Сравнительный анализ моделей безопасности в SQL Server 2000 и SQL Server 2005

Андрей Синкин
Системный инженер
Microsoft

Мартин Рахманов
Старший инженер-программист
Рексофт

Компоненты SQL Server

- Реляционный сервер
  - Внутризапросный параллелизм
  - Распределенные фрагментированные представления
  - Службы тиражирования
  - Средства создания резервных копий БД
  - Механизмы отказоустойчивости (Log Shipping, MSCS)
  - Графические средства администрирования и отладки
  - Утилиты настройки и оптимизации
- Службы репликации
- Службы формирования отчетов (Reporting Services)
- Службы оповещения (Notification Services)
- Службы анализа данных (OLAP, DataMining)
- Инструменты управления (Management tools)
- Программные интерфейсы доступа и разработки
  - ODBC, OLE DB, ADO, OLE DB for OLAP, ADO MD, ADOX, интерфейсы дистрибутора и согласования, SQL DMO, DSO, …

- По своей природе веб-приложения электронной коммерции чувствительны к защите информации. Это послужило причиной внесения в сервер SQL Server 2000 новых значительных улучшений системы безопасности, не только обеспечивающих наиболее высокий в отрасли уровень безопасности, но также упрощающих применение средств, необходимых для достижения этого уровня. Прежде всего, SQL Server 2000 устанавливается по умолчанию с более высоким уровнем безопасности, при этом используются средства, встроенные в новейшую интегрированную систему безопасности операционной системы Windows 2000. Это упрощает и ускоряет изоляцию сервера в производственной среде.
- SQL Server 2000 также включает в себя набор новых средств обеспечения безопасности: мощную и гибкую систему безопасности сервера на **ролевой** основе, **профили** БД и приложений, интегрированные средства **аудита** безопасности (отслеживающие 18 различных видов событий и дополнительные события), поддержку **шифрования** файлов и сетевых сообщений (включая SSL), а также поддержку **протокола Kerberos** и возможность **делегирования** полномочий. SQL Server 2000 прошел проверку в рамках программы правительства США Trusted Product Evaluation Program и Агентство национальной безопасности подтвердило его соответствие уровню безопасности C2

Редакции SQL Server

- SQL Server 2005 Enterprise Edition
- SQL Server 2005 Standart Edition
- SQL Server 2005 Workgroup Edition
- SQL Server 2005 Express Edition (Free)
- SQL Server 2005 Developer Edition

Безопасность

- SQL Profiler - мониторинг событий класса безопасности
  - Add/drop SQL login, Add/remove database user, Add/remove database role member, Password change, GRD - statement perms, GRD – object perms, ...
  - Для каждого записывается время, пользователь, хост, успех/неудача и т.д.
- Шифрация трафика для всех сетевых библиотек при помощи SSL / TLS
- Поддержка делегирования на основе Kerberos, интеграция с Active Directory

Кластеризация

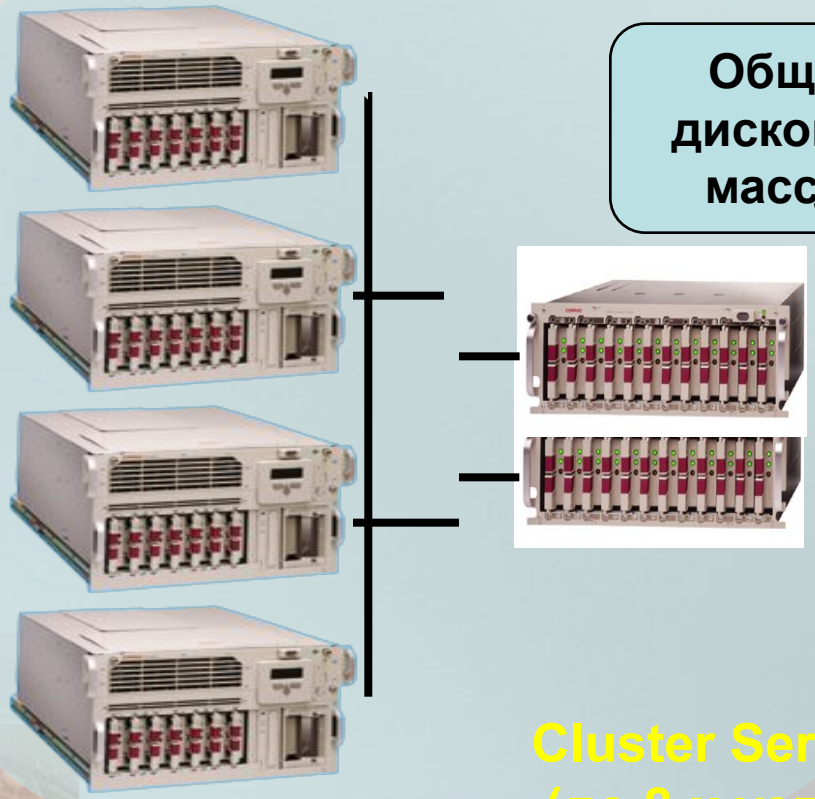
Fiber Channel

SQL Server 2005 EE

SQL Server 2005 EE

SQL Server 2005 EE

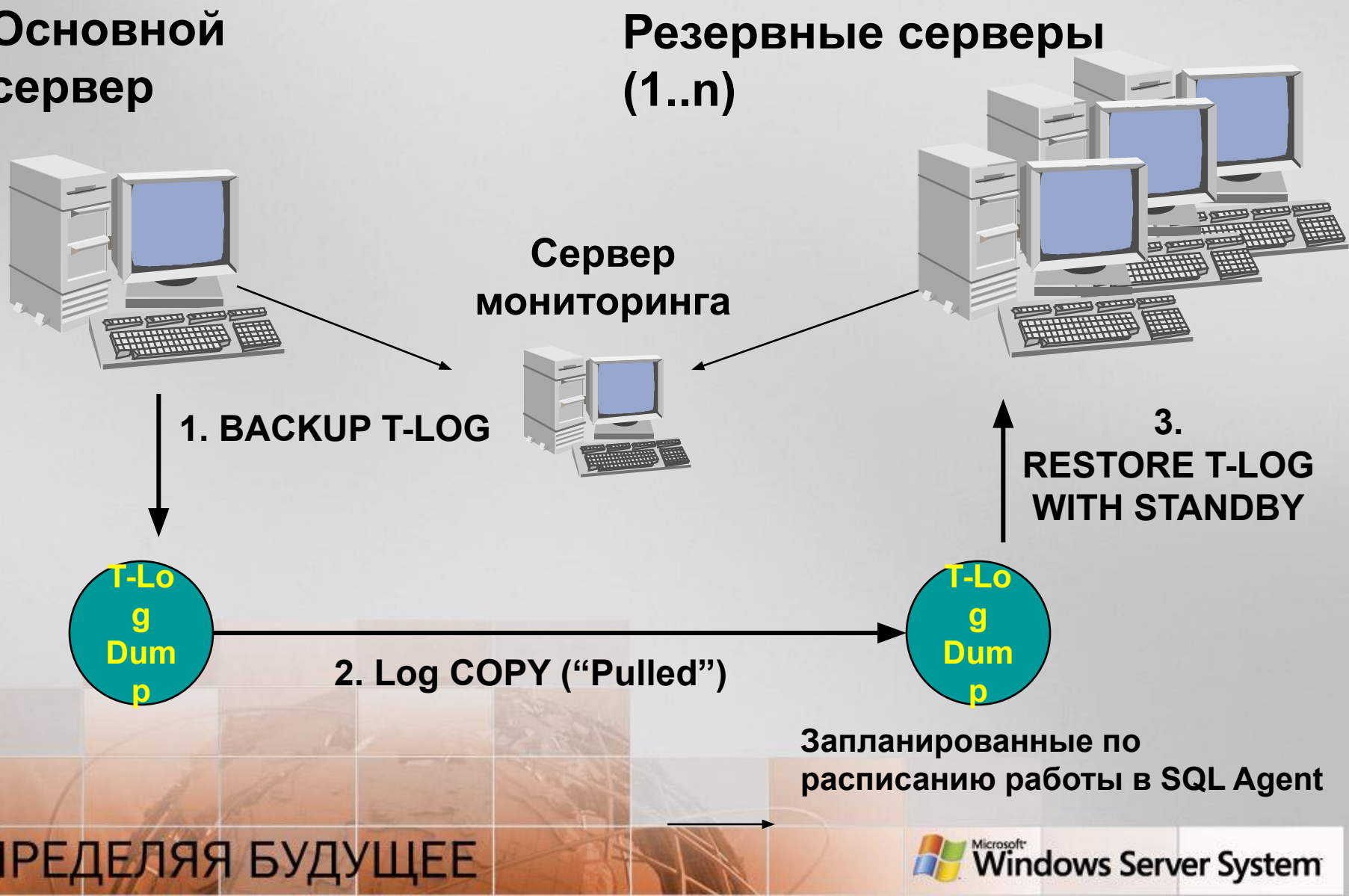
SQL Server 2005 EE



Общий дисковый массив

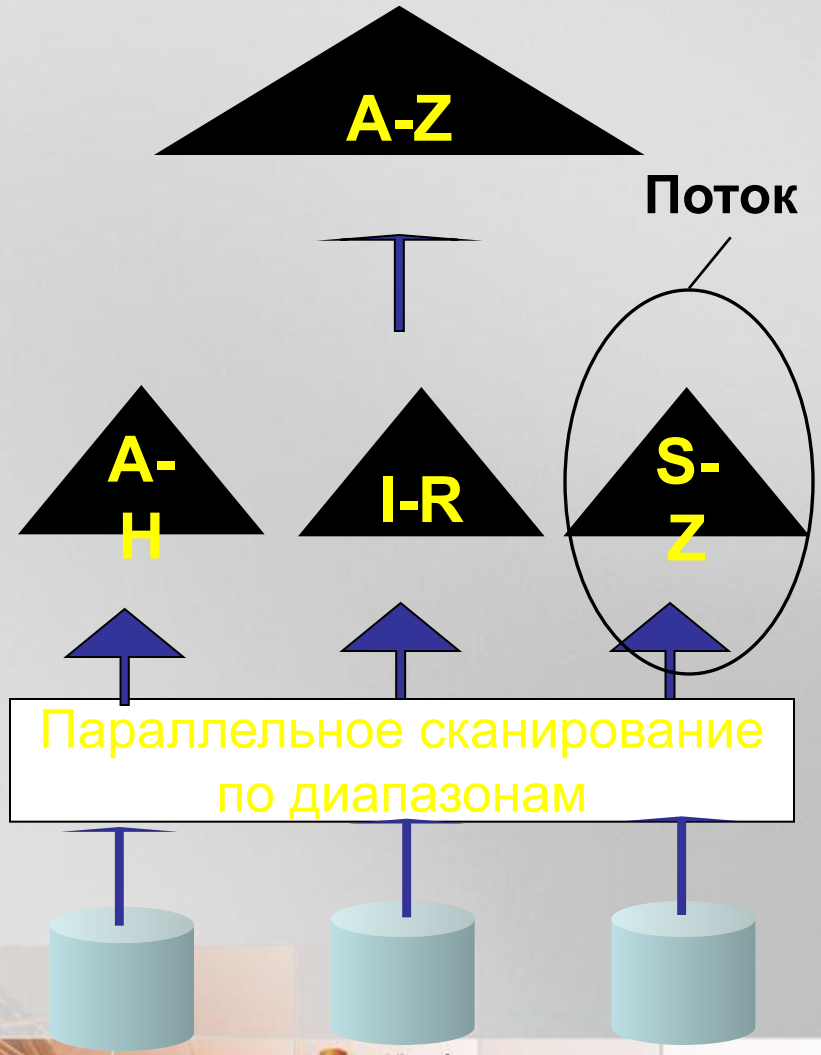
Cluster Service (до 8-х узлов)

Передача журналов (Log Shipping)



Расширенный список параллельных операций

- Операция создания индекса выполняется одновременно на нескольких потоках
  - Линейная масштабируемость в зависимости от числа процессоров
- Каждый поток получает свой диапазон значений
  - На основе известной статистики распределения индексных ключей поддерживается баланс нагрузки между потоками
  - На заключительном этапе поддеревья объединяются в единый индекс



Индексированные представления

- Обычное представление – всего лишь удобная форма записи сложного оператора SELECT
  - Чтобы обращаться к нему как к якобы таблице
  - Хранится только определение SQL-запроса, который выполняется всякий раз при обращении к представлению
- Как только над представлением создается индекс, его результаты «материализуются»
  - И обновляются при модификации данных в исходных таблицах
  - Т.е. ведут себя как все приличные индексы
- Представление может содержать агрегаты, операторы связывания таблиц или их комбинацию
- Первый индекс над представлением должен быть
  - Кластерным -> для сохранения представления как таблицы
  - Уникальным -> для поддержки индекса актуальным при внесении изменений в таблицы
- Кто выигрывает от индексированных представлений
  - Приложения, изобилующие запросами с многочисленными операторами связывания, группировки, агрегации
    - Т.е. OLAP-приложения
  - Оптимизатор может использовать индекс над представлением, даже если оно явно не фигурирует в запросе
- Кто не выигрывает
  - Приложения, для которых характерны постоянные обновления в БД
    - Т.е. OLTP-приложения
    - Т.к. частая коррекция представлений снижает производительность
  - Группировки по высокоселективным полям
    - Т.к. размер представления будет ненамного меньше самой таблицы

Поддержка XML

- SQL -> XML
  - SELECT … FOR XML
- XML -> SQL
  - OpenXML в T-SQL
  - UpdateGrams
    - XML-описание операций INSERT, UPDATE, DELETE
    - Bulk Load XML-файлов в БД
- XML View Mapper
  - Отображение XML-документа на таблицу

Содержание

- Обзор модели безопасности SQL Server 2000
  - Ограничения модели безопасности и способы их устранения
  - Рекомендации по настройке
- Что нового в SQL Server 2005?
  - Новая модель разрешений
    - Безопасность метаданных
    - Более гранулярные разрешения
    - Покрывающие разрешения
  - Разделение схемы и владельца
  - Контекст выполнения хранимого кода

Составляющие модели безопасности SQL Server 2000

- Аутентификация
  - Учётные записи Windows и SQL Server
- Авторизация
  - Проверки разрешений на доступ к объектам и на выполнение операций
- Аудит
  - Трассировка, журналы сервера и системы

Модель безопасности SQL Server 2000

Microsoft Платформа 2005

Запрос на сетевое подключение

Подключение к компьютеру с SQL Server

Запрос на подключение к SQL Server

Определение прав на подключение

Переход к БД и проверка прав

Создание контекста работы с БД

Попытка выполнить действия

Проверка прав на выполнение действий

Режимы доступа к SQL Server

- Windows® Authentication
  - Доступ разрешен только с использованием бюджета Windows NT®/ Windows 2000/2003
  - Обеспечивает единую регистрацию
- Mixed security
  - Принимает доступ под бюджетом Windows
  - Принимает доступ через авторизацию на SQL Server
  - Сложнее поддается защите

Учётная запись и Пользователь

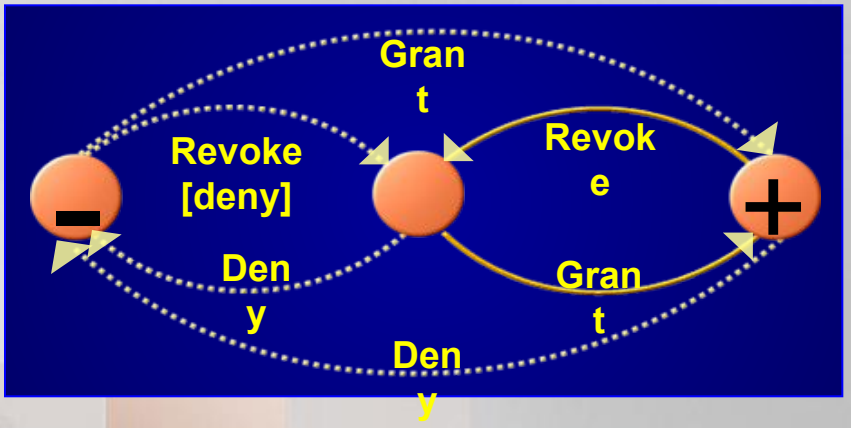
- Учётная запись (login) дает право на подключение
  - Хранится в БД **master**
  - Относится к серверу в целом
  - Сама по себе не дает прав
    - Исключение: Членство в фиксированной серверной роли
- Пользователь (user) БД ассоциируется с правами
  - С ним ассоциируется схема (коллекция объектов БД)
  - Права назначаются пользователям БД
  - Действует в рамках конкретной БД

Роли SQL Server

- Фиксированные роли сервера
  - Гибкое администрирование сервера
- Фиксированные роли БД
  - Гибкое администрирование БД
- Пользовательские роли БД
  - Пользовательские комбинации прав
- Прикладные роли
  - Ассоциация прав с приложением, а не с пользователем

Разрешения

- SQL Server поддерживает три команды для работы с разрешениями (Data Control Language - DCL):
  - GRANT назначает разрешение
  - DENY запрещает разрешение
  - REVOKE отзывает сделанное ранее при помощи GRANT или DENY действие



Содержание

- Обзор модели безопасности SQL Server 2000
  - Ограничения модели безопасности и способы их устранения
  - Рекомендации по настройке
- Что нового в SQL Server 2005?
  - Новая модель разрешений
    - Безопасность метаданных
    - Более гранулярные разрешения
    - Покрывающие разрешения
  - Разделение схемы и владельца
  - Контекст выполнения хранимого кода

Ограничения модели безопасности SQL Server 2000 и способы их устранения

1. Неограничен доступ к метаданным для любого пользователя БД. Решение: разрабатывать приложение так, чтобы доступ к хранимому коду на стороне SQL Server не позволил злоумышленнику нанести существенный ущерб
2. Невозможно назначить пользователю БД разрешения на выполнение определённых команд без повышения уровня его привилегий. Решение: в SQL2k решение отсутствует
3. По умолчанию непривилегированный пользователь может выполнить ряд расширенных хранимых процедур, предоставляющих доступ к важной информации. Решение: отозвать разрешения на выполнение ряда хранимых процедур для встроенной группы **public**
4. Отсутствуют встроенные средства шифрования данных. Решение: использовать средства сторонних производителей
5. Нет возможности ограничить количество попыток соединения с сервером при указании неверной пары учётная запись/пароль для учётных записей SQL Server. Решение: использовать для соединения с сервером только учётные записи Windows, либо использовать средства сторонних производителей

Есть db_datareader и db_datawriter, а db_executor нет..

SQL Query Analyzer - [Query - SINIKIN01.pubs.EUROPE\asinkin - Untitled1*]

File Edit Query Tools Window Help

pubs

```
-- Формирование набора команд для разрешения выполнения
-- хранимых процедур пользователем [EUROPE\asinkin]
SELECT 'gra
QUOTENAME(P
WHERE OBJEC
```

```
grant exec on
grant exec on
grant exec on
grant exec on

(4 row(s) aff
```

SQL Query Analyzer - [Query - SINIKIN01.pubs.EUROPE\asinkin - Untitled2*]

File Edit Query Tools Window Help

pubs

```
use master
go
create procedure sp_grantexec(@user sysname,@pattern sysname = NULL,@debug int = 0)
as
set nocount on
declare @ret int
declare @sql nvarchar(4000)
declare @db  sysname ; set @db = DB_NAME()
declare @u   sysname ; set @u = QUOTENAME(@user)

set @sql ='select ''grant exec on '' + QUOTENAME(ROUTINE_SCHEMA) + ''.'' +
          QUOTENAME(ROUTINE_NAME) + '' TO ' + @u + ''' FROM INFORMATION_SCHEMA.ROUTINES ' +
          'WHERE OBJECTPROPERTY(OBJECT_ID(ROUTINE_NAME),''IsMSShipped'') = 0'

if @pattern is not null
set @sql = @sql + N' AND ROUTINE_NAME LIKE ''' + @pattern + ''''

if @debug = 1 print @sql
else
exec @ret = master.dbo.xp_execresultset @sql,@db

If @ret <> 0
begin
   raiserror('Error executing command %s',16,1,@sql)
   return -1
end
```

```
/* CREATE A NEW ROLE */
CREATE ROLE db_executor

/* GRANT EXECUTE TO THE ROLE */
GRANT EXECUTE TO db_executor
```

ОПРЕДЕ

Содержание

- Обзор модели безопасности SQL Server 2000
  - Ограничения модели безопасности и способы их устранения
  - Рекомендации по настройке
- Что нового в SQL Server 2005?
  - Новая модель разрешений
    - Безопасность метаданных
    - Более гранулярные разрешения
    - Покрывающие разрешения
  - Разделение схемы и владельца
  - Контекст выполнения хранимого кода

Рекомендации по настройке

- Установить самые свежие пакеты обновлений и исправлений
- Отключить ненужные сетевые протоколы
- Включить протоколирование (аудит) неудачных попыток подключения
- Разместить файлы данных на файловой системе NTFS и настроить доступ к ним
- Использовать только аутентификацию Windows
- Указать сложные пароли для учётных записей SQL Server
- Использовать непривилегированные (не административные, как минимум) учётные записи для запуска службы mssqlserver и службы SQL Server Agent.
- Подробнее в разделе “SQLSecurity Checklist” на сайте www.sqlsecurity.com

Рекомендации по настройке

- Microsoft Baseline Security Analyzer
  - Слишком много пользователей входят в роль *sysadmin*
  - Права на выполнение *CmdExec* дано не только роли *sysadmin*
  - Пустой или простой пароль
  - Администраторы Windows входят в роль *sysadmin*?
  - Некорректные разрешения (ACLs) на папки данных SQL Server
  - В логах установки остается пароль *sa* (sqlstp.log и др.)
  - Пользователь *guest* по умолчанию наделен излишними правами
  - SQL Server запущен на контроллере домена
  - Доступ на чтение определенных ключей реестра для группы *Everyone*
  - Наделение учетной записи для запуска SQL Server излишними полномочиями
  - Отсутствие пакетов исправлений и обновлений

http://www.microsoft.com/technet/security/tools/mbsahome.mspx

Service Pack Installation May Save Standard Security Password in File

Содержание

- Обзор модели безопасности SQL Server 2000
  - Ограничения модели безопасности и способы их устранения
  - Рекомендации по настройке
- Что нового в SQL Server 2005?
  - Новая модель разрешений
    - Безопасность метаданных
    - Более гранулярные разрешения
    - Покрывающие разрешения
  - Разделение схемы и владельца
  - Контекст выполнения хранимого кода

SQL Server 2005

Microsoft Платформа 2005

.NET Framework
- Common Language Runtime Integration
- User-defined Aggregates
- User-defined Data Types
- User-defined Functions
- SQL Server In-Proc Data Provider
- Extended Triggers

Data Types
- File Stream Storage Attribute
- Managed SQL Types
- New XML Datatype

SQL Server Engine
- New Message Service Broker
- HTTP Support (Native HTTP)
- Database Tuning Advisor
- Enhanced Read ahead & scan
- Extended Indexes
- Multiple Active Result Sets
- Persisted Computed Columns
- Queuing Support
- Snapshot Isolation Level
- Scale Up Partitioning
- VIA support
- NUMA support

Database Failure and Redundancy
- Fail-over Clustering (up to 8 node)
- Enhanced Multi-instance Support
- Database Mirroring
- Database Viewpoints

XML
- XQUERY Support (Server & Mid Tier)
- XML Data Manipulation Language
- FOR XML Enhancements
- XML Schema (XSD) Support
- MSXML 6.0 (Native)
- XQuery Designer

Database Maintenance
- Backup and Restore Enhancements
- Checksum Integrity Checks
- Dedicated Administrator Connection
- Dynamic AWE
- Fast Recovery
- Highly-available Upgrade
- Online Index Operations
- Online Restore
- Parallel DBCC
- Parallel Index Operations

Management Tools
- MDX Query Editor
- MDX Intellisense
- T-SQL Intellisense
- Version Control Support
- XML/A
- SQLCMD Command Line Tool

Performance Tuning
- Profiler Enhancements
- Profiling Analysis Services
- Exportable Showplan
- Exportable Deadlock Traces

Full-text Search
- Indexing of XML Datatype

MDAC
- Side by Side installation
- Microsoft Installer base setup
- Support for Active Directory Deployment

SQL Client .NET Data Provider
- Server Cursor Support
- Asynch

Security
- All Permissions Grantable
- Fine Grain Administration Rights
- Separation of Users and Schema

Replication
- Auto-tuning Replication Agents
- Oracle Publication
- Improved Blob Change Tracking

OLAP and Data Mining
- Analysis Management Objects
- Windows Integrated Backup and Restore
- Web Services/XML for Analysis
- DTS and DM Integration
- Eight new DM algorithms
- Auto Packaging and Deployment

Data Transformation Services
- New Architecture (DTR + DTP)
- Complex Control Flows
- Control Flow Debugging
- For Each Enumerations
- Property Mappings
- Full Data Flow Designer
- Full DTS Control Flow Designer
- Graphical Presentation of Pkg Execution
- Immediate Mode and Project Mode
- Package (Advanced) Deployment Tools
- Custom Tasks and Transformations

Reporting Services
- Multiple Output Formats
- Parameters (Static, Dynamic, Hierarchical)
- Bulk Delivery of Personalized Content
- Support Multiple Data Sources
- Sharepoint Support
- Visual Design Tool
- Charting, Sorting, Filtering, Drill-Through
- Scheduling, Caching
- Complete Scripting Engine
- Scale Out architecture
- XML Report Definition

ОПРЕДЕЛЯЯ БУДУЩЕЕ

Microsoft Windows Server System

Что нового в модели безопасности SQL Server 2005

- Более жесткие настройки по умолчанию
- Сокрытие метаданных
- Парольная политика
- Новые разрешения и уровни разрешений
- Разделение схемы и владельца
- Контекст выполнения хранимого кода
- Безопасность .NET кода
- Встроенное шифрование данных
- Триггеры на DDL

Настройки по умолчанию

- Требуется явное включение дополнительной функциональности
  - Microsoft .NET Framework
  - SQL Service Broker Network Connectivity
  - Analysis Services http connectivity
- Следующие службы находятся в режиме запуска Manual
  - SQL Server Agent
  - full-text search
  - Новый сервис Data Transformation Services

- При установке требуется задание пароля учётной записи sa даже если сервер будет применять исключительно режим аутентификации Windows

Сокрытие метаданных

- Системные объекты теперь находятся в скрытой базе mssqlsystemresource
- Catalog Views – замена и расширение системных таблиц, данные из Catalog Views фильтруются в зависимости от того, кто делает запрос
- Разрешение `VIEW DEFINITION` позволяет обойти сокрытие метаданных и его можно выдать на трех уровнях: базы, схемы, объекта
- Шифрование хранимого кода стало надёжным

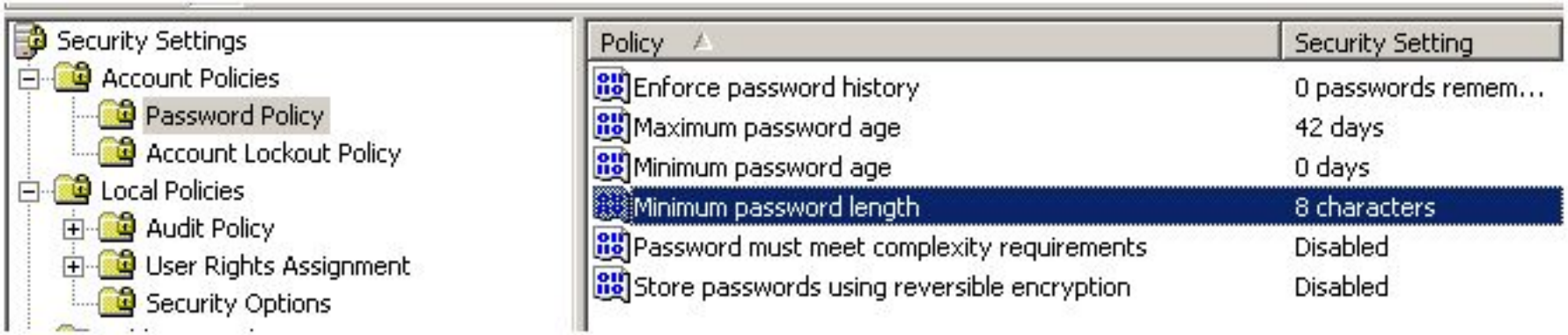
Парольная политика

Для учётной записи SQL Server можно указать следующие параметры команды CREATE/ALTER LOGIN:

- Необходимость сменить пароль при первом соединении с сервером (MUST_CHANGE)
- Необходимость проверки срока действия пароля (CHECK_EXPIRATION)
- Необходимость применения локальной парольной политики Windows (CHECK_POLICY)

- CHECK_EXPIRATION и CHECK_POLICY работают полноценно на Windows 2003 Server и более новых системах, а на Windows 2000 это сводится к проверке жестко зашитых правил

Парольная политика



SQLQuery1.sql-SS2K5B2.master* | Summary

```
if exists(SELECT * FROM sys.server_principals WHERE name = 'james')
  DROP LOGIN james
GO
CREATE LOGIN james
  WITH PASSWORD = 'hm2!D34' MUST_CHANGE, CHECK_EXPIRATION = ON, CHECK_POLICY = ON
GO
```

```
Msg 15116, Level 16, State 1, Line 1
Password validation failed. The password does not meet policy requirements because it is too short.
```

Новые разрешения

Разрешения можно выдавать на четырех уровнях:

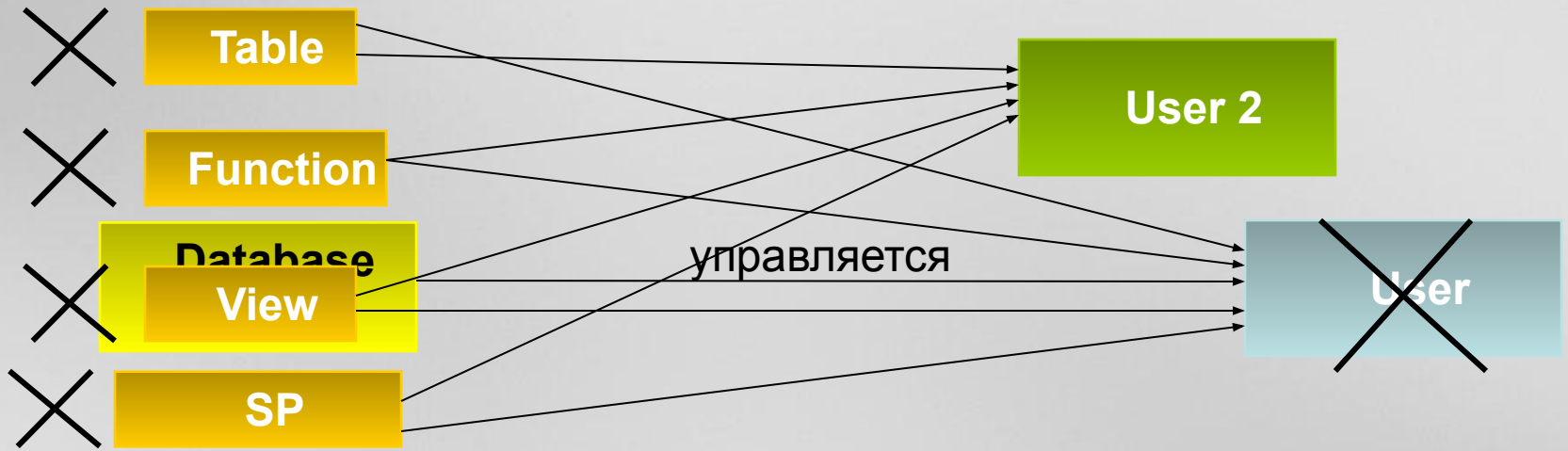
- Сервера
- Базы
- Схемы
- Объекта

Выданные разрешения уровня сервера и уровня базы данных можно получить через опрос представлений sys.server_permissions и sys.database_permissions

Содержание

- Обзор модели безопасности SQL Server 2000
  - Ограничения модели безопасности и способы их устранения
  - Рекомендации по настройке
- Что нового в SQL Server 2005?
  - Новая модель разрешений
    - Безопасность метаданных
    - Более гранулярные разрешения
    - Покрывающие разрешения
  - Разделение схемы и владельца
  - Контекст выполнения хранимого кода

Проблема неразделения схемы и пользователя



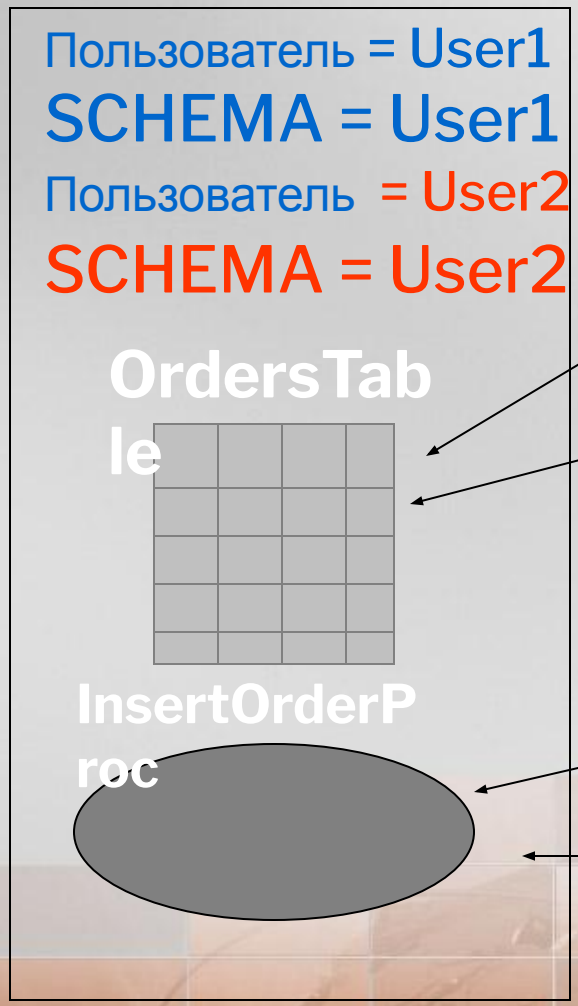
Разрешение имен

H-p: Select * from Foo

- **User.foo**
- **dbo.foo**

Удаление пользователей может привести к необходимости изменения кода приложения!

Удаление пользователя => изменение кода приложения



SELECT custID FROM User1.OrdersTable

App1

Изменение App1

SELECT custID FROM User2.OrdersTable

Exec User1.InsertOrderProc (@orderid)

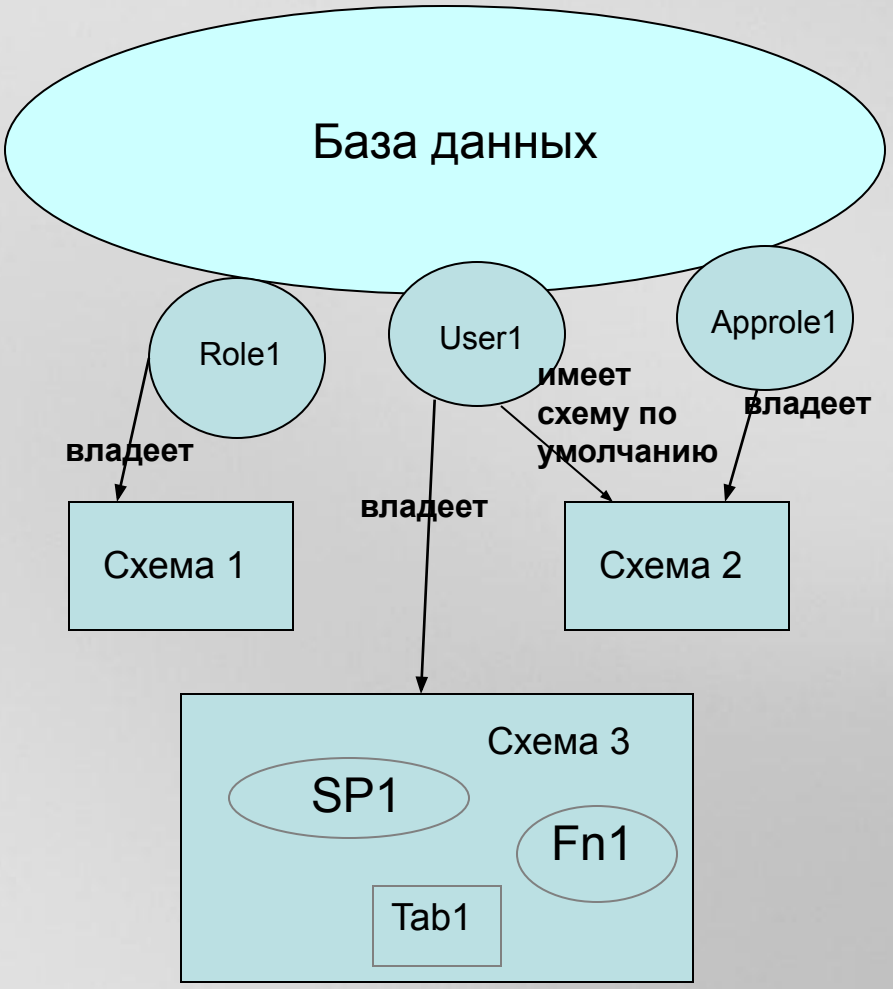
App2

Изменение App2

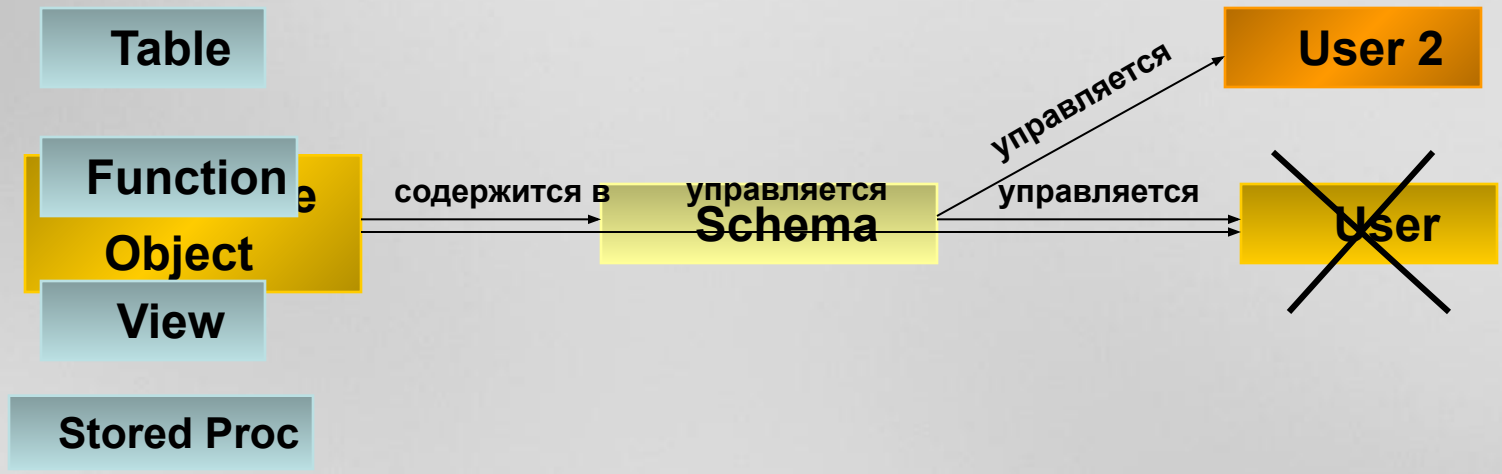
Exec User2.InsertOrderProc (@orderid)

Разделение схемы и владельца

- База данных может содержать множество схем
- Каждая схема имеет только одного владельца – пользователя или роль
- Каждый пользователь имеет схему по умолчанию
- Большинство объектов БД находятся в схемах
- Создание объекта внутри схемы требует полномочий CREATE и полномочий ALTER или CONTROL на эту схему
- Цепочка владения по-прежнему основана на владельцах, а не на схемах

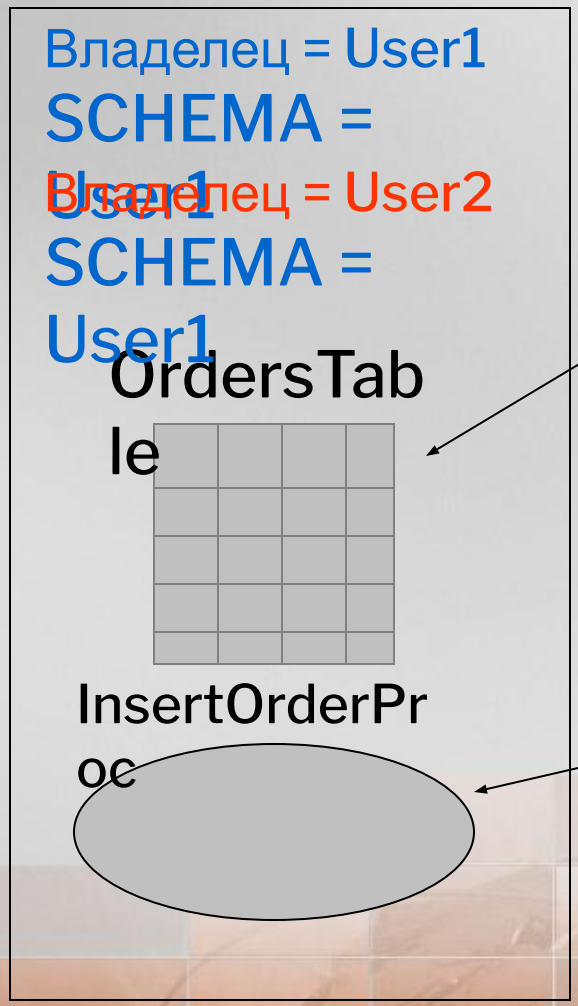


Разделение схемы и владельца



Удаление пользователя не потребует изменения кода приложения

Удаление пользователя не приводит к изменению кода приложения



SELECT custID FROM User1.OrdersTable

App1

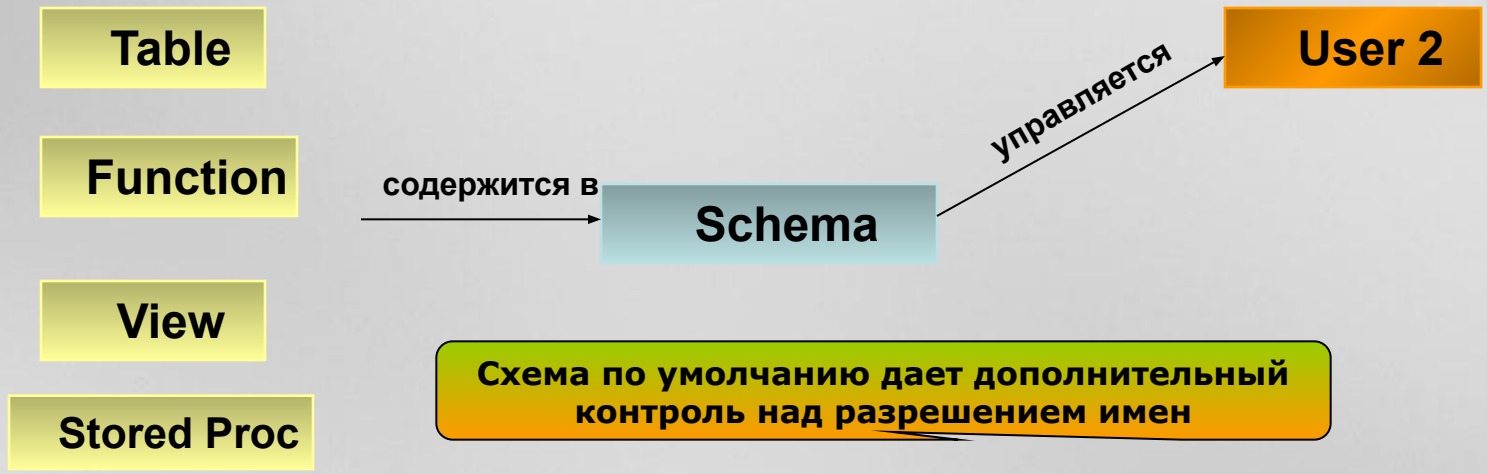
Exec User1.InsertOrderProc (@orderid)

App2

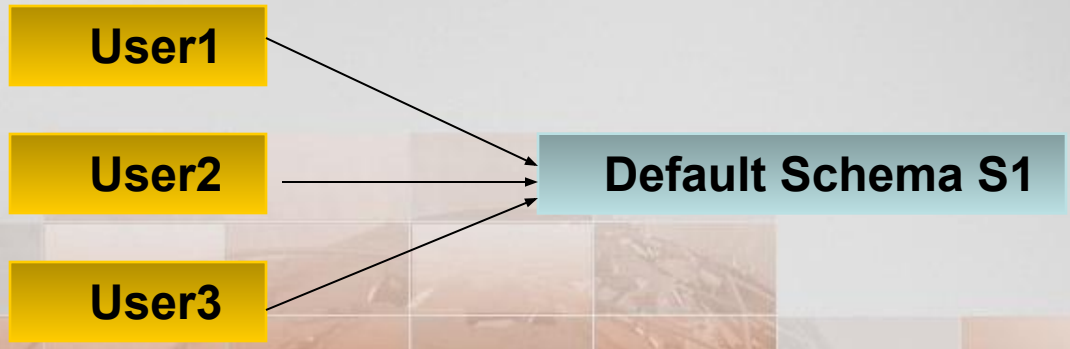
Разделение схемы и владельца

- Разделение владельцев (principals) и схем
  - Владелец (Principal)
    - Сущность от которой защищают объекты
    - Доступны через представление sys.database_principals
  - Схема (Schema)
    - Контейнер объектов; 3-я часть полного наименования
    - Доступны через представление sys.schemas
- Понятие схемы по умолчанию
  - Присуще пользователю или роли приложения
  - Используется при разрешении имен; механизм для поиска объектов
  - Содержится в представлении sys.database_principals
- Удаление пользователя не требует изменения кода приложения

Разделение схемы и владельца



Default Schema



Разрешение имен

Select * from foo

- S1.foo
- Dbo.foo

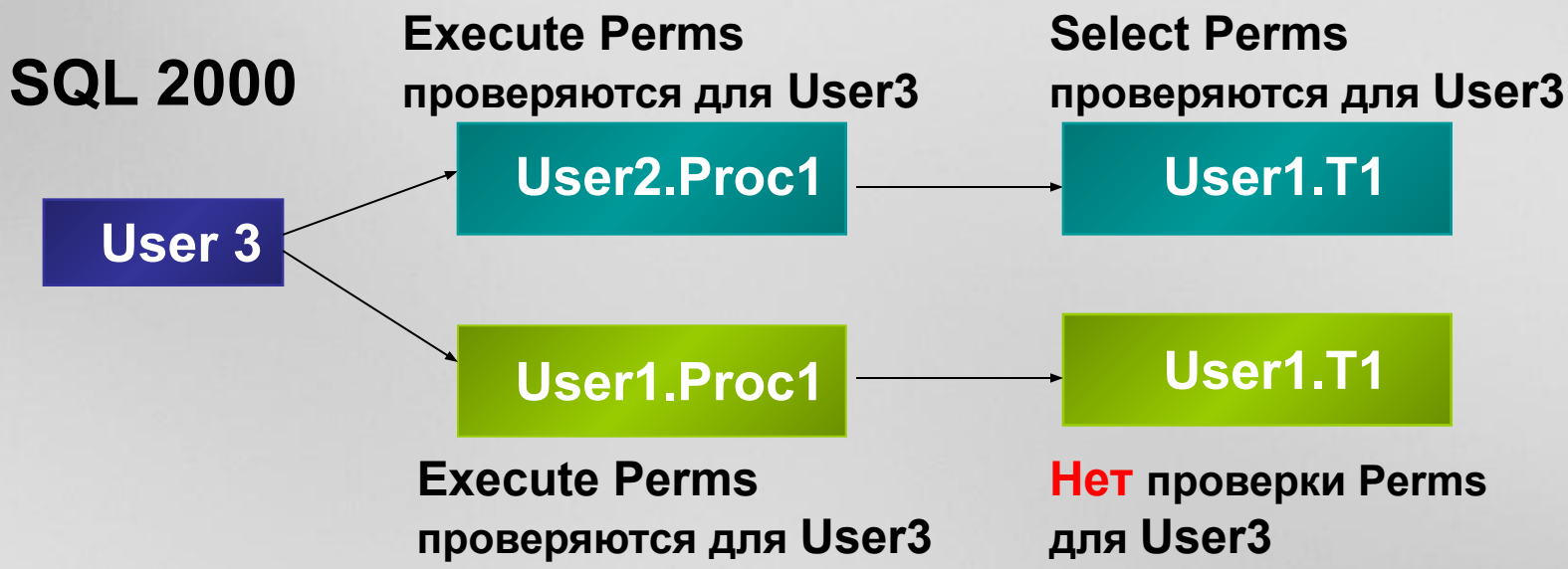
Схема по умолчанию

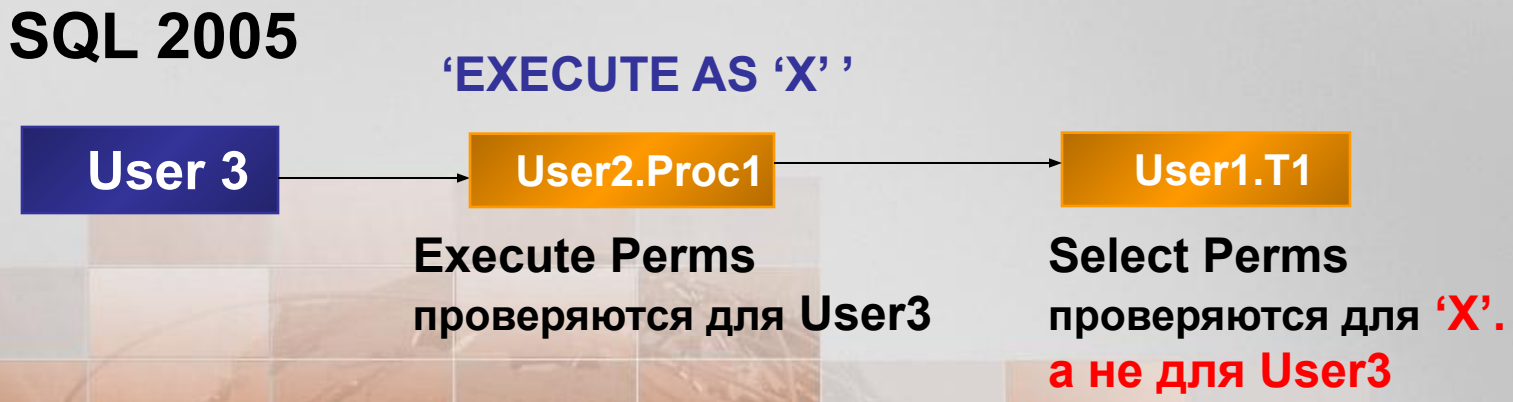
- **Используется для разрешения имен**
  - Не всем пользователям нужно управлять схемами
- **Один и тот же процесс разрешения имён для нескольких пользователей**
  - Схема dbo может является не обязательно единственной общей схемой в плане разрешения имен
- **Зачем это нужно?**
  - для того, чтобы объект мог быть доступен из любого контекста, его не обязательно создавать в схеме dbo
  - Разрешение создания объектов в схеме dbo может привести к некоторому риску безопасности при использовании цепочек владения

Содержание

- Обзор модели безопасности SQL Server 2000
  - Ограничения модели безопасности и способы их устранения
  - Рекомендации по настройке
- Что нового в SQL Server 2005?
  - Новая модель разрешений
    - Безопасность метаданных
    - Более гранулярные разрешения
    - Покрывающие разрешения
  - Разделение схемы и владельца
  - Контекст выполнения хранимого кода

Контекст выполнения хранимого кода





Контекст выполнения хранимого кода

- **Возможность указывать контекст выполнения**
  - процедуры, функции, триггеры
- **Цепочка владения теперь не является единственным механизмом упрощающим назначение прав**
  - Правила цепочек владения по-прежнему применимы
- **Разрешения проверяются для тек.контекста**
  - В отличие от цепочек владения применимо и к командам DDL
- **Применимо и к динамическому SQL**

Контекст выполнения хранимого кода

Для хранимых процедур и определяемых пользователем функций (кроме inline table-valued) предусмотрено четыре выражения для указания контекста выполнения, т.е. пользователя, который будет использоваться при проверке разрешений на объекты, на которые ссылается процедура или функция.

`CALLER` – выполнять под вызвавшим пользователем

`SELF` – под создавшим процедуру

`USER = username` - под указанным пользователем

`OWNER` – под текущим владельцем процедуры

Контекст выполнения хранимого кода

По умолчанию – `EXECUTE AS CALLER`. Для указания имени пользователя *`username`* (отличного от своего) необходимо выполнение одного из условий:

- Входить в фиксированную серверную роль `sysadmin`
- Входить в фиксированную роль базы данных `db_owner`
- Обладать разрешением на имперсонализацию учетной записи, соответствующей пользователю *`username`*.

Можно использовать `EXECUTE AS USER = username` в качестве “обертки” команд, разрешения на которые нельзя передавать. Например так можно делегировать TRUNCATE.

Создание набора разрешений с помощью *EXECUTE AS*

- **Сценарий:**
  - Database Admin хочет дать возможность делать усечение (truncate) ряда таблиц каждую ночь.
- **Проблема:**
  - Truncate непередаваемое разрешение
  - Минимальное покрывающее разрешение - ALTER, но оно дает больше прав чем нужно
- **Решение: нам поможет *EXECUTE AS!***
  - Создать процедуру, которая усекает нужные таблицы
  - Указать в строке execute as пользователя с правами ALTER
  - Дать разрешение на выполнение процедуры нужному пользователю
- **Результат:**
  - Мы только что сделали Truncate назначаемым разрешением!

Безопасность .NET кода

Для .NET сборок предусмотрено указание одного из трех уровней безопасности при загрузке в SQL Server командой `CREATE ASSEMBLY`:

`SAFE` - доступ ко внешним ресурсам не допускается

`EXTERNAL_ACCESS` – допускается доступ к файлам, сетевым ресурсам, реестру, переменным окружения

`UNSAFE` - доступ ко всем ресурсам, в том числе к неуправляемому коду

Если сборка в процессе работы выйдет за указанные при ее загрузке пределы, то CLR сгенерирует исключение и выполнение прекратится.

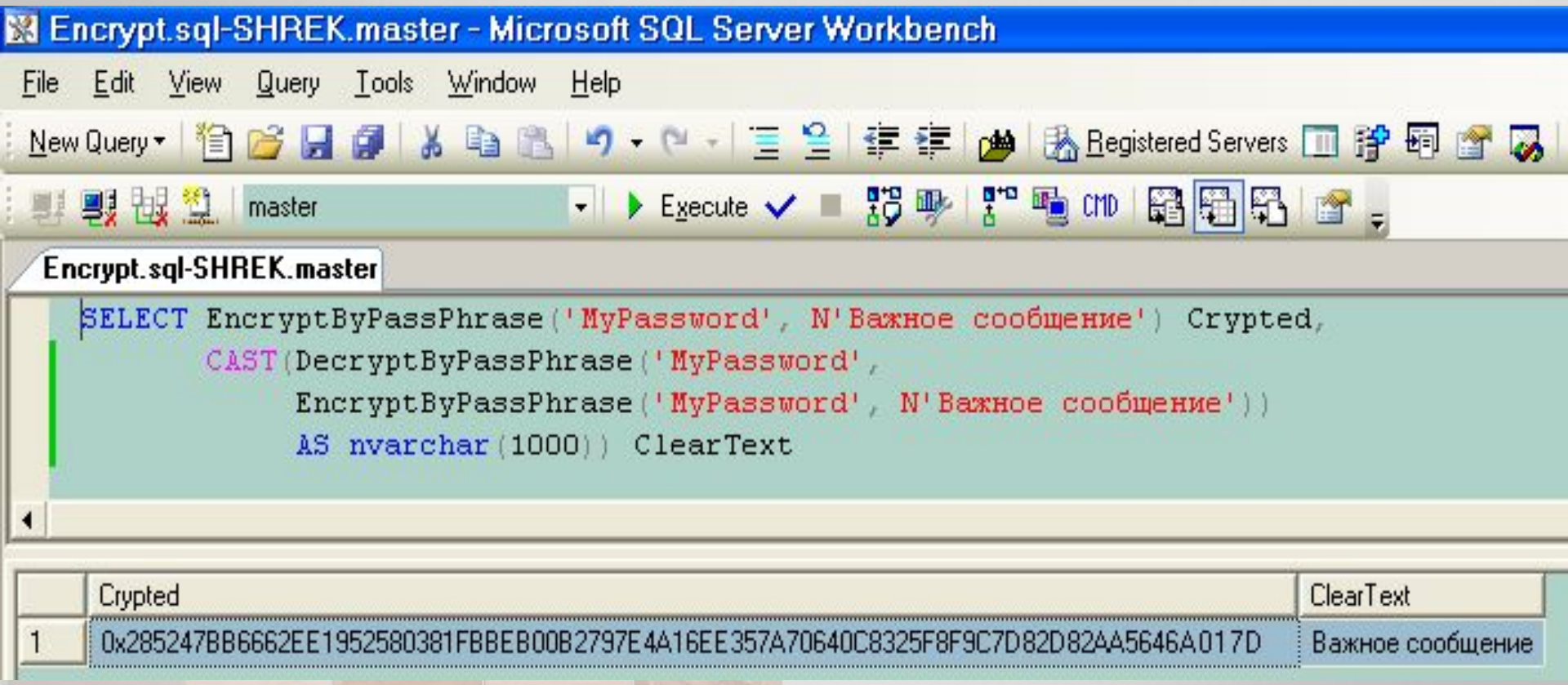
Поддержка криптографии

В SQL Server 2005 есть встроенные средства шифрования, цифровой подписи и верификации

Поддерживаемые типы ключей:

- Симметричные ключи
  - RC4, RC2, DES, AES
- Асимметричные ключи
  - Rivest-Shamir-Adelman Encryption (RSA)

Поддержка криптографии



Триггеры на DDL

Появилась возможность создавать триггеры для DDL, что позволяет вести расширенный аудит. Создаются на базу или сервер, не могут быть `INSTEAD OF`.

SafetyDDL.sql-SHREK.SampleDB - Microsoft SQL Server Workbench

File Edit View Query Tools Window Help

New Query | SampleDB | Execute | CMD | Registere

SafetyDDL.sql-SHREK.SampleDB

```
CREATE TRIGGER safety
    ON DATABASE
   FOR DROP_TABLE
AS
   PRINT 'You must disable Trigger "safety" to drop tables!'
   ROLLBACK
GO
```

SQLQuery1-SHREK.SampleDB* - Microsoft SQL Server Work

File Edit View Query Tools Window Help

New Query | SampleDB | Execute

SQLQuery1-SHREK.SampleDB*

```
DROP TABLE dbo.SampleTable
```

```
You must disable Trigger "safety" to drop tables!
Msg 3609, Level 16, State 2, Line 1
Transaction ended in trigger. Batch has been aborted.
```

ОПРЕДЕЛЯЯ БУДУЩЕЕ

Конечные точки (Endpoints)

Абстракция номера порта, транспортного протокола и принципала

Как происходит соединение:

- Клиент указывает номер порта
- Сервер вычисляет Endpoint, который соответствует указанному порту
- Сервер проверят, что этот Endpoint правильно сконфигурирован
- Сервер проверят, что указанные принципал имеет разрешение на соединение с данным Endpoint

Человеку свойственно ошибаться
SQL Injection

Как избежать:

- Отказаться от динамических запросов в пользу хранимых процедур или параметризованных запросов.
- Использовать регулярные выражения для проверки пользовательского ввода до того, как он будет отправлен в СУБД.
- Использовать функции для экранирования специальных символов
- Проверка пользовательского ввода
- Соответствие типов
- Никогда не строить T-SQL команды непосредственно из введенных пользователем данных
- Использовать процедуры для проверки ввода либо вынести проверку на уровень приложения
- Никогда не принимать строки, участвующие в создании имени файла и содержащие: AUX, CLOCK$, COM1, .., COM8, CON, CONFIG$, LPT1, .., LPT8, NUL и PRN

Подведение итогов

- SQL Server 2000 – промышленная СУБД с надежным механизмом контроля доступа к информации
- SQL Server 2005 – упростит жизнь разработчикам и администраторам при создании и поддержке надежных и безопасных приложений

Ресурсы

Для разработчиков и профессионалов в IT

Официальная страница о SQL Server 2005
- http://www.microsoft.com/sql/2005

Один из самых интересных сайтов, посвящ
безопасности SQL Server
- http://www.sqlsecurity.com

Российское сообщество по SQL Server
- http://www.sql.ru
- Каждый месяц в Microsoft Russia проводится семинар, посвященный SQL Server (уже 18 - в Москве, 4 – в Питере, до 70 участников)

Microsoft Платформа 2005

Microsoft®

ОПРЕДЕЛЯЯ БУДУЩЕЕ

Microsoft Windows Server System