

# Лекция 8

## Безопасность систем баз данных

# План лекции

- Основные термины
- Специфика БД
- Модели управления доступом
- Репликация БД
- Секционирование и сегментирование
- Аудит событий безопасности БД

# Понятие защищенной базы данных

**Безопасность данных** – это состояние защищенности, при котором обеспечиваются конфиденциальность, доступность и целостность данных:

- конфиденциальность отвечает за обеспечение доступа к данным, только санкционированным пользователями;
- целостность исключает несанкционированное изменение структуры и содержания данных;
- доступность позволяет обеспечить доступ к данным, санкционированным пользователями, по их первому требованию.

# Конфиденциальность информации

– необходимость предотвращения разглашения или утечки какой-либо информации.

Конфиденциальность информации достигается предоставлением к ней доступа с наименьшими привилегиями исходя из принципа минимальной необходимой осведомлённости.

Меры обеспечения конфиденциальности:

- классификация информации, предназначенной для публичного или внутреннего пользования
- шифрование информации

# Целостность информации

– термин, означающий, что данные не были изменены при выполнении какой-либо операции над ними, будь то передача, хранение или отображение.

Меры обеспечения:

- ограничение круга лиц с правами на изменения лишь тем, кому такой доступ необходим для выполнения служебных обязанностей
- применение транзакций

# Доступность информации

– состояние информации, при котором субъекты, имеющие права доступа, могут реализовывать их беспрепятственно.

Основными факторами, влияющими на доступность информационных систем, являются DoS-атаки, атаки программ-вымогателей, саботаж. Кроме того, источником угроз доступности являются непреднамеренные человеческие ошибки, отказ в обслуживании в результате превышения допустимой мощности или недостатка ресурсов оборудования и другие факторы.

# Угрозы безопасности баз данных

- несанкционированный доступ к данным за счет уязвимостей в клиентских приложениях БД
  - решение: администрирование прав доступа
- потеря данных вследствие аппаратных или программных сбоев серверов БД случайного или преднамеренного характера
  - решение: резервное копирование данных
- остановка или значительное снижение производительности сервера БД, вызванное большим количеством активных пользователей или преднамеренными атаками
  - решение: репликация данных, масштабирование БД
- снижение производительности сервера БД, вызванное преднамеренными действиями уполномоченных пользователей
  - решение: средства мониторинга и протоколирования событий
- беспрепятственный доступ к данным в случае успешной атаки или хищения
  - решение: шифрование критических данных

# Администрирование СУБД

Многие вопросы обеспечения безопасности решаются путем администрирования СУБД, в частности:

- аутентификация и авторизация пользователя
- криптографическая защита БД
- резервное копирование данных
- репликация и балансировка нагрузки
- аудит событий безопасности БД
- модернизация системного и прикладного ПО



# Аутентификация и авторизация

Любой пользователь, получающий доступ к БД, на этапе создания пользовательской сессии подлежит обязательной идентификации. Все дальнейшие его действия так или иначе будут требовать предъявления этого идентификатора.

**Аутентификация** — это процедура проверки подлинности пользователя. Обычно пользователь подтверждает то, что он является именно тем, за кого он себя выдает, путем ввода в систему уникальной (неизвестной другим) информации о себе (пароль, биометрия, смарт-карты и т.д.)

# Аутентификация и авторизация

Если пользователь успешно прошел аутентификацию, СУБД осуществляет его авторизацию. **Авторизация** – это процедура предоставления пользователю определенных ресурсов и прав на их использование. Всё дальнейшее взаимодействие пользователя с объектами БД строго регламентируется в соответствии с назначенными правами.

# Криптографическая защита

**Шифрование** – обратимое преобразование информации в целях сокрытия от неавторизованных лиц с предоставлением в это же время авторизованным пользователям доступа к ней.

Защищенная СУБД должна уметь шифровать: хранящиеся в ней данные (включая служебную информацию), исходный код запросов, хранимых процедур и триггеров, данные, передаваемые к другим компьютерам по незащищенным каналам.

# Репликация баз данных

— это копирование изменений данных с главного сервера баз данных на одном или нескольких зависимых серверах.

Репликация помогает:

- масштабироваться и справляться с нагрузкой, вызываемой одновременными операциями чтения и записи в БД
- обеспечить отказоустойчивость систем; в случае отказа реплики все запросы чтения можно перевести на главный сервер
- удобно резервировать данные
- использовать отложенные вычисления — неповоротливые SQL-запросы (анализ данных) можно выполнять на отдельной реплике, не боясь помешать оптимальной работе всей системы

# Аудит событий безопасности

– процесс получения и анализа данных о происходящих в системе событиях и степени их соответствия требованиям к защите данных. Для этого многие СУБД автоматически ведут журнал аудита. В нём содержится описание набора событий (авторизации пользователя, доступа к тем или иным данным и операций с ними; создания, модификации и уничтожения объектов БД; выполнение нестандартных SQL-команд и т.д.)

Журнал аудита сам по себе должен быть защищен от несанкционированного доступа.

# Модель управления доступом

Целью управления доступом является ограничение действий или операций, которые может выполнять легальный пользователь информационной системы.

Модель управления доступом определяет порядок доступа субъектов к объектам.

Три основных модели управления доступом:

- дискреционная (избирательная)
- мандатная
- ролевая

# Дискреционная модель

– управление доступом субъектов к объектам на основе списков управления доступом или матрицы доступа.

Субъект с определенным правом доступа может передать это право любому другому субъекту.

Для каждой пары (субъект – объект) должно быть задано явное и недвусмысленное перечисление допустимых типов доступа.

	Объект №1	Объект №2	Объект №3	...
Пользователь #1	READ, WRITE	READ, WRITE	READ	...
Пользователь #2	READ, WRITE	READ	-	...
Пользователь #3	READ	READ	-	...
...	...	...	...	...

# Дискреционная модель

- 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 (администратор). Это нужно для того, чтобы исключить возможность недоступных объектов.



# Дискреционная модель

## Достоинства:

- простота реализации
- гибкость (пользователь может описать доступ к своим ресурсам)

## Недостатки:

- сложность администрирования

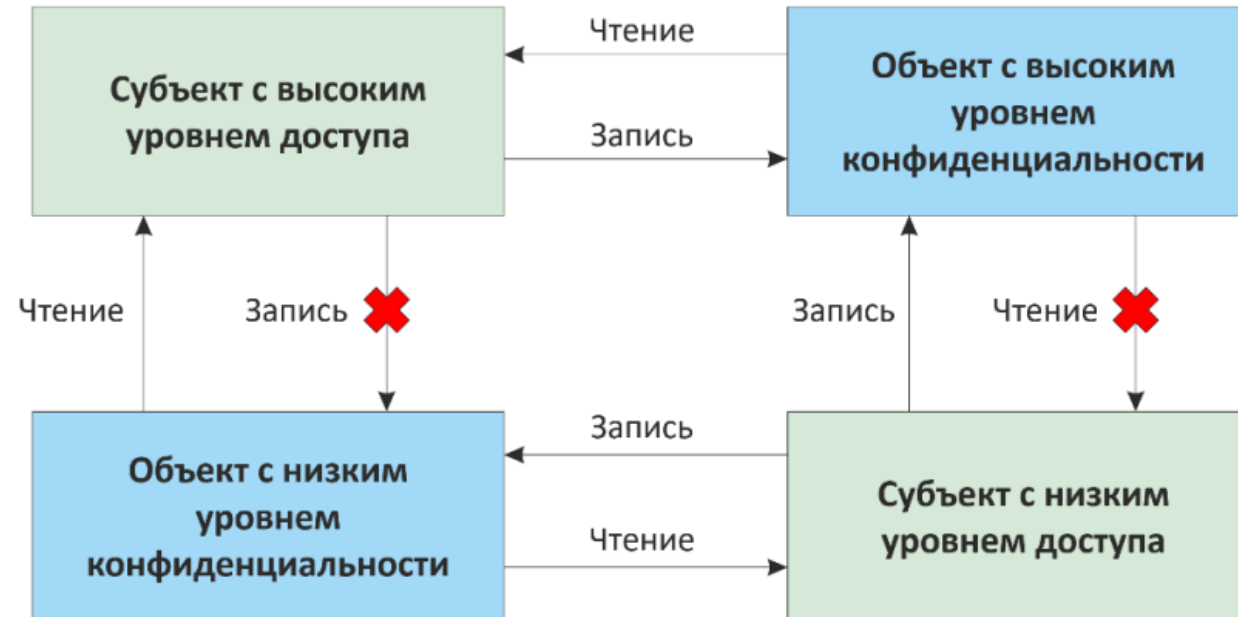
# Мандатное управление доступом

Разграничение доступа субъектов к объектам, основанное на назначении метки (мандата) конфиденциальности для информации, содержащейся в объектах, и выдаче официальных разрешений (допуска) субъектам на обращение к информации такого уровня конфиденциальности.

Метка конфиденциальности – это элемент иерархически упорядоченного набора (unclassified<confidential<secret<topsecret)

# Мандатное управление доступом

- субъект может читать только те объекты, класс доступа которых равен, либо меньше его класса доступа
- субъект может записывать только те объекты, класс доступа которых равен либо выше его класса доступа



# Мандатное управление доступом

Достоинства:

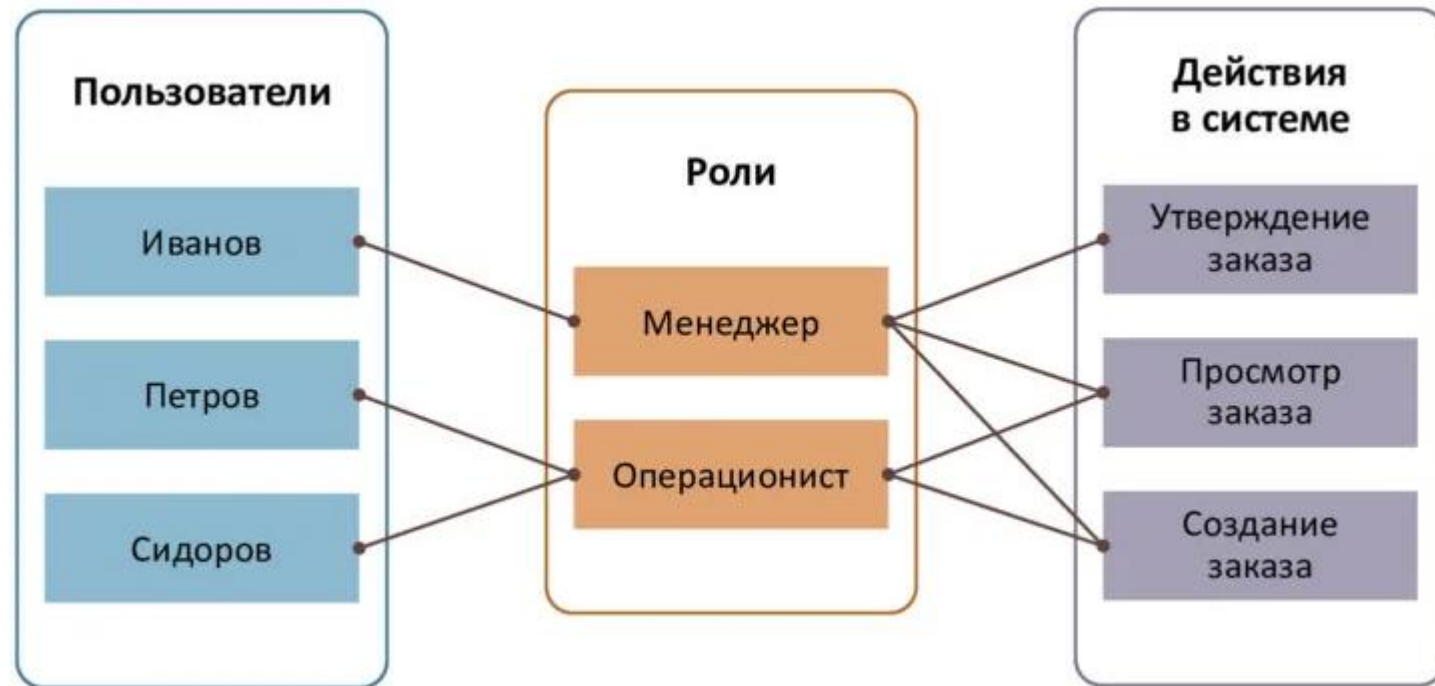
- упрощение администрирования

Недостатки:

- проблема разграничения пользователей одного уровня
- пользователь не может назначать доступ к объекту

# Ролевая модель

– развитие дискреционной модели, где права доступа субъектов системы группируются с учётом специфики их применения, образуя роли. Ролью называется именованная совокупность привилегий, которые могут быть предоставлены пользователям или другим ролям.



# Ролевая модель

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

- 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

## Достоинства:

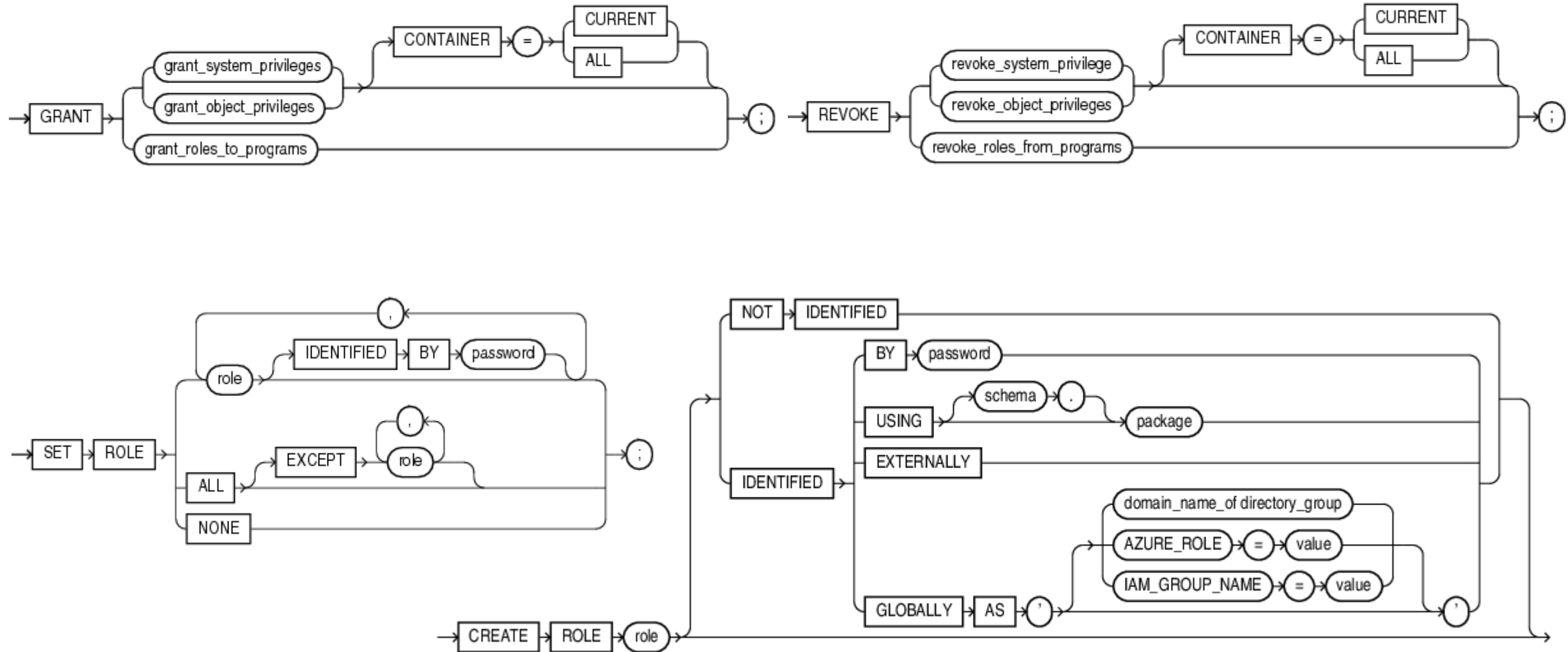
- гибкость правил разграничения
- простота добавления нового пользователя

## Недостатки:

- сложность конструирования ролей
- возможность внесения дублирования и избыточности при предоставлении пользователям прав доступа

В обычных версиях СУБД 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.

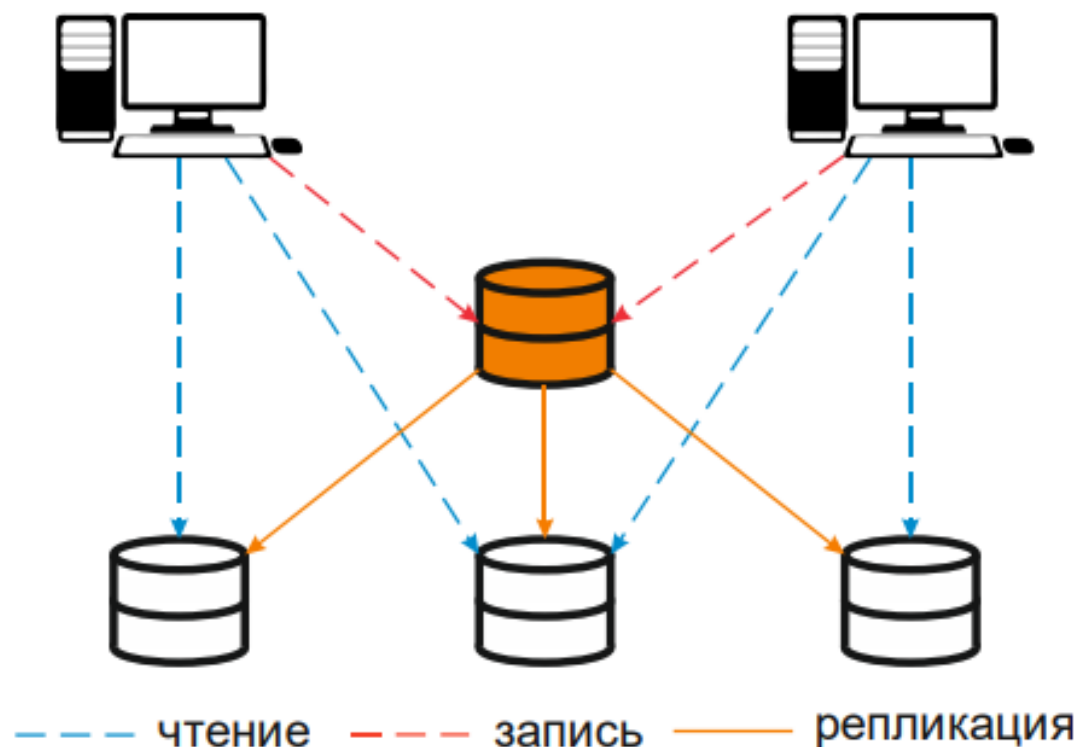
# Реализация в SQL





# Типы репликации БД

В контексте репликации у серверов могут быть две роли: ведущий сервер и подчиненный (ведомый) сервер. При репликации данные копируются с ведущего сервера на подчиненные серверы.



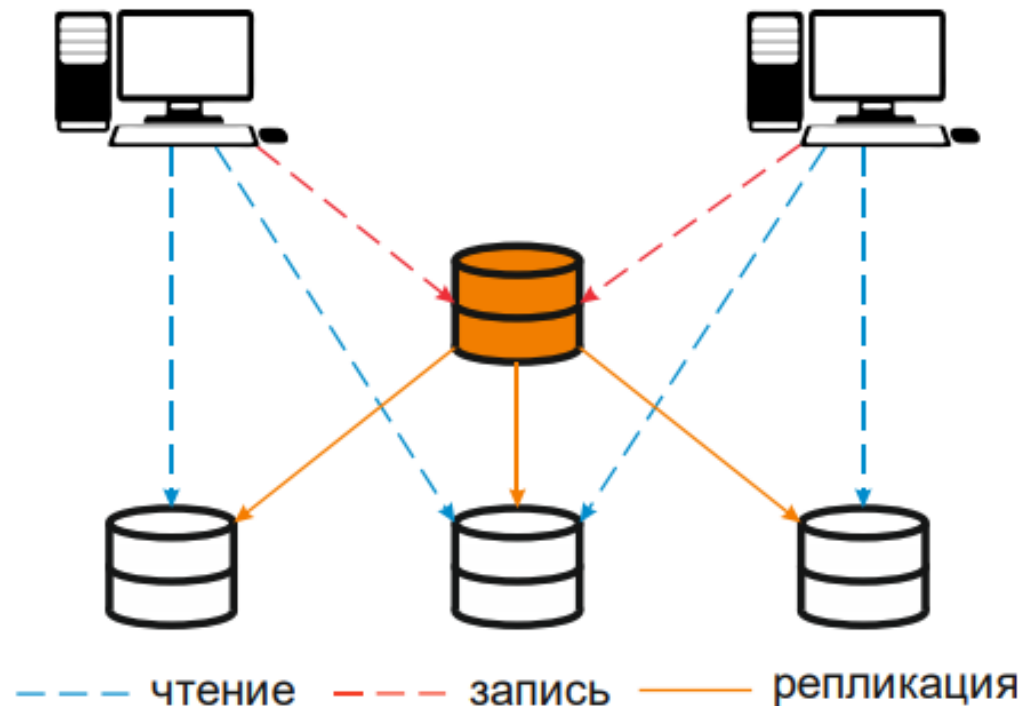
# Типы репликации БД

Количество серверов с соответствующими ролями определяет различные виды топологии репликации:

- репликация с одним ведущим сервером; данные всегда отправляются на один конкретный узел
- репликация с несколькими ведущими серверами; может существовать несколько узлов, играющих роль ведущих, и каждый ведущий узел должен сохранять данные в кластере
- репликация без ведущих серверов; все узлы могут принимать данные при записи

# Репликация с одним ведущим сервером

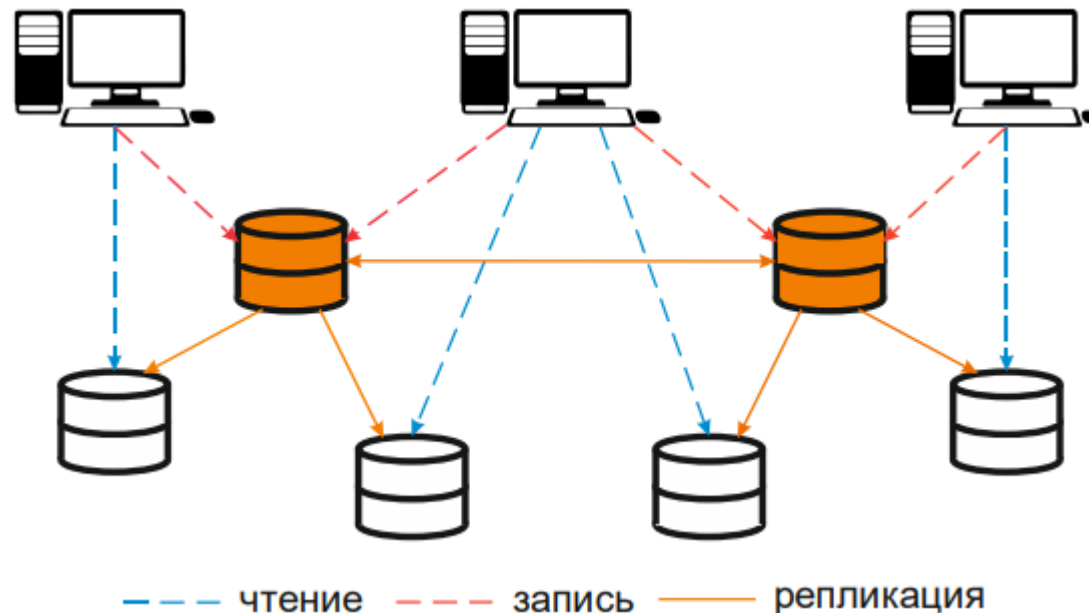
- наиболее распространенная топология репликации
- достоинство – один ведущий сервер
- недостаток – один ведущий сервер



# Репликация с несколькими ведущими серверами

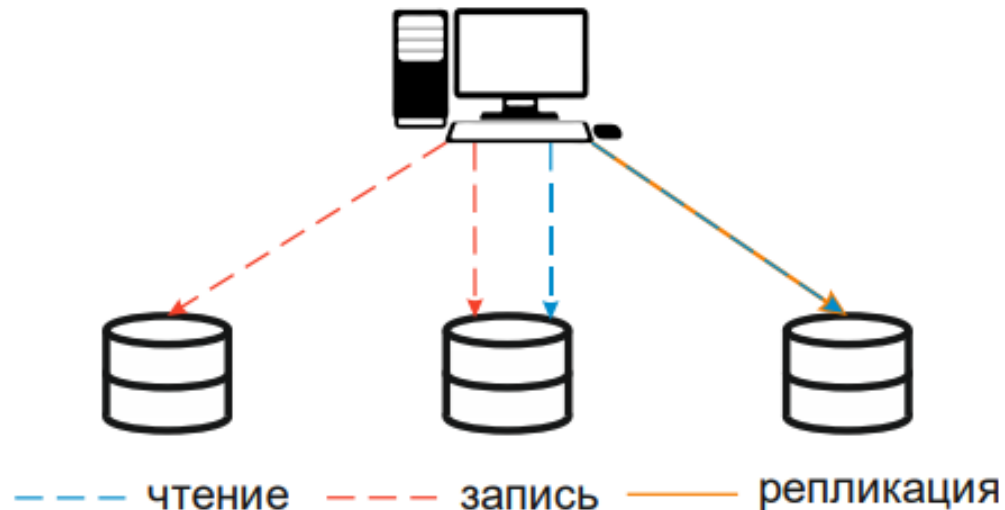
Основная проблема подхода в возникновении конфликтов изменения данных и необходимости их разрешения

- см. проблему конфликта транзакций, решения те же:
  - временные метки
  - синхронная репликация: транзакция, модифицирующая данные, считается подтверждённой только тогда, когда все серверы подтвердят эту транзакцию.



# Репликация без ведущих серверов

При записи клиент посылает запрос на запись одновременно нескольким репликам, и как только он получает подтверждение от некоторых из них (в предельном случае от одной), считается, что запись прошла успешно и клиент может продолжать работу.



# Репликация без ведущих серверов

В отсутствии ведущего сервера, обеспечивающего синхронизацию, проблема согласования данных решается следующим образом:

- 1) запросы на чтение идут на несколько реплик сразу;
- 2) реплики возвращают свои локальные значения и некоторый номер версии данных, чтобы решить, какое из полученных значений является актуальным;
- 3) одновременно выполняется и синхронизация значений между репликами: если при чтении данных обнаруживается (по номеру версии), что на некотором узле значение не актуально, клиент посылает этому узлу запрос на запись с актуальным значением.

Такой механизм называют чтением с восстановлением (Read Repair).

# Секционирование и сегментирование

Если проблема заключается не в количестве одновременных запросов, а в размере базы данных и скорости выполнения одного запроса, необходим подход разделения данных.

Выделяют два подхода к разделению данных:

- секционирование (partitioning) – это разделение хранимых объектов баз данных (таких как таблицы, индексы, материализованные представления) на отдельные части с отдельными параметрами физического хранения. Используется в целях повышения управляемости, производительности и доступности для больших баз данных

# Секционирование и сегментирование

Если проблема заключается не в количестве одновременных запросов, а в размере базы данных и скорости выполнения одного запроса, необходим подход разделения данных.

Выделяют два подхода к разделению данных:

- сегментирование (sharding) – подход, предполагающий разделение баз данных, отдельных её объектов или индексов поисковых систем на независимые сегменты, каждый из которых управляется отдельным экземпляром сервера базы данных, размещаемым, как правило, на отдельном вычислительном узле



# Секционирование


Можно отметить следующие особенности применения секционирования таблиц:

- секционирование позволяет хранить в одной таблице больше данных, чем может храниться на одном диске или разделе файловой системы
- в определённых ситуациях секционирование значительно увеличивает быстродействие, особенно когда большой процент часто запрашиваемых строк относится к одной секции
- редко используемые данные можно перенести на более дешёвые(медленные) носители

id	Имя	Отдел
0001	Иванов А.В.	11
0002	Сидоров Н.Е.	23
0003	Петров Н.З.	23
0004	Лебедев А.Н.	44
0005	Дроздов И.Ю.	11
0006	Воронцов М.И.	23



id	Имя	Отдел
0001	Иванов А.В.	11
0002	Сидоров Н.Е.	23
0003	Петров Н.З.	23



id	Имя	Отдел
0004	Лебедев А.Н.	44
0005	Дроздов И.Ю.	11
0006	Воронцов М.И.	23

# Сегментирование (шардирование)

Метод улучшает время отклика на запрос за счет разделения данных на более мелкие группы, позволяет делать параллельно больше работы одновременно.

Недостатки метода: проблемы при объединении, поддержание ссылочной целостности – как следствие, не все движки баз данных обеспечивают встроенную поддержку метода.

# Секционирование и сегментирование

	Сегментирование	Секционирование
Распределение данных	Для нескольких экземпляров базы данных (сегментов).	В пределах одного экземпляра базы данных (секций).
Масштабируемость	Отличная горизонтальная масштабируемость.	Ограничена емкостью одной базы данных.
Производительность запросов	Высокая производительность благодаря параллельной обработке.	Улучшена производительность для целенаправленных запросов.
Техническое обслуживание	Комплексное управление распределенными системами.	Эффективное управление данными в рамках одной базы данных.
Операции объединения	Может быть сложным и медленным в разных сегментах.	Обычно объединения внутри раздела выполняются проще.
Согласованность данных	Проблемы с поддержанием согласованности.	Управление согласованностью более простое.
Метод	Могут происходить по объектам, интервалам, по списку значений или по хешу	

# Аудит событий безопасности БД

– процесс получения и анализа данных о происходящих в системе событиях и степени их соответствия требованиям к защите данных.

События записываются в журналы событий или файлы аудита. Когда аудит активен, каждая контролируемая операция с базой данных порождает аудиторский след с информацией о том, какой объект базы данных был задействован, а также кто и когда выполнял эту операцию.

Аудиторский след, накопленный в течение определенного отрезка времени, позволяет строить модели типичного поведения, которые используются для автоматического выявления нетипичного поведения, основываясь на статистическом анализе или на методах машинного обучения.

# Журнал аудита

В журнале содержится:

- описание стандартного набора событий (авторизации пользователя, доступа к тем или иным данным и операций с ними; создания, модификации и удаления объектов БД; выполнение SQL-команд и т.д.);
- настраиваемый перечень атрибутов в отдельной записи журнала аудита (даты и времени события, идентификатор пользователя, имя и сетевой адрес компьютера, описание события, связанные с событием объекты, признак успешного или неудачного завершения события).

Журнал аудита сам по себе должен быть защищен от несанкционированного доступа.

# Методы аудита

Существует несколько подходов, которые в той или иной степени позволяют решать задачи аудита:

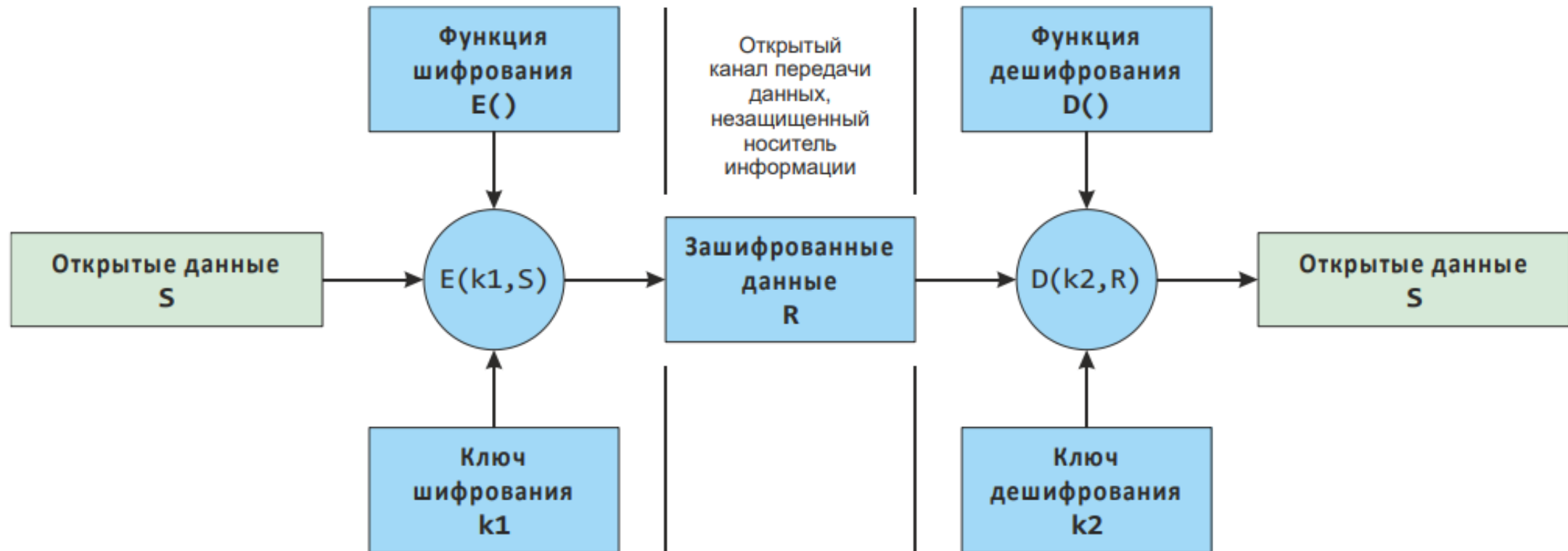
- трассировка(все события, связанные с контролируруемыми объектами);
- анализ журналов транзакций;
- аудиторские следы в данных;
- мониторинг сетевого трафика сервера БД;
- мониторинг сервера БД.

# Шифрование

Шифрование базы данных – использование технологии шифрования для преобразования информации, хранящейся в базе данных, в шифротекст, что делает её прочтение невозможным для лиц, не обладающих ключами шифрования.

Кроме того, необходимо шифровать обмен данными между БД и клиентами.

# Шифрование





# Подходы в шифровании

- шифрование на уровне хранилища
  - данные шифруются перед записью на диск и дешифруются во время чтения в память
  - не обеспечивает сохранность информации при передаче
- шифрование на уровне базы данных
  - для шифрования значений отдельных столбцов
- шифрование на уровне приложения
  - шифрование данных происходит перед их записью в базу
    - обеспечена защиты данных при передаче