**Идентификация** — установление тождественности неизвестного объекта известному на основании совпадения признаков; опознание.

**Аутентификация** - проверка подлинности пользователя путём сравнения введённого им пароля с паролем, сохранённым в [базе данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) пользователей

(это проверка соответствия субъекта и того, за кого он пытается себя выдать, с помощью некой уникальной информации, в простейшем случае - с помощью имени входа и пароля.)

**Авториза́ция** — предоставление определённому лицу или группе лиц прав на выполнение определённых действий; а также процесс проверки (подтверждения) данных прав при попытке выполнения этих действий

(проверка и определение полномочий на выполнение некоторых действий в соответствии с ранее выполненной аутентификацией.)

**Сервер баз данных** (БД) выполняет обслуживание и управление базой данных и отвечает за целостность и сохранность данных, а также обеспечивает операции ввода-вывода при доступе клиента к информации.

Архитектура клиент-сервер состоит из клиентов и серверов. Основная идея состоит в том, чтобы размещать серверы на мощных машинах, а приложениям, использующим языковые компоненты СУБД, обеспечить доступ к ним с менее мощных машин-клиентов посредством внешних интерфейсов.

***Пользователь (user)*** - это некоторое имя, определенное в базе данных и связанное с некоторым субъектом, который может соединяться с базой данных и выполнять доступ к ее объектам.

***Привилегией*** некоторый поддерживаемый системой признак, который определяет, разрешение на выполнение какой-либо конкретной операции (общей или относящейся к конкретному объекту).

***Роль (role)*** - это объект базы данных, представляющий собой именованный набор привилегий, который может предоставляться пользователю или другой роли. Администратор может создавать новые роли и изменять состав ролей. Как объект базы данных, роль могут управляться средствами языка SQL.

В MSSQL существует два вида ролей:

***Роли уровня сервера*** предоставляют различные степени доступа к операциям и задачам сервера баз данных. Они не зависят от конкретных баз данных и модифицировать их нельзя.

***Роли уровня базы данных*** позволяют назначить набор прав доступа для работы с конкретной базой данных отдельному пользователю или группе. В MSSQL существует три типа таких ролей:

* Роли уровня сервера.

sysadmin - возможность исполнить любое действие в SQL Server;

serveradmin - возможность изменить параметры SQL Server и завершить его работу;

setupadmin - возможность инсталлировать систему репликации и управлять выполнением расширенных хранимых процедур;

securityadmin - возможность контролировать параметры учетных записей для подключения к серверу и предоставлять права доступа к базам данных;

processadmin - возможность управлять ходом процессов в MSSQL Server;

dbcreator - возможность создавать и модифицировать базы данных;

diskadmin - возможность управлять файлами баз данных на диске.

* Роли уровня базы данных.

Заранее определенные роли.

* db\_owner - имеет полный доступ ко всем объектам базы данных;
* db\_accessadmin - осуществляет контроль за доступом к базе данных путем добавления и удаления пользователей;
* db\_datareader - имеет полный доступ к выборке данных;
* db\_datawriter - может выполнять операторы INSERT, DELETE и UPDATE для любой таблицы базы данных;
* db\_ddladmin - имеет возможность создавать, модифицировать и удалять объекты базы данных;
* db\_securityadmin - управляет системой безопасности базы данных;
* db\_backupoperator - позволяет создавать резервные копии базы данных;
* db\_denydatareader - отказывает в разрешении на выполнение оператора SELECT для всех таблиц базы данных;
* db\_denydatawriter - отказывает в разрешении на выполнение операторов модификации данных (INSERT, DELETE, UPDATE) для любых таблиц базы данных;
* public - каждому пользователю при открытии доступа к базе данных автоматом назначается роль public.

Определяемые пользователем роли.

Каждый пользователь базы данных является членом роли базы данных **public** . Если для пользователя не были предоставлены или запрещены конкретные разрешения на защищаемый объект, он наследует разрешения роли public на этот объект. *Пользователей базы данных нельзя удалить из роли* ***public****.*

В роли уровня базы данных(при наличии определенных разрешений) можно добавить любую учетную запись базы данных и другие роли SQL Server. Каждый член предопределенной роли базы данных может добавлять других пользователей в ту же самую роль.

***Менеджер СУБД (database manager)***является собственно программным кодом СУБД. Для каждого менеджера может быть создан один или несколько экземпляров.

MSSQL понимают под ***экземпляром*** логическую среду, включающую в себя несколько баз данных и постоянно хранимые параметры. Экземпляр, таким образом, является монитором нескольких баз данных. Экземпляры выполняются совершенно независимо друг от друга. Такой механизм позволяет осуществлять раздельное администрирование и настройку параметров экземпляров, повышает безопасность и надежность обработки данных.

Базой данных называется коллекция данных, рассматриваемая как единое целое. База данных характеризуется занимаемой ею памятью (табличными пространствами) и каталогом.

**Владение и разделение пользовательских схем в SQL Server**

Владельцы объектов имеют неотзываемые разрешения на их администрирование. Нельзя удалять права доступа у владельцев объектов. Также нельзя удалять пользователей из базы данных, если они владеют в ней объектами.

По умолчанию, если разработчик создает объект в схеме, он принадлежит участнику безопасности, являющемуся владельцем схемы, а не разработчику. Владение объектом можно передать с помощью инструкции Transact-SQL ALTER AUTHORIZATION.

SQL Server поставляется с десятью предварительно определенными схемами, имеющими одинаковое имя со встроенными пользователями и ролями базы данных. Они существуют главным образом ради обратной совместимости. **Невозможно удалить следующие схемы**:

* **dbo**
* **guest**
* **sys**
* **INFORMATION\_SCHEMA**

Схемы **sys** и **INFORMATION\_SCHEMA** зарезервированы для системных объектов. **В этих схемах нельзя создавать и удалять объекты**.

**Схема dbo**

Схема **dbo** представляет собой схему по умолчанию для вновь созданной базы данных. Владельцем схемы **dbo** является учетная запись пользователя **dbo**. Схемой по умолчанию для пользователей, созданных с помощью команды Transact-SQL CREATE USER, является **dbo**.

Пользователи, которым назначена схема **dbo**, не наследуют разрешения пользовательской учетной записи **dbo**. Пользователи не наследуют разрешения схемы, их наследуют объекты базы данных, содержащиеся в схеме.

**Схемы**

Компонент **Database Engine** — основная служба для хранения, обработки и защиты данных. Компонент Компонент Database Engine обеспечивает управляемый доступ и быструю обработку транзакций, достаточную даже для самых требовательных к предоставляемым данным приложений.

***Схема*** - это коллекция объектов базы данных, имеющая одного владельца и формирующая одно пространство имен. (Две таблицы в одной и той же схеме не могут иметь одно и то же имя.)

Схемы используются в модели безопасности компонента Database Engine для упрощения взаимоотношений между пользователями и объектами, и, следовательно, схемы имеют очень большое влияние на взаимодействие пользователя с компонентом Database Engine.

Объекты схемы (schema objects) - таблицы, представления, индексы и т.д. - скалярные (не составные) объекты базы данных.

**Имя схемы** не может быть изменено.

Все объекты базы данных принадлежат той или иной схеме. По умолчанию, в базе данных создается одна схема с именем dbo. Все объекты, вне зависимости от того, какой пользователь их создал, размещаются в этой схеме. Так как эта схема является «домашней», или схемой по умолчанию, для всех пользователей базы данных. В случае необходимости, можно создать дополнительные схемы, разместить в них объекты, и сделать новые схемы – схемами по умолчанию для пользователей.

Схема является коллекцией объектов базы данных, принадлежащих одному пользователю и образующих одно пространство имен. Пространством имен является набор объектов, в котором не может быть повторяющихся имен. Например, две таблицы могут иметь одинаковое имя, если только они находятся в разных схемах. Объект базы данных, например таблица, принадлежит схеме, а схема принадлежит пользователю или роли базы данных.

* Владение схемами и защищаемыми объектами в области схемы можно передать. Дополнительные сведения см. в разделе [ALTER AUTHORIZATION (Transact-SQL)](https://technet.microsoft.com/ru-ru/library/ms187359(v=sql.105).aspx).
* Объекты можно перемещать между схемами.
* Одна схема может содержать объекты, принадлежащие нескольким пользователям базы данных.
* Несколько баз данных могут делить одну схему по умолчанию.
* Разрешениями на схемы и содержащимися в них защищаемыми объектами можно управлять с большей точностью, чем в более ранних выпусках. Дополнительные сведения см. в разделах [GRANT, предоставления разрешения на схему (Transact-SQL)](https://technet.microsoft.com/ru-ru/library/ms187940(v=sql.105).aspx) и [GRANT, предоставление разрешений на объект (Transact-SQL)](https://technet.microsoft.com/ru-ru/library/ms188371(v=sql.105).aspx).
* Схема может принадлежать любому участнику базы данных. Это могут быть роли и роли приложений.
* Пользователь базы данных может быть удален без удаления объектов в соответствующей схеме.
* Если при создании объекта базы данных указать допустимого участника домена (пользователя или группу) в качестве владельца объекта, то этот участник будет добавлен в базу данных в качестве схемы. Новая схема будет принадлежать этому участнику домена.

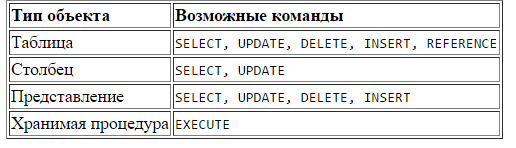
Новые представления каталогов

Начиная с версии SQL Server 2005, схемы могут иметь только одного владельца, но один пользователь может владеть несколькими схемами.  
Инструкция CREATE SCHEMA является атомарной. Иными словами, если в процессе выполнения этой инструкции происходит ошибка, не выполняется ни одна из содержащихся в ней подынструкций.

Схему можно удалить только при условии, что она не содержит никаких объектов.

**MSSQL: объектные и командные права доступа**

На уровне MSSQL существует два типа прав доступа (привилегий): объектные и командные. Объектные права доступа определяют, кто может получать доступ и работать с данными в таблицах и представлениях, и кто может запускать хранимые процедуры. Командные права доступа определяют, кто может удалять и создавать объекты в базе данных.



Командные права доступа определяют, кто может выполнять административные действия. Командные права могут быть назначены только системным администраторам, пользователям, которым назначена роль sysadmin, или владельцам баз данных. Ниже приведены командные права доступа, которые можно предоставить или аннулировать:

* CREATE DATABASE - право создания базы данных.
* CREATE DEFAULT - право создания стандартного значения для столбца таблицы.
* CREATE PROCEDURE - право создания хранимой процедуры.
* CREATE RULE - право создания правила для столбца таблицы.
* CREATE TABLE - право создания таблицы.
* CREATE VIEW - право создания представления.
* BACKUP DATABASE - право создания резервной копии базы данных.
* BACKUP TRANSACTION - право создания резервной копии журнала транзакций.

|  |  |
| --- | --- |
| sys.objects (Transact-SQL) | Содержит одну строку для каждого определенного пользователем объекта в области схемы, который создан в базе данных. |
| sys.schemas (Transact-SQL) | Содержит по одной строке для каждой схемы базы данных. |
| sp\_addlogin [ @loginame = ] 'login'  [ , [ @passwd = ] 'password' ] | Создает новое имя входа на SQL Server, позволяющее пользователю подключаться к экземпляру SQL Server с применением проверки подлинности SQL Server. |
| sp\_password [ [ @old = ] 'old\_password' , ]  { [ @new =] 'new\_password' }  [ , [ @loginame = ] 'login' ] | Добавляет или изменяет пароль для Microsoft SQL Server входа. |
| sp\_defaultdb [ @loginame = ] 'login', [ @defdb = ] 'database' | Изменяет базу данных по умолчанию для Microsoft SQL Server входа. |
| sp\_grantlogin [@loginame=] 'login' | Создает имя входа на SQL Server. |
| sp\_droplogin [ @loginame = ] 'login' | Удаляет имя входа SQL Server. Это предотвращает доступ к экземпляру SQL Server с этим именем входа. |
| DROP LOGIN login\_name | Нельзя удалить текущее имя входа. Также нельзя удалить имя входа, владеющее любым защищаемым объектом уровня сервера или заданием агента SQL Server.  Можно удалить имена входа, сопоставленные пользователям базы данных; однако это приведет к появлению пользователей, утративших связь с учетными записями |
| CREATE LOGIN |  |
|  |  |
|  |  |
| GRANT, предоставление разрешений на объект | Предоставляет разрешения на таблицу, представление, функцию с табличным значением, хранимую процедуру, расширенную хранимую процедуру, скалярную функцию, агрегатную функцию  **ALL** эквивалентно предоставлению всех разрешений применимых к указанному объекту  **WITH GRANT OPTION** Показывает, что участнику будет дана возможность предоставлять указанное разрешение другим участникам |
| DENY | Запрещает разрешение для участника. Предотвращает наследование разрешения участником через его членство в группе или роли.  **CASCADE** Обозначает, что разрешение запрещается для указанного участника и всех других участников, которым этот участник предоставил разрешение. Требуется, если участник имеет разрешение с параметром GRANT OPTION.  Инструкция DENY завершится ошибкой, если не указан аргумент CASCADE при отзыве разрешения у участника, которому это разрешение было предоставлено с параметром GRANT OPTION. |
| REVOKE | Отменяет разрешения, предоставленные или запрещенные для пользователя базы данных, роли базы данных или роли приложения.  **GRANT OPTION** Показывает, что отменяемое разрешение также отменяется для других участников, для которых оно было предоставлено или запрещено данным участником |
| deny select on награда to public  grant select on награда to test  --as test  select \* from Награда  --**error** | grant select on награда to public  deny select on награда to test  --as test  select \* from Награда  --**error** |
| deny select on награда to test  grant select on награда to public  --as test  select \* from Награда  --**error** | grant select on награда to public  deny select on награда to test  --as test  select \* from Награда  --**error** |