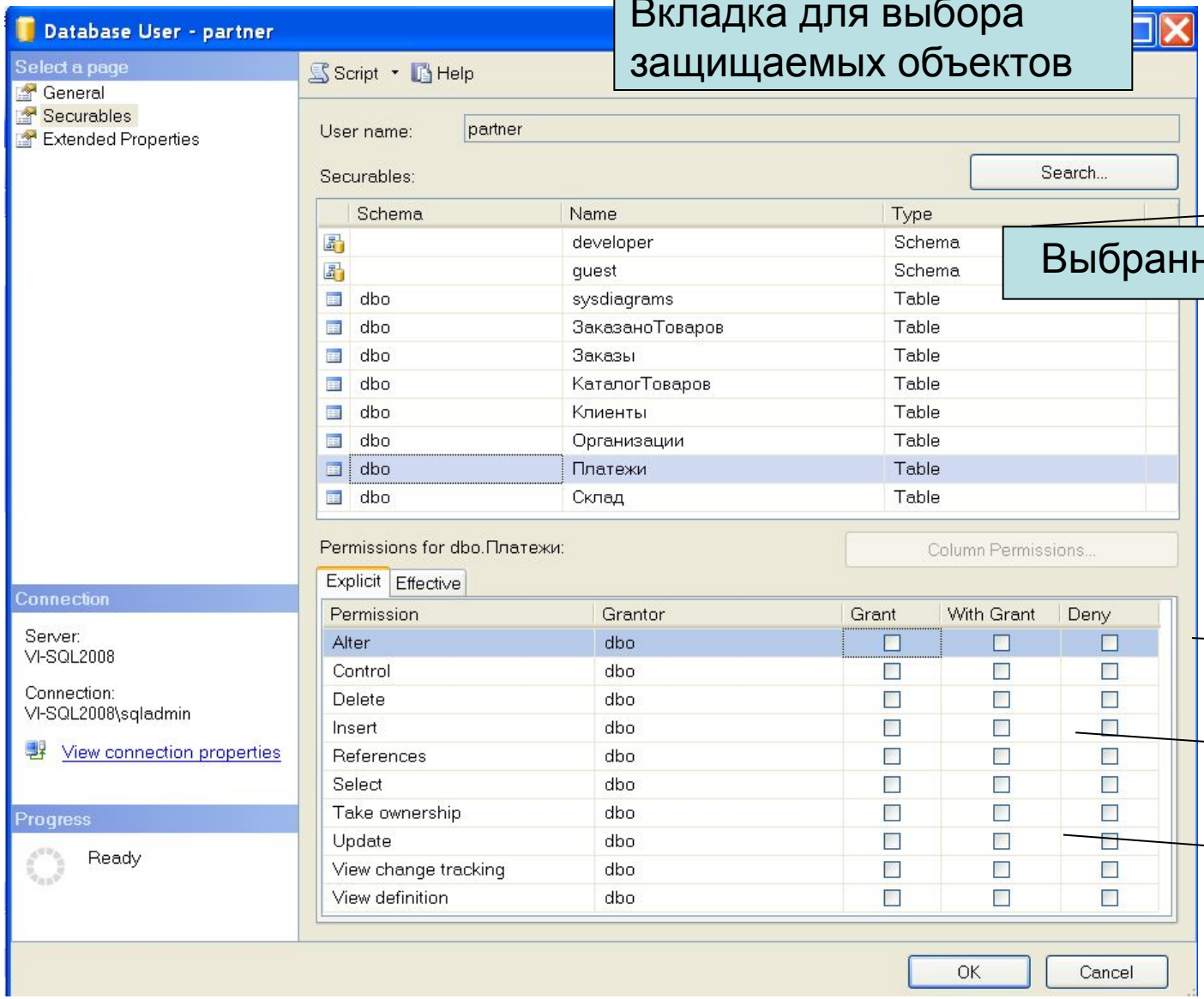
Права

Вкладка для задания прав уровня сервера для учетной записи developer в SSMS



Права

Вкладка для задания прав пользователя БД partner в SSMS



Вкладка для выбора защищаемых объектов

Выбор объектов

Выбранный объект

Закладка прав

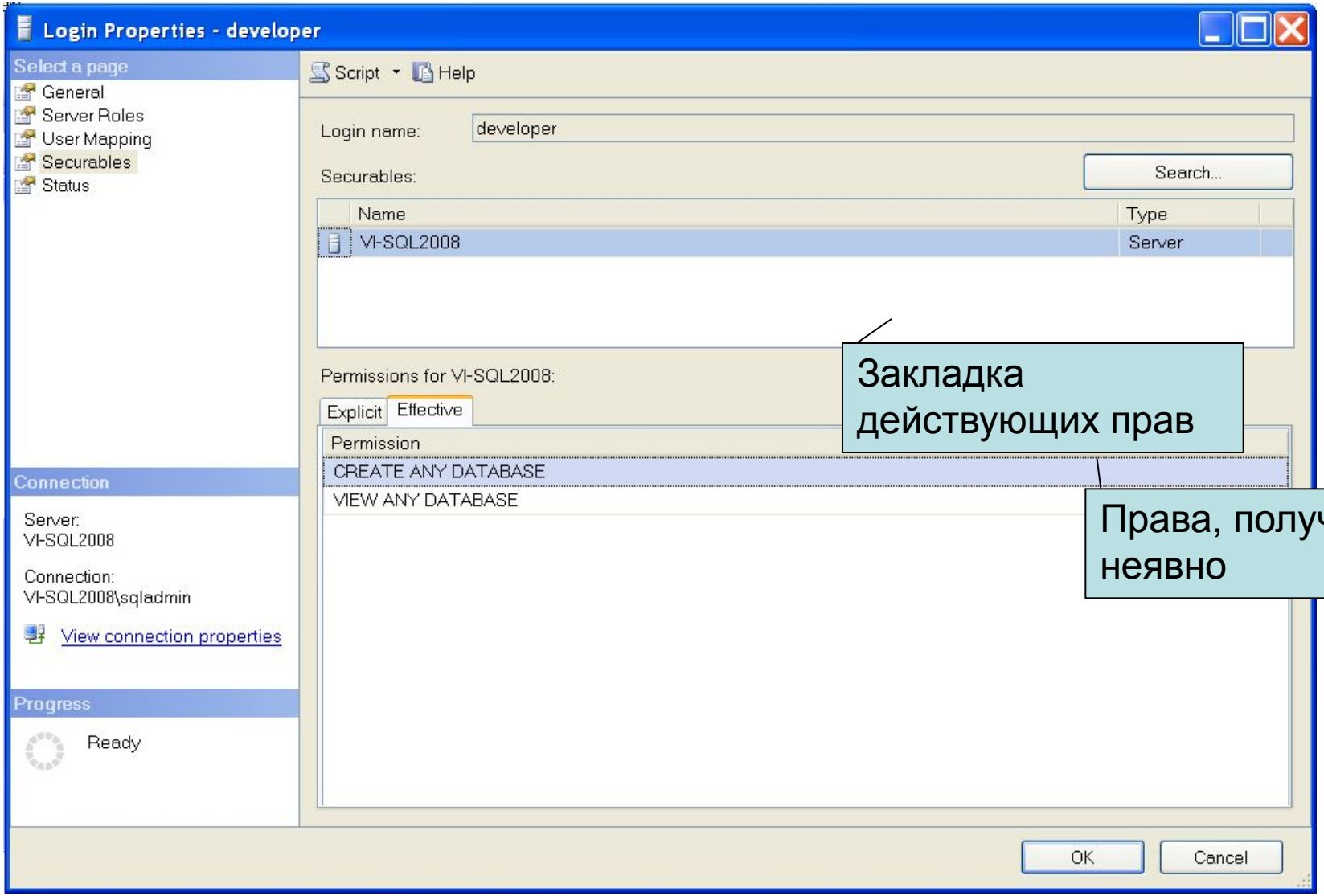
Запрещение

Разрешение

Разрешение на передачу прав

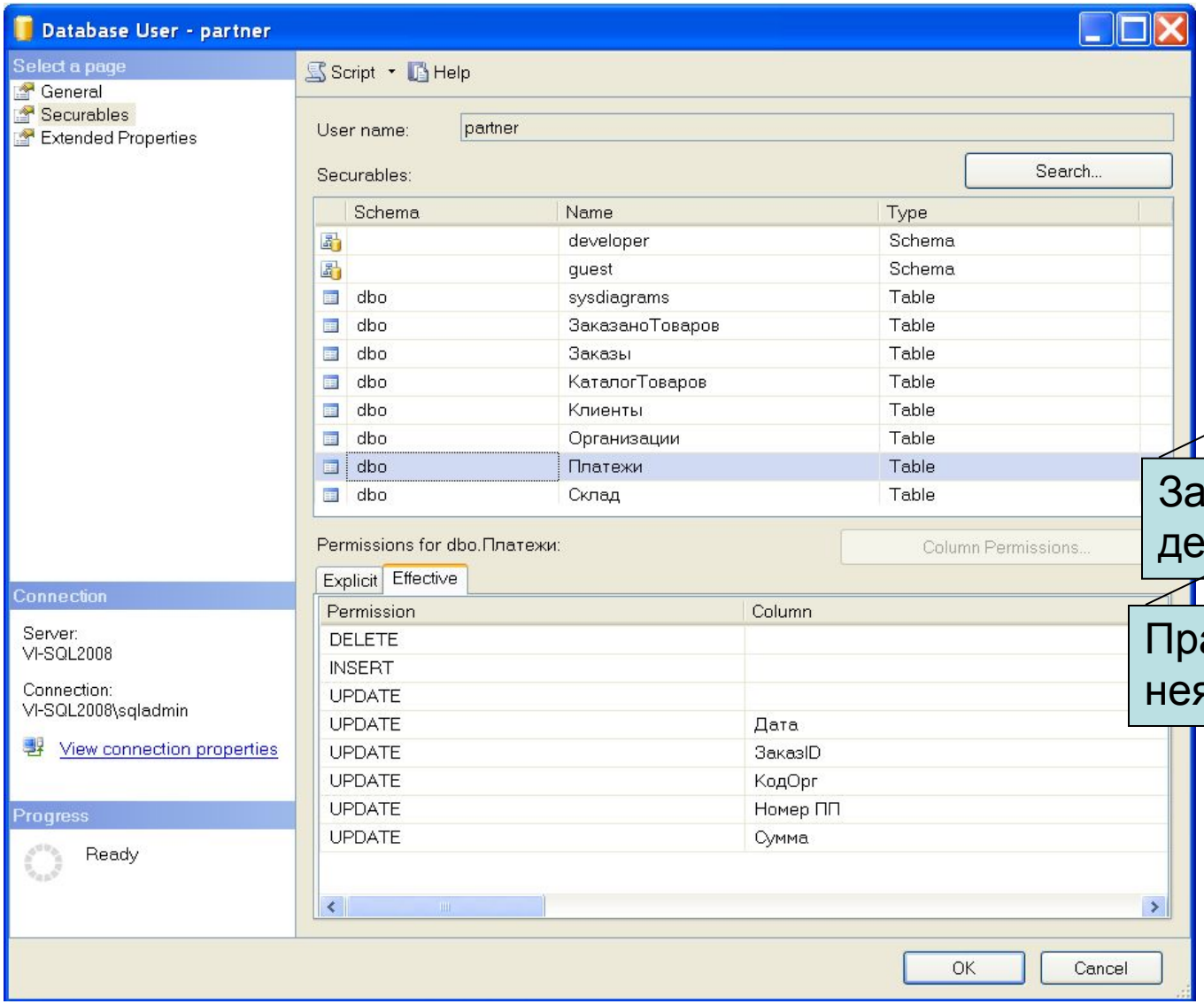
Права

Вкладка для просмотра действующих прав уровня сервера для учетной записи developer в SSMS



Права

Вкладка для просмотра действующих прав пользователя БД partner в SSMS



Закладка действующих прав

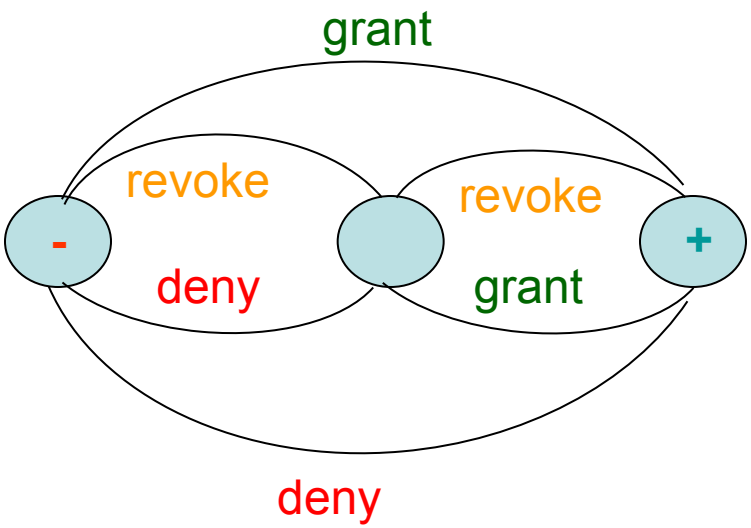
Права, полученные неявно

Команды SQL управления доступом

GRANT Предоставляет разрешения на объектам участнику

DENY Запрещает разрешения на объектам участнику

REVOKE Отклоняет разрешения на объектам участнику



Команды SQL управления доступом

содержат

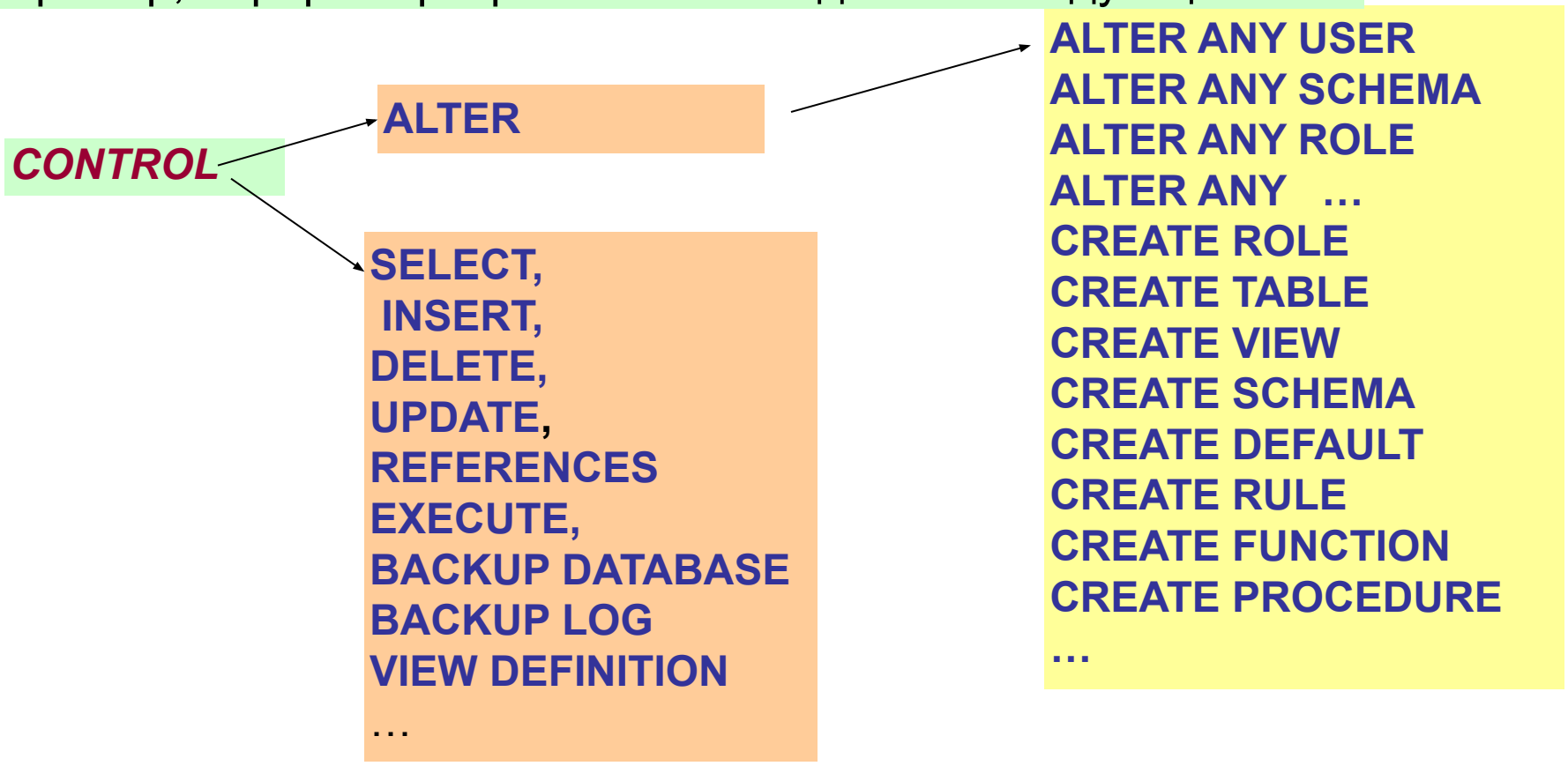
Защищаемые объекты	участников
Уровень сервера	
Имя входа База данных	Уч. запись (имя входа) Серверная роль
Уровень базы данных	
Роль Роль приложения Пользователь Схема Сборка Сертификат Асимметричный ключ …	Пользователь Роль базы данных Роль приложения
Уровень схемы	
Таблица Представление функция Процедура Тип Синоним …	

Разрешения (права)

Все разрешения находятся в иерархической подчиненности.

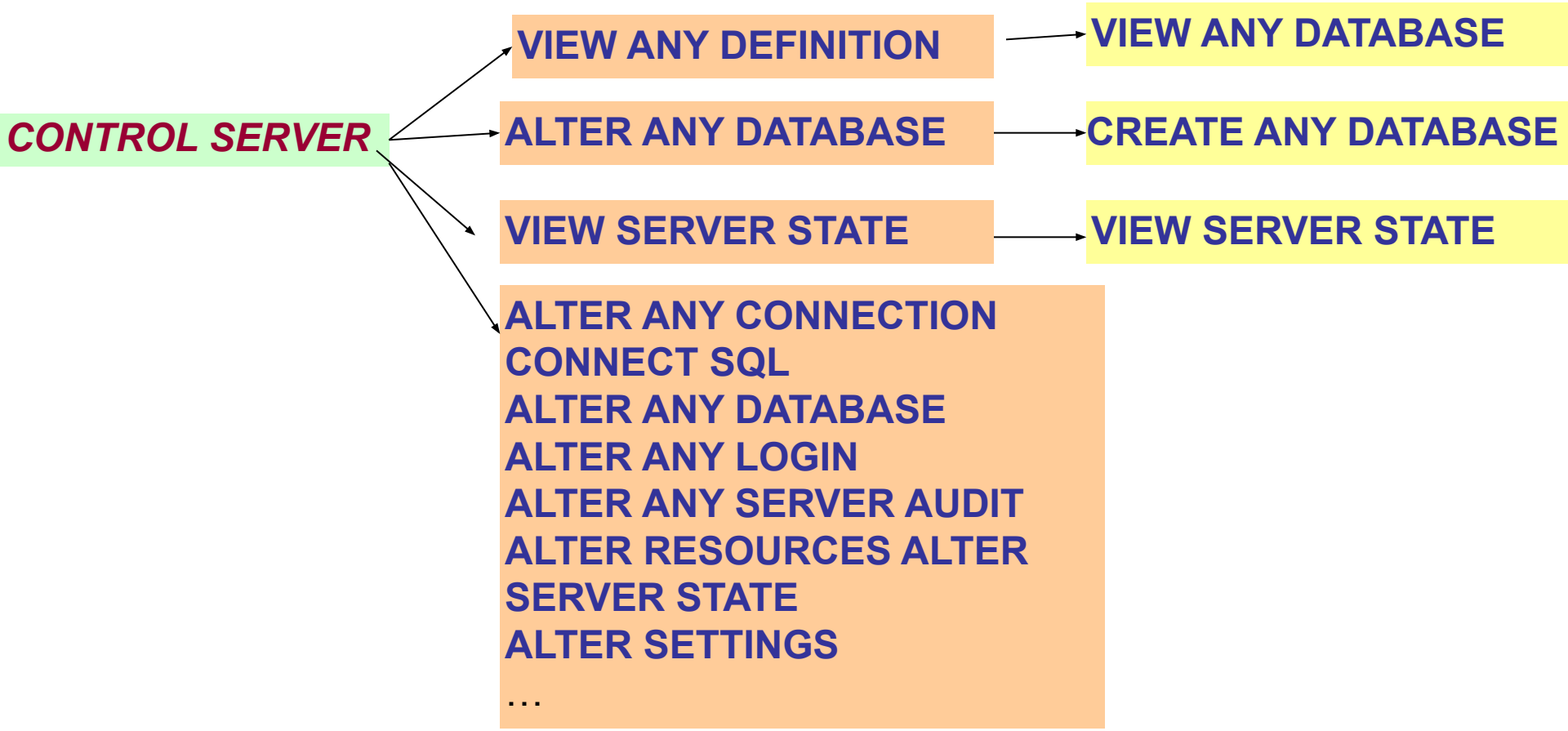
Получив разрешение на одном уровне наследуется все разрешения в нижестоящей в иерархии разрешения.

Например, иерархия разрешений базы данных следующая



Разрешения (права)

Иерархия разрешений сервера следующая



Команда GRANT

GRANT
{ ALL [PRIVILEGES] } | ***permission*** [(column,…)] ,…]
[ON [***class*** ::] ***securable***]
TO ***principal*** ,…
[WITH GRANT OPTION]
[AS ***principal***]

ALL [PRIVILEGES] – разрешает все возможные для защищаемого объекта разрешения

permission *– это разрешения, которые могут иметь защищаемые объекты*

class– *это* класс защищаемого объекта, для которого предоставляется разрешение**: LOGIN**, **USER, ROLE, APPLICATION ROLE**, **OBJECT (по умолчанию), SCHEMA …**

securable– это защищаемый объект, на который предоставляется разрешение: на **уровнях сервера, базы данных, схемы или конкретного объекта схемы**.

principal – это имя участника: принципал сервера или базы данных

WITH GRANT OPTION– позволяет пользователю, которому предоставляются права назначать их другим пользователям

AS *principal* - определяет участника от которого участник, выполняющий данный запрос, наследует право на предоставление разрешения.

Команда Grant

Разрешения на объекты базы данных :

```
GRANT  { ALL [ PRIVILEGES ] | permission ,… }
   {
      [ ( column ,… ) ] ON {[ OBJECT :: ]  [ schema. ] table | view }
      | ON { [ schema . ] table | view } [ ( column,…) ]
      | ON { [ schema . ] stored_procedure | extended_procedure }
      | ON { [ schema . ] user_defined_function }
   }
TO database_principal,…]
[ WITH GRANT OPTION ]
[ AS database_principa ]
```

permission *- это* **CONTROL, ALTER, SELECT, INSERT, DELETE, UPDATE, REFERENCES, EXECUTE, VIEW DEFINITION** …

database_principa – это пользователь базы данных, или роль базы данных, или роль приложения и др. участники

WITH GRANT OPTION– позволяет пользователю, которому предоставляются права назначать их другим пользователям

AS *database_principal* - определяет принципал базы данных, у которого участник, выполняющий данный запрос, наследует право предоставлять данное разрешение.

Команда Grant для схемы

Разрешения на схему:

GRANT
permission ,…
ON SCHEMA :: schema_name
TO *database_principal*,…
[**WITH GRANT OPTION**]
[**AS** { *user* | *roler* | *role app* }]

permission *– это*
CONTROL, ALTER, CREATE SEQUENCE, EXECUTE, SELECT, INSERT, DELETE, UPDATE, REFERENCES VIEW CHANGE TRACKING, VIEW DEFINITION,

database_principa – это пользователь базы данных, или роль базы данных, или роль приложения и некоторые др. участники

WITH GRANT OPTION– позволяет пользователю, которому предоставляются права назначать их другим пользователям

В БД используются десятки - сотни таблиц и др. объектов . Предоставлять каждому пользователю разрешения на каждый из этих объектов очень неудобно. Поэтому лучше использовать разрешения на уровне схемы или всей базы данных (например, через встроенные роли баз данных **db_datareader** и **db_datawriter**)

Пример команды Grant

```
GRANT CREATE DATABASE, CREATE TABLE
TO bokov, dirina, IIT7\spfuser

 USE pubs
 GO
 GRANT SELECT
 ON authors
 TO public
 GO
 GRANT INSERT, UPDATE, DELETE
 ON authors
 TO bokov, dirina
 GO
```

Команды DENY, REVOKE

DENY | **REVOKE** [GRANT OPTION FOR]
{ ALL [PRIVILEGES] } | ***permission*** [(column,…)] ,…]
[ON [***class*** ::] ***securable***]
TO ***principal*** ,…
[CASCADE]
[AS ***principal***]

ALL [PRIVILEGES] – разрешает все возможные для защищаемого объекта разрешения

permission *– это разрешения, которые могут иметь защищаемые объекты*

class– *это* класс защищаемого объекта, для которого предоставляется разрешение: **OBJECT (по умолчанию), SCHEMA**

securable– это защищаемый объект, на который предоставляется разрешение: на **уровнях сервера, базы данных, схемы или конкретного объекта схемы**.

principal – это имя участника: принципал сервера или базы данных

C**ASCADE -** обозначает, что разрешение запрещается для указанного участника и всех других участников, которым этот участник предоставил разрешение (если участник имеет разрешение с параметром WITH GRANT OPTION). Для **REVOKE** требуется указывать GRANT OPTION FOR

AS *principal* - определяет участника от которого участник, выполняющий данный запрос, наследует право на предоставление разрешения.

Пример команды Deny

```
DENY CREATE DATABASE, CREATE TABLE
TO bokov, dirina, IIT7\spfuser

  USE pubs
  GO
  GRANT SELECT
  ON authors
  TO public
  GO
  DENY SELECT, INSERT, UPDATE, DELETE
  ON authors
  TO bokov, dirina
  GO
```

Порядок действий для предоставления доступа к БД

1. Создать учетную запись (роль Public)

```
create login XXX
with password =…, …
```

2. Создать пользователя БД (роль Public)

```
create user XXX
for login XXX
with default_schema = …
```

3. Назначить ему права на БД

   а) неявно через права на БД (включение в роли БД)

```
sp_addrolemember ‘db_datareader’, XXX
…
```

   б) неявно через права на схему(ы) БД

```
grant insert,  … on schema::sss1 to XXX
grant insert,  … on schema::sss2 to XXX
…
```

   в) явно на каждый объект БД

```
grant insert,  … on object::ooo1 to XXX;
grant insert,  … on object::ooo2 to XXX;
…
```