**4 ЛР. Разграничение доступа**

**4LR. Kiểm soát truy cập**

1. **Модели управления доступом. Способы реализации контроля доступа к БД.**

**Các mô hình kiểm soát truy cập Các phương pháp thực hiện kiểm soát truy cập cơ sở dữ liệu.**

Модели управления доступом к базам данных (БД) представляют собой концептуальные схемы, которые определяют правила, какие объекты пользователь может просматривать, изменять или удалять в БД. Существуют различные модели управления доступом, некоторые из них:

Các mô hình kiểm soát truy cập cơ sở dữ liệu là các khung khái niệm xác định các quy tắc mà người dùng có thể xem, sửa đổi hoặc xóa trong cơ sở dữ liệu. Có nhiều mô hình kiểm soát truy cập khác nhau, một số trong số đó là:

***Модель дискреционного контроля доступа*** *(DAC)* - более старая модель, в которой пользователь сам устанавливает права доступа для других пользователей или групп пользователей.

**Mô hình Kiểm soát truy cập tùy ý** (DAC) là mô hình cũ hơn trong đó người dùng đặt quyền truy cập cho người dùng hoặc nhóm người dùng khác.

***Модель обязательного контроля доступа*** *(MAC)* - порядок доступа к данным устанавливается только администратором на основании политики безопасности, касающейся конфиденциальности и защиты данных.

**Mô hình Kiểm soát truy cập bắt buộc (MAC) -** quyền truy cập dữ liệu chỉ được quản trị viên đặt dựa trên các chính sách bảo mật liên quan đến bảo mật và bảo vệ dữ liệu.

***Модель ролевого контроля доступа (RBAC)*** - модель, в которой права доступа к данным привязаны к определенным ролям пользователей или группам пользователей.

**Mô hình kiểm soát truy cập dựa trên vai trò (RBAC**) là mô hình trong đó quyền truy cập vào dữ liệu được gắn với các vai trò người dùng hoặc nhóm người dùng cụ thể.

***Модель контроля доступа на основе атрибутов (ABAC)*** - модель, в которой права доступа к данным определяются на основе атрибутов объектов или пользователей.

**Mô hình kiểm soát truy cập dựa trên thuộc tính (ABAC)** là mô hình trong đó quyền truy cập dữ liệu được xác định dựa trên thuộc tính của đối tượng hoặc người dùng.

Способы реализации контроля доступа к БД могут быть основаны на моделях управления доступом и могут включать в себя:

Các phương pháp triển khai kiểm soát truy cập cơ sở dữ liệu có thể dựa trên các mô hình kiểm soát truy cập và có thể bao gồm:

- Установку ограничений на доступ к объектам в базе данных, таких как таблицы, представления и процедуры.

- Đặt các hạn chế về quyền truy cập vào các đối tượng trong cơ sở dữ liệu, chẳng hạn như bảng, dạng xem và thủ tục.

- Определение прав доступа для отдельных пользователей или групп пользователей.

- Xác định quyền truy cập cho người dùng cá nhân hoặc nhóm người dùng.

- Шифрование данных для защиты от несанкционированного доступа.

- Mã hóa dữ liệu để bảo vệ chống truy cập trái phép.

- Аудит доступа для отслеживания доступа к данным и обнаружения нарушений безопасности.

- Kiểm tra truy cập để giám sát việc truy cập dữ liệu và phát hiện các vi phạm bảo mật.

- Использование программного обеспечения для автоматического контроля доступа к данным, в том числе программ для протоколирования и мониторинга доступа к БД.

- Sử dụng phần mềm kiểm soát tự động việc truy cập dữ liệu, bao gồm các chương trình ghi nhật ký và giám sát việc truy cập cơ sở dữ liệu.

1. **Синтаксис по разграничению доступа к объектам с помощью команды grant. К каким объектам БД может быть ограничен доступ? Виды привилегий**

**Cú pháp hạn chế quyền truy cập vào các đối tượng bằng lệnh cấp. Những đối tượng cơ sở dữ liệu nào có thể bị hạn chế quyền truy cập? Các loại đặc quyền**

Команда GRANT используется в системе управления базами данных (СУБД) для назначения определенных привилегий пользователям или пользователям-группам. Синтаксис команды GRANT может быть следующим:

**Lệnh GRAN**T được sử dụng trong hệ thống quản lý cơ sở dữ liệu (DBMS) để gán các đặc quyền cụ thể cho người dùng hoặc nhóm người dùng. Cú pháp lệnh GRANT có thể như sau:

*GRANT* ***permission\_list*** *ON* ***object\_name*** *TO* ***role\_specification*** *[WITH GRANT OPTION];*

*где:*

* *‘permission\_list’ - список разрешений, которые необходимо предоставить (например, SELECT, INSERT, UPDATE, DELETE);*
* *‘object\_name’ - имя объекта базы данных (например, таблица, представление, процедура);*
* *‘role\_specification’ can be:*

*Text

Description automatically generated*

* *‘WITH GRANT OPTION’ - опция, которая позволяет пользователю, получившему разрешение на доступ к объекту, предоставлять доступ к этому объекту другим пользователям.*

*Một tùy chọn cho phép người dùng có quyền truy cập vào một đối tượng có thể cấp quyền truy cập vào đối tượng đó cho những người dùng khác.*

Пример команды, предоставляющей разрешение на чтение таблицы "users" пользователю "user1":

*Một lệnh ví dụ cấp quyền đọc bảng "users" cho người dùng "user1":*

*GRANT SELECT ON users TO user1;*

В данном примере группе пользователей Role1 назначено право на выполнения хранимой процедуры spTest:

*GRANT EXECUTE ON dbo.spTest TO Role1;*

В данном примере всем пользователям базы данных назначено право на выполнение операции UPDATE для таблицы Product:

*GRANT UPDATE ON Product TO PUBLIC;*

В реляционных базах данных доступ может быть ограничен к следующим объектам:

Trong cơ sở dữ liệu quan hệ, quyền truy cập có thể bị giới hạn ở các đối tượng sau:

* Таблицы: доступ к чтению, записи, удалению или изменению данных в таблицах может быть ограничен.
* Bảng: Quyền truy cập vào việc đọc, viết, xóa hoặc thay đổi dữ liệu trong bảng có thể bị hạn chế.
* Представления: доступ к чтению или изменению представлений может быть ограничен.
* Lượt xem: Quyền truy cập vào chế độ xem đọc hoặc chỉnh sửa có thể bị hạn chế.
* Хранимые процедуры: доступ к выполнению хранимых процедур может быть ограничен.
* Thủ tục lưu trữ: Quyền truy cập để thực hiện các thủ tục được lưu trữ có thể bị hạn chế.
* Индексы: доступ к индексам может быть ограничен.
* Chỉ mục: Việc truy cập vào chỉ mục có thể bị hạn chế.
* Секреты: доступ к конфиденциальным данным о пользователях или подсистемах может быть ограничен.
* Bí mật: Quyền truy cập vào dữ liệu nhạy cảm về người dùng hoặc hệ thống con có thể bị hạn chế.
* Роли: управление правами доступа в зависимости от роли пользователя.
* Vai trò: Quản lý quyền truy cập dựa trên vai trò của người dùng.

Привилегии включают следующие виды:

Đặc quyền bao gồm các loại sau:

* SELECT: разрешает выборку данных из объектов БД.
* SELECT: cho phép lựa chọn dữ liệu từ các đối tượng cơ sở dữ liệu.
* INSERT: разрешает добавление новых записей в таблицу.
* INSERT: Cho phép thêm bản ghi mới vào bảng.
* UPDATE: разрешает обновление записей в таблице.
* UPDATE: Cho phép cập nhật các bản ghi trong bảng.
* DELETE: разрешает удаление записей из таблицы.
* DELETE: Cho phép xóa bản ghi khỏi bảng.
* EXECUTE: разрешает выполнение хранимых процедур.
* EXECUTE: Cho phép thực hiện các thủ tục được lưu trữ.
* REFERENCES: разрешает ссылаться на столбцы таблицы.
* TÀI LIỆU THAM KHẢO: Cho phép tham chiếu các cột trong bảng.
* ALTER: разрешает изменять структуру объекта в БД.
* ALTER: cho phép bạn thay đổi cấu trúc của một đối tượng trong cơ sở dữ liệu.
* CREATE: разрешает создавать новые объекты в БД.
* TẠO: cho phép bạn tạo các đối tượng mới trong cơ sở dữ liệu.
* DROP: разрешает удалять объекты из БД.
* DROP: cho phép bạn xóa các đối tượng khỏi cơ sở dữ liệu.

Эти привилегии можно назначить пользователям, группам или ролям в БД для ограничения доступа к объектам.

Những đặc quyền này có thể được gán cho người dùng, nhóm hoặc vai trò trong cơ sở dữ liệu để hạn chế quyền truy cập vào các đối tượng.

<https://www.postgresql.org/docs/current/sql-grant.html>

1. **Ролевая модель доступа. Атрибуты роли**.

Mô hình truy cập dựa trên vai trò Thuộc tính vai trò.

В PostgreSQL пользователи и группы – это роли. Одна роль может быть членом другой роли. Роли в PostgreSQL не имеют связи с пользователями в операционной системе. Роли это глобальные объекты для всего кластера баз данных.

Trong PostgreSQL, người dùng và nhóm là các vai trò. Một vai trò có thể là thành viên của một vai trò khác. Vai trò trong PostgreSQL không có kết nối với người dùng trong hệ điều hành. Vai trò là các đối tượng chung cho toàn bộ cụm cơ sở dữ liệu.

Ролевая модель доступа (RBAC - Role-Based Access Control) - это способ управления доступом к данным в базах данных, основанный на ролях, которые подразумевают набор разрешений, определяющих доступ к объектам базы данных.

Mô hình truy cập dựa trên vai trò (RBAC - Kiểm soát truy cập dựa trên vai trò) là một cách kiểm soát quyền truy cập vào dữ liệu trong cơ sở dữ liệu, dựa trên vai trò, bao hàm một tập hợp các quyền xác định quyền truy cập vào các đối tượng cơ sở dữ liệu.

Каждый пользователь базы данных имеет определенную роль, которая определяет его права доступа к объектам базы данных (таблицам, представлениям, процедурам и т.д.). Каждый объект базы данных имеет определенную роль, которая определяет, какие пользователи могут получить к нему доступ.

Mỗi người dùng cơ sở dữ liệu có một vai trò cụ thể xác định quyền truy cập của mình vào các đối tượng cơ sở dữ liệu (bảng, dạng xem, thủ tục, v.v.). Mỗi đối tượng cơ sở dữ liệu có một vai trò cụ thể xác định người dùng nào có thể truy cập nó.

RBAC позволяет управлять доступом к базам данных с помощью группировки пользователей в роли и назначения прав доступа этим ролям, а не отдельным пользователям. Это упрощает управление доступом и обеспечивает более гибкую систему доступа к данным.

RBAC cho phép bạn kiểm soát quyền truy cập vào cơ sở dữ liệu bằng cách nhóm người dùng thành các vai trò và gán quyền truy cập cho các vai trò đó thay vì cho từng người dùng. Điều này giúp đơn giản hóa việc kiểm soát truy cập và cung cấp khả năng truy cập dữ liệu linh hoạt hơn.

У ролей есть несколько атрибутов. Эти атрибуты указываются при создании роли:

Vai trò có một số thuộc tính. Các thuộc tính này được chỉ định khi tạo vai trò:

<https://postgrespro.ru/docs/postgresql/9.6/role-attributes>

* *Право подключения*. Только роли с атрибутом LOGIN могут использоваться для начального подключения к базе данных. Для создания такой роли можно использовать любой из вариантов:
* Quyền kết nối. Chỉ những vai trò có thuộc tính LOGIN mới có thể được sử dụng cho kết nối ban đầu với cơ sở dữ liệu. Để tạo vai trò như vậy, bạn có thể sử dụng bất kỳ tùy chọn nào:



Команда CREATE USER эквивалентна CREATE ROLE за исключением того, что CREATE USER по умолчанию предполагает атрибут LOGIN, в то время как CREATE ROLE нет.

Lệnh CREATE USER tương đương với CREATE ROLE ngoại trừ việc CREATE USER giả định thuộc tính LOGIN theo mặc định, trong khi CREATE ROLE thì không.

* *Статус суперпользователя*. Суперпользователь базы данных обходит все проверки прав доступа, за исключением права на вход в систему. Это опасная привилегия и она не должна использоваться небрежно. Лучше всего выполнять большую часть работы не как суперпользователь. Для создания нового суперпользователя используется ***CREATE ROLE имя SUPERUSER***. Это нужно выполнить из под роли, которая также является суперпользователем.
* Trạng thái siêu người dùng. Siêu người dùng cơ sở dữ liệu bỏ qua tất cả các kiểm tra truy cập ngoại trừ quyền đăng nhập. Đây là một đặc quyền nguy hiểm và không nên sử dụng một cách bất cẩn. Tốt nhất bạn không nên thực hiện hầu hết công việc của mình với tư cách là siêu người dùng. Để tạo superuser mới, hãy sử dụng tên TẠO ROLE SUPERUSER. Việc này phải được thực hiện từ một vai trò cũng là siêu người dùng.

// Если при создании роли не указать атрибуты, то роль получит запрещающие атрибуты (*NOLOGIN, NOSUPERUSER*) автоматом.

// Nếu bạn không chỉ định các thuộc tính khi tạo vai trò, vai trò đó sẽ tự động nhận các thuộc tính cấm (NOLOGIN, NOSUPERUSER).

* *Создание базы данных.* Роль должна явно иметь разрешение на создание базы данных (за исключением суперпользователей, которые пропускают все проверки). Для создания такой роли используется *CREATE ROLE* ***имя*** *CREATEDB*.
* Tạo cơ sở dữ liệu. Vai trò này rõ ràng phải có quyền tạo cơ sở dữ liệu (ngoại trừ siêu người dùng bỏ qua tất cả các bước kiểm tra). Để tạo một vai trò như vậy, hãy sử dụng tên CREATE ROLE CREATEDB.
* *Создание роли*. Роль должна явно иметь разрешение на создание других ролей (за исключением суперпользователей, которые пропускают все проверки). Для создания такой роли используется *CREATE ROLE имя CREATEROLE*. Роль с правом CREATEROLE может не только создавать, но и изменять и удалять другие роли, а также выдавать и отзывать членство в ролях. Однако для создания, изменения, удаления ролей суперпользователей и изменения членства в них требуется иметь статус суперпользователя; права CREATEROLE в таких случаях недостаточно.
* Tạo một vai trò. Vai trò này phải có quyền rõ ràng để tạo các vai trò khác (ngoại trừ siêu người dùng, bỏ qua tất cả các bước kiểm tra). Để tạo một vai trò như vậy, hãy sử dụng tên CREATE ROLE CREATEROLE. Vai trò có quyền CREATEROLE không chỉ có thể tạo mà còn sửa đổi và xóa các vai trò khác cũng như cấp và thu hồi tư cách thành viên trong các vai trò. Tuy nhiên, bạn phải là siêu người dùng để tạo, chỉnh sửa, xóa và thay đổi tư cách thành viên trong vai trò siêu người dùng; Quyền CREATEROLE là không đủ trong những trường hợp như vậy.
* *Запуск репликации*. Роль должна иметь явное разрешение на запуск потоковой репликации (за исключением суперпользователей, которые пропускают все проверки). Роль, используемая для потоковой репликации, также должна иметь атрибут LOGIN. Для создания такой роли используется *CREATE ROLE имя REPLICATION LOGIN*.
* Bắt đầu sao chép. Vai trò này phải có quyền rõ ràng để chạy sao chép phát trực tuyến (ngoại trừ siêu người dùng bỏ qua tất cả các bước kiểm tra). Vai trò được sử dụng để sao chép luồng cũng phải có thuộc tính ĐĂNG NHẬP. Để tạo một vai trò như vậy, hãy sử dụng tên TẠO VAI TRÒ ĐĂNG NHẬP.
* *Пароль.* Пароль имеет значение, если метод аутентификации клиентов требует, чтобы пользователи предоставляли пароль при подключении к базе данных. Методы аутентификации password и md5 используют пароли. База данных и операционная система используют раздельные пароли. Пароль указывается при создании роли: *CREATE ROLE имя PASSWORD 'строка'*.
* Mật khẩu. Mật khẩu rất quan trọng nếu phương thức xác thực ứng dụng khách yêu cầu người dùng cung cấp mật khẩu khi kết nối với cơ sở dữ liệu. Phương thức xác thực mật khẩu và md5 sử dụng mật khẩu. Cơ sở dữ liệu và hệ điều hành sử dụng mật khẩu riêng biệt. Mật khẩu được chỉ định khi tạo vai trò: TẠO ROLE tên PASSWORD 'string'.

Text

Description automatically generated

1. **Методы аутентификации и способы хранения аутентификационных данных.**

**Phương pháp xác thực và phương pháp lưu trữ dữ liệu xác thực.**

Graphical user interface, text, application, email

Description automatically generated

<https://postgrespro.ru/docs/postgrespro/10/auth-methods>

Существует несколько способов хранения аутентификационных данных, включая:

Có một số cách để lưu trữ dữ liệu xác thực, bao gồm:

* Хеширование: В этом методе пароль хешируется (преобразуется в набор символов фиксированной длины) и сохраняется на сервере. При аутентификации пользователь вводит свой пароль, который затем хешируется и сравнивается с сохраненным значением хеша.
* Băm: Trong phương pháp này, mật khẩu được băm (chuyển đổi thành một tập hợp các ký tự có độ dài cố định) và được lưu trữ trên máy chủ. Trong quá trình xác thực, người dùng nhập mật khẩu của họ, mật khẩu này sau đó được băm và so sánh với giá trị băm được lưu trữ.
* Шифрование: В этом методе пароль шифруется (преобразуется в неразборчивую форму) и сохраняется на сервере. При аутентификации пользователь вводит свой пароль, который затем расшифровывается и сравнивается с исходным значением пароля.
* Mã hóa: Trong phương pháp này, mật khẩu được mã hóa (chuyển đổi thành dạng khó hiểu) và được lưu trữ trên máy chủ. Trong quá trình xác thực, người dùng nhập mật khẩu của họ, sau đó mật khẩu này được giải mã và so sánh với giá trị mật khẩu ban đầu.
* Хранение на клиентской стороне: Аутентификационные данные сохраняются на компьютере пользователя в виде куки-файлов или локального хранилища браузера.
* Bộ nhớ phía máy khách: Dữ liệu xác thực được lưu trữ trên máy tính của người dùng dưới dạng cookie hoặc bộ nhớ cục bộ của trình duyệt.
* Хранение в базе данных: Аутентификационные данные могут быть сохранены в базе данных на сервере.
* Lưu trữ cơ sở dữ liệu: Dữ liệu xác thực có thể được lưu trữ trong cơ sở dữ liệu trên máy chủ.
* Использование токенов: В этом методе сервер выдает токен, который сохраняется на клиентской стороне и используется для аутентификации вместо пароля.
* Sử dụng mã thông báo: Trong phương pháp này, máy chủ phát hành mã thông báo được lưu trữ ở phía máy khách và được sử dụng để xác thực thay vì mật khẩu.

Важно выбирать подходящий метод хранения аутентификационных данных в зависимости от требований безопасности и конфиденциальности.

Điều quan trọng là chọn phương pháp lưu trữ dữ liệu xác thực phù hợp dựa trên yêu cầu bảo mật và quyền riêng tư của bạn.

1. **Примеры механизмов обеспечения целостности БД.**

**Ví dụ về các cơ chế đảm bảo tính toàn vẹn của cơ sở dữ liệu.**

Некоторые из механизмов обеспечения целостности БД включают в себя:

* Ограничения целостности: это правила, которые определяют допустимые значения, которые могут быть сохранены в определенных полях таблицы. Например, можно установить ограничение, которое запрещает вставку записи, если определенное поле равно NULL.
* Ràng buộc toàn vẹn: Đây là các quy tắc xác định các giá trị hợp lệ có thể được lưu trữ trong các trường nhất định của bảng. Ví dụ: bạn có thể đặt ràng buộc ngăn không cho bản ghi được chèn vào nếu một trường nhất định là NULL.
* Триггеры: это программы, которые выполняются автоматически в ответ на определенные действия пользователя. Например, можно настроить триггер, который удаляет запись из связанной таблицы при удалении записи из основной таблицы.
* Trình kích hoạt: Đây là các chương trình thực thi tự động để đáp ứng với một số hành động nhất định của người dùng. Ví dụ: bạn có thể thiết lập trình kích hoạt xóa bản ghi khỏi bảng liên quan khi bản ghi bị xóa khỏi bảng chính.
* Хранимые процедуры: это блоки кода, которые можно вызывать из приложения для выполнения определенных задач. Хранимые процедуры могут обеспечивать целостность данных, например, проверять, что поля таблицы заполнены корректно.
* Thủ tục lưu trữ: Đây là các khối mã có thể được gọi từ một ứng dụng để thực hiện các tác vụ cụ thể. Các thủ tục được lưu trữ có thể đảm bảo tính toàn vẹn của dữ liệu, ví dụ: kiểm tra xem các trường trong bảng có được điền chính xác hay không.
* Уникальные индексы: это индексы, которые требуют, чтобы значения в указанном поле были уникальными. Это обеспечивает целостность данных, поскольку таким образом невозможно вставить две записи с одинаковыми значениями в поле, указанном в уникальном индексе.
* Chỉ mục duy nhất: Đây là những chỉ mục yêu cầu các giá trị trong một trường được chỉ định là duy nhất. Điều này đảm bảo tính toàn vẹn dữ liệu vì không thể chèn hai bản ghi có cùng giá trị vào một trường được chỉ định trong một chỉ mục duy nhất.
* Внешние ключи: это связи между таблицами, которые гарантируют, что запись в основной таблице имеет соответствующую запись в связанной таблице. Это обеспечивает целостность данных при изменении или удалении записи в одной из таблиц.
* Khóa ngoại: Đây là mối quan hệ giữa các bảng nhằm đảm bảo rằng một bản ghi trong bảng chính có bản ghi tương ứng trong bảng liên quan. Điều này đảm bảo tính toàn vẹn dữ liệu khi một bản ghi trong một trong các bảng bị thay đổi hoặc bị xóa.

1. **Основные виды угроз в СУБД**

**Các loại mối đe dọa chính trong DBMS**

В СУБД существуют различные типы угроз, которые могут привести к нарушению безопасности и целостности данных. Основные из них:

Có nhiều loại mối đe dọa khác nhau trong DBMS có thể làm tổn hại đến tính bảo mật và tính toàn vẹn của dữ liệu. Những cái chính:

* Несанкционированный доступ к данным: это может произойти, когда злоумышленник получает доступ к базе данных через сеть, используя уязвимый порт или слабый пароль.
* Truy cập dữ liệu trái phép: Điều này có thể xảy ra khi kẻ tấn công giành quyền truy cập vào cơ sở dữ liệu qua mạng bằng cổng dễ bị tấn công hoặc mật khẩu yếu.
* Внедрение вредоносных программ: злоумышленники могут использовать различные виды вредоносных программ для атаки на СУБД, такие как вирусы, черви, троянские программы и т.д.
* Tiêm phần mềm độc hại: Kẻ tấn công có thể sử dụng nhiều loại phần mềm độc hại khác nhau để tấn công DBMS, chẳng hạn như vi-rút, sâu, ngựa Trojan, v.v.
* Недостаточное управление доступом: это может произойти, когда пользователи получают неограниченный доступ к данным базы данных без должного контроля прав доступа.
* Kiểm soát truy cập không đầy đủ: Điều này có thể xảy ra khi người dùng có quyền truy cập không hạn chế vào dữ liệu cơ sở dữ liệu mà không có kiểm soát truy cập thích hợp.
* Отказ в обслуживании: это может произойти, когда злоумышленники создают множество запросов к базе данных, которые приводят к перегрузке системы, что может привести к отказу в обслуживании.
* Từ chối dịch vụ: Điều này có thể xảy ra khi kẻ tấn công tạo nhiều truy vấn cơ sở dữ liệu làm quá tải hệ thống, điều này có thể dẫn đến việc từ chối dịch vụ.
* Недостаточное резервное копирование: если данные не резервируются или хранятся в неправильном формате, то это может привести к их потере или повреждению.
* Sao lưu không đầy đủ: Nếu dữ liệu không được sao lưu hoặc lưu trữ ở định dạng sai, có thể dẫn đến mất hoặc hỏng dữ liệu.
* Несанкционированное изменение данных: злоумышленники могут использовать различные способы для изменения данных в базе данных, такие как SQL-инъекции или подделку запросов.
* Sửa đổi dữ liệu trái phép: Kẻ tấn công có thể sử dụng nhiều phương pháp khác nhau để sửa đổi dữ liệu trong cơ sở dữ liệu, chẳng hạn như chèn SQL hoặc giả mạo truy vấn.

1. **Классификация методов защиты на основе функций СУБД**
2. **Phân loại các phương pháp bảo vệ dựa trên chức năng DBMS**

Методы защиты данных в СУБД можно классифицировать на основе их функций. Вот некоторые из них:

Các phương pháp bảo vệ dữ liệu trong DBMS có thể được phân loại dựa trên chức năng của chúng. Dưới đây là một số trong số họ:

* Аутентификация и авторизация: Эти функции помогают убедиться в том, что пользователь имеет право доступа к конкретным данным. Аутентификация используется для проверки подлинности пользовательской учетной записи, а авторизация определяет разрешения на доступ к данным.
* Xác thực và ủy quyền: Những tính năng này giúp đảm bảo rằng người dùng có quyền truy cập dữ liệu cụ thể. Xác thực được sử dụng để xác minh danh tính của tài khoản người dùng và ủy quyền xác định quyền truy cập dữ liệu.
* Шифрование: Шифрование данных является одним из наиболее распространенных методов защиты данных в СУБД. Это процесс преобразования данных в такой формат, который не может быть понят другими, кроме того, кто обладает ключом шифрования.
* Mã hóa: Mã hóa dữ liệu là một trong những phương pháp phổ biến nhất để bảo vệ dữ liệu trong DBMS. Đó là quá trình chuyển đổi dữ liệu sang định dạng mà người khác không thể hiểu được ngoại trừ người giữ khóa mã hóa.
* Мониторинг и журналирование: Эти функции помогают отслеживать доступ к данным и действия пользователей в СУБД. Мониторинг и журналирование могут быть использованы для быстрого выявления угроз безопасности и ответа на них.
* Giám sát và ghi nhật ký: Các tính năng này giúp theo dõi hoạt động truy cập dữ liệu và người dùng trong DBMS. Giám sát và ghi nhật ký có thể được sử dụng để nhanh chóng xác định và ứng phó với các mối đe dọa bảo mật.
* Резервное копирование и восстановление: Резервное копирование и восстановление данных являются неотъемлемой частью защиты данных в СУБД. Они позволяют восстановить данные в случае их потери или повреждения из-за технических сбоев, кибератак или иных событий.
* Sao lưu và phục hồi: Sao lưu và phục hồi dữ liệu là một phần không thể thiếu trong việc bảo vệ dữ liệu trong DBMS. Chúng cho phép bạn khôi phục dữ liệu trong trường hợp mất mát hoặc hư hỏng do lỗi kỹ thuật, tấn công mạng hoặc các sự kiện khác.
* Фильтрация и контроль доступа: Фильтрация и контроль доступа помогают определить, какие пользователи имеют доступ к какой информации в СУБД. Они могут быть использованы для ограничения доступа к конфиденциальным данным и защите от несанкционированного доступа.
* Lọc và kiểm soát truy cập: Kiểm soát lọc và truy cập giúp xác định người dùng nào có quyền truy cập vào thông tin nào trong DBMS. Chúng có thể được sử dụng để hạn chế quyền truy cập vào dữ liệu bí mật và bảo vệ khỏi sự truy cập trái phép.

Это не все функции СУБД, связанные с защитой данных, но они дают представление о том, какие методы защиты могут быть использованы и какие функции СУБД они могут обеспечить.

Đây không phải là tất cả các chức năng DBMS liên quan đến bảo vệ dữ liệu, nhưng chúng đưa ra ý tưởng về những phương pháp bảo mật nào có thể được sử dụng và những chức năng DBMS nào chúng có thể cung cấp.

1. **Функции первичного ключа в реляционной модели базы данных**

**Các hàm khóa chính trong mô hình cơ sở dữ liệu quan hệ**

Первичный ключ - это уникальный идентификатор записи в реляционной модели базы данных. Его функции включают:

Khóa chính là mã định danh duy nhất cho một bản ghi trong mô hình cơ sở dữ liệu quan hệ. Các tính năng của nó bao gồm:

* Уникальность: каждый первичный ключ должен быть уникальным в пределах таблицы.
* Duy nhất: Mỗi khóa chính phải là duy nhất trong bảng.
* Идентификация: первичный ключ помогает идентифицировать каждую запись в таблице.
* Nhận dạng: Khóa chính giúp xác định từng bản ghi trong bảng.
* Ограничение целостности: первичный ключ используется для выполнения ограничения целостности, которое гарантирует, что каждая запись в таблице имеет уникальный идентификатор.
* Ràng buộc toàn vẹn: Khóa chính được sử dụng để thực thi ràng buộc toàn vẹn nhằm đảm bảo rằng mỗi bản ghi trong bảng có một mã định danh duy nhất.
* Связь: первичный ключ часто используется для создания связей между таблицами базы данных.
* Mối quan hệ: Khóa chính thường được sử dụng để tạo mối quan hệ giữa các bảng trong cơ sở dữ liệu.
* Индексирование: первичный ключ также используется для создания индексов, что повышает производительность поиска и сортировки данных в таблице.
* Lập chỉ mục: Khóa chính cũng được sử dụng để tạo chỉ mục, giúp cải thiện hiệu suất tìm kiếm và sắp xếp dữ liệu trong bảng.
* Обеспечение целостности данных: первичный ключ помогает предотвратить дублирование данных в таблице и обеспечивает ее целостность.
* Đảm bảo tính toàn vẹn dữ liệu: Khóa chính giúp ngăn chặn dữ liệu trùng lặp trong bảng và đảm bảo tính toàn vẹn của nó.

1. **Правила и условия контроля целостности в реляционной модели**

**Các quy tắc và điều kiện để kiểm soát tính toàn vẹn trong mô hình quan hệ**

Целостность в реляционной модели данных подразумевает соблюдение определенных правил и ограничений для обеспечения корректности информации в базе данных.

Tính toàn vẹn trong mô hình dữ liệu quan hệ ngụ ý việc tuân thủ các quy tắc và hạn chế nhất định để đảm bảo tính chính xác của thông tin trong cơ sở dữ liệu.

Основные правила и условия контроля целостности в реляционной модели включают:

Các quy tắc và điều kiện cơ bản để kiểm soát tính toàn vẹn trong mô hình quan hệ bao gồm:

* Правило уникальности: в каждой таблице должен быть определен уникальный ключ (primary key), который гарантирует уникальность каждой записи в таблице.
* Quy tắc duy nhất: Mỗi bảng phải có một khóa duy nhất được xác định (khóa chính), đảm bảo tính duy nhất của mỗi bản ghi trong bảng.
* Правило ссылочной целостности: в таблицах должны использоваться внешние ключи (foreign key), которые связывают записи с другими таблицами в базе данных.
* Quy tắc toàn vẹn tham chiếu: Các bảng phải sử dụng khóa ngoại để liên kết các bản ghi với các bảng khác trong cơ sở dữ liệu.
* Правило целостности домена: определенный столбец (атрибут) должен содержать только те значения, которые соответствуют определенному домену данных для этого столбца.
* Quy tắc toàn vẹn miền: Một cột (thuộc tính) cụ thể chỉ được chứa những giá trị khớp với miền dữ liệu cụ thể cho cột đó.
* Правило целостности сущности: каждая запись в таблице должна быть уникальной и иметь достаточно информации для ее идентификации.
* Quy tắc toàn vẹn thực thể: Mỗi bản ghi trong bảng phải là duy nhất và có đủ thông tin để xác định nó.
* Правило целостности проверки: определенные условия (check conditions) могут быть применены к столбцам таблицы, чтобы гарантировать корректность ввода данных.
* Quy tắc toàn vẹn kiểm tra: Một số điều kiện kiểm tra nhất định có thể được áp dụng cho các cột trong bảng để đảm bảo nhập dữ liệu chính xác.
* Правило целостности обновления: связанные записи в других таблицах должны быть обновлены или удалены, если связующий внешний ключ был изменен.
* Cập nhật quy tắc toàn vẹn: Các bản ghi liên quan trong các bảng khác phải được cập nhật hoặc xóa nếu khóa ngoại liên kết đã thay đổi.
* Правило целостности удаления: удаление записи из таблицы, на которую ссылаются другие таблицы, должно быть ограничено или запрещено, чтобы избежать разрушения связей в базе данных.
* Quy tắc toàn vẹn khi xóa: Việc xóa một bản ghi khỏi bảng được tham chiếu bởi các bảng khác phải bị hạn chế hoặc bị cấm để tránh phá vỡ các mối quan hệ trong cơ sở dữ liệu.

Введение и контроль этих правил и ограничений может быть выполнено с использованием соответствующих механизмов баз данных, таких как ограничения целостности (integrity constraints), триггеры (triggers), индексы (indexes) и хранимые процедуры (stored procedures).