



# Доступ в DB2

Белькова Евгения,  
программист отдела  
тестирования,  
группа DB2 Tools



# Понятие сервера и клиента

**Сервер базы данных** управляет одной или большим числом баз данных и обслуживает запросы клиентов, которые хотят получить доступ к этим базам данных.

Определяется версией установленного продукта DB2 UDB.



# Понятие сервера и клиента

DB2-клиент встроен в каждый программный продукт DB2 UDB.

**Клиент** предоставляет способность выполнять доступ к базам данных DB2 и осуществлять администрирование этих баз данных.



# Понятие сервера и клиента

**Локальное приложение** – выполняется на сервере, на котором размещена база данных.

**Удаленное приложение** – выполняется на сервере, но обращается к базе данных, расположенной на другой машине.

Если вы работаете на клиентской машине, вы можете запускать только удаленные приложения, так как на клиентской машине не может быть баз данных..



# Доступ

Доступом к данным и функциям системы DB2 UDB управляют три уровня безопасности:

- Аутентификация
- Авторизация
- Права (полномочия и привилегии)



# Аутентификация

Проверка допустимости доступа к базе данных или экземпляру в первую очередь производится вне системы.

**Аутентификация** – верификация пользователя путем проверки его идентификатора пользователя или пароля.

Этот процесс гарантирует, что пользователь является именно тем, за кого себя выдает.



# Типы аутентификации

Определяет, каким образом и где происходит проверка пользователя.

Тип аутентификации сохраняется в файле конфигурации менеджера баз данных на сервере.

Первоначально он задается при создании экземпляра.

Тип аутентификации распространяется на весь экземпляр, определяя доступ к серверу баз данных и всем базам данных, которыми он управляет.



# Тип SERVER

Задается по умолчанию.

Аутентификация происходит на сервере при помощи средств защиты локальной операционной системы.

При попытке установления связи задаются имя пользователя и пароль, производится их сравнение со всеми комбинациями имени пользователя и пароля, действительными на этом сервере.





# Тип SERVER

## **Примечание:**

Программа сервера определяет, является ли соединение локальным или удаленным.

При локальных соединениях для успешной аутентификации типа SERVER не требуется ID пользователя и пароля.



# Тип SERVER\_ENCRYPT

Аналогично типу SERVER.

Но! перед отправкой на сервер пароли шифруются системой DB2 на стороне клиента.



# Тип CLIENT

Предполагается, что пользователь прошел проверку на клиенте, на котором размещено приложение.

Дополнительная аутентификация на сервере не требуется.



# Тип Kerberos

Клиент, и сервер DB2 UDB должны работать в операционных системах, где поддерживается протокол защиты Kerberos.

Используется обычное шифрование для создания общего секретного ключа.

Этот ключ становится паролем пользователя и используется для проверки личности пользователя во всех случаях, когда требуются локальные или сетевые службы.



# Тип Kerberos

Пароль не передается, вводится во все сервера сети и его можно не проверять (single-sign-on). Взлом одного сервера не грозит провалу всей системы, так как при отправке ключа он не распознается.

Kerberos Domain Controller  
Key Distribution Center



# Аутентификация

**ЗАДАНИЕ:** Настройка уровня аутентификации.

1. Находясь в Центре Управления, щелкните правой кнопкой мыши по экземпляру DB2 и выберите в контекстном меню пункт Конфигурировать параметры.
2. Найдите параметр AUTHENTICATION и выберите его.
3. Найдите параметр TRUST\_ALLCLNT (Доверять всем клиентам).
4. Попробуйте изменить эти параметры, поэкспериментируйте с ними.

Примечание:

Для того, чтобы настройки вступили в силу, необходимо перезапустить экземпляр.

Для этого правой кнопкой мыши щелкните по экземпляру и выберите команду Стоп, а затем Старт.



# Доступ к DB2 UDB

Для решения задач администрирования DB2 необходимо располагать именем пользователя Windows.

Имя пользователя должно принадлежать группе Администраторы и должно быть допустимым именем пользователя системы DB2.

В большинстве случаев DB2 создает новое имя пользователя во время инсталляции (db2admin), которое потом может использоваться для администрирования.

# Доступ к DB2 UDB

## *Примечание:*

По умолчанию полномочия системного администратора (SYSADM) предоставляются любому допустимому имени пользователю DB2, которое принадлежит группе администраторов в среде Windows.

SYSADM\_GROUP – группа пользователей, обладающая правами администратора.





# Доступ к DB2 UDB

## **ЗАДАНИЕ:**

1. Находясь в Центре Управления, щелкните правой кнопкой мыши по экземпляру DB2 и выберите в контекстном меню пункт Конфигурировать параметры.
2. В разделе Управление найдите параметр SYSADM\_GROUP.

Можно указать группу, пользователи которой будут обладать правами SYSADM



# Авторизация

Следующий уровень безопасности.

**Авторизация** – проверка полномочий, в процессе которой пользователь должен быть распознан системой DB2 через имя авторизации SQL или идентификатора авторизации (authid).

Авторизация – это определение прав доступа для конкретного пользователя.



# Авторизация

Доступ внутри системы управляются посредством административных полномочий и привилегий пользователей в рамках менеджера баз данных (экземпляра).



# Права

К правам относятся полномочия и привилегии.

*Привилегии* позволяют пользователям создавать ресурсы баз данных и обращаться к ним.

*Уровни полномочий* предоставляют способ объединения привилегий и высокоуровневых операций обслуживания и утилит менеджера баз данных.

# Полномочия

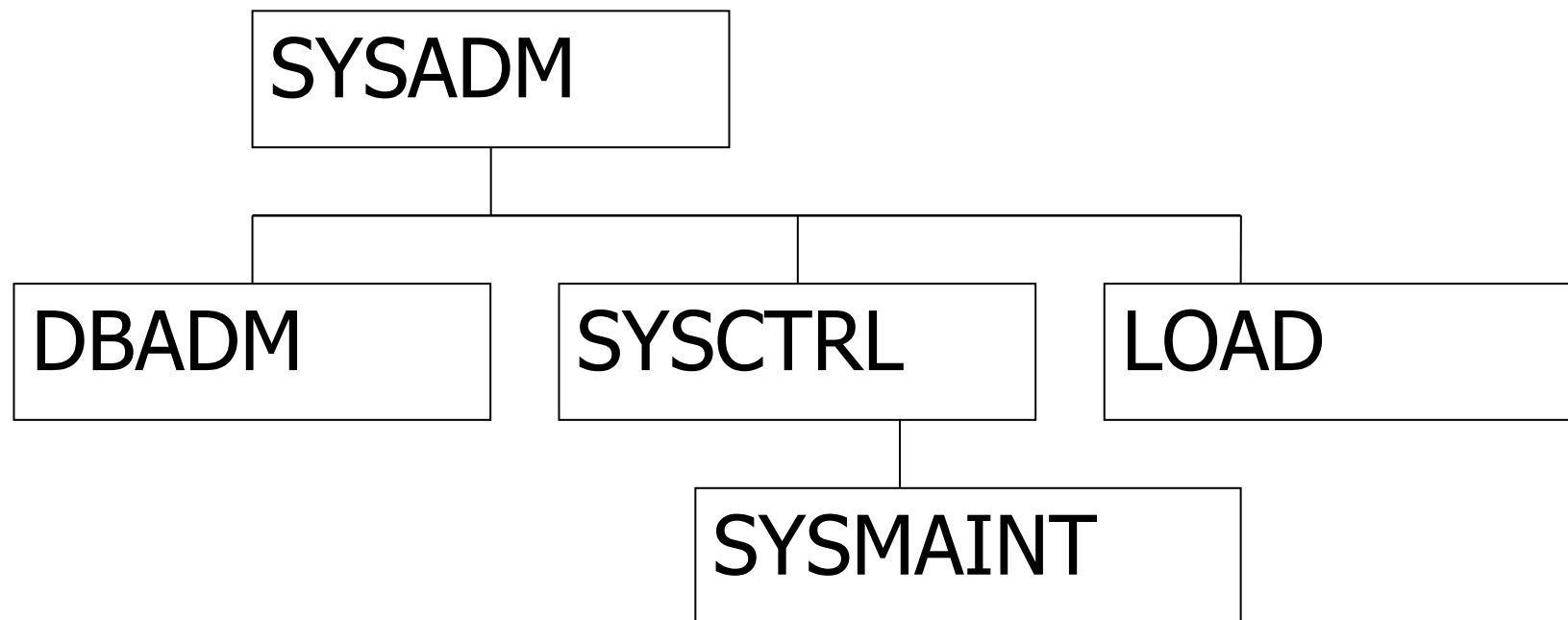
*Полномочия базы данных* позволяют пользователям выполнять действия на уровне этой базы данных.

Привилегии и полномочия баз данных могут вместе служить для управления доступом к менеджеру баз данных и его объектам баз данных.

Пользователи могут обращаться только к тем объектам, в отношении которых они обладают соответствующей авторизацией, т.е. имеют необходимые для действия полномочия или привилегии.

# Полномочия

Уровни полномочий в DB2:



# Уровни полномочий

## **SYSADM**

- Системный администратор.
- Наивысший уровень административных полномочий.
- Обеспечивает контроль над всеми ресурсами системы.
- Включает все привилегии для всех баз данных в рамках экземпляра.
- Обладает правом предоставлять и отзывать полномочия и привилегии.

# Уровни полномочий

## **DBADM**

- Второй после SYSADM уровень.
- Администратор базы данных.
- Применяется ТОЛЬКО к специфической базе данных и обладает всеми привилегиями в рамках базы данных.
- Право предоставления и отзыва привилегий пользователям конкретной БД вне зависимости кто эти привилегии назначал.



# Уровни полномочий

## **SYSCTRL**

- Наивысший уровень полномочий управления системой.
- Затрагивает ТОЛЬКО системные ресурсы.
- Операции обслуживания баз данных.
- НО! Непосредственного доступа к данным базы данных не имеет.



# Уровни полномочий

## **SYSMANT**

- Второй после SYSCTRL уровень.
- Операции обслуживания баз данных в рамках экземпляра.
- НО! Непосредственного доступа к данным базы данных не имеет.

## **LOAD**

- Полномочие выполнения операций обслуживания баз данных на уровне базы данных.



# Полномочия

## **ЗАДАНИЕ:**

1. Находясь в Центре Управления, щелкните правой кнопкой мыши по базе данных и выберите в контекстном меню пункт Полномочия.
2. Попробуйте создать пользователей с различными полномочиями и привилегиями.



# Привилегии

**Привилегия** – это право доступа к конкретному объекту базы данных.

Контролируются пользователями, имеющие полномочия SYSADM или DBADM.

В рамках собственной базы данных каждый пользователь имеет право предоставлять или отзывать привилегии на объекты внутри этой базы данных.

# Привилегии

Некоторые привилегии назначаются по умолчанию при создании объектов.

Например, привилегии

- CONNECT,
- CREATETAB,
- BINDADD,
- IMPLICIT\_SCHEMA

предоставляются всем пользователям.



# Привилегии

CONNECT разрешает пользователю получать доступ к базе данных.

BINDADD предоставляет пользователю возможность создавать новые пакеты в базе данных.



# Привилегии

*Пакет* (package) – объект базы данных, который содержит информацию, необходимую для наиболее эффективного доступа менеджера баз данных к хранимым данным для целей конкретной прикладной программы. В результате предоставления пользователям привилегии по выполнению того или иного пакета, отпадает необходимость явно предоставлять привилегии в отношении объектов, на которые ссылается пакет.



# Привилегии

`CREATETAB` разрешает пользователю создавать новые таблицы в базе данных.

`IMPLICIT_SCHEMA` дает возможность неявного создания схемы во время создания нового объекта.





# Схема

Схема – некоторая совокупность именованных объектов.

Обеспечивает логическую классификацию объектов в базе данных.

Схема – тоже объект базы данных.

Имя схемы (schema) используется в качестве составного имени объекта, состоящего из двух частей.

# Схема

Схема может создаваться явным и неявным способом.

*Явный способ создания схемы:*

- С помощью оператора CREATE SCHEMA;
- При создании объекта необходимо указывать имя схемы в качестве первой части составного имени.
- Владелец – текущий пользователь.



# Схема

Пример явного способа:

```
CREATE SCHEMA A;
```

```
CREATE TABLE A.TABLE1
```

```
(
```

```
    NAME varchar(40) NOT NULL,
```

```
    TYPE varchar(20),
```

```
    URL  varchar(128)
```

```
);
```

```
INSERT INTO A.TABLE1 VALUES('A','B','C');
```

```
SELECT * FROM A.TABLE1;
```



# Схема

## **Примечание:**

Если вы в первой части составного имени используете несуществующее имя схемы, схема с таким именем будет автоматически создана.

# Схема

*Неявный способ:*

- Требуется наличие привилегии `IMPLICIT_SCHEMA`;
- Если при создании объекта имя схемы не указывается, схема создается по умолчанию.
- По умолчанию именем схемы становится ID пользователя, который создает этот объект.
- Владелец – текущий пользователь.



# Схема

Пример неявного способа:

```
CREATE TABLE TABLE1
```

```
(
```

```
    NAME varchar(40) NOT NULL,
```

```
    TYPE varchar(20),
```

```
    URL  varchar(128)
```

```
);
```

```
INSERT INTO TABLE1 VALUES('A','B','C');
```

```
SELECT * FROM TABLE1; - это соответствует  
запросу SELECT * FROM ID.TABLE1;
```



# GRANT

Наделение полномочий и привилегий осуществляется оператором GRANT.

В общем виде:

```
GRANT <privilege> ON <object DB2> TO <id>
```

Пример:

```
GRANT DBADM ON DATABASE DBMY TO  
USER1;
```

```
GRANT SELECT ON A.TABLE1 TO USER1;
```



# REVOKE

Отзыв полномочий – REVOKE.

В общем виде:

REVOKE <privilege> ON <object DB2> FROM <id>

Пример:

REVOKE DBADM ON DATABASE DBMY FROM  
USER1;

REVOKE SELECT ON A.TABLE1 FROM USER1;





Лекция закончена.

Спасибо за внимание.

?Вопросы и пожелания?

---

С уважением,  
Белькова Евгения