**1) Sinh viên hãy trình bày về bài toán xác thực (Audentication) và uỷ quyền  
(Authorization) trong lĩnh vực bảo mật cơ sở dữ liệu. Trình bày các phương pháp xác  
thực và uỷ quyền hiện nay đang sử dụng? Mô tả 1 phương pháp mà bạn cho là tối ưu  
nhất hiện nay? Giải thích vì sao bạn chọn phương pháp này? Cho ví dụ minh hoạ.**

**- Xác thực (Authentication):**

* Xác thực là quá trình xác định xem một người dùng hoặc một hệ thống có đúng là ai họ tuyên bố không. Trong bảo mật cơ sở dữ liệu, việc xác thực thường liên quan đến việc xác định danh tính của người dùng (user) hoặc ứng dụng trước khi cho phép họ truy cập vào cơ sở dữ liệu.
* Các phương pháp xác thực phổ biến bao gồm xác thực bằng mật khẩu, xác thực bằng chứng chỉ số, xác thực bằng vân tay, hoặc sử dụng các phương tiện xác thực hai yếu tố (2FA) như mã OTP.

**- Uỷ quyền (Authorization):**

* Uỷ quyền là quá trình quyết định xem một người dùng được phép làm gì sau khi họ đã được xác thực thành công. Trong bảo mật cơ sở dữ liệu, uỷ quyền xác định quyền hạn của người dùng trong việc truy cập và thao tác với các đối tượng trong cơ sở dữ liệu như bảng, cột, hoặc thậm chí là dòng dữ liệu cụ thể.
* Các phương pháp uỷ quyền bao gồm phân quyền dựa trên vai trò (role-based access control - RBAC), phân quyền dựa trên mức độ tối thiểu (least privilege), và phân quyền dựa trên các quy tắc kinh nghiệm (rule-based access control).

**- Phương pháp em cho là tối ưu nhất:** RBAC là một phương pháp linh hoạt và hiệu quả trong việc quản lý quyền truy cập vào các tài nguyên trong hệ thống thông tin. Cơ bản, RBAC phân chia quyền hạn thành các vai trò và gán các vai trò đó cho người dùng. Mỗi vai trò có một tập hợp các quyền hạn cụ thể, và người dùng được gán vào các vai trò tương ứng với các nhiệm vụ của họ trong tổ chức.

**- Lý do RBAC được coi là tối ưu nhất bởi vì:**

* **Dễ quản lý:** RBAC giúp giảm bớt sự phức tạp trong quản lý quyền truy cập bằng cách tập trung vào việc quản lý các vai trò thay vì quản lý các quyền truy cập của từng người dùng một cách riêng lẻ.
* **Minh bạch và dễ hiểu:** RBAC tạo ra một mô hình quyền hạn rõ ràng và dễ hiểu, giúp người quản trị và người dùng dễ dàng hiểu và kiểm soát quyền hạn của mình.
* **Bảo mật cao:** RBAC giúp giảm thiểu rủi ro bảo mật bằng cách giảm khả năng lỗi trong việc cấu hình quyền truy cập, đồng thời ngăn chặn các nguy cơ xâm nhập từ người dùng không được ủy quyền.
* **Linh hoạt:** RBAC cho phép dễ dàng thay đổi quyền hạn của người dùng bằng cách chỉ cần thay đổi vai trò của họ, thay vì phải điều chỉnh từng quyền hạn riêng lẻ.

**🡺 Ví dụ:** Trong một hệ thống quản lý bán hàng, có các vai trò như quản trị viên, nhân viên bán hàng và khách hàng. Quản trị viên có quyền truy cập và quản lý toàn bộ hệ thống, nhân viên bán hàng chỉ có quyền truy cập vào các tính năng bán hàng và quản lý đơn hàng, trong khi khách hàng chỉ có quyền xem và đặt hàng sản phẩm. Điều này được thực hiện thông qua việc sử dụng RBAC.

**2). Sinh viên hãy giải thích vì sao điều khiển truy cập tùy quyền không kiểm soát được luồng thông tin. Cho ví dụ minh hoạ //Hướng dẫn: xem ví dụ về cách tấn công Trojan**

**Điều khiển truy cập tùy quyền** là một công nghệ được sử dụng để giới hạn quyền truy cập vào các tài nguyên hệ thống, nhằm ngăn chặn việc truy cập trái phép hoặc lộ thông tin quan trọng.

Điều khiển truy cập tùy quyền chỉ giới hạn quyền truy cập vào các tài nguyên cụ thể và cho phép người dùng chỉ truy cập vào những thông tin mà họ được phép xem. Tuy nhiên, điều này **không kiểm soát được luồng thông tin do các yếu tố như:**

1. Các vấn đề về mã độc: Mã độc có thể truy cập vào tài nguyên mà người dùng không được phép truy cập và gửi thông tin đó đến các nơi khác.

2. Tấn công từ bên ngoài: Kẻ tấn công có thể xâm nhập vào hệ thống và truy cập vào các tài nguyên mà họ không được phép truy cập.

3. Lỗi hệ thống: Các lỗi trong hệ thống có thể cho phép người dùng truy cập vào các tài nguyên mà họ không được phép truy cập.

4. Sự nhầm lẫn: Người dùng có thể nhầm lẫn và truy cập vào các tài nguyên mà họ không được phép truy cập.

Vì vậy, điều khiển truy cập tùy quyền không đảm bảo rằng luồng thông tin sẽ được kiểm soát hoàn toàn và người dùng cần phải cẩn thận khi sử dụng các tài nguyên.

**Ví dụ,** một loại tấn công Trojan có tên là "keylogger" có thể được cài đặt trên máy tính của một người dùng mà không cần quyền truy cập cao. Keylogger sẽ ghi lại tất cả các phím được nhấn trên bàn phím và gửi thông tin này đến máy chủ từ xa. Do đó, dù cho điều khiển truy cập tùy quyền đã được cài đặt để hạn chế quyền truy cập vào tài khoản người dùng, nhưng keylogger vẫn có thể lấy được thông tin quan trọng thông qua luồng thông tin từ máy tính của người dùng đó.

**Ví dụ về điều khiển truy cập tùy quyền không kiểm soát được luồng thông tin đó là trong trường hợp của Trojan.**

Nhân viên mới vào làm của công ty vô tình nhận được email có chứa Trojan, do không nghi ngờ và không được đào tạo nghiệp vụ chuyên sâu nên đã mở và tải email xuống máy tính của công ty việc này đã vô tình làm cho Trojan xâm nhập vào máy tính và hệ thống của công ty.Thường thì Trojan được phát tán qua email lừa đảo hoặc được tải xuống từ các trang web không đáng tin cậy. Sau khi nó xâm nhập vào máy tính, Trojan có thể thực hiện các hoạt động không mong muốn như lấy trộm thông tin đăng nhập, quét mạng hoặc phát tán các chương trình độc hại khác.

Trojan có thể xâm nhập vào máy tính của nhân viên đó và lấy cắp các thông tin đăng nhập bằng cách chiếm đoạt các quyền truy cập của người dùng. Ví dụ, một Trojan có thể thực hiện một hành động quyền truy cập như một quản trị viên. Nếu nhân viên đó đang đăng nhập vào máy tính với tư cách quản trị viên, Trojan sẽ có quyền truy cập vào tất cả các tài nguyên của hệ thống, bao gồm các tài khoản người dùng và các tệp tin quan trọng khác.

Trong trường hợp này, Trojan được coi là một ví dụ điển hình của điều khiển truy cập tùy quyền không kiểm soát được luồng thông tin. Nó có thể truy cập và thực hiện các hoạt động không mong muốn mà không cần sự đồng ý hoặc kiểm soát của người dùng chính thức. Điều này giúp kẻ tấn công có thể mở khóa các thông tin nhạy cảm mà không phải vượt qua các rào cản bảo vệ thông tin khắt khe trong một hệ thống.

Để bảo vệ khỏi các tấn công Trojan và các hình thức tấn công khác, các tổ chức cần áp dụng các giải pháp bảo mật, bao gồm việc giám sát và kiểm soát truy cập, cài đặt phần mềm chống virus và bảo mật chặt chẽ hệ thống.

**3). Sinh viên hãy trình bày về bài toán kiểm toán truy cập (Auditing) trong lĩnh vực bảo mật cơ sở dữ liệu. Cho ví dụ minh hoạ**

**Bài toán kiểm toán truy cập (Auditing)** trong lĩnh vực bảo mật cơ sở dữ liệu đó là quá trình theo dõi và ghi lại các hoạt động truy cập vào cơ sở dữ liệu từ các người dùng, nhất là những truy cập không được ủy quyền hoặc có nguy cơ gây hại. Nó giúp đảm bảo an toàn và bảo mật thông tin trong cơ sở dữ liệu.

Các yếu tố được lưu trữ trong quá trình kiểm toán truy cập bao gồm người dùng, hoạt động thực hiện, thời điểm, nơi thực hiện và kết quả của hoạt động. Các thông tin này sẽ giúp quản trị viên tổng hợp các hoạt động truy cập, xác định các hành vi khả nghi hay gian lận và giúp phát hiện các cuộc tấn công cũng như giải quyết các vấn đề kỹ thuật trong hệ thống.

**Ví dụ:** Khi một ngân hàng muốn theo dõi các hoạt động truy cập của nhân viên vào cơ sở dữ liệu của khách hàng. Đây là để đảm bảo rằng thông tin của khách hàng được giữ an toàn và các hoạt động truy cập vào cơ sở dữ liệu của họ là hợp lệ và được ủy quyền.

Khi một nhân viên ngân hàng đăng nhập vào hệ thống cơ sở dữ liệu của khách hàng, một bản ghi sẽ được tạo để theo dõi hoạt động của nhân viên đó. Bản ghi này sẽ bao gồm các thông tin liên quan đến nhân viên, thời gian truy cập, các hoạt động được thực hiện, như xem thông tin khách hàng, sửa đổi thông tin hoặc tạo tài khoản mới.

Việc ghi lại các hoạt động này cho phép ngân hàng theo dõi và phát hiện các hành vi đáng ngờ. Nếu một hoạt động không hợp lệ được thực hiện, ví dụ như nhân viên truy cập thông tin của khách hàng không được ủy quyền hoặc thực hiện một hành vi không đúng quy trình, ngân hàng có thể tìm ra nguyên nhân của sự cố và đưa ra giải pháp để khắc phục tình trạng đó và ngăn chặn lại xảy ra trong tương lai.

Ví dụ này cho thấy rằng phương pháp kiểm toán truy cập rất quan trọng để đảm bảo an toàn và bảo mật thông tin của khách hàng. Nó giúp phát hiện và ngăn chặn các hoạt động không hợp lệ hoặc đáng ngờ cũng như giúp giải quyết các vấn đề kỹ thuật trong hệ thống cơ sở dữ liệu.

**4). Giải thích tính “tùy ý” trong mô hình điều khiển truy cập tùy ý? Trong mô hình điều  
khiển truy cập tùy ý, những loại người dùng nào có quyền phân quyền/thu hồi quyền  
hạn? Trong mô hình điều khiển truy cập tùy ý, tại sao ưu điểm là tính linh hoạt, tính  
dễ dàng trong quản lý quyền truy cập. Cho ví dụ minh hoạ?**

- Trong mô hình điều khiển truy cập tùy ý (ABAC - Attribute-Based Access Control), **tính "tùy ý"** xuất phát từ khả năng quản lý quyền truy cập dựa trên các thuộc tính linh hoạt của người dùng, tài nguyên và ngữ cảnh. Điều này có nghĩa là quyền truy cập được xác định không chỉ dựa trên vai trò của người dùng, mà còn dựa trên các thuộc tính khác như thông tin cá nhân, ngữ cảnh của giao dịch, và trạng thái của tài nguyên.

- Trong mô hình ABAC, những loại người dùng có **quyền phân quyền/thu hồi quyền hạn** bao gồm các quản trị viên hệ thống, quản trị viên quyền truy cập, người quản lý chính sách an ninh thông tin, hoặc người sử dụng có đủ đặc quyền đặc biệt. Các quản trị viên này có thể xác định và quản lý các chính sách quyền truy cập, gán các thuộc tính và điều kiện cho các vai trò và người dùng, và cấp phép hoặc thu hồi quyền truy cập dựa trên nhu cầu của tổ chức.

**- Tính linh hoạt và dễ dàng** trong mô hình điều khiển truy cập tùy ý ABAC (Attribute-Based Access Control) nằm ở cách thức quản lý quyền truy cập được định nghĩa dựa trên các thuộc tính và quy tắc định nghĩa trước.

* Các thuộc tính, quy tắc và môi trường giúp cho quyền truy cập có thể được cấu hình linh hoạt hơn và dễ dàng hơn, do đó giúp cho công việc cấu hình và quản lý quyền truy cập đơn giản và dễ dàng hơn. Các loại thuộc tính khác nhau, từ các thuộc tính kiểu số đến các thuộc tính văn bản và cả các thuộc tính động, có thể được sử dụng để xác định quyền truy cập.
* Các đối tượng và tài nguyên có thể được nhóm lại dựa trên các thuộc tính quy định trước để đáp ứng các mục đích khác nhau. Nhóm người dùng và quy tắc truy cập có thể được thiết lập tự động cho từng đối tượng trong cùng một nhóm, giúp giảm thời gian cấu hình và giảm thiểu sự nhầm lẫn.

**Ví dụ:** Một tổ chức sử dụng mô hình ABAC để quản lý quyền truy cập vào hệ thống của mình. Thay vì chỉ xác định quyền dựa trên vai trò như "quản trị viên" hoặc "nhân viên", họ cũng xác định quyền dựa trên các thuộc tính như phòng ban, cấp bậc, vị trí địa lý, hoặc thậm chí là tuổi. Ví dụ, một người dùng có tuổi dưới 18 không thể truy cập vào dữ liệu nhạy cảm như thông tin tài chính. Điều này thể hiện tính linh hoạt và khả năng tùy chỉnh cao của mô hình ABAC.

**5). Tại sao khi cài đặt, người ta sử dụng danh sách khả năng thay thế cho cấu trúc ma trận phân quyền?**

Khi cài đặt hệ thống quản lý quyền truy cập, người ta thường sử dụng danh sách khả năng thay thế cho cấu trúc ma trận phân quyền vì một số lý do chính sau:

* **Tính linh hoạt:** Danh sách khả năng thay thế cho phép mỗi đối tượng hoặc người dùng được gán một danh sách các quyền truy cập cụ thể. Điều này tạo ra một mô hình quản lý quyền truy cập linh hoạt hơn, cho phép các quản trị viên dễ dàng thêm, sửa đổi hoặc loại bỏ các quyền mà không cần thay đổi cấu trúc toàn bộ ma trận.
* **Đơn giản hóa quản lý:** Trong mô hình ma trận phân quyền truy cập, việc quản lý quyền truy cập có thể trở nên phức tạp khi số lượng người dùng và tài nguyên tăng lên. Sử dụng danh sách khả năng thay thế giúp giảm bớt sự phức tạp này bằng cách tập trung vào các quyền truy cập cụ thể cho mỗi đối tượng.
* **Dễ dàng hiểu và duy trì:** Danh sách khả năng thay thế thường dễ dàng hiểu hơn và ít phức tạp hơn so với ma trận phân quyền truy cập. Điều này làm cho quản lý và duy trì hệ thống trở nên đơn giản hơn, đặc biệt là đối với các tổ chức có quy mô nhỏ hoặc không có nhu cầu phức tạp.
* **Hiệu suất:** Trong môi trường có số lượng lớn người dùng và tài nguyên, việc sử dụng danh sách khả năng thay thế thường có hiệu suất tốt hơn so với ma trận phân quyền truy cập. Điều này giúp hệ thống hoạt động một cách hiệu quả hơn và giảm thiểu tải cho máy chủ.

**6). Giải thích tính “bắt buộc” trong mô hình điều khiển truy cập tùy ý? Trong mô hình điều khiển truy cập bắt buộc, tại sao nó ít được vận dụng trong thực tế? Tại sao mô hình điều khiển truy cập bắt buộc được áp dụng trong môi trường quân đội. Trong mô hình điều khiển truy cập bắt buộc, tại sao nó ít mang tính nghiêm ngặt/khắc khe?**

- Trong mô hình điều khiển truy cập tùy ý, **tính "bắt buộc"** ám chỉ việc quản lý quyền truy cập dựa trên các quy định và luật lệ cụ thể, mà không phụ thuộc vào quan điểm hoặc quyết định của các cá nhân.

- Mặc dù mô hình điều khiển truy cập bắt buộc có thể cung cấp một cấu trúc an ninh mạnh mẽ, nhưng **nó thường ít được vận dụng trong thực tế** do các hạn chế sau:

* **Đòi hỏi sự cấu trúc hóa cao:** Mô hình này yêu cầu việc xác định rõ ràng các quy tắc và luật lệ, đồng thời thiết lập một cơ sở hạ tầng phức tạp để thực hiện và duy trì hệ thống. Điều này đòi hỏi nhiều công sức và tài nguyên.
* **Khó thích ứng với môi trường linh hoạt:** Trong một số tổ chức, đặc biệt là trong môi trường công nghiệp và thương mại, yêu cầu về tính linh hoạt và khả năng thích ứng nhanh chóng có thể làm cho mô hình bắt buộc trở nên không linh hoạt.

**-** Mặt khác, mô hình điều khiển truy cập bắt buộc thường **được áp dụng trong môi trường quân đội vì:**

* **Tính bảo mật cao:** Trong môi trường quân sự, việc bảo vệ thông tin và tài nguyên quan trọng là hết sức cần thiết. Mô hình bắt buộc cung cấp một cấu trúc an ninh chặt chẽ, đảm bảo rằng quyền truy cập được kiểm soát một cách nghiêm ngặt.
* **Tính quy định**: Mô hình này phù hợp với môi trường có các quy định và quy tắc nghiêm ngặt, nơi mà việc tuân thủ và thực thi các quy định là rất quan trọng.

- Tuy nhiên, mô hình điều khiển truy cập bắt buộc cũng **ít mang tính nghiêm ngặt hoặc khắc khe** trong một số trường hợp vì:

* **Cần sự linh hoạt:** Trong một số tình huống, đặc biệt là trong môi trường doanh nghiệp, yêu cầu về tính linh hoạt và khả năng thích ứng có thể khiến cho việc áp dụng mô hình bắt buộc trở nên không thực tế.
* **Chi phí và độ phức tạp:** Xây dựng và duy trì một hệ thống điều khiển truy cập bắt buộc có thể đòi hỏi nhiều tài nguyên và chi phí, điều này có thể không phù hợp với môi trường có ngân sách hạn chế.

**7). Ý tưởng trọng tâm của mô hình RBAC là gì? Vì sao RBAC hỗ trợ nguyên tắc bảo mật: đặc quyền ít nhất (Least Privilege)? Vì sao RBAC hỗ trợ nguyên tắc bảo mật: sự tách biệt các nhiệm vụ (Separation of duties). Vì sao RBAC hỗ trợ nguyên tắc bảo mật: trừu tượng hóa dữ liệu (Data Abstraction).**

**- Ý tưởng trọng tâm của mô hình RBAC** (Role-Based Access Control) là quản lý quyền truy cập dựa trên vai trò của người dùng trong tổ chức. Thay vì gán quyền trực tiếp cho từng người dùng, RBAC gán quyền theo vai trò, và người dùng được gán vào các vai trò này.

**- RBAC hỗ trợ nguyên tắc bảo mật: đặc quyền ít nhất (Least Privilege) bởi vì** người dùng chỉ nhận được những quyền hạn cần thiết để thực hiện công việc của họ. Thay vì cấp cho người dùng toàn bộ quyền truy cập, RBAC chỉ cấp cho họ những quyền cần thiết để thực hiện công việc hiện tại, giảm thiểu rủi ro liên quan đến việc sử dụng quyền truy cập quá nhiều.

- RBAC cũng hỗ trợ nguyên tắc bảo mật: **sự tách biệt các nhiệm vụ (Separation of duties)** bằng cách phân chia quyền hạn giữa các vai trò khác nhau. Mỗi vai trò có thể được thiết kế để thực hiện một loạt các nhiệm vụ riêng biệt, và không có vai trò nào được cấp đủ quyền hạn để thực hiện tất cả các nhiệm vụ một cách độc lập. Điều này giúp ngăn chặn sự lạm dụng quyền lợi và giảm thiểu rủi ro tiềm ẩn từ việc xâm phạm thông tin hoặc lợi ích của tổ chức.

- Cuối cùng, RBAC hỗ trợ nguyên tắc bảo mật: **trừu tượng hóa dữ liệu** (Data Abstraction) bằng cách phân chia quyền truy cập dựa trên cấp độ trừu tượng của dữ liệu. RBAC không chỉ xác định quyền truy cập dựa trên các tài nguyên cụ thể, mà còn có thể áp dụng các quyền truy cập cho các tập hợp dữ liệu hoặc loại dữ liệu cụ thể, giúp kiểm soát quyền truy cập một cách hiệu quả và linh hoạt.

**8). Điểm đặc trưng của mô hình Hierarchical RBAC là gì? Điểm đặc trưng của mô hình Static Separation of Duty Relations là gì? Điểm đặc trưng của mô hình Dynamic Separation of Duty Relations là gì? Ứng dụng của từng mô hình? Cho ví dụ minh hoạ**

**- Mô hình Hierarchical RBAC:**

* **Điểm đặc trưng:** Mô hình Hierarchical RBAC sử dụng cấu trúc phân cấp để quản lý quyền truy cập. Các vai trò được tổ chức thành các cấp độ khác nhau, trong đó vai trò ở mức cao hơn có thể kế thừa quyền truy cập từ vai trò ở mức thấp hơn.
* **Ứng dụng:** Thích hợp cho các tổ chức có cấu trúc tổ chức phân cấp, nơi mà quyền truy cập phải tuân thủ các nguyên tắc phân quyền trong tổ chức, ví dụ như các công ty có cấp bậc quản lý.

**- Mô hình Static Separation of Duty Relations:**

* **Điểm đặc trưng:** Mô hình Static Separation of Duty Relations sử dụng các quy tắc cố định để xác định các mối quan hệ phân tách nhiệm vụ. Các quy tắc này không thay đổi theo thời gian hoặc ngữ cảnh.
* **Ứng dụng:** Thường được áp dụng trong các môi trường yêu cầu tính nhất quán trong việc tách biệt các nhiệm vụ, chẳng hạn như các tổ chức tài chính hay các hệ thống thanh toán trực tuyến.

**- Mô hình Dynamic Separation of Duty Relations:**

* **Điểm đặc trưng:** Mô hình Dynamic Separation of Duty Relations cho phép các quy tắc phân tách nhiệm vụ có thể thay đổi dựa trên các điều kiện hoặc ngữ cảnh cụ thể, như thời gian, sự kiện xảy ra hoặc tình trạng của hệ thống.
* **Ứng dụng:** Thích hợp cho các môi trường linh hoạt và thay đổi, nơi mà yêu cầu phân tách nhiệm vụ có thể thay đổi theo thời gian, chẳng hạn như trong các hệ thống quản lý nhân sự trong doanh nghiệp lớn.

**Ví dụ minh họa:**

**Mô hình Hierarchical RBAC:** Trong một công ty, vai trò "Quản lý" có thể kế thừa quyền truy cập từ vai trò "Nhân viên" trong cùng một bộ phận.

**Mô hình Static Separation of Duty Relations:** Một hệ thống thanh toán trực tuyến yêu cầu mỗi giao dịch phải được duyệt bởi ít nhất hai người dùng khác nhau.

**Mô hình Dynamic Separation of Duty Relations:** Trong một tổ chức, quy tắc phân tách nhiệm vụ có thể thay đổi theo thời gian, ví dụ như một nhân viên được phép tạo và duyệt yêu cầu nghỉ phép trong vòng ba tháng làm việc, nhưng sau đó phải có một quản lý khác duyệt nếu nghỉ lâu hơn.

**9). So sánh điểm giống/khác nhau giữa hai khái niệm kế thừa vai trò tổng quát (General Role Hierarchies) và kế thừa vai trò bị giới hạn (Limited Role Hierarchies).**

**- Giống nhau:**

* Cả kế thừa vai trò tổng quát và kế thừa vai trò bị giới hạn đều là các kỹ thuật được sử dụng trong RBAC để quản lý quyền truy cập.
* Cả hai kỹ thuật đều cho phép các vai trò con kế thừa các quyền và truy cập của các vai trò cha.

**- Khác nhau:**

* Kế thừa vai trò tổng quát cho phép các vai trò con kế thừa tất cả các quyền và truy cập của các vai trò cha, trong khi kế thừa vai trò bị giới hạn chỉ cho phép các vai trò con kế thừa một phần các quyền và truy cập của các vai trò cha.
* Kế thừa vai trò tổng quát cho phép các vai trò con thừa hưởng các quyền và truy cập của tất cả các vai trò cha trong cùng hệ thống, trong khi kế thừa vai trò bị giới hạn chỉ cho phép các vai trò con thừa hưởng các quyền và truy cập của các vai trò cha trong cùng nhóm hoặc cùng loại.
* Kế thừa vai trò tổng quát có tính linh hoạt cao hơn và cho phép quản trị viên tùy chỉnh các quyền và truy cập theo cấu trúc phân cấp, trong khi kế thừa vai trò bị giới hạn có tính linh hoạt thấp hơn và không cho phép quản trị viên tùy chỉnh quyền và truy cập của các vai trò con.
* Kế thừa vai trò tổng quát có tính phức tạp cao hơn trong việc quản lý và bảo trì hệ thống, trong khi kế thừa vai trò bị giới hạn đơn giản hơn trong việc quản lý và bảo trì hệ thống.

**10). Trong mô hình RBAC, xung đột (Conflit) quyền hạn là gì? Bạn hãy nêu một chiến lược (strategy) hoặc giải pháp (solution) giải quyết xung đột. Cho ví dụ minh hoạ**

- Trong mô hình RBAC (Role-Based Access Control), xung đột quyền hạn xảy ra khi một người dùng hoặc một đối tượng yêu cầu quyền hạn mà không rõ ràng hoặc có sự mâu thuẫn với quyền hạn hiện tại của họ hoặc quyền hạn của vai trò mà họ đang được gán.

**- Chiến lược hoặc giải pháp để giải quyết xung đột quyền hạn là:** sử dụng quy tắc ưu tiên. Quy tắc này xác định mức độ ưu tiên của các quyền hạn khi có xung đột xảy ra. Cụ thể, quy tắc này quy định rằng một quyền hạn cụ thể sẽ được ưu tiên hơn so với một quyền hạn khác khi có xung đột xảy ra.

**🡺 Ví dụ:** của một hệ thống quản lý bán hàng, trong đó có các vai trò khác nhau như Quản lý, Nhân viên Bán hàng và Kế toán. Một người dùng có thể thuộc hai vai trò, ví dụ như Quản lý và Nhân viên Bán hàng. Nếu quản lý muốn xem và chỉnh sửa thông tin khách hàng, nhưng vai trò Nhân viên Bán hàng chỉ cho phép xem thông tin này mà không thể chỉnh sửa, xung đột quyền hạn sẽ xảy ra.

Trong trường hợp này, giải pháp có thể là thiết lập quy tắc ưu tiên, mà ở đó quyền hạn của vai trò Quản lý sẽ được ưu tiên hơn so với vai trò Nhân viên Bán hàng. Do đó, người dùng sẽ có quyền sửa đổi thông tin khách hàng khi họ thuộc cả hai vai trò, đồng thời đảm bảo tính nhất quán và tuân thủ các quy định trong hệ thống.

**11). Nêu các phương pháp bảo mật CSDL NoSQL mà bạn biết. Trình bày 1 phương pháp  
mà bạn nghĩ là tối ưu nhất? Giải thích tại sao bạn chọn phương pháp này**

**- Có một số phương pháp bảo mật cho cơ sở dữ liệu NoSQL, trong đó có:**

* **Mã hóa dữ liệu (Data Encryption):** Sử dụng mã hóa dữ liệu để bảo vệ thông tin quan trọng khỏi sự truy cập trái phép. Mã hóa có thể áp dụng cho dữ liệu trong quá trình truyền và lưu trữ.
* **Kiểm soát truy cập (Access Control):** Thiết lập các quy tắc và chính sách để kiểm soát quyền truy cập vào cơ sở dữ liệu. Điều này bao gồm việc quản lý người dùng, vai trò, và phân quyền truy cập.
* **Ghi log và giám sát (Logging and Monitoring):** Theo dõi các hoạt động truy cập vào cơ sở dữ liệu bằng cách ghi log và giám sát. Điều này giúp phát hiện và phản ứng nhanh chóng đối với các hành vi đe dọa.
* **Bảo mật mạng (Network Security):** Áp dụng các biện pháp bảo mật để bảo vệ mạng và cơ sở dữ liệu khỏi các cuộc tấn công từ mạng.
* **Kiểm tra và xác thực (Authentication and Authorization):** Xác thực danh tính của người dùng và xác định quyền hạn truy cập của họ dựa trên thông tin xác thực.

**Giải thích vì sao chọn phương pháp này:**

* Vì phương pháp tối ưu nhất trong số này phụ thuộc vào nhu cầu cụ thể của tổ chức và tính chất của cơ sở dữ liệu. Tuy nhiên, trong nhiều trường hợp, việc kết hợp mã hóa dữ liệu và kiểm soát truy cập có thể được xem là phương pháp tối ưu.
* Mã hóa dữ liệu bảo vệ thông tin khi dữ liệu đang truyền qua mạng hoặc được lưu trữ trên đĩa, trong khi kiểm soát truy cập đảm bảo rằng chỉ những người dùng được ủy quyền mới có thể truy cập vào dữ liệu.
* Điều này tạo ra một lớp bảo vệ kép giữa việc bảo vệ dữ liệu và kiểm soát quyền truy cập, đồng thời cung cấp một cơ sở bảo mật vững chắc cho cơ sở dữ liệu NoSQL.

**12). Trình bày các kiểu tấn công vào CSDL hiện nay? Hậu quả của các kiểu tấn công này? Nêu những rủi ro mà Doanh nghiệp sẽ gặp phải trong bảo mật CSDL hiện nay? Nêu ví dụ minh hoạ. Mô tả các giải pháp phòng chống để ngăn ngừa các kiểu tấn công mà bạn đã trình bày.**

**Có nhiều kiểu tấn công vào cơ sở dữ liệu (CSDL) hiện nay, bao gồm:**

* **Tấn công SQL Injection:** Tấn công này xảy ra khi kẻ tấn công chèn mã SQL độc hại vào các trường dữ liệu đầu vào của ứng dụng, từ đó thực hiện các thao tác không được ủy quyền trên CSDL.
* **Tấn công XSS (Cross-Site Scripting):** Kỹ thuật này cho phép kẻ tấn công chèn mã JavaScript độc hại vào các trang web hoặc ứng dụng web, từ đó thực hiện các hành động không được phép trên trình duyệt của người dùng.
* **Tấn công Brute Force:** Kẻ tấn công cố gắng đoán mật khẩu bằng cách thử tất cả các khả năng có thể cho đến khi tìm ra mật khẩu đúng.
* **Tấn công DDoS (Distributed Denial of Service):** Tấn công này nhằm vào việc làm quá tải hệ thống bằng cách gửi lượng lớn yêu cầu từ nhiều nguồn khác nhau đến một máy chủ hoặc hệ thống, từ đó làm cho dịch vụ trở nên không khả dụng.

**- Hậu quả của các kiểu tấn công này** có thể là mất dữ liệu quan trọng, tiết lộ thông tin cá nhân, làm ngưng hoạt động hệ thống, gây thiệt hại về uy tín và tài chính cho doanh nghiệp.

**- Ví dụ minh họa:** cho SQL Injection

* Kẻ tấn công có thể chèn một đoạn mã SQL độc hại vào một trường dữ liệu đầu vào của một biểu mẫu trên trang web. Ví dụ, trong một trang đăng nhập, kẻ tấn công có thể nhập một chuỗi như sau vào trường "Tên đăng nhập": ‘’ OR 1=1
* Nếu ứng dụng không được bảo vệ đúng cách, đoạn mã này có thể được thêm vào truy vấn SQL mà ứng dụng sử dụng để xác thực, dẫn đến việc cho phép kẻ tấn công truy cập vào hệ thống mà không cần mật khẩu hợp lệ.

**- Rủi ro mà doanh nghiệp có thể gặp phải trong bảo mật CSDL hiện nay bao gồm:**

* Mất dữ liệu quan trọng hoặc lợi ích kinh doanh.
* Tiết lộ thông tin cá nhân của khách hàng hoặc người dùng.
* Giảm hiệu suất hoạt động do ảnh hưởng của các tấn công DDoS hoặc tấn công Brute Force.
* Mất uy tín và khách hàng do thất bại trong việc bảo vệ dữ liệu của họ.

**- Để phòng chống các kiểu tấn công này, doanh nghiệp có thể triển khai các giải pháp như:**

* Sử dụng các công cụ và phần mềm bảo mật CSDL để phát hiện và ngăn chặn các tấn công SQL Injection và XSS.
* Áp dụng chính sách mật khẩu mạnh và cơ chế chặn đăng nhập sau một số lần thất bại để ngăn chặn tấn công Brute Force.
* Triển khai giải pháp bảo mật mạng để phòng chống tấn công DDoS.
* Thường xuyên cập nhật và bảo trì hệ thống, cùng với việc giáo dục nhân viên về các biện pháp bảo mật và phòng ngừa tấn công.

**13. Bạn hãy trình bày và mô tả các cơ chế mã hoá trong SQL Server sau:**

* **Transact-SQL functions**
* **Asymmetric keys**
* **Symmetric keys**
* **Certificates**
* **Transparent Data Encryption**

Trong SQL Server, có nhiều cơ chế mã hóa khác nhau được sử dụng để bảo vệ dữ liệu. Dưới đây là mô tả của mỗi cơ chế mã hóa:

**- Transact-SQL Functions:**

* Trình bày: SQL Server cung cấp một loạt các hàm mã hóa Transact-SQL để thực hiện các thao tác mã hóa và giải mã trên dữ liệu.
* Cơ chế: Các hàm như ENCRYPTBYKEY và DECRYPTBYKEY cho phép mã hóa và giải mã dữ liệu sử dụng khóa được lưu trữ trong CSDL.

**- Asymmetric Keys:**

* Trình bày: Asymmetric keys là một cặp khóa bao gồm khóa công khai và khóa riêng tư. Khóa công khai được sử dụng để mã hóa dữ liệu, trong khi khóa riêng tư được sử dụng để giải mã.
* Cơ chế: Trong SQL Server, asymmetric keys thường được sử dụng để bảo vệ các khóa đối xứng (symmetric keys) và chứng thực (certificates).

**- Symmetric Keys:**

* Trình bày: Symmetric keys sử dụng cùng một khóa để mã hóa và giải mã dữ liệu.
* Cơ chế: SQL Server hỗ trợ việc tạo và quản lý symmetric keys để bảo vệ dữ liệu, và chúng thường được sử dụng để mã hóa cột, bảng hoặc cả cơ sở dữ liệu.

**- Certificates:**

* Trình bày: Certificates là các đối tượng bảo mật trong SQL Server, chứa khóa công khai và thông tin xác thực.
* Cơ chế: Certificates thường được sử dụng để bảo vệ symmetric keys và để xác thực dữ liệu và thực thể trong hệ thống SQL Server.

**- Transparent Data Encryption (TDE):**

* Trình bày: TDE là một tính năng trong SQL Server cho phép mã hóa toàn bộ cơ sở dữ liệu trong lúc dữ liệu đang được lưu trữ trên đĩa.
* Cơ chế: Khi TDE được kích hoạt, dữ liệu và log của cơ sở dữ liệu sẽ được mã hóa tự động, giúp bảo vệ dữ liệu trước khi nó được ghi ra đĩa và khi nó đang ở trong bộ nhớ.

🡺 Các cơ chế mã hóa này cung cấp các phương tiện khác nhau để bảo vệ dữ liệu trong SQL Server, cho phép các quản trị viên cơ sở dữ liệu lựa chọn cách thích hợp nhất cho nhu cầu bảo mật của họ.

**14)Trình bày tổng quan về việc sử dụng SQL Server Audit (Overview of Using SQLServer Audit)? Nêu các thành phần của SQL Server Audit? SQL Server Audit thường dùng trong những trường hợp nào? Cho ví dụ minh hoạ?**

**- Tổng quan về việc sử dụng SQL Server Audit:**

* **Theo dõi hoạt động:** SQL Server Audit cho phép theo dõi các sự kiện như truy cập đến cơ sở dữ liệu, thao tác thay đổi dữ liệu, hay thậm chí là các sự kiện hệ thống như khởi động lại SQL Server.
* **Ghi lại hoạt động:** Các sự kiện được theo dõi có thể được ghi lại vào một tập tin log hoặc bảng dữ liệu cụ thể, cho phép quản trị viên kiểm tra, phân tích và lưu trữ thông tin hoạt động của hệ thống.
* **Phân tích và báo cáo:** Dữ liệu ghi lại từ SQL Server Audit có thể được sử dụng để tạo báo cáo, phân tích xu hướng và thực hiện các kiểm tra an ninh để đảm bảo rằng hệ thống đang hoạt động một cách an toàn và tuân thủ các quy định.

**- Các thành phần của SQL Server Audit:**

* Audit: Định nghĩa một hoặc nhiều nhóm sự kiện mà bạn muốn theo dõi.
* Audit Specification: Xác định các chi tiết cụ thể của các sự kiện được theo dõi, bao gồm cơ sở dữ liệu, đối tượng và loại sự kiện.
* Target: Xác định nơi lưu trữ dữ liệu log, có thể là tập tin log hoặc bảng dữ liệu.
* Database Audit Specification: Kết hợp Audit và Audit Specification và áp dụng chúng cho cơ sở dữ liệu cụ thể.
* Server Audit Specification: Kết hợp Audit và Audit Specification và áp dụng chúng cho toàn bộ SQL Server.

**- SQL Server Audit thường được sử dụng trong các trường hợp cần theo dõi và ghi lại các hoạt động xảy ra trên cơ sở dữ liệu SQL Server. Dưới đây là một số trường hợp phổ biến:**

* **Tuân thủ và Tuân thủ Luật pháp:** Doanh nghiệp thường cần tuân thủ các quy định pháp lý hoặc tiêu chuẩn ngành trong việc bảo vệ thông tin và dữ liệu của khách hàng. SQL Server Audit có thể được sử dụng để theo dõi các hoạt động truy cập vào cơ sở dữ liệu và xác định xem liệu các quy tắc và quy định đã được tuân thủ hay không.
* **Bảo mật và Phòng chống Xâm nhập: SQL** Server Audit cung cấp khả năng ghi lại các hoạt động như đăng nhập, truy cập đối với các đối tượng cụ thể và thậm chí các thay đổi cấu trúc của cơ sở dữ liệu. Việc theo dõi này giúp phát hiện kịp thời các hoạt động không mong muốn hoặc có thể là dấu hiệu của một cuộc tấn công.
* **Quản lý Hiệu suất:** SQL Server Audit cũng có thể được sử dụng để theo dõi hiệu suất của cơ sở dữ liệu bằng cách ghi lại các hoạt động như thời gian truy vấn, số lượng truy cập, và các hoạt động khác có thể ảnh hưởng đến hiệu suất.

**Ví dụ:** Một công ty tài chính muốn đảm bảo rằng dữ liệu khách hàng của họ được bảo vệ và tuân thủ các quy định pháp lý như GDPR. Họ triển khai SQL Server Audit để theo dõi các hoạt động truy cập vào cơ sở dữ liệu, bao gồm cả việc truy vấn, chỉnh sửa, và xóa thông tin cá nhân của khách hàng. Khi phát hiện bất kỳ hoạt động không hợp lệ hoặc có nguy cơ, họ có thể ngay lập tức xử lý và báo cáo về sự việc.

**15. Trình bày những gì bạn biết về minorring, logshipping và replication. Cho ví dụ minh hoạ**

**1. Mirroring:**

Mirroring (đồng bộ hóa) là một kỹ thuật để sao chép quy trình xử lý và cơ sở dữ liệu trên các máy chủ để cung cấp khả năng cao cho hệ thống. Khi sử dụng mirroring, có hai máy chủ: một máy chủ chính (principal) và một máy chủ đồng bộ (mirror). Các thay đổi dữ liệu được thực hiện trên máy chủ chính sẽ được sao chép đến máy chủ đồng bộ bằng cách sử dụng mạng. Nếu máy chủ chính bị lỗi, máy chủ đồng bộ sẽ được viết để lấy chỗ của máy chủ chính để tiếp tục thành công. Ví dụ về mirroring:

```

-- Tạo Principal Database

CREATE DATABASE Database1;

GO

-- Tạo Mirror Database với tùy chọn Mirror

CREATE DATABASE Database1\_Mirror

ON

PRIMARY

(NAME = 'MinhTien',

FILENAME = 'D:\BMCSDL\MinhTien\_Mirror.mdf')

LOG ON

(NAME = 'Database1\_log',

FILENAME = 'D:\BMCSDL\MinhTien\_log.ldf')

GO

-- Enable Mirroring

ALTER DATABASE Database1

SET PARTNER = 'TCP://[Mirror\_Server\_Name]:[Port]';

GO

ALTER DATABASE Database1\_Mirror

SET PARTNER = 'TCP://[Principal\_Server\_Name]:[Port]';

GO

```

**2. Log Shipping:**

Log Shipping là kỹ thuật khác để sao chép dữ liệu. Kỹ thuật này là một giải pháp phổ biến cho việc tạo bản sao phụ bản của cơ sở dữ liệu trên máy chủ khác, giúp bảo vệ dữ liệu bằng cách sao lưu các giao dịch vào tệp log và chuyển chúng đến máy chủ sao lưu. Khi máy chủ chính gặp sự cố và dữ liệu bị mất, ta có thể khôi phục chúng từ bản sao phụ bản. Ví dụ về Log Shipping:

```

-- Tạo sao lưu

BACKUP DATABASE MinhTien TO DISK = 'D:\Backup\ MinhTien.bak';

BACKUP LOG MinhTien TO DISK = 'D:\Backup\ MinhTien.log';

GO

-- Tạo nơi chưa nhật ký

RESTORE DATABASE MinhTien\_Copy

WITH NORECOVERY

FROM DISK = 'D:\Backup\ MinhTien.bak';

GO

RESTORE LOG MinhTien\_Copy

WITH NORECOVERY

FROM DISK = 'D:\Backup\ MinhTien.log';

GO

-- Enable Log Shipping

USE master

GO

EXEC sp\_add\_log\_shipping\_primary\_database

@database = N' MinhTien ',

@backup\_directory = N'\\[Backup\_Server\_Name]\BackupFolder',

@backup\_share = N'\\[Backup\_Server\_Name]\BackupShare',

@backup\_retention\_period = 1440,

@monitor\_server = N'[Monitor\_Server\_Name]',

@monitor\_server\_security\_mode = 1,

@backup\_threshold = 5,

@threshold\_alert\_enabled = 0,

@history\_retention\_period = 5760

GO

```

**3. Replication:**

Replication là kỹ thuật sao chép và phân phối cơ sở dữ liệu trên nhiều máy chủ để đáp ứng nhu cầu di chuyển dữ liệu giữa các hoạt động của một tổ chức. Sử dụng Replication, ta có thể tạo ra nhiều bản sao cơ sở dữ liệu độc lập nhau. Các thay đổi sẽ được sao chép và phân phối đến các bản sao khác để đảm bảo dữ liệu đồng bộ. Ví dụ về Replication:

```

-- Tạo bản sao (Subscriber)

USE [master]

GO

EXEC sp\_addsubscriber @subscriber = N'Subscriber\_Server\_Name'

, @description = N'Subscriber\_Server\_Description'

, @security\_mode = 0

, @frequency\_type = 4

, @frequency\_interval = 1

, @frequency\_relative\_interval = 1

, @frequency\_recurrence\_factor = 0

, @active\_start\_time\_of\_day = 190000

, @active\_end\_time\_of\_day = 235959

, @active\_start\_date = 20161011

, @active\_end\_date = 20161231

GO

-- Thêm bản sao db (Subscriber\_db) vào publication

USE [Publisher\_db]

GO

EXEC sp\_addsubscription @publication = N'publication\_name'

, @subscriber = N'Subscriber\_Server\_Name'

, @destination\_db = N'Subscriber\_db'

, @subscription\_type = N'Push'

, @sync\_type= N'automatic'

, @article = N'all'

, @update\_mode = N'read only'

GO

```

* **Database Mirroring:** Database Mirroring là một phương pháp để tạo ra một bản sao chính xác của cơ sở dữ liệu trên một hoặc nhiều máy chủ sao lưu. Trong quá trình này, một máy chủ chính (primary server) gửi các bản sao của giao dịch (transaction logs) đến một hoặc nhiều máy chủ sao lưu (mirror servers) để đảm bảo tính nhất quán của dữ liệu. Khi máy chủ chính gặp sự cố, máy chủ sao lưu có thể nhanh chóng đảm nhận vai trò của máy chủ chính.
* **Log Shipping:** Log Shipping là một phương pháp đơn giản để sao lưu và phục hồi dữ liệu trong SQL Server. Quá trình này bao gồm sao chép và gửi các bản ghi giao dịch từ cơ sở dữ liệu nguồn (source database) đến một hoặc nhiều cơ sở dữ liệu đích (destination databases) thông qua các tập tin log. Log Shipping thường được sử dụng để sao lưu dữ liệu từ một máy chủ SQL Server đến một máy chủ khác để bảo vệ dữ liệu và dự phòng.
* **Replication:** Replication là một phương pháp cho phép sao chép và phân phối dữ liệu từ một cơ sở dữ liệu nguồn đến một hoặc nhiều cơ sở dữ liệu đích trong thời gian thực hoặc gần thời gian thực. Replication cho phép cập nhật dữ liệu ở một nơi và tự động sao chép các thay đổi đó đến các nơi khác mà không cần phải truy cập trực tiếp vào cơ sở dữ liệu nguồn. Các kiểu sao chép phổ biến trong Replication bao gồm sao chép theo hàng ngang (snapshot replication), sao chép theo hàng dọc (transactional replication) và sao chép theo hàng xáo trộn (merge replication).

**🡺 Ví dụ:** Một công ty có một cơ sở dữ liệu quản lý đơn hàng trên máy chủ chính ở trung tâm dữ liệu chính. Họ triển khai Database Mirroring để sao lưu và đồng bộ hóa dữ liệu với một máy chủ sao lưu ở một trung tâm dữ liệu khác nhằm đảm bảo sự liên tục của hệ thống. Đồng thời, họ sử dụng Log Shipping để sao lưu dữ liệu hàng ngày từ máy chủ chính sang một máy chủ phục hồi ở trung tâm dữ liệu phụ để dự phòng dữ liệu.

**16) Cho bảng dữ liệu SINHVIEN(masv, hodem, ten, sodienthoai, diachi, sotaikhoan) chứa nhiều dữ liệu phục vụ hệ thống phần mềm đã được vận hành nhiều năm. Do nhu cầu nâng cấp bảo mật bằng phương pháp mã hóa dữ liệu sử dụng giải pháp được Microsoft SQL Server cung cấp sẵn, thực hiện các yêu cầu sau đây:**

**a. Tạo bảng SINHVIEN\_MAHOA để chứa dữ liệu được chuyển đổi từ bảng SINHVIEN, viết câu lệnh chuyển đổi dữ liệu bản rõ từ bảng SINHVIEN sang dữ liệu bản mã và lưu trữ vào bảng SINHVIEN\_MAHOA.**

create database test

use test

create table SINHVIEN (

masv int identity,

hodem nvarchar(100),

ten nvarchar(100),

sodienthoai nvarchar(20),

diachi nvarchar(50),

sotaikhoan nvarchar(100)

)

insert into SINHVIEN (hodem, ten, sodienthoai, diachi, sotaikhoan)

values (N'Nguyễn Thành', N'Phát', '091234567', N'Gò Vấp', '01122233')

insert into SINHVIEN (hodem, ten, sodienthoai, diachi, sotaikhoan)

values (N'Trần Ngọc', N'Thái', '09321654', N'Q2', '0223344')

insert into SINHVIEN (hodem, ten, sodienthoai, diachi, sotaikhoan)

values (N'Phạm Thanh', N'Trang', '09456789', N'Q5', '0556677')

select \* from SINHVIEN

create table SINHVIEN\_MAHOA (

masv\_mahoa varbinary(max),

hodem\_mahoa varbinary(max),

ten\_mahoa varbinary(max),

sodienthoai\_mahoa varbinary(max),

diachi\_mahoa varbinary(max),

sotaikhoan\_mahoa varbinary(max)

)

select \* from SINHVIEN\_MAHOA

-- Chuyển dữ liệu từ bảng SINHVIEN sang bảng SINHVIEN\_MAHOA

insert into SINHVIEN\_MAHOA (masv\_mahoa, hodem\_mahoa, ten\_mahoa, sodienthoai\_mahoa, diachi\_mahoa, sotaikhoan\_mahoa)

SELECT ENCRYPTBYPASSPHRASE('1', cast(masv as varchar(100))), ENCRYPTBYPASSPHRASE('1', hodem), ENCRYPTBYPASSPHRASE('1', ten),

ENCRYPTBYPASSPHRASE('1', sodienthoai), ENCRYPTBYPASSPHRASE('1', diachi), ENCRYPTBYPASSPHRASE('1', sotaikhoan)

FROM SINHVIEN

**b. Với bảng dữ liệu SINHVIEN\_MAHOA chứa dữ liệu đã được mã hóa, trình bày giải pháp được thực hiện ở tầng cơ sở dữ liệu sao cho tầng ứng dụng (app) ít sửa đổi nhất? Cho ví dụ với các trường hợp xem, thêm, xóa, sửa dữ liệu sinh viên?**

**Sử dụng**

**Stored Procedures:**

* Tạo các stored procedures trong cơ sở dữ liệu để thực hiện các thao tác (xem, thêm, xóa, sửa) trên bảng SINHVIEN\_MAHOA.
* Ứng dụng chỉ gọi các stored procedures thay vì viết trực tiếp các câu lệnh SQL.

**Views:**

* Tạo views trên bảng SINHVIEN\_MAHOA để ẩn chi tiết mã hóa.
* Ứng dụng truy vấn views thay vì trực tiếp truy vấn bảng.

**Triggers:**

* Sử dụng triggers để tự động thực hiện các thay đổi khi dữ liệu được thêm, sửa hoặc xóa.

-- Xem

select convert(varchar(max),DECRYPTBYPASSPHRASE('1', masv\_mahoa)) as masv,

convert(nvarchar(max),DECRYPTBYPASSPHRASE('1', hodem\_mahoa)) as hodem,

convert(nvarchar(max),DECRYPTBYPASSPHRASE('1', ten\_mahoa)) as ten,

convert(nvarchar(max),DECRYPTBYPASSPHRASE('1', sodienthoai\_mahoa)) as sodienthoai,

convert(nvarchar(max),DECRYPTBYPASSPHRASE('1', diachi\_mahoa)) as diachi,

convert(nvarchar(max),DECRYPTBYPASSPHRASE('1', sotaikhoan\_mahoa)) as sotaikhoan

from SINHVIEN\_MAHOA

-- Thêm

insert into SINHVIEN\_MAHOA

values (ENCRYPTBYPASSPHRASE('1', '4'), ENCRYPTBYPASSPHRASE('1', N'Thanh Long'),

ENCRYPTBYPASSPHRASE('1', N'Lang'), ENCRYPTBYPASSPHRASE('1', N'09643754'),

ENCRYPTBYPASSPHRASE('1', N'Q7'), ENCRYPTBYPASSPHRASE('1', N'0322233'))

-- Xóa

delete SINHVIEN\_MAHOA

where convert(varchar(max), DECRYPTBYPASSPHRASE('1', masv\_mahoa)) = 1

-- Sửa

update SINHVIEN\_MAHOA

set ten\_mahoa = ENCRYPTBYPASSPHRASE('1', N'Thông')

where CONVERT(nvarchar(max), DECRYPTBYPASSPHRASE('1', ten\_mahoa)) = N'Thái'

**17) Giải thích mô hình Audit sau:**

**A diagram of a cloud computing system

Description automatically generated**

**Mô hình Audit trong hình minh họa là một mô hình kiểm tra và giám sát sự hoạt động của hệ thống lưu trữ dữ liệu trên đám mây. Dưới đây là giải thích chi tiết:**

**1. Người Dùng (Users):**

* Được biểu thị bằng biểu tượng máy tính để bàn, laptop và điện thoại di động.
* Tương tác với hệ thống lưu trữ dữ liệu trên đám mây thông qua luồng dữ liệu và luồng tin nhắn bảo mật.

**2. Kiểm Toán Viên Bên Thứ Ba (Optional Third Party Auditor):**

* Biểu thị bằng biểu tượng tòa nhà.
* Là một bên thứ ba độc lập, thường được tạo ra để kiểm tra tính toàn vẹn và bảo mật của dữ liệu lưu trữ trên đám mây.
* Kết nối với hệ thống lưu trữ dữ liệu thông qua luồng dữ liệu và luồng tin nhắn bảo mật.

**3. Nhà Cung Cấp Dịch Vụ Đám Mây (Cloud Service Provider):**

* Được biểu thị bằng biểu tượng đám mây.
* Chứa các máy chủ lưu trữ dữ liệu trên đám mây.

**4. Máy Chủ Lưu Trữ Dữ Liệu Trên Đám Mây (Cloud Storage Servers):**

* Biểu thị bằng biểu tượng hình trụ.
* Lưu trữ dữ liệu của người dùng trên đám mây.

Mô hình này cho phép kiểm toán viên bên thứ ba kiểm tra tính toàn vẹn và bảo mật của dữ liệu lưu trữ trên đám mây mà không cần can thiệp vào hệ thống lưu trữ. Điều này giúp đảm bảo rằng dữ liệu của người dùng được bảo vệ và tuân thủ các quy định liên quan đến bảo mật và quyền riêng tư.