САМАРСКИЙ УНИВЕРСИТЕТ
SAMARA UNIVERSITY

# Безопасность систем баз данных

## Лекция 2
## Управление доступом к данным

Агафонов Антон Александрович
к.т.н., доцент кафедры ГИиИБ

Самара

# План лекции

- Управление доступом. Привилегии
- Модели управления доступом
  - Дискреционная
  - Мандатная
  - Ролевая
- Управление привилегиями средствами языка SQL

# Управление доступом

**Целью управления доступом** является ограничение действий или операций, которые может выполнять легальный пользователь информационной системы. Управление доступом ограничивает ряд действий, которые пользователь может выполнять над ресурсами информационной системы непосредственно, а также то, какими программными компонентами информационной системы он может управлять (запускать, останавливать, конфигурировать).

Ключевые понятия:

- **Субъект доступа** (access subject) — лицо или процесс, действия которого регламентируются правилами разграничения доступа.
- **Объект доступа** (access object) — единица информации автоматизированной системы, доступ к которой регламентируется правилами разграничения доступа. Объектами доступа в СУБД является практически все, что содержит конечную информацию: таблицы, представления, а также более мелкие элементы данных: столбцы и строки таблиц и даже поля строк (значения), а также программируемые объекты базы данных.

# Управление доступом. Привилегии

Привилегия – разрешение на использование определенной услуги управления данными для доступа к объекту данных, предоставляемое идентифицированному пользователю.

Принципы, на которых основаны политики управления доступом:

- **Принцип минимальных привилегий** – принцип организации доступа к ресурсам, когда каждый модуль (процесс, пользователь или программа) должны иметь доступ к такой информации и ресурсам, которые минимально необходимы для успешного выполнения его задач. Противоположностью этому является **принцип максимальных привилегий**, предусматривающий максимальную доступность данных в БД.
- **Принцип открытой** или **закрытой системы**. В открытой системе разрешен доступ к тем объектам, для которых явно не настроен запрет. В закрытой системе доступ к объекту разрешен только при наличии явного разрешения, в противном случае доступ к объектам запрещен.
- **Принцип централизованного и децентрализованного администрирования**. В варианте централизованного администрирования привилегий доступа одна сущность контролирует доступ ко всем объектам, в то время как в децентрализованной системе различные сущности контролируют доступ к различным объектам.

# Управление доступом

Для противодействия несанкционированному доступу в большинстве современных СУБД реализована многоуровневая система обеспечения безопасности, включающая три процедуры:

- **идентификация** – назначение пользователю (процессу) уникального имени;
- **аутентификация** – процедура проверки подлинности пользователя, представившего свой идентификатор. Обычно пользователь подтверждает то, что он является именно тем, за кого он себя выдает, путем ввода в систему уникальной (неизвестной другим) информации о себе. Наиболее распространенный способ подтверждения – ввод символьного пароля;
- если пользователь успешно прошел процедуру аутентификации, то сервер осуществляет его **авторизацию** – процедуру предоставления пользователю определенных ресурсов и прав на их использование

Набор прав (привилегий) назначается администратором БД. Обычно пользователь вводится в группу с заранее предопределенными ролями (администратор данных, администратор БД, пользователь БД, гость). Все дальнейшее взаимодействие пользователя с объектами БД строго регламентируется в соответствии с назначенной ролью.

# Модели управления доступом

**Модель управления доступом** – это структура, которая определяет порядок доступа субъектов к объектам. Для реализации правил и целей этой модели используются технологии управления доступом и механизмы безопасности.

Основные модели управления доступом:

- **Дискреционная (избирательная) модель** (discretionary access control, DAC)
- **Мандатная (принудительная) модель** (mandatory access control, MAC)
- **Ролевая модель** (role based access control, RBAC)

# Дискреционная модель управление доступом

**Дискреционное управление доступом** — управление доступом субъектов к объектам на основе списков управления доступом или матрицы доступа.

Дискреционное управление доступам определяет разграничение доступа между поименованными субъектами и поименованными объектами, при этом субъект с определенным правом доступа может передать это право любому другому субъекту.

Правила доступа определяют для каждого пользователя и каждого объекта в системе типы доступа, разрешенные пользователю к данному объекту. Запрос пользователя на доступ к объекту проверяется по указанным правилам; при наличии привилегии доступа, доступ разрешается; в противном случае доступ отклоняется.

# Дискреционная модель управление доступом

Для каждой пары (субъект — объект) должно быть задано явное и недвусмысленное перечисление допустимых типов доступа (читать, писать и т. д.), то есть тех типов доступа, которые являются санкционированными для данного субъекта (индивида или группы индивидов) к данному ресурсу (объекту).

Субъект доступа «Пользователь № 1» имеет право доступа только к объекту доступа № 3 на чтение. Субъект «Пользователь № 2» имеет право доступа как к объекту доступа № 1 (на чтение и изменение), так и к объекту доступа № 2 (на чтение).

| | **Объект доступа №1** | **Объект доступа №2** | **Объект доступа №3** |
|---|---|---|---|
| **Пользователь №1** | – | – | Чтение |
| **Пользователь №2** | Чтение, запись | Чтение | – |
| **Пользователь №3** | – | – | – |

# Дискреционная модель управление доступом

Использование матрицы доступа основывается на следующих предположениях:

- Все субъекты и объекты доступа должны быть однозначно идентифицированы.
- Для любого объекта должен быть определён пользователь-владелец.
- Владелец обладает определенными правами доступа (может передавать свои права другим).
- В системе существует привилегированный пользователь, обладающий правом полного доступа к любому объекту. Это нужно для того, чтобы исключить возможность недоступных объектов.

Владелец БД является администратором, ему предоставлены все привилегии. Он имеет право регистрации новых пользователей и предоставление им привилегий.

Привилегии делятся на системные и объектные. Системные привилегии позволяют создавать и модифицировать объекты БД. Объектные привилегии позволяют использовать объекты (выполнять запросы выборки, добавления, модификации данных).

# Дискреционная модель управление доступом

Достоинства:

- ▲ Наглядность
- ▲ Простота реализации и понимания

Недостатки дискреционной модели применительно к БД:

- ▼ Проблемы разграничения доступа к различным строкам одной таблицы. Частичное решение проблемы — запрет доступа к таблице и разрешение доступа к отдельным представлениям, созданным на базе этой таблицы и содержащим различные условия выборки строк.
- ▼ Проблемы разграничения доступа к отдельным столбцам одной таблицы.
- ▼ Отсутствие средств защиты от несанкционированного распространения конфиденциальной информации. Пользователь, легально получивший доступ (с правом чтения `Select`) к таблице с конфиденциальной информацией, может сделать эту информацию доступной другим пользователям путем вставки (`Insert`) этой информации в другую (например, временную) общедоступную таблицу. Этот недостаток делает дискреционное управление доступом уязвимым для вредоносных атак типа «Троянский конь».

# Мандатная модель управления доступом

**Мандатное (принудительное) управление доступом** определяет разграничение доступа субъектов к объектам, основанное на назначении метки конфиденциальности для информации, содержащейся в объектах, и выдаче официальных разрешений (допуска) субъектам на обращение к информации такого уровня конфиденциальности.

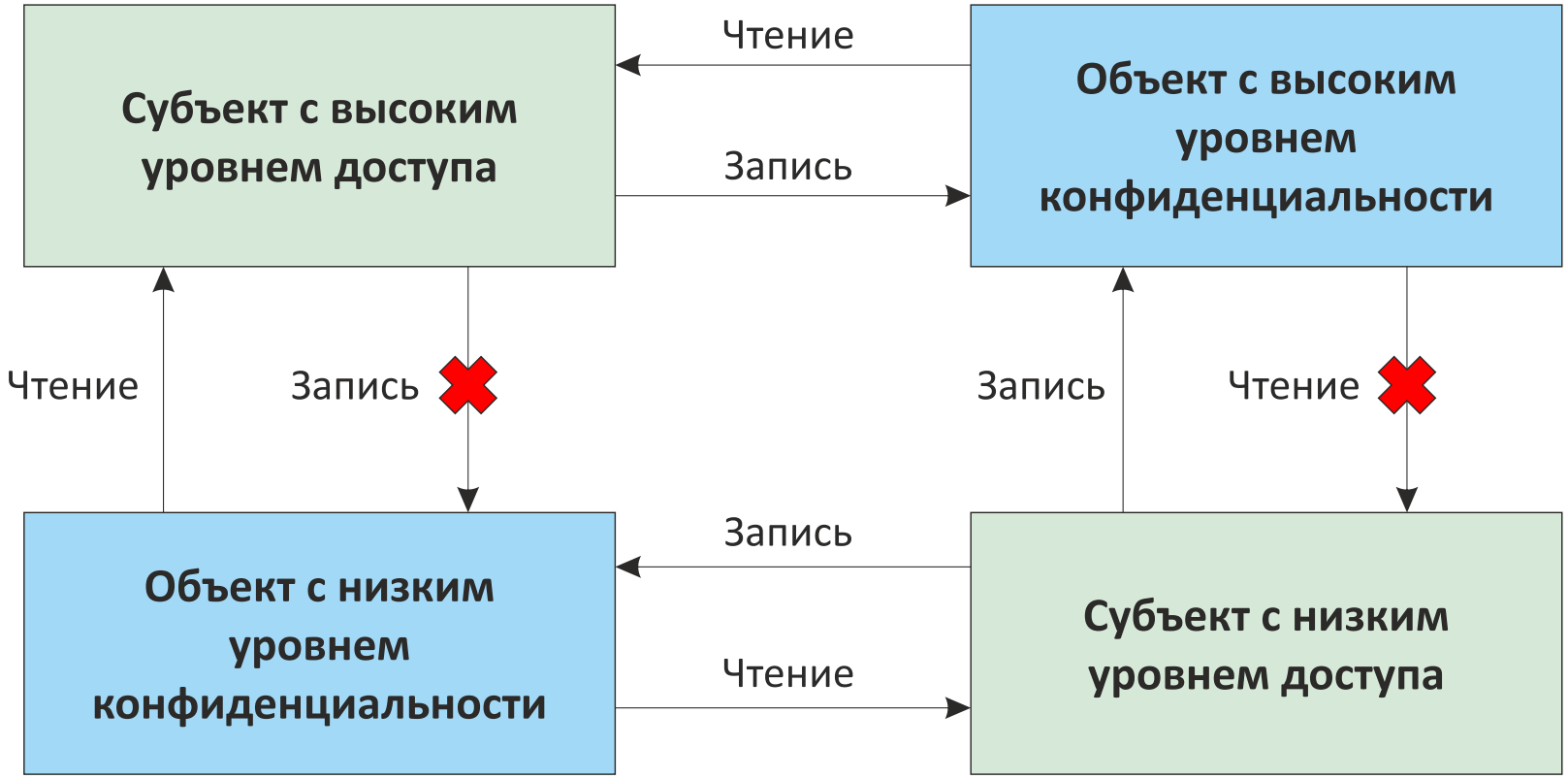
**Метки конфиденциальности** - это элемент иерархически упорядоченного набора.
Например можно рассмотреть следующие уровни: $TS$ - сверхсекретный (top secret), $S$ - секретный (secret), $C$ - конфиденциальный (confidentional) и $U$ - неклассифицированный (unclassified), где $TS > S > C > U$. Пользователям не допускается создавать объекты с уровнем безопасности ниже, чем его собственный.

Контроль доступа в системах обязательной защиты основан на следующих двух принципах:

- Чтение вниз (Read down): субъект может читать только те объекты, класс доступа которых равен либо меньше его класса доступа.
- Запись вверх (Write up): субъект может записывать только те объекты, класс доступа которых равен либо выше его класса доступа.

Удовлетворение этих принципов предохраняет конфиденциальную информацию, от копирования в объекты с более низким уровнем допуска.

Субъект не может читать информацию объекта, метка конфиденциальности которого выше его собственной, но также ему запрещена запись информации в объекты с более низким уровнем безопасности, что не позволит такому субъекту понизить уровень секретности информации, к которой он получил легальный доступ.



# Мандатная модель управления доступом

Достоинства:

- ▲ Мандатная модель доступа не подвержена атаке типа «Троянский конь»

Недостатки:

- ▼ Снижение эффективности работы информационной системы, так как проверка доступа к объектам осуществляется не только при открытии объектов, но и перед выполнением любой операции.

# Ролевая модель управления доступом

**Управление доступом на основе ролей** — развитие политики избирательного управления доступом, при этом права доступа (привилегии) субъектов системы на объекты группируются с учётом специфики их применения, образуя **роли**.

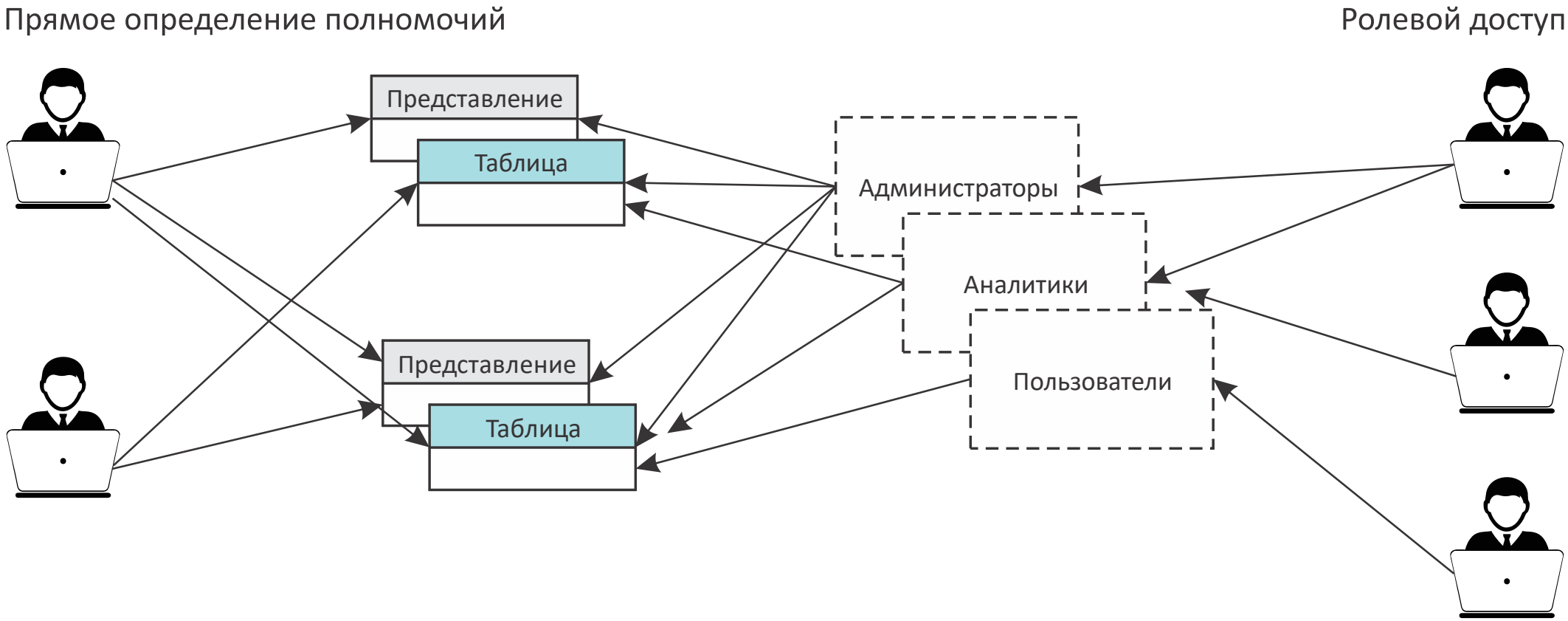
Роль – это именованная совокупность привилегий, которые могут быть предоставлены пользователям или другим ролям.

Формирование ролей позволяет определить чёткие и понятные для пользователей правила разграничения доступа, а также реализовать гибкие механизмы управления привилегиями. В отличие от дискреционной модели привилегии доступа определяются не для субъектов, а для ролей, а субъекты в свою очередь уже получают членство в различных ролях и наследуют определенные для нее привилегии.

Множества субъектов, ролей и привилегий связаны по типу «многие ко многим», что позволяет сформулировать следующие утверждения, характеризующую ролевую модель:

- Один субъект может иметь несколько ролей.
- Одну роль могут иметь несколько субъектов.
- Одна роль может иметь несколько разрешений.
- Одно разрешение может принадлежать нескольким ролям.

# Ролевая модель управления доступом



# Ролевая модель управления доступом

Несмотря на то, что роль является совокупностью привилегий доступа на объекты, ролевое управление доступом отличается от дискреционного:

- порядок предоставления доступа субъекту определяется в зависимости от имеющихся (или отсутствующих) у него ролей в каждый момент времени, что скорее характерно мандатной модели управления доступом;
- правила ролевого разграничения доступа являются более гибкими, чем при мандатном подходе к разграничению.

Использование ролевой модели во многом упрощает разграничение прав доступа и по своему определению наилучшим образом подходит для реализации принципа наименьших привилегий, а также четкого и прозрачного разграничения полномочий.

# Ролевая модель управления доступом

В ролевой модели также вводится понятие «сессия». **Сессия** – это подмножество ролей, которые активировал пользователь после входа в систему.

1. Для каждой роли создается набор полномочий, которые предоставляют пользователю определенные права.
2. Каждому пользователю предоставляется список ролей.
3. После авторизации пользователя создается сессия.

# Применение различных моделей доступа

В обычных версиях СУБД в настоящее время поддерживается смешанная модель управления доступом, основанная на ролевой и дискреционной моделях. Такой подход совмещает гибкость и удобство управления доступом на уровне ролей с возможностью точного управления на уровне пользователя.

Средства мандатной защиты предоставляются специальными (trusted) версиями СУБД:

- В Oracle Database есть подсистема Oracle Label Security (LBAC, Label-Based Access Control system)
- В PostgreSQL в версии 9.2 появилась начальная поддержка SELinux.



# Управление привилегиями средствами языка SQL

Язык SQL включает операторы GRANT и REVOKE, предназначенные для организации разграничения доступа к объектам базы данных. Механизм защиты построен на использовании идентификаторов пользователей и предоставляемых им прав владения и привилегий.

**Идентификатором пользователя** называется обычный идентификатор языка SQL, применяемый для обозначения некоторого пользователя базы данных. Каждому пользователю должен быть назначен собственный идентификатор, присваиваемый администратором базы данных. Каждый выполняемый СУБД SQL-оператор выполняется от имени какого-либо пользователя. Идентификатор пользователя определяет, на какие объекты базы данных пользователь может ссылаться и какие операции с этими объектами он имеет право выполнять.

Каждой создаваемой в БД роли (так же, как и пользователю) назначается уникальный идентификатор.

# Предоставление привилегий

Предоставление прав на защищаемый объект санкционированному пользователю или роли осуществляется при посредничестве инструкции GRANT. Обычно его использует владелец таблицы с целью предоставления доступа к ней другим пользователям. Оператор GRANT имеет следующий формат:

```
GRANT {привилегия на объект [,…] | имя роли [,…]}
ON имя объекта
TO {получатель привилегии [,…]}
[WITH GRANT OPTION | WITH ADMIN OPTION]
```

# Предоставление привилегий

| Привилегия | Описание | Применимо к объектам |
|---|---|---|
| ALL PRIVILEGIES | Назначить все привилегии | Ко всем объектам |
| SELECT \| INSERT \| UPDATE \| DELETE | Право на просмотр, вставку, редактирование и удаление данных в таблице (столбце) | Таблицы, столбцы, представления (только SELECT) |
| REFERENCES | Право управления ограничением внешнего ключа (FOREIGIN KEY), право использовать столбцы в любом ограничении | Таблицы и столбцы |
| USAGE | Дает право использовать данный объект для определения другого объекта | Домены, пользовательские типы данных, наборы символов, порядки сравнения и сортировки, трансляции |
| UNDER | Право на создание подтипов или объектных таблиц | Структурные типы |
| TRIGGER | Право на создание триггера | Таблицы |
| EXECUTE | Запуск на выполнение | Хранимые процедуры и функции |

# Предоставление привилегий

| Привилегия | Описание |
|---|---|
| CREATE <тип объекта> | Создание объекта некоторого типа |
| ALTER <тип объекта> | Изменение структуры объекта |
| DROP <тип объекта> | Удаление объекта |

# Отзыв привилегий

Для отмены привилегий, предоставленных пользователям посредством оператора GRANT, используется оператор REVOKE. С помощью этого оператора могут быть отменены все или некоторые из привилегий, полученных указанным пользователем раньше. Оператор REVOKE имеет следующий формат :

```
REVOKE [GRANT OPTION FOR] {привилегия на объект [,…] | имя роли [,…]}
ON имя объекта
FROM {получатель привилегии [,…]}
[RESTRICT | CASCADE]
```

# Отзыв привилегий

| Опция | Описание |
|---|---|
| `ALL PRIVILEGIES` | Отзываются все привилегии, предоставленные ему ранее тем пользователем, который ввел данный оператор |
| `GRANT OPTION FOR` | Отзывается только право передачи привилегии, но не сама привилегия. Без этого указания отзывается и привилегия, и право назначать привилегии. |
| `CASCADE` | Отзыв привилегий не только непосредственно у указанного пользователя, но и у всех пользователей, которым он выдавал привилегии используя опцию `GRANT OPTION`. |
| `RESTRICT` | Отзыв привилегий касается только непосредственно пользователя, указанного в операторе `REVOKE`, но при наличии у этого пользователя делегированных с помощью опции `GRANT OPTION` привилегий возникнет ошибка. |

# Коллизии

| user1 | user2 | user3 | user4 | user5 |
|---|---|---|---|---|
| `GRANT SELECT ON orders TO user2 WITH GRANT OPTION;` | Получение права от `user1` | | | |
| | `GRANT SELECT ON orders TO user3 WITH GRANT OPTION;` | Получение права от `user2`. Получение права от `user5`. | | `GRANT SELECT ON orders TO user3 WITH GRANT OPTION;` |
| | | `GRANT SELECT ON orders TO user4;` | Получение права от `user3`. | |
| `REVOKE SELECT ON orders FROM user2 CASCADE` | Отмена права | Сохранение права | Сохранение права | Сохранение права |

# Конфликты доступа

Разрешения, предоставленные роли или группе, наследуются их членами.

Хотя пользователю может быть предоставлен доступ через членство в одной роли, роль другого уровня может иметь запрещение на действие с объектом. В таком случае возникает конфликт доступа.

При разрешении конфликтов доступа руководствуются следующим принципом: разрешение на предоставление доступа имеет самый низкий приоритет, а на запрещение доступа – самый высокий. Это значит, что доступ к данным может быть получен только явным его предоставлением при отсутствии запрещения доступа на любом другом уровне иерархии системы безопасности.

Если доступ явно не предоставлен, пользователь не сможет работать с данными.

САМАРСКИЙ УНИВЕРСИТЕТ
SAMARA UNIVERSITY

# БЛАГОДАРЮ ЗА ВНИМАНИЕ

Агафонов А.А.
к.т.н., доцент кафедры ГИиИБ