Практическое занятие 3. Авторизация пользователей для доступа к ресурсам БД

1. В чем разница между командами REVOKE и DENY?

Команды REVOKE и DENY предназначены для управления правами доступа в базах данных, но действуют по-разному:

-REVOKE удаляет явно присвоенную привилегию. Она отменяет предыдущую выдачу права, возвращая ситуацию к состоянию, когда оно не было выдано вообще. Другими словами, команда снимает существующие привилегии и возвращает состояние доступа к настройкам по умолчанию (наследование родительских прав).

-DENY блокирует возможность доступа, независимо от наследуемых прав. Даже если привилегия была предоставлена вышестоящей группой или унаследована, команда DENY переопределяет её и полностью запрещает доступ.

2. Как реализовать принцип минимальных привилегий на практике?

1. Назначать права постепенно:

2.Использовать роли: Создавать роли с набором необходимых привилегий и присваивать их пользователям. Это упрощает управление правами и снижает вероятность ошибочного назначения больших полномочий.

3.Проверять специфику: Действительно ли необходимы запрашиваемые привилегии, изучить, каким минимальным уровнем доступа можно обойтись.

4.Мониторинг и аудит: Периодически проводить ревизию назначенных прав,

5.Обучение пользователей: Объяснять сотрудникам принципы безопасной работы с системами и правилами назначения прав.

3. Какие преимущества предоставляют роли по сравнению с назначением прав отдельным пользователям?

Использование ролей вместо индивидуального назначения прав отдельным пользователям предоставляет ряд важных преимуществ:

- Упрощённое управление

Настройка ролей позволяет собрать пользователей с одинаковыми обязанностями в единую категорию. Достаточно задать права доступа для роли, и они автоматически распространяются на всех её участников. Такой подход значительно сокращает объём рутинных операций по назначению прав.

- Централизованное обновление

Изменяя права доступа для роли, вы одновременно вносите изменения для всех её членов. Если изменилась политика доступа, вам не придётся менять права каждого пользователя по отдельности — достаточно обновить роль.

- Четкость и структурированность

Роли позволяют разделять пользователей на группы по функционалу и ответственности. Такое разделение повышает прозрачность системы и снижает риск ошибок, вызванных неправильным назначением прав.

- Масштабируемость

По мере роста числа пользователей внедрение ролей упрощает администрирование. Вместо сложного отслеживания прав сотен индивидуальных пользователей вы можете управлять гораздо меньшим количеством ролей, что экономит время и ресурсы.

- Повышение безопасности

Поскольку роли облегчают контроль над привилегиями, вы сможете лучше соблюдать принцип минимальных привилегий, выдавая только необходимые права и предотвращая излишнюю экспансию полномочий.

Таким образом, применение ролей существенно упрощает управление доступом, повышает безопасность и надёжность системы, делая её удобной для эксплуатации и модернизации.

4. Когда следует использовать разрешения на уровне столбцов?

когда необходимо предоставить доступ только к определённой части данных в таблице. Такие случаи возникают чаще всего, когда:

-Таблица содержит смешанные типы данных (например, публичные и приватные данные).

-Нужно защитить конфиденциальные данные, сохраняя доступ к остальным столбцам.

-Запросы касаются только части полей таблицы, и предоставление полного доступа нежелательно по соображениям безопасности.

Примером может служить таблица сотрудников, содержащая как общедоступные поля (ФИО, должность), так и закрытые (зарплата, персональные номера). Предоставление доступа на уровне столбцов позволяет разделить права чтения и модификации между группами пользователей.

5. Как организовать безопасный доступ к чувствительным данным?

-Использовать принцип минимальных привилегий: Назначать пользователям только те права доступа, которые необходимы для выполнения их задач.

-Применять шифрование: Храним конфиденциальные данные в зашифрованном виде и используем защищённые соединения (SSL/TLS) для передачи данных.

-Осуществлять контроль доступа на уровне столбцов: Используем механизмы маскировки данных (data masking) и ограничения доступа к отдельным столбцам, чтобы обезопасить частную информацию.

-Организовать аудит и ведение журналов доступа: Вести журналы попыток доступа к важным данным и следить за попытками их просмотра или изменения.

-Разработать политики хранения и удаления данных: Определить строгие правила, регламентирующие хранение и уничтожение данных, чтобы минимизировать риск разглашения устаревшей информации.

-Выполнять автоматизированные проверки и тестирование на предмет уязвимостей: Регулярно тестировать систему на безопасность и оценивать защиту базы данных с целью выявления слабых мест и своевременного их устранения.