Раздел 6. Администрирование серверов баз данных

Тема 6.1

Обеспечение безопасности данных *(на примере MS SQL Server)*

Вопросы лекции:

1. **Принципы системы безопасности БД.**
2. Управление учетными записями сервера БД.
3. Управление пользователями БД.
4. Управление правами и ролями.

Принципы системы безопасности серверов БД

Системы безопасности большинства современных серверов:

1) основаны на принципах избирательного подхода

2) применении 2-х уровневой модели доступа

3) использовании

- аутентификации
- авторизации
- шифрования

Аутентификация – это установление соответствия лица названному им идентификатору

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

Шифрование – это процесс кодирования информации

Система безопасности Microsoft SQL Server

1) основана на принципах избирательного подхода

2) реализуется 2-х уровневой моделью – уровень сервера и уровень БД

3) использует :

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

- учетную запись БД - пользователь (user) или принципал базы данных (database principal) для авторизации;

  - схему (schema);

  - роли (roles);

  - группы (groups);

Понятия модели безопасности

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

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

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 и **они обязательно авторизуются в системе**.

Уровни безопасности сервера

1-й уровень – сервера

2-й уровень – базы данных

Режимы аутентификации, реализуемые сервером

- средствами MS SQL Server

- средствами ОС Windows

Аутентификация Windows

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

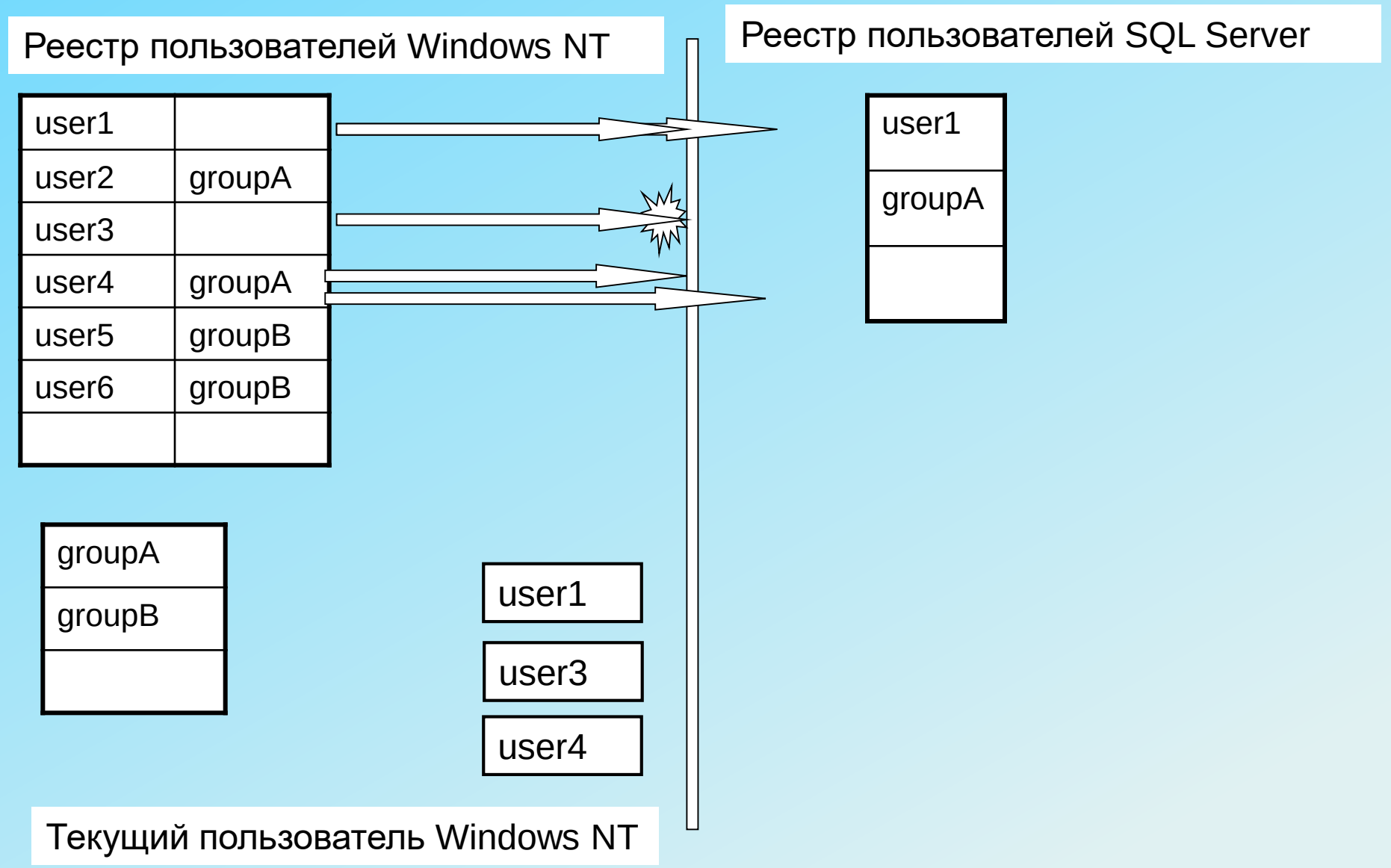
При подключении к SQL Server он выполняет считывание LoginID из БД системы безопасности домена Windows. Проверка правильности ввода имени и пароля не производится, т.к. она выполнилась котроллером домена Windows NT.

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 внутренних 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 системных таблицах.

Учетные записи SQL Server

Учетные записи сервера (logins) - стандартные и пользовательские

Стандартные учетные записи, которые создаются при установке сервера:

- **BUILTIN\Administrators**	Учетная запись Windows NT, обеспечивающая доступ всем членам группы Administrators к SQL Server(у) с полными правами
- **NT AUTHORITY\SYSTEM**	имя локальной системной учетной записи Windows NT
- **NT AUTHORITY\LOCAL SERVICE**	Учетная запись, используемая Windows для подключения к 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, имеющими права на создание учетных записей!

Раздел 6. Администрирование серверов баз данных

Тема 6.1

Обеспечение безопасности данных *(на примере MS SQL Server)*

Вопросы лекции:

1. Принципы системы безопасности сервера БД.
2. **Управление учетными записями сервера БД.**
3. Управление пользователями БД.
4. Управление правами и ролями.

Управление учетными записями

Учетные записи могут создаваться 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

Учетные записи SQL Server могут создаваться 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

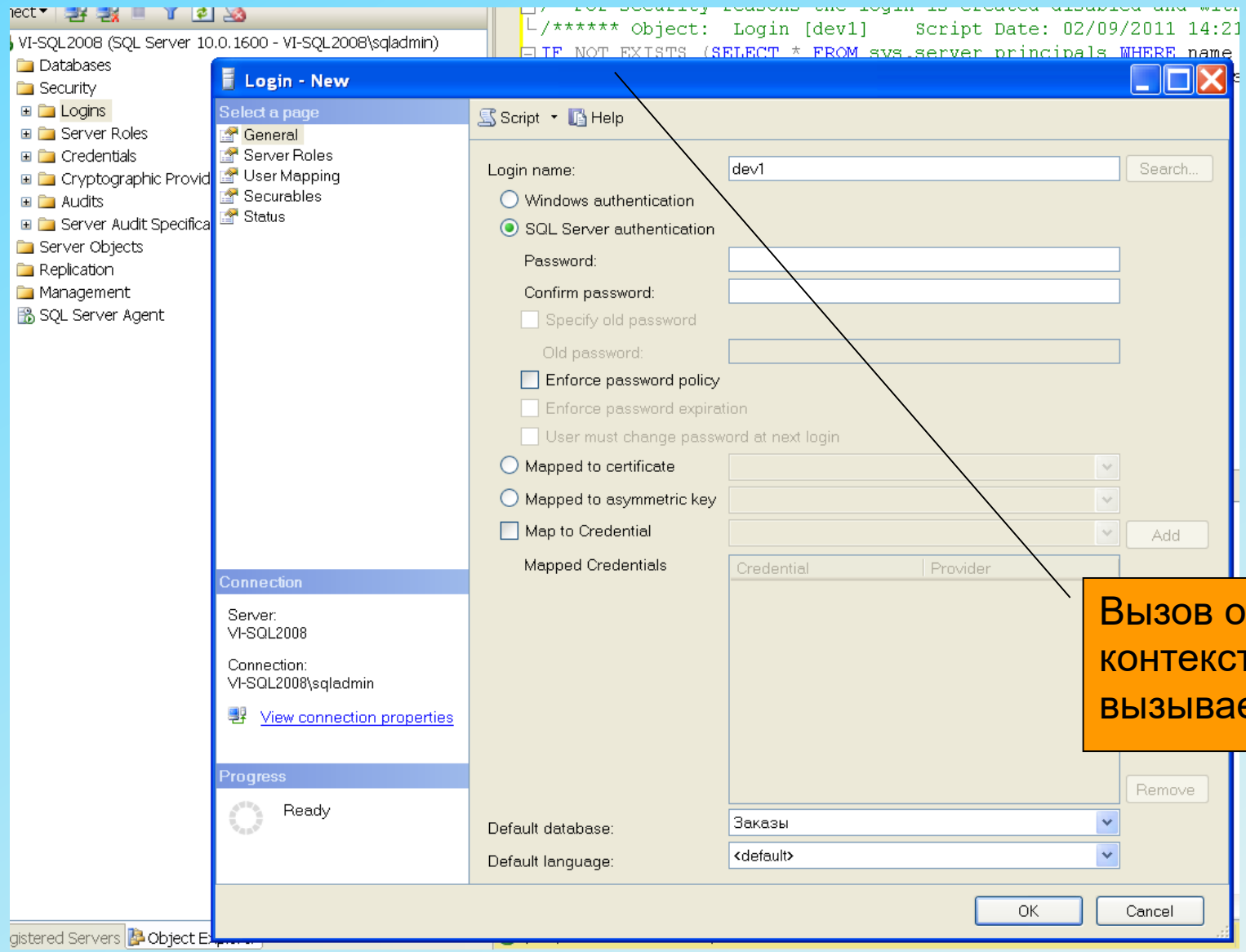
[, [@encryptopt =] '*encryption_option*'] — Отмена режима шифрования пароля (skip)

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 через GUI в среде SSMS



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

Раздел 6. Администрирование серверов баз данных

Тема 6.1

Обеспечение безопасности данных *(на примере MS SQL Server)*

Вопросы лекции:

1. Принципы системы безопасности сервера БД.
2. Управление учетными записями сервера БД.
3. **Управление пользователями БД.**
4. Управление правами и ролями.

Пользователи БД

2-й уровень безопасности MS SQL SERVER предусматривает получение **доступа к конкретным БД** сервера.

Доступ к БД сервера получают пользователи БД

Пользователь (**USER**) – это объект БД, с помощью которого определяются все разрешения доступа ко всем остальным объектам БД (таблицы, представления, ХП, триггеры и т.д.)

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

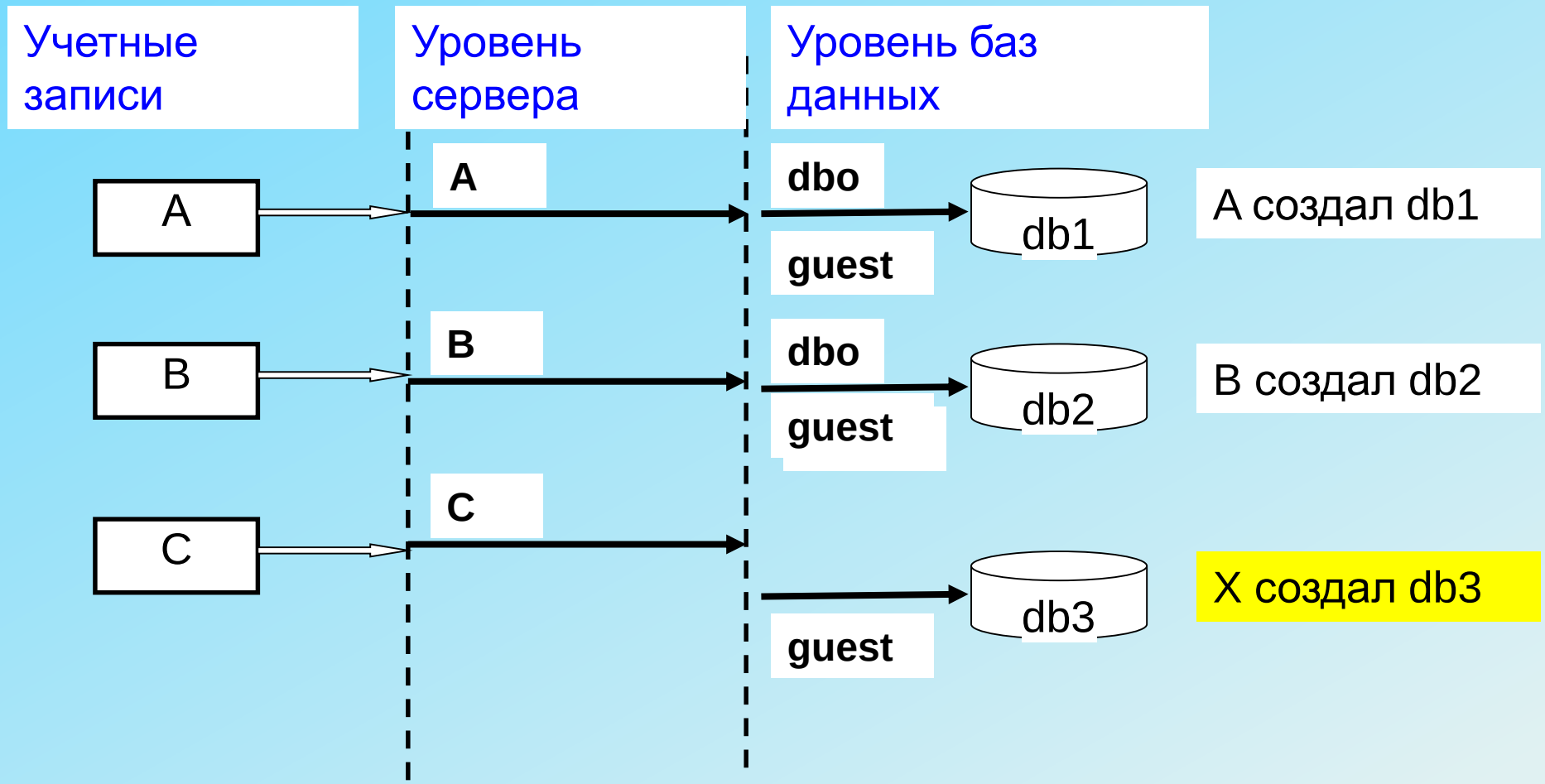
Пользователи БД

“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



Пользователи БД

В ранних версиях MS SQL Server имя пользователя базы данных использовалось для идентификации принадлежности созданных им объектов

С версии SQL Server 2008 все объекты принадлежат схемам

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

Полный формат имени в SQL Server 2008 и поздних версиях

NameServer.NameDatabase.NameSchema.NameTable.NameColunm

Принадлежность объекта к схеме

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

Пользователи БД

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

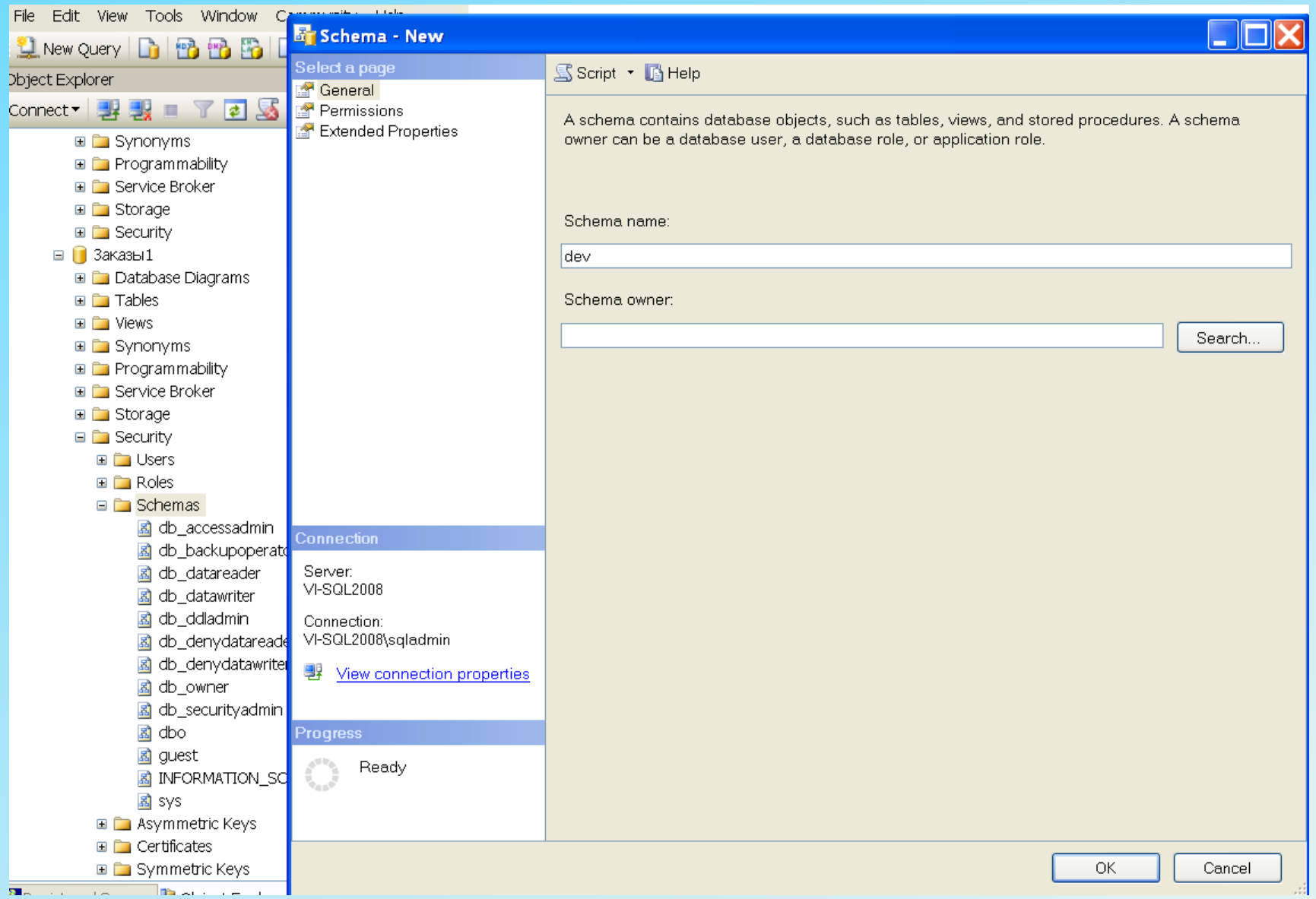
- **нескольким пользователям можно назначить одну и ту же схему по умолчанию, что может быть удобно при разработке приложений;**
- **несколько пользователей (через группы Windows или роли баз данных) могут владеть одной и той же схемой. При этом один пользователь может являться владельцем сразу нескольких 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:

- **использованием графического интерфейса SQL Server Management Studio**
- **командой Transact SQL**

Создание схемы в MS SSMS



Диалоговое окно SSMS для создания схемы БД

Создание схемы через Transact-SQL

имя схемы в пределах базы данных

```
CREATE SCHEMA schema_name AUTHORIZATION owner_name
```

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

Например,

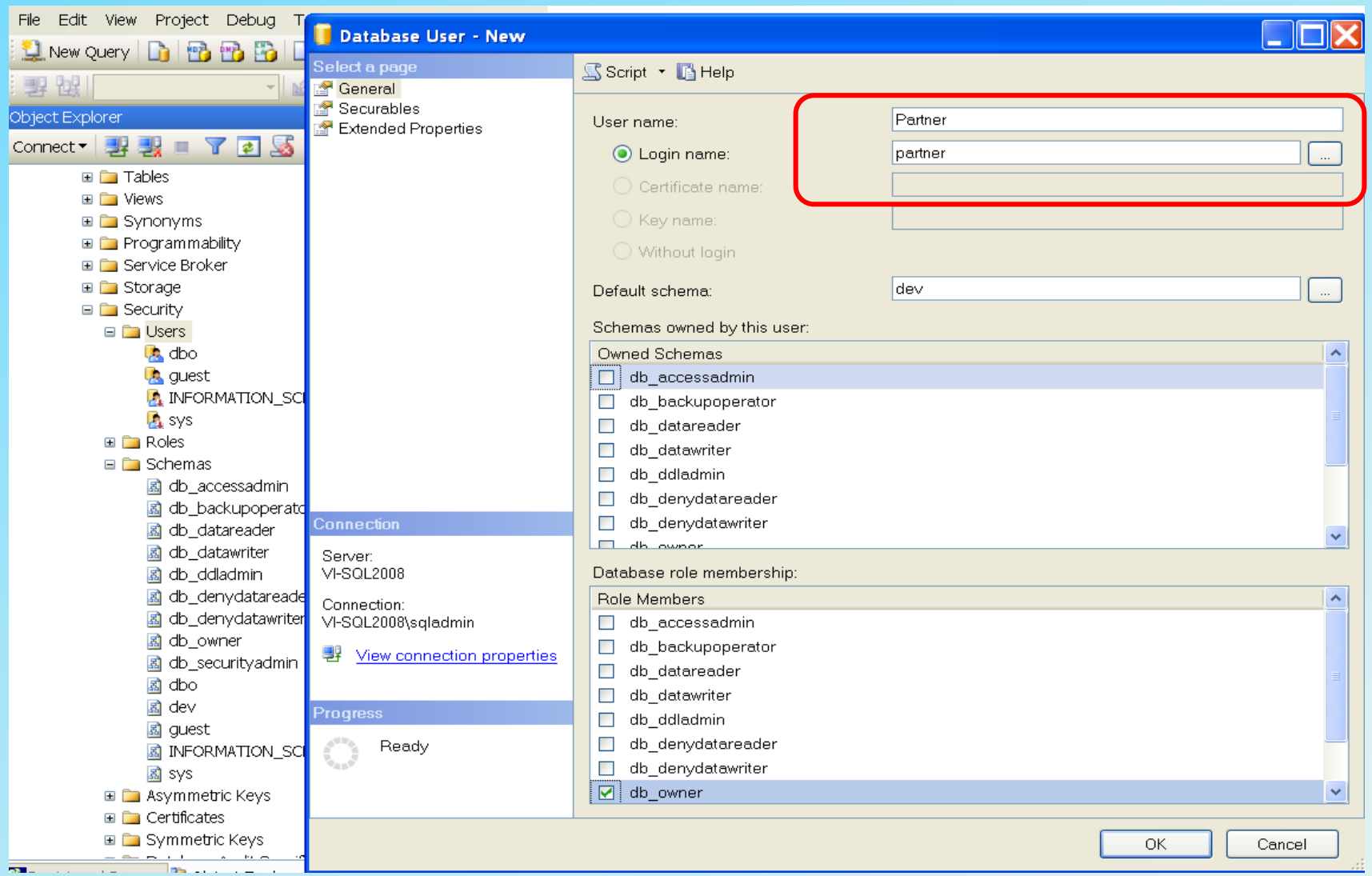
```
CREATE SCHEMA dev AUTHORIZATION dbo
```

Создание пользователей БД

Создание пользователя БД выполняется:

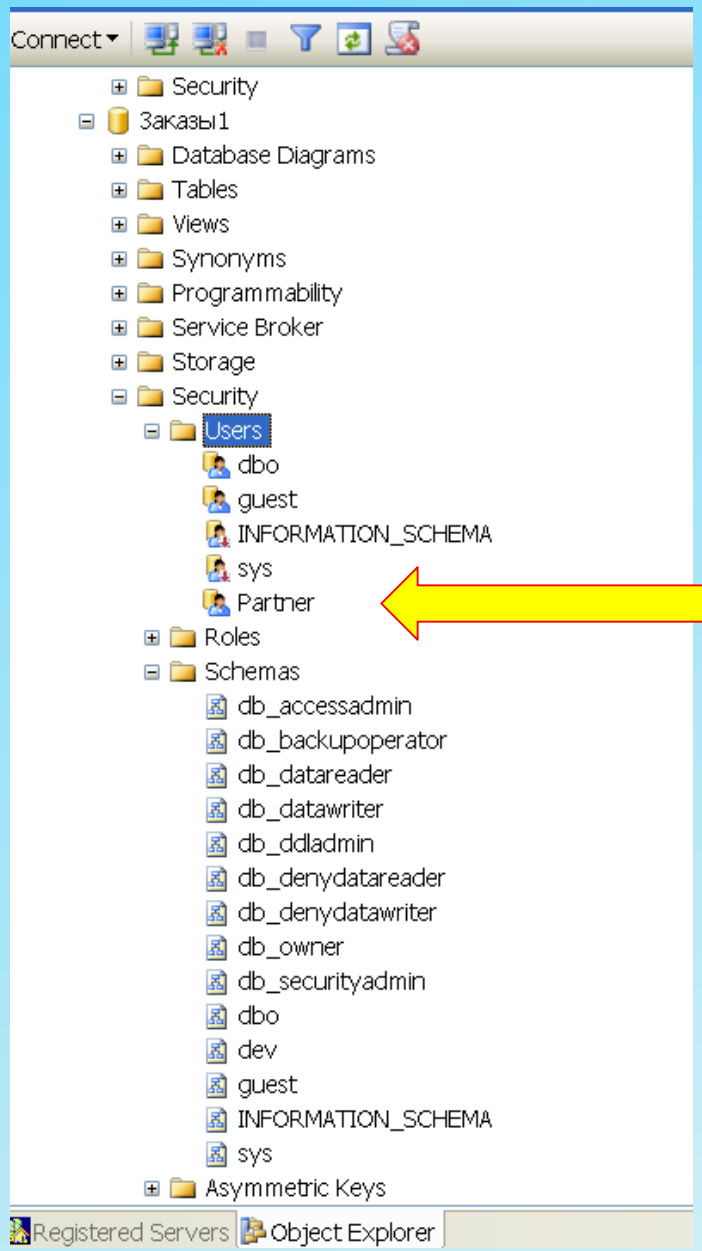
- использованием GUI SSMS
- командой T-SQL
- использованием ХП СУБД

Создание пользователей БД через интерфейс MS SSMS



Диалоговое окно SSMS для создания пользователя БД

Создание пользователей БД



Результат создания пользователя БД Заказы1

Создание пользователя через T-SQL

Имя пользователя базы данных

```
CREATE USER user_name FOR LOGIN login_name

[ WITH DEFAULT_SCHEMA = schema_name ]
```

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

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

Раздел 6. Администрирование серверов баз данных

Тема 6.1

Обеспечение безопасности данных *(на примере MS SQL Server)*

Вопросы лекции:

1. Принципы системы безопасности сервера БД.
2. Управление учетными записями сервера БД.
3. Управление пользователями БД.
4. **Управление правами и ролями.**

Роли и права

Роль – это **именованный набор** (комбинация) различных **прав**

Права – это конкретные разрешения на доступ и определенные 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 роли 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

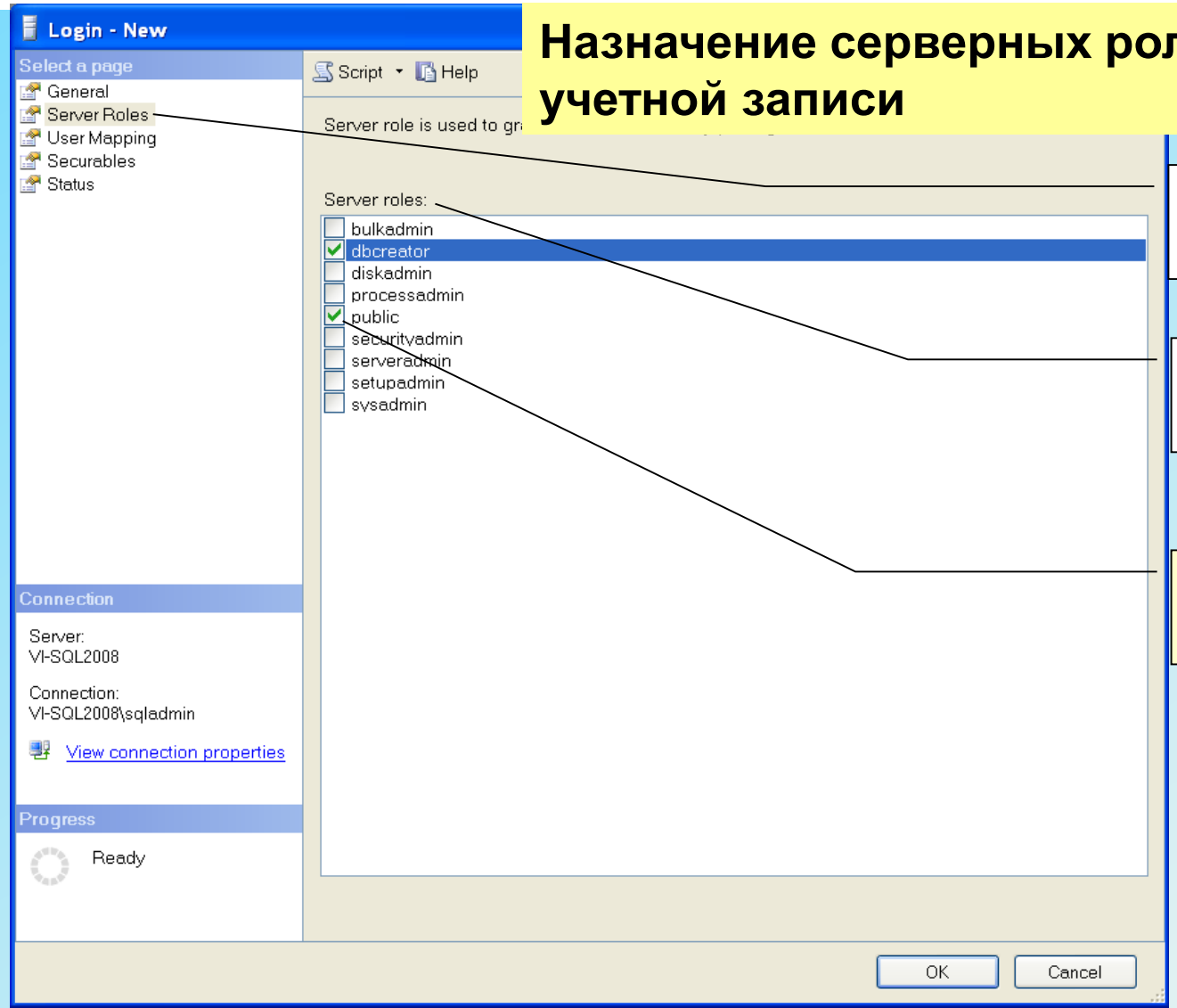
Стандартные роли сервера (fixed role server) определяют **права учетной записи** по администрированию сервера.

securityadmin	Можно управлять учетными записями и правами на создание БД, читать журнал ошибок
dbcreator	Можно создавать и модифицировать БД
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



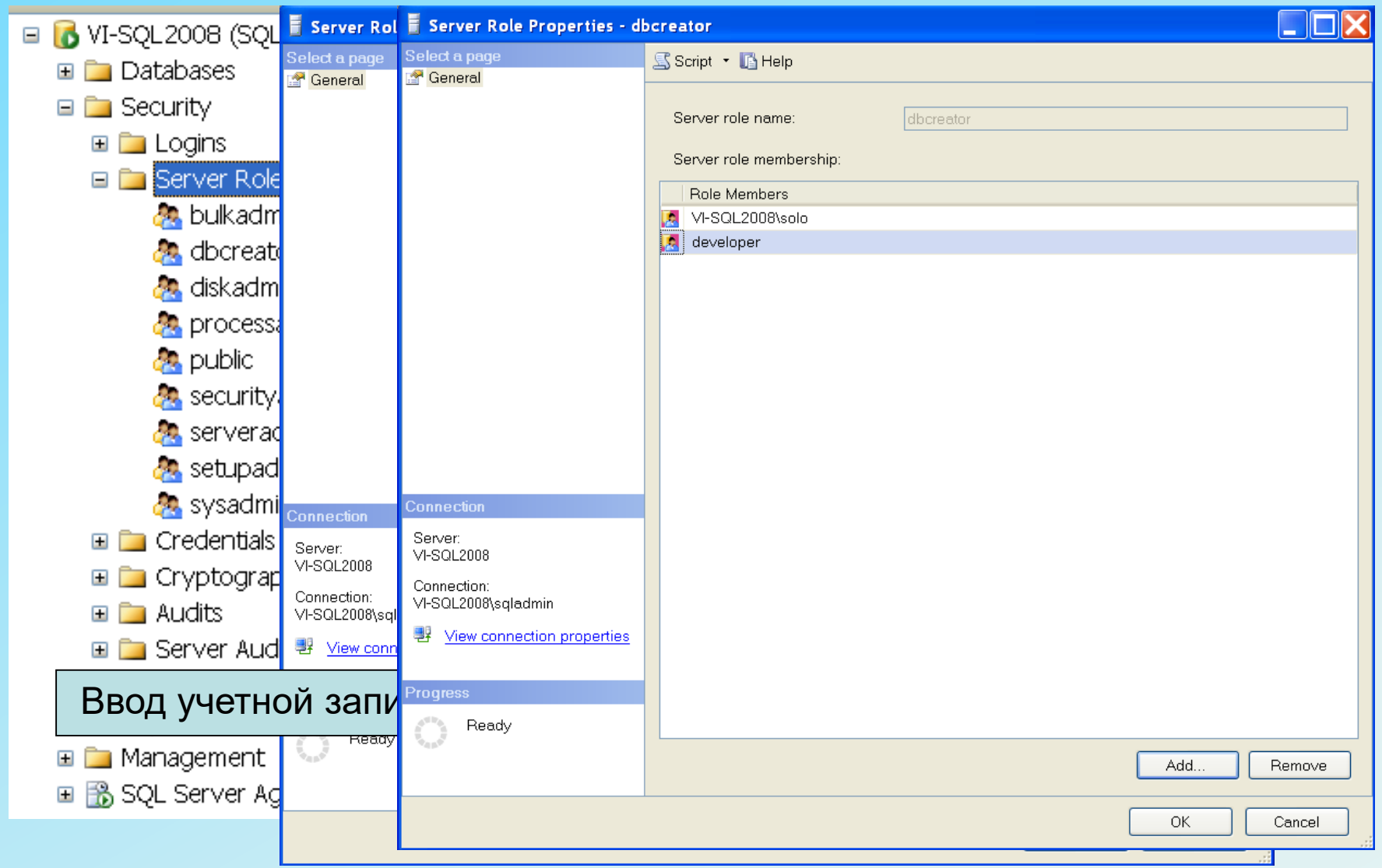
Закладка для назначения серверной роли

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*'
, [**@rolename =**] '*role*'

имя уч.записи сервера

имя серверной роли

Например,

```
sp_addsrvrolemember 'developer', 'dbcreator'
```

Роли уровня БД

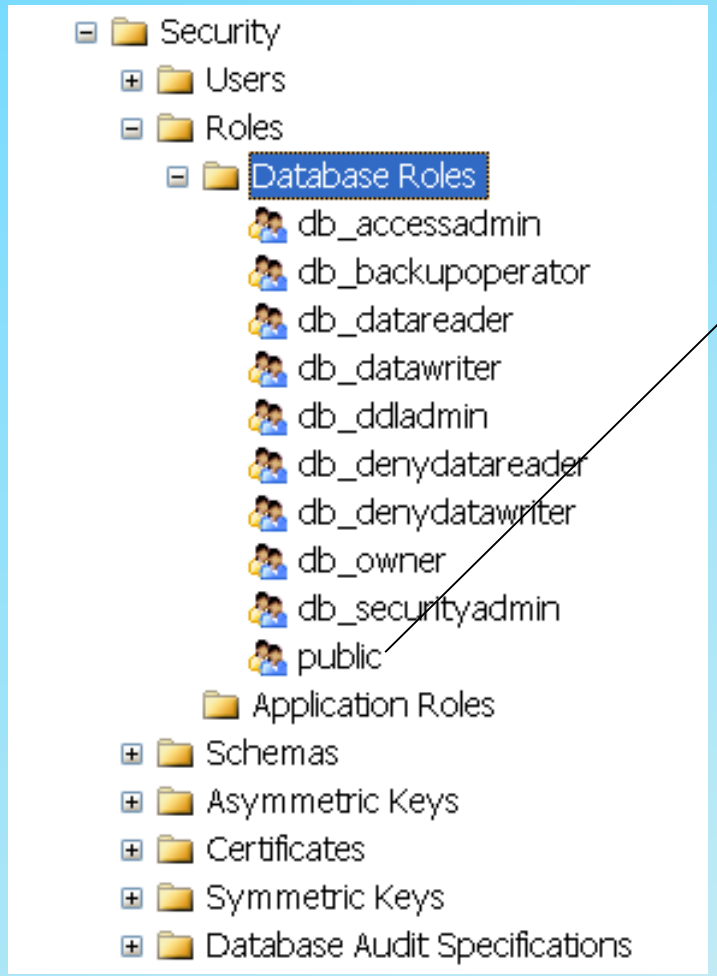
- фиксированные
- пользовательские
  - стандартные
  - приложения

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



Специальная роль

Все пользователи базы данных получают права этой роли автоматически.

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

Роли БД в проводнике объектов в MS SSMS

Пользовательские роли БД

- **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.

Права пользователей

Явные

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

Неявные

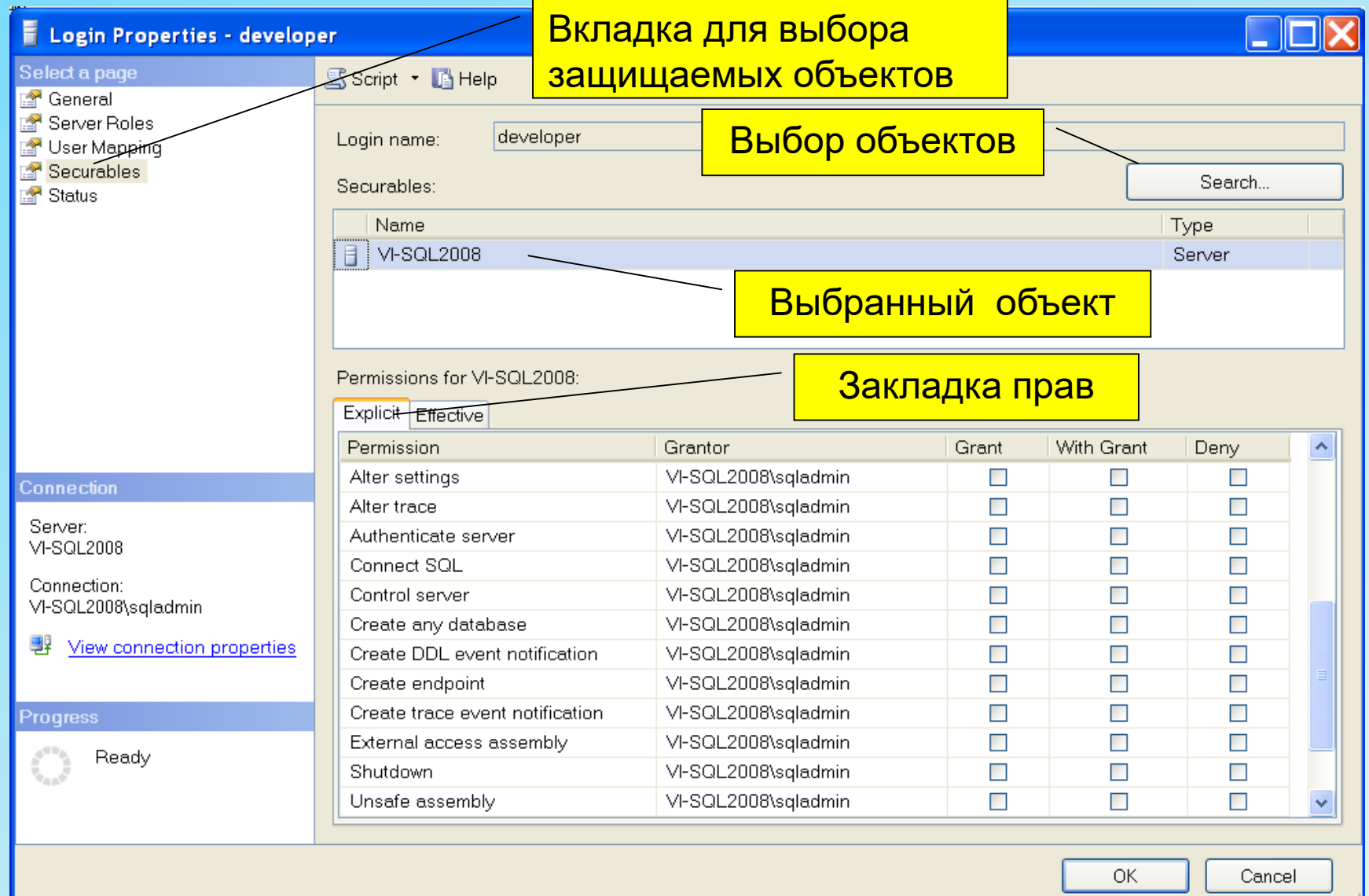
Это права полученные при определенных обстоятельствах.

Права выдаются (назначаются):

- администратором сервера
- владельцем БД
- владельцем объекта

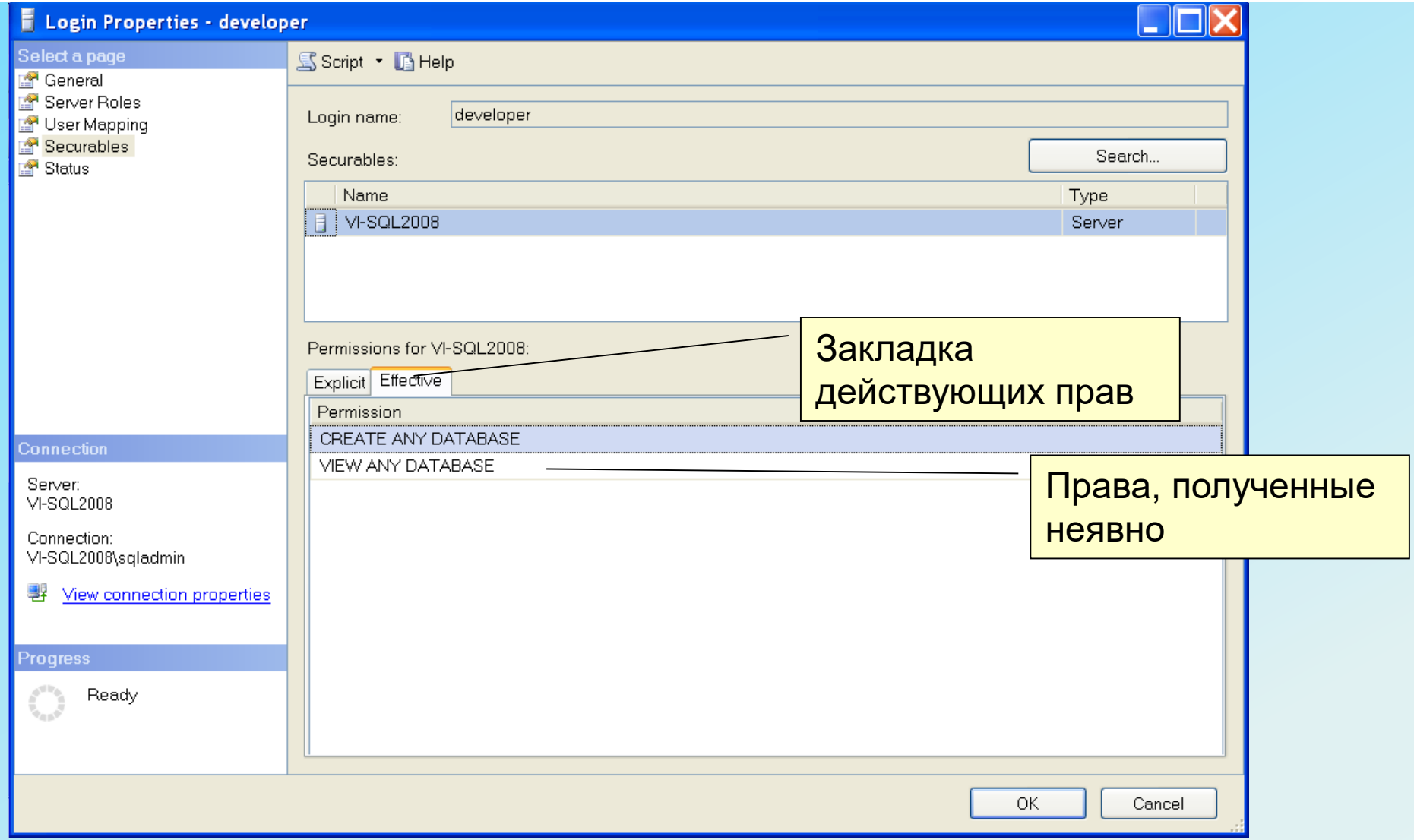
Назначение прав пользователей

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



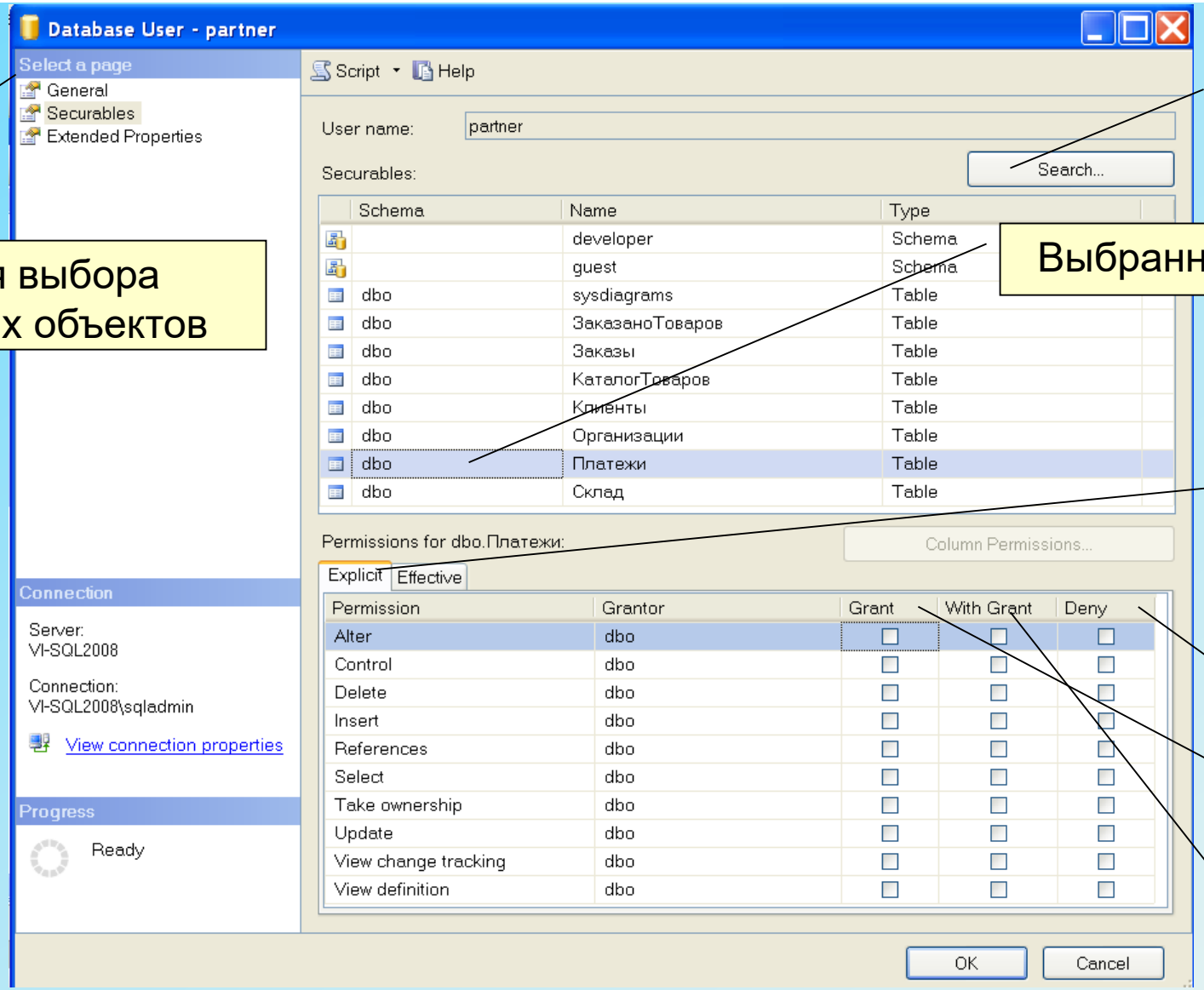
Назначение прав пользователей

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



Назначение прав пользователей

Вкладка для задания прав пользователя БД partner в SSMS



Вкладка для выбора защищаемых объектов

Выбор объектов

Выбранный объект

Закладка прав

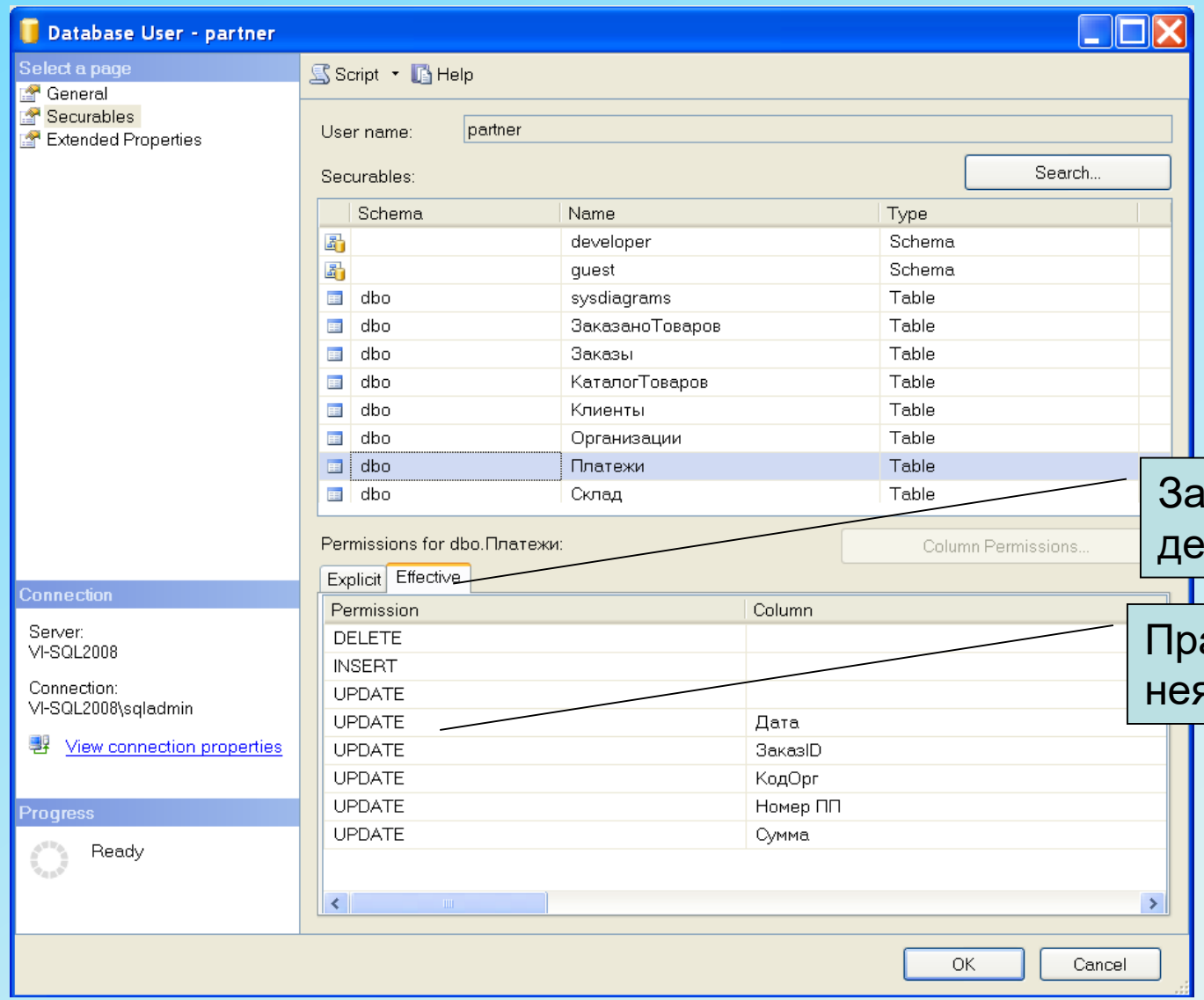
Запрещение

Разрешение

Разрешение на передачу прав

Назначение прав пользователей

Вкладка для просмотра действующих прав пользователя БД partner в SSMS

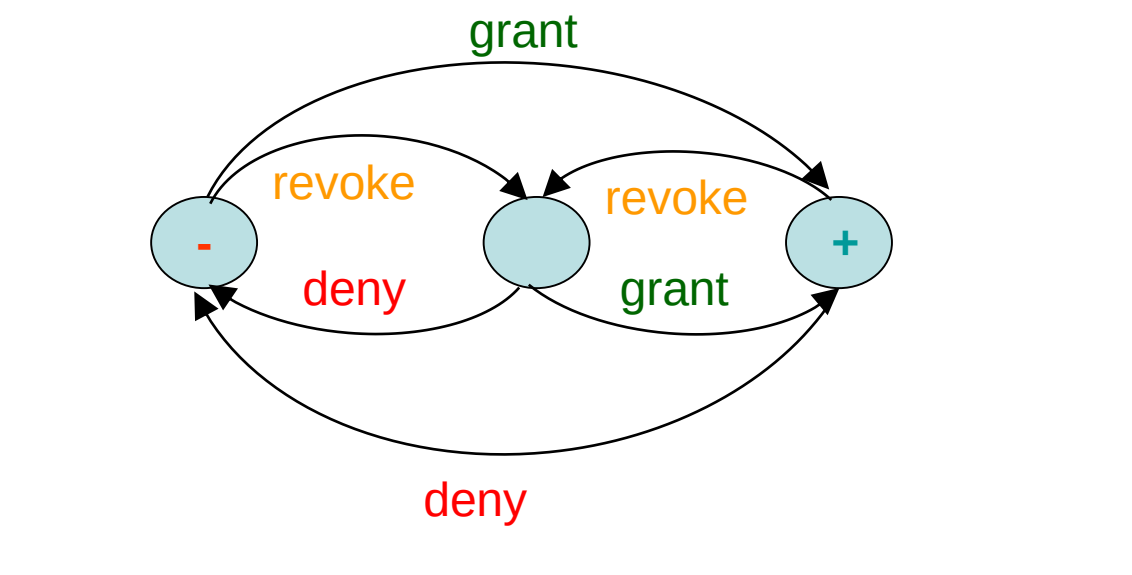


Закладка действующих прав

Права, полученные неявно

Команды SQL управления доступом к объектам БД

GRANT	Предоставляет право доступа к объектам БД
DENY	Запрещает право доступа к объектам БД
REVOKE	Отклоняет право доступа к объектам БД



Команда Grant

Командные разрешения:

```
GRANT { ALL | statement [ ,...n ] }
TO security_account [ ,...n ]
```

Объектные разрешения:

```
GRANT
   { ALL [ PRIVILEGES ] | permission [ ,...n ] }
   {
      [ ( column [ ,...n ] ) ] ON { table | view }
      | ON { table | view } [ ( column [ ,...n ] ) ]
      | ON { stored_procedure | extended_procedure }
      | ON { user_defined_function }
   }
TO security_account [ ,...n ]
[ WITH GRANT OPTION ]
[ AS { group | role } ]
```

Statement *- это*
CREATE DATABASE
CREATE DEFAULT
CREATE FUNCTION
CREATE PROCEDURE
CREATE RULE
CREATE TABLE
CREATE VIEW
BACKUP DATABASE
BACKUP LOG

permission *- это*
SELECT, INSERT, DELETE, or UPDATE, а также REFERENCES и EXECUTE

security_account – это учетные записи сервера, пользователи и группы Windows NT, которым предоставлен доступ к серверу

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

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
```

Команда DENY

Командные запрещения:

DENY { ALL | *statement* [**,**...*n*] }
TO *security_account* [**,**...*n*]

Объектные запрещения :

DENY
{ ALL [PRIVILEGES] | *permission* [**,**...*n*] }
{
[**(** *column* [**,**...*n*] **)**] **ON** { *table* | *view* }
| **ON** { *table* | *view* } [**(** *column* [**,**...*n*] **)**]
| **ON** { *stored_procedure* | *extended_procedure*
| **ON** { *user_defined_function* }
}
TO *security_account* [**,**...*n*]
[**CASCADE**]

Statement *- это*
CREATE DATABASE
CREATE DEFAULT
CREATE FUNCTION
CREATE PROCEDURE
CREATE RULE
CREATE TABLE
CREATE VIEW
BACKUP DATABASE
BACKUP LOG

permission *- это*
SELECT, INSERT, DELETE, or UPDATE, а также REFERENCES и **EXECUTE**

security_account – это учетные записи сервера, пользователи и группы Windows NT, которым предоставлен доступ к серверу

CASCADE– отзывает права у других пользователей, кому данный пользователь их предоставил

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
```