# Dạng 1: Bài tập thu hồi quyền đệ quy và không đệ quy

\* Lý thuyết:

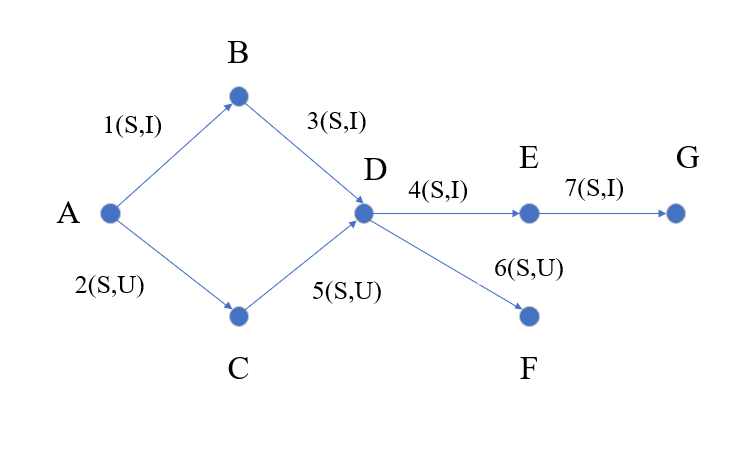
- **Thu hồi quyền đệ quy**

* Khi người dùng A thu hồi quyền truy cập của người B thì tất cả các quyền mà B đã gán cho người khác đều được thu hồi.
* Thu hồi quyền đệ quy trong system R dựa vào nhãn thời gian mỗi lần cấp quyền truy nhập cho người dùng.

- **Thu hồi quyền không đệ quy:**

* Khi A đã thu hồi quyền truy nhập trên B thì tất cả quyền truy nhập mà B đã cấp cho chủ thể khác được thay bằng A đã cấp cho chủ thể này.
* Thực tế khi một người dùng A thay đổi công việc hay vị trí thì đôi khi tổ chức chỉ muốn lấy lại quyền truy nhập của A mà không muốn lấy lại các quyền truy nhập mà A đã cấp => áp dụng thu hồi không đệ quy.
* Cần lưu ý đến nhãn thời gian.

**VD1: Thu hồi quyền đệ quy**



Bước cuối cùng: B thu hồi quyền của D đệ quy (CASCADE). Vậy E, G, F còn quyền gì?

Lời giải:

- B gán quyền (S, I) cho D ở nhãn thời gian 3 => D có quyền (S, I).

- Ở nhãn thời gian 4, D gán quyền (S, I) cho E, cả hai quyền này đều của B gán cho D => E có quyền (S, I).

- Ở nhãn thời gian 5, C gán quyền (S,U) cho D => D có quyền (S, I, U) trong đó có 2 quyền (S) của B và C gán cho D.

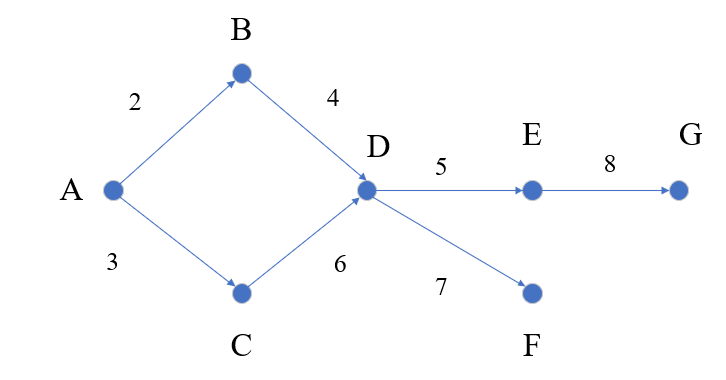
- Ở nhãn thời gian 6, D gán cho F quyền (S, U) => F có quyền (S, U) trong đó có 2 quyền (S) của B và C gán cho D.

- Ở nhãn thời gian 7, E gán cho G quyền (S, I) => G có quyền (S, I)

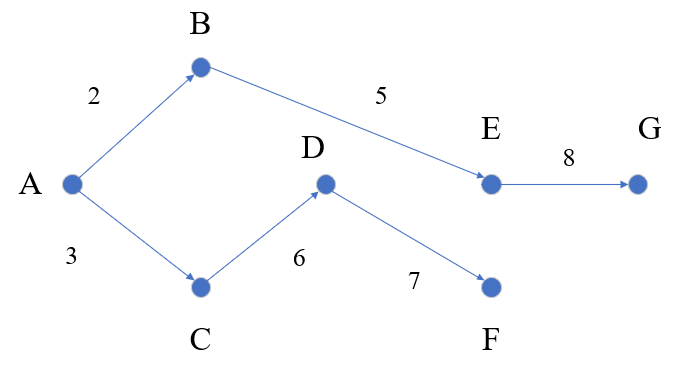
=> B thu hồi quyền đệ quy của D ở nhãn thời gian 3 thì:

* E, G: Không còn quyền
* F còn quyền (S, U) của C gán cho D

**VD2: Thu hồi quyền không đệ quy**



Sau khi B thu hồi quyền không đệ quy của D



**VD3:**

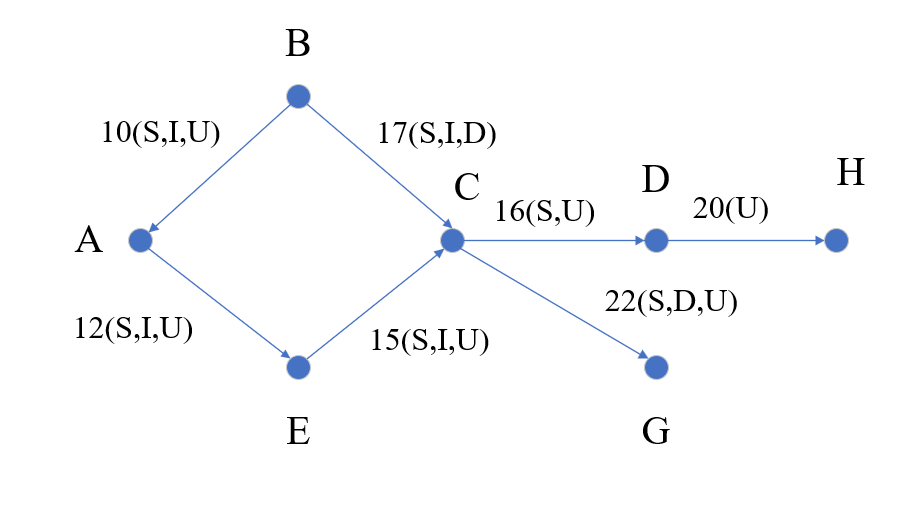
a) Vẽ sơ đồ cho dãy các câu lệnh gán quyền trong bảng dưới đây.

b) Vẽ lại sơ đồ và giải thích khi E thu hồi không đệ quy cả ba quyền Select(S), Insert (I), Update (U) đã gán cho C.

|  |  |  |
| --- | --- | --- |
| Người dùng | Câu lệnh | Nhân thời gian |
| B | Grant Select, Insert, Update to A with grant option | 10 |
| B | Grant Select, Insert, Delete to C with grant option | 17 |
| A | Grant Select, Insert, Update to E with grant option | 12 |
| E | Grant Select, Insert, Update to C with grant option | 15 |
| C | Grant Select, Update to D with grant option | 16 |
| C | Grant Select, Delete, Update to G with grant option | 22 |
| D | Grant Update to H without grant option | 20 |

LG

a)



b)

- Ở nhãn thời gian 15, E gán cho C quyền (S, I, U) => C có quyền (S, I, U)

- Ở nhãn thời gian 16, C gán cho D quyền (S, U) => D có quyền (S, U)

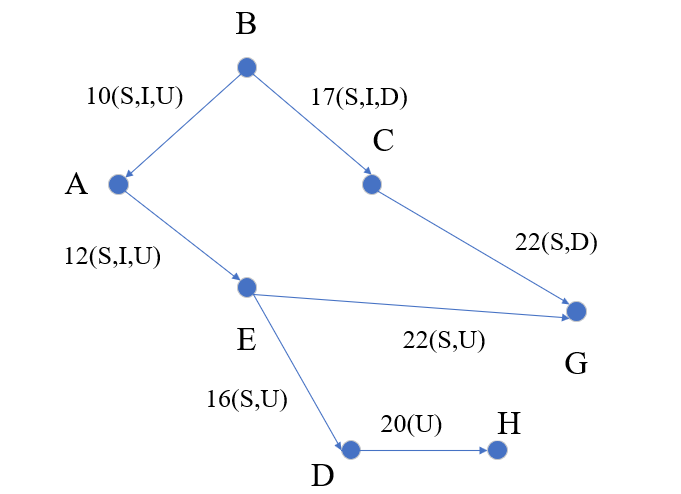
- Ở nhãn thời gian 17, B gán cho C quyền (S, I, D) => C có quyền (S, I, U, D) trong đó có 2 quyền (S, I) của E và B gán cho C.

- Ở nhãn thời gian 20, D gán cho H quyền (U) => H có quyền (U)

- Ở nhãn thời gian 22, C gán cho G quyền (S,D,U) => G có quyền (S, D, U) trong đó có 2 quyền (S) của E và B gán cho C.

=> E thu hồi không đệ quy 3 quyền (S, I, U) đã gán cho C ở nhãn thời gian 15 thì khi đó quyền của các phần tử còn lại là:

* C chỉ còn quyền (S, I, D) của B gán ở nhãn thời gian 17
* G còn quyền (S, D) của C gán và quyền (S, U) của E gán ở nhãn thời gian 22
* D còn quyền (S, U) của E gán ở nhãn thời gian 16
* H còn quyền (U) của D gán ở nhãn thời gian 20



VD4:

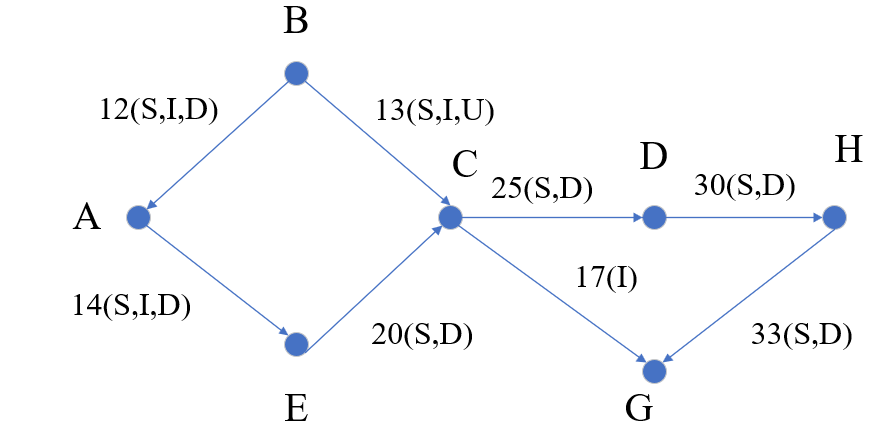
a) Vẽ sơ đồ cho dãy các câu lệnh gán quyền trong bảng dưới đây.

b) Vẽ lại sơ đồ và giải thích khi E thu hồi đệ quy cả hai quyền Select(S), Delete (D) đã gán cho C.

|  |  |  |
| --- | --- | --- |
| Người dùng | Câu lệnh | Nhân thời gian |
| B | Grant Select, Insert, Delete to A with grant option | 12 |
| B | Grant Select, Insert, Update to C with grant option | 13 |
| A | Grant Select, Insert, Delete to E with grant option | 14 |
| E | Grant Select, Delete to C with grant option | 20 |
| C | Grant Select, Delete to D with grant option | 25 |
| C | Grant Insert to G with grant option | 17 |
| D | Grant Select, Delete to H with grant option | 30 |
| H | Grant Select, Delete to G without grant option | 33 |

LG

a)



b)

- Ở nhãn thời gian 13, B gán cho C quyền (S, I, U) => C có quyền (S, I, U)

- Ở nhãn nhời gian 17, C gán cho G quyền (I) => G có quyền (I)

- Ở nhãn thời gian 20, E gán cho C quyền (S, D) => C có quyền (S, I, U, D) trong đó có 2 quyền (S) của B và E gán cho C.

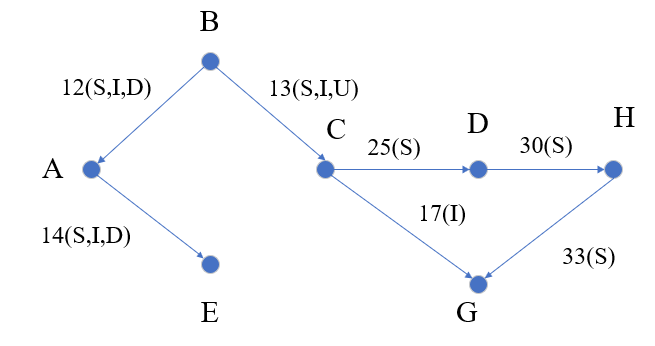
- Ở nhãn thời gian 25, C gán cho D quyền (S, D) => D có quyền (S, D) trong đó có 2 quyền (S) của B và E gán cho C.

- Ở nhãn thời gian 30, D gán cho H quyền (S, D) => H có quyền (S,D) trong đó có 2 quyền S của B và E gán cho C.

- Ở nhãn thời gian 33, H gán cho G quyền S, D => G có quyền (S, I, D) trong đó có 2 quyền S của B và E gán cho C.

=> E thu hồi đệ quy cả hai quyền Select(S), Delete (D) đã gán cho C thì khi đó các phần tử sẽ còn lại các quyền:

* C còn quyền (S, I, U) của B gán ở nhãn thời gian 13
* D còn quyền (S) của C gán ở nhãn thời gian 25
* H còn quyền (S) của D gán ở nhãn thời gian 30
* G còn quyền (I) của C gán ở nhãn thời gian 17 và quyền (S) của H gán ở nhãn thời gian 33



**Câu 1**

1. **Giả sử C là một công thức đặc trưng của một cơ sở dữ liệu thống kê, hãy giải thích q(C), X(C), |X(C)| là gì? Cho ví dụ và viết câu lệnh SQL để thực hiện.**

**-** q(C): đếm số lượng bản ghi(hoặc dòng dữ liệu) trong cơ sở dữ liệu thỏa mãn công thức đặc trưng **C**. q(C) được tính bằng cách lọc các bản ghi trong CSDL theo điều kiện được biểu diễn bởi công thức C

**-** X(C): tập hợp các bản ghi trong CSDL thỏa mãn công thức đặc trưng **C**. X(C) chính là danh sách đầy đủ các bản ghi sau khi áp dụng bộ lọc theo C

**-** |X(C)|: số lượng phần tử (bản ghi) trong tập hợp X(C). Về bản chất, |X(C)| tương đương với q(C) vì cả hai đều thể hiện số lượng bản ghi thỏa mãn điều kiện C.

**Ví dụ:** Giả sử ta có một bảng dữ liệu **Employees** với các cột ID, Name, Department, Salary. Công thức đặc trưng C:”Lương lớn hơn 10 triệu và làm việc trong phòng IT” thì:

**-** q(C): số lượng nhân viên có lương lớn hơn 10 triệu và làm việc trong phòng IT.

**-** X(C): Danh sách các nhân viên (các bản ghi) có lương lớn hơn 10 triệu và làm trong phòng IT.

**-** |X(C)|: số lượng bản ghi có trong X(C)

**Câu lệnh SQL:**

**Tính q(C):**

SELECT COUNT(\*) as qC FROM Employees

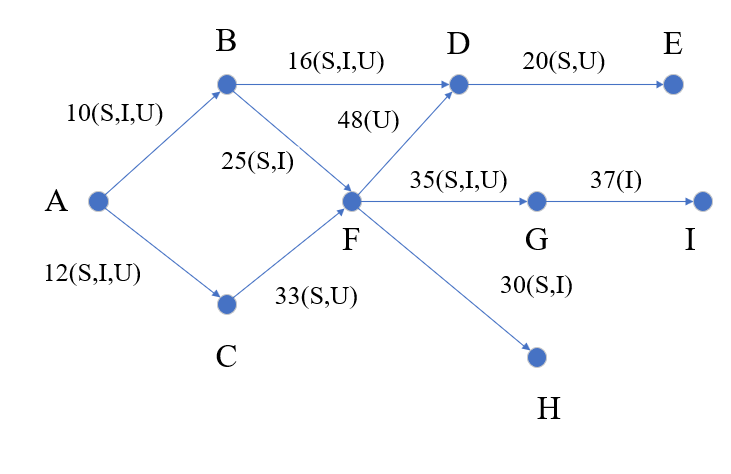
WHERE Salary > 10000000 AND Department = ‘IT’;

**Lấy tập X(C):**

SELECT \* FROM Employees WHERE Salary > 10000000 AND Department = ‘IT’;

b) Vẽ sơ đồ cho dãy các câu lệnh gán quyền trong bảng dưới đây.

|  |  |  |
| --- | --- | --- |
| Người dùng | Câu lệnh | Nhãn thời gian |
| A | Grant Select, Insert, Update to B with grant option | 10 |
| A | Grant Select, Insert, Delete to C with grant option | 12 |
| B | Grant Select, Insert, Update to D with grant option | 16 |
| B | Grant Select, Insert to F with grant option | 25 |
| D | Grant Select, Update to E without grant option | 20 |
| C | Grant Select, Update to F with grant option | 33 |
| F | Grant Select, Insert, Update to G with grant option | 35 |
| F | Grant Select, Insert to H without grant option | 30 |
| F | Grant Update to D without grant option | 48 |
| G | Grant Insert to I with grant option | 37 |



c) Vẽ lại sơ đồ và giải thích khi B thu hồi quyền đệ quy của D cả ba quyền Select (S), Insert (I), Update (U) và B thu hồi quyền đệ quy của F cả hai quyền S và I.

- Ở nhãn thời gian 10, A gán cho B quyền (S,I,U) => B có quyền (S,I,U)

- Ở nhãn thời gian 16, B gán cho D quyền (S,I,U) => D có quyền (S,I,U)

- Ở nhãn thời gian 20, D gán cho E quyền (S,U) => E có quyền (S,U)

- Ở nhãn thời gian 25, B gán cho F quyền (S,I) => F có quyền (S,I)

- Ở nhãn thời gian 30, F gán cho H quyền (S,I) => H có quyền (S,I)

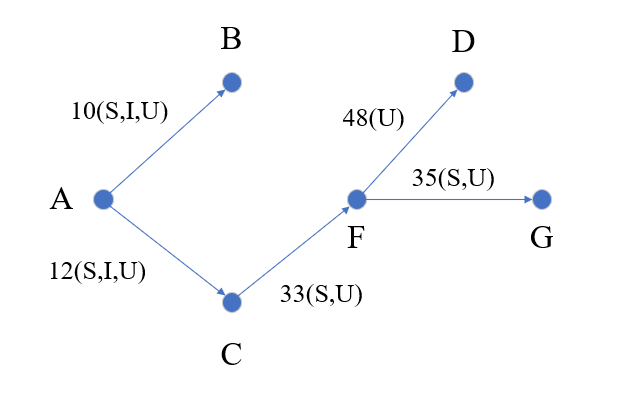
- Ở nhãn thời gian 33, C gán cho F quyền (S,U)=> F có quyền (S,I,U) trong đó có 2 quyền (S) của B và C gán cho F

- Ở nhãn thời gian 35, F gán cho G quyền (S,I,U) => G có quyền (S,I,U) trong đó có 2 quyền (S) của B và C gán cho F

- Ở nhãn thời gian 37, G gán cho I quyền (I) => I có quyền (I)

=> Khi B thu hồi quyền đệ quy của D cả ba quyền Select (S), Insert (I), Update (U) và B thu hồi quyền đệ quy của F cả hai quyền S và I. Khi đó quyền của các phần tử là:

* D còn quyền (U) của F gán ở nhãn thời gian 48
* E hết quyền
* F còn quyền (S,U) của C gán ở nhãn thời gian 33
* G còn quyền (S, U) của F gán ở nhãn thời gian 35
* I hết quyền
* H hết quyền



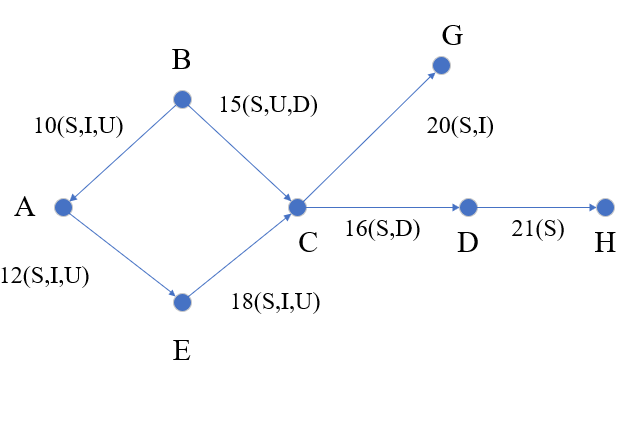
**Câu 2:**

a) Khi nói đến mã hóa CSDL, chúng ta cần quan tâm đến những vấn đề gì?

* Mục tiêu của mã hóa:
  + Bảo mật dữ liệu: Ngăn chặn truy cập trái phép vào dữ liệu nhạy cảm, đặc biệt khi CSDL bị đánh cắp.
  + Tuân thủ quy định của pháp luật: Đảm bảo tuân thủ các quy định như GDPR, HIPAA, PCI-DSS, trong đó yêu cầu mã hóa dữ liệu nhạy cảm.
  + Đảm bảo quyền riêng tư: Bảo vệ thông tin cá nhân và nhạy cảm của người dùng.
* Dữ liệu cần mã hóa: Xác định các loại dữ liệu quan trọng cần được mã hóa (thông tin cá nhân, dữ liệu đăng nhập, …)
* Phương pháp mã hóa
* Quản lý khóa mã hóa:
  + Lưu trữ khóa an toàn: Khóa mã hóa cần được lưu trữ ở nơi an toàn
  + Phân quyền: Chỉ những người và hệ thống được cấp quyền mới có thể truy cập khóa.
  + Thay đổi khóa định kỳ: Xoay vòng hoặc thay đổi khóa định kỳ để giảm thiểu rủi ro nếu khóa bị lộ.
  + Hủy khóa cũ: Khi không còn sử dụng, khóa cũ cần được vô hiệu hóa.
* Hiệu năng hệ thống: Mã hóa có thể làm giảm hiệu năng của hệ thống, đặc biệt với các truy vấn trên dữ liệu lớn. Cần:
  + Chỉ mã hóa những cột, bảng cần thiết
  + Sử dụng mã hóa ở mức lưu trữ
  + Tối ưu hóa thuật toán mã hóa.

b) Vẽ sơ đồ cho dãy các câu lệnh gán quyền trong bảng dưới đây:

|  |  |  |
| --- | --- | --- |
| Người dùng | Câu lệnh | Nhãn thời gian |
| B | Grant Select, Insert, Update to A with grant option | 10 |
| B | Grant Select, Update, Delete to C with grant option | 15 |
| A | Grant Select, Insert, Update to E with grant option | 12 |
| E | Grant Select, Insert, Update to C with grant option | 18 |
| C | Grant Select, Delete to D with grant option | 16 |
| C | Grant Select, Insert to G with grant option | 20 |
| D | Grant Select to H without grant option | 21 |



c) Vẽ lại sơ đồ và giải thích khi E thu hồi quyền không đệ quy của C cả 3 quyền Select (S), Insert (I) và Update (U).

- Ở nhãn thời gian 15, B gán cho C quyền (S,U,D) => C có quyền (S,U,D)

- Ở nhãn thời gian 16, C gán cho D quyền (S,D) => D có quyền (S,D)

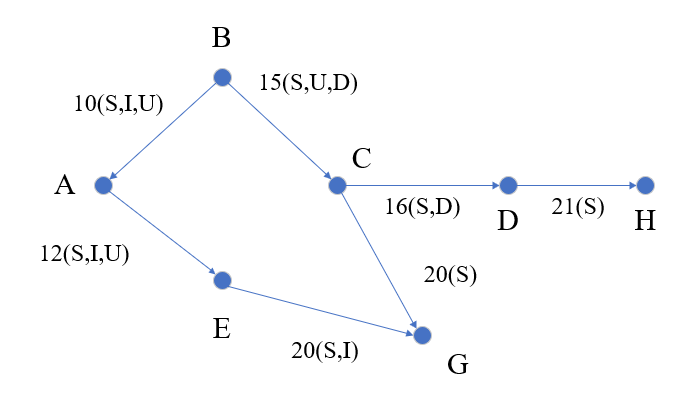
- Ở nhãn thời gian 18, E gán cho C quyền (S,I,U) => C có quyền (S,I,U,D) trong đó có 2 quyền (S,U) của B và E gán cho C

- Ở nhãn thời gian 20, C gán cho G quyền (S,I) => G có quyền (S,I) trong đó có 2 quyền (S) của B và E gán cho C.

- Ở nhãn thời gian 21, D gán cho H quyền (S) => H có quyền (S)

=> E thu hồi quyền không đệ quy của C cả 3 quyền Select (S), Insert (I) và Update (U). Khi đó các phần tử còn lại các quyền là:

* C còn quyền (S,U,D) của B gán cho ở nhãn thời gian 15
* G còn quyền (S) của C gán cho ở nhãn thời gian 20 và quyền (S,I) của E gán cho ở nhãn thời gian 20
* D giữ nguyên quyền
* H giữ nguyên quyền

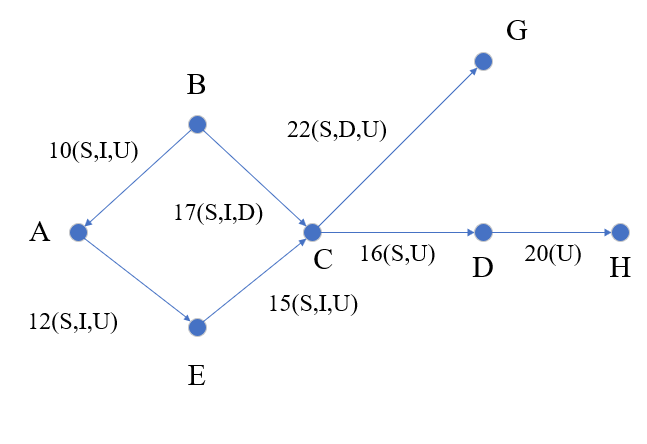


**Câu 3:**

a) Nêu định nghĩa về ngôn ngữ DDL, ngôn ngữ DML và cho ví dụ?

b) Vẽ sơ đồ cho dãy các câu lệnh gán quyền trong bảng dưới đây:

|  |  |  |
| --- | --- | --- |
| Người dùng | Câu lệnh | Nhãn thời gian |
| B | Grant Select, Insert, Update to A with grant option | 10 |
| B | Grant Select, Insert, Delete to C with grant option | 17 |
| A | Grant Select, Insert, Update to E with grant option | 12 |
| E | Grant Select, Insert, Update to C with grant option | 15 |
| C | Grant Select, Update to D with grant option | 16 |
| C | Grant Select, Delete, Update to G with grant option | 22 |
| D | Grant Update to H without grant option | 20 |



c) Vẽ lại sơ đồ và giải thích khi E thu hồi quyền không đệ quy cả 3 quyền Select (S), Insert (I) và Update (U) đã gán cho C.

- Ở nhãn thời gian 15, E gán cho C quyền (S,I,U) => C có quyền (S,I,U)

- Ở nhãn thời gian 16, C gán cho D quyền (S,U) => D có quyền (S,U)

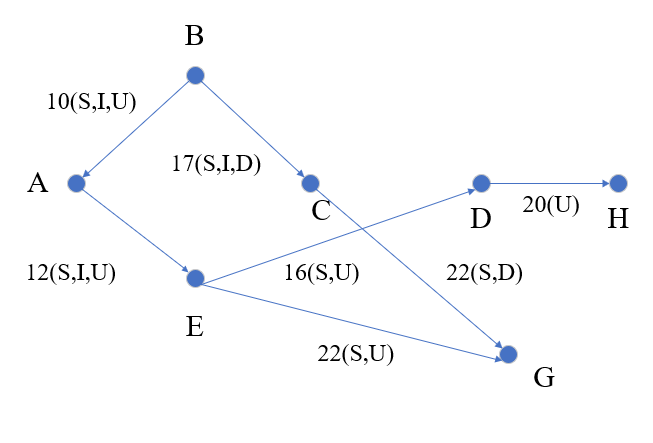
- Ở nhãn thời gian 17, B gán cho C quyền (S,I,D) => C có quyền (S,I,U,D) trong đó có 2 quyền (S,I) của E và B đều gán cho C

- Ở nhãn thời gian 20, D gán cho H quyền (U) => H có quyền (U)

- Ở nhãn thời gian 22, C gán cho G quyền (S,D,U) => G có quyền (S,D,U) trong đó có 2 quyền (S) của B và E gán cho C.

E thu hồi quyền không đệ quy cả 3 quyền Select (S), Insert (I) và Update (U) đã gán cho C. Khi đó các quyền sẽ bị thay đổi như sau:

* C còn quyền (S,I,D) của B gán cho ở nhãn thời gian 17
* G còn quyền (S,D) của C gán cho ở nhãn thời gian 22 và quyền (S) của E gán cho ở nhãn thời gian 22
* D còn quyền (S,U) của E gán cho ở nhãn thời gian 16
* H vẫn còn nguyên quyền

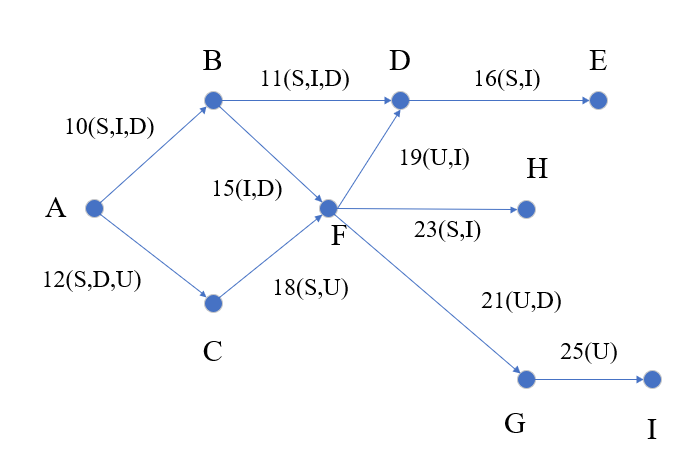


**Câu 4:**

a) Trình bày sự khác nhau giữa thu hồi quyền đệ quy và thu hồi quyền không đệ quy? Cho ví dụ.

b) Vẽ sơ đồ cho dãy các câu lệnh gán quyền trong bảng dưới đây.

|  |  |  |
| --- | --- | --- |
| Người dùng | Câu lệnh | Nhân thời gian |
| A | Grant Select, Insert, Delete to B with grant option | 10 |
| A | Grant Select, Delete, Update to C with grant option | 12 |
| B | Grant Select, Insert, Delete to D with grant option | 11 |
| B | Grant Insert, Delete to F with grant option | 15 |
| D | Grant Select, Insert to E without grant option | 16 |
| C | Grant Select, Update to F with grant option | 18 |
| F | Grant Update, Delete to G with grant option | 21 |
| F | Grant Select, Insert to H without grant option | 23 |
| F | Grant Insert, Update to D with grant option | 19 |
| G | Grant Update to I with grant option | 25 |



c) Vẽ lại sơ đồ và giải thích khi B thu hồi quyền không đệ quy quyền Delete (D), Insert (I) của F. Sau đó thu hồi quyền không đệ quy Update (U) của D thì những quyền gì bị mất?

- Ở nhãn thời gian 10, A gán cho B quyền (S,I,D) => B có quyền (S,I,D)

- Ở nhãn thời gian 11, B gán cho D quyền (S,I,D) => D có quyền (S,I,D)

- Ở nhãn thời gian 15, B gán cho F quyền (I,D) => F có quyền (I,D)

- Ở nhãn thời gian 16, D gán cho E quyền (S,I) => E có quyền (S,I)

- Ở nhãn thời gian 18, C gán cho F quyền (S,U) => F có quyền (S,I,U,D)

- Ở nhãn thời gian 19, F gán cho D quyền (U,I) => D có quyền (S,I,U,D)

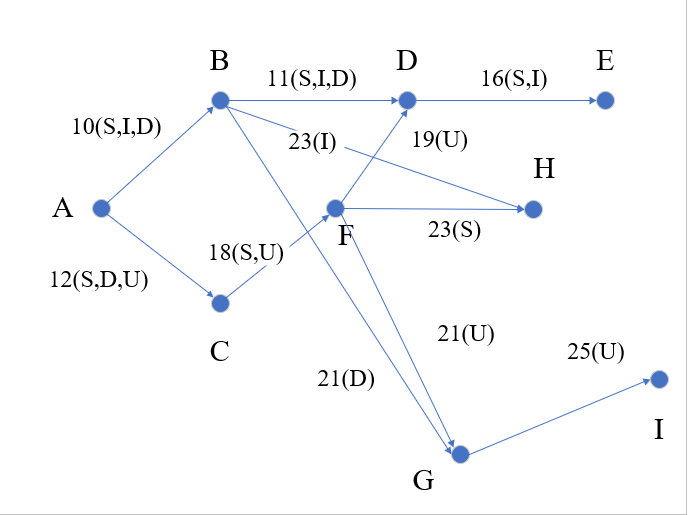
- Ở nhãn thời gian 21, F gán cho G quyền (U,D) => G có quyền (U,D)

- Ở nhãn thời gian 23, F gán cho H quyền (S,I) => H có quyền (S,I)

- Ở nhãn thời gian 25, G gán cho I quyền (U) => I có quyền U

=> B thu hồi quyền không đệ quy quyền Delete (D), Insert (I) của F thì các quyền thay dổi như sau:

* D còn quyền (S,I,D) của B gán cho ở nhãn thời gian 11, quyền (U) của F gán cho và quyền (I) của B gán cho ở nhãn thời gian 19
* H còn quyền (I) của B gán cho và quyền (S) của F gán cho ở nhãn thời gian 23
* G còn quyền (U) của F gán cho và quyền (D) của B gán cho ở nhãn thời gian 21



**Câu 5:**

a) Thế nào là thống kê nhạy cảm với một cơ sở dữ liệu thống kê, cho ví dụ?

b) Cho cơ sở dữ liệu thống kê về công nhân của một công ty như sau:

**CONGNHAN**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Ten | ChucVu | Phong | Tuoi | GioiTinh | Luong |
| 01 | Phương | Nhân viên | Marketing | 30 | F | 3700 |
| 02 | Tuyến | Trưởng phòng | Kế hoạch | 42 | M | 6500 |
| 03 | Phương | Nhân viên | Kế hoạch | 19 | M | 3800 |
| 04 | Huyền | Giám sát viên | Marketing | 28 | F | 4500 |
| 05 | Phương | Nhân viên | Marketing | 25 | M | 2500 |
| 06 | Thắng | Phó phòng | Marketing | 39 | M | 3900 |

Áp dụng biện pháp kiểm soát kích cỡ tập truy vấn với k = 2. Bạn hãy trình bày tấn công Trình theo dõi để tính được các thống kê trên **C = (Ten = ‘Phương’) ∧ (GioiTinh = F)** Viết một số câu lệnh SQL để thực hiện.

**\* Tấn công trình theo dõi**

**\* Kiểu 1 (Tính T = A ^ thỏa mãn k< |T| < N-k)**

- Đặt C = (A ∧ B) và T = A ∧ , trong đó:

A = (Ten = ‘Phương’) => Count(A) = 3

B = (GioiTinh = F)

=> = (GioiTinh ≠ F)

Ta có: Count (C) = 1 => Câu truy vấn này bị cấm!

- Tính Count(T) = Count(A ∧ ) = 2 (thỏa mãn 2 ≤ Count(T) = 2 ≤ 3)

- Tính gián tiếp Count(C):

Count C = Count(A ∧ B) = Count(A) - Count(A ∧ ) = Count(A) – Count(T) = 3 – 2 = 1

- Tính gián tiếp Sum(C, Luong):

Sum(C, Luong) = Sum(A ∧ B, Luong) = Sum(A, Luong) – Sum (A ∧ , Luong)

Sum(C, Luong) = (3700+3800+2500) – (3800+2500) = 3700

**\* Kiểu 2 (Chọn T thỏa mã: k <|T|, || < N – k)**

- Chọn T = (Tuoi<30) => Count(T) = 3 > 2, Count() = 3 < 4

- Count(All) = Count(T) + Count() = 3 + 3 = 6

- Tính: Count(C) = Count(C∨T) + Count(C∨) – Count(All) = 4 + 3 - 6 = 1

Sum(Luong, C) = Sum(Luong, C ∨ Tuoi<30) + Sum(Luong, C ∨ Tuoi≥30) – Sum(Luong, All)

= (3700+3800+4500+2500) + (3700+6500+3900)- (3700+6500+3800+4500+2500+3900) = 3700

**\* Một số câu lệnh SQL**

- Select Count(\*) From CONGNHAN

Where (Ten = ‘Phương’) and (GioiTinh = ‘F’);

- Select Sum(Luong) From CONGNHAN

Where (Ten = ‘Phương’) and (GioiTinh = ‘F’);

**Câu 6:**

a) Working knowledge và Supplementary knowledge trong SDB là gì?

b) Cho cơ sở dữ liệu thống kê về công nhân của một công ty như sau:

**CONGNHAN**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Ten | ChucVu | Phong | Tuoi | GioiTinh | Luong |
| 01 | Nam | Nhân viên | Thuế | 54 | F | 3500 |
| 02 | Bằng | Nhân viên | Kế hoạch | 22 | M | 3000 |
| 03 | Sang | Trưởng phòng | Kế hoạch | 40 | M | 5600 |
| 04 | Tuyến | Giám sát viên | Lập trình | 19 | M | 5000 |
| 05 | Phượng | Giám sát viên | Lập trình | 33 | F | 4500 |
| 06 | Hương | Phó phòng | Lập trình | 38 | M | 3200 |

C = (Phong = ‘Lập trình’) ∧ (ChucVu = ‘Giám sát viên’) ∧ (GioiTinh = M)

- **Các thống kê trên C có phải nhạy cảm không? Vì sao?**

Các thống kê trên C là thống kê nhạy cảm vì Count(C) = 1 (Số nhân viên ở phòng lập trình có chức vụ là giám sát viên và giới tính M bằng 1) => Count(C) = 1 bị cấm.

- **Giả sử áp dụng biện pháp kiểm soát kích cỡ tập truy vấn với k = 2, bạn hãy trình bày tấn công Trình theo dõi để tính được các thống kê trên C. Viết một số câu lệnh SQL để thực hiện.**

**\* Kiểu 1**

- Đặt C = (A ∧ B) và T = A ∧ , trong đó:

A = (Phong = ‘Lập trình’) => Count(A) = 3

B = (ChucVu = ‘Giám sát viên’) ∧ (GioiTinh = M)

=> = (ChucVu ≠ ‘Giám sát viên’) ∨ (GioiTinh ≠ M)

- Tính Count(T) = Count(A ∧ ) = 2 (thỏa mãn 2 ≤ Count(T) = 2 ≤ 3)

- Tính gián tiếp Count(C):

Count C = Count(A ∧ B) = Count(A) - Count(A ∧ ) = Count(A) – Count(T) = 3 – 2 = 1

- Tính gián tiếp Sum(C, Luong):

Sum(C, Luong) = Sum(A ∧ B, Luong) = Sum(A, Luong) – Sum (A ∧ , Luong)

Sum(C, Luong) = (5000+4500+3200) – (4500+3200) = 5000

**\* Kiểu 2**

- Chọn T = (Tuoi<38) => Count(T) = 3 > 2, Count() = 3 < 4

- Count(All) = Count(T) + Count() = 3 + 3 = 6

- Tính: Count(C) = Count(C∨T) + Count(C∨) – Count(All) = 3 + 4 - 6 = 1

Sum(Luong, C) = Sum(Luong, C ∨ Tuoi<38) + Sum(Luong, C ∨ Tuoi≥38) – Sum(Luong, All) = (3000+5000+4500) + (3500+5600+5000+3200) – (3500+3000+5600+5000+4500+3200) = 5000

**\* Một số câu lệnh SQL**

- Select Count(\*) From CONGNHAN

Where (Phong = ‘Lập trình’) and (ChucVu = ‘Giám sát viên’) and (GioiTinh = ‘M’);

- Select Sum(Luong) From CONGNHAN

Where (Phong = ‘Lập trình’) and (ChucVu = ‘Giám sát viên’) and (GioiTinh = ‘M’);

**Câu 7:**

a) Cơ sở dữ liệu thống kê (Statistical database) là gì? Phân biệt cơ sở dữ liệu thống kê và cơ sở dữ liệu quan hệ thông thường?

b) Áp dụng với cơ sở dữ liệu thống kê về nhân viên sau (**NhanVien**)

**CONGNHAN**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Ten | Chức vụ | Phòng | Tuổi | Giới tính | Lương |
| 01 | Hà | Nhân viên | Tài vụ | 24 | F | 4200 |
| 02 | Thúy | Trưởng phòng | Tổ chức | 42 | M | 7000 |
| 03 | Bình | Nhân viên | Tổ chức | 33 | M | 3800 |
| 04 | Mai | Giám sát viên | Marketing | 50 | F | 6000 |
| 05 | Hồng | Nhân viên | Tổ chức | 33 | F | 2500 |

C = (Phong = ‘Tổ chức) ∧ (Tuoi = 33) ∧ (GioiTinh = ‘F’)

**- Theo bạn các câu truy vấn Count(C) và Sum(C, Luong) có phải nhạy cảm không? Vì sao?**

Count(C) và Sum(C, Luong) là các câu truy vấn nhạy cảm vì Count(C) = 1

**- Tìm cách tính các thống kê trên C và viết câu lệnh SQL để thực hiện.**

**\* Kiểu 1**

- Đặt C = (A ∧ B) và T = A ∧ , trong đó:

A = (Phong = ‘Tổ chức’) => Count(A) = 3

B = (Tuoi = 33) ∧ (GioiTinh = ‘F’)

=> = (Tuoi ≠ 33) ∨ (GioiTinh ≠ ‘F’)

- Tính Count(T) = Count(A ∧ ) = 2 (thỏa mãn 2 ≤ Count(T) = 2 ≤ 3)

- Tính gián tiếp Count(C):

Count C = Count(A ∧ B) = Count(A) - Count(A ∧ ) = Count(A) – Count(T) = 3 – 2 = 1

- Tính gián tiếp Sum(C, Luong):

Sum(C, Luong) = Sum(A ∧ B, Luong) = Sum(A, Luong) – Sum (A ∧ , Luong)

Sum(C, Luong) = (7000+3800+2500) – (7000+3800) = 2500

**\* Kiểu 2**

- Chọn T = (Tuoi<42) => Count(T) = 3 > 2, Count() = 2 < 3

- Count(All) = Count(T) + Count() = 3 + 2 = 5

- Tính: Count(C) = Count(C∨T) + Count(C∨) – Count(All) = 3 + 3 - 5 = 1

Sum(Luong, C) = Sum(Luong, C ∨ Tuoi<42) + Sum(Luong, C ∨ Tuoi≥42) – Sum(Luong, All) = (2500+3800+4200) + (2500+6000+7000) – (4200+7000+3800+6000+2500) = 2500

**- Theo bạn, hệ thống dùng kiểm soát nào để chống tấn công này, mô tả?**

**Câu 8:**

Xét bài toán quản lý như sau:

Trường Cao Đẳng Cộng Đồng có nhu cầu tin học hóa công tác quản lý của mình. Trường có nhiều khoa, mỗi khoa có nhiều Ngành đào tạo. Mỗi Ngành có nhiều lớp học. Thông tin về lớp học gồm: Mã lớp, tên lớp, sĩ số.

Một sinh viên khi nhập học được xếp vào một lớp và thuộc lớp này trong suốt khóa học. Thông tin về sinh viên gồm: Mã SV, Họ tên, Ngày sinh, Giới tính, Địa chỉ.

Sinh viên học nhiều môn học. Kết thúc mỗi môn học, sinh viên có một điểm thi. Sinh viên phải thi lại lần 1 nếu điểm lần đầu không qua (điểm < 5), sẽ phải thi lại lần 2 hoặc lần 3 nếu vẫn trượt. Mỗi môn học có thông tin: Mã môn, Tên môn, Số trình.

Mỗi sinh viên khi vào trường có một hồ sơ lý lịch kèm theo. Trong đó có thông tin về sinh viên như: Số lý lịch, ngày vào Đoàn, ngày vào Đảng, quá trình học tập, họ tên bố, họ tên mẹ, anh chị em,…

a) Vẽ mô hình E-R và viết các lược đồ CSDL cho bài toán quản lý hoạt động trên.

b) Viết các câu lệnh SQL để tạo bảng: SinhVien, Diem, Monhoc

c) Nếu cấp trên yêu cầu bạn thiết kế an toàn cho cơ sở dữ liệu trên, bạn sẽ làm gì?

**Bài tập 3**

1. Kiến trúc khóa toàn vẹn Integrity Lock là gì?

2. Cho bảng dữ liệu sau:

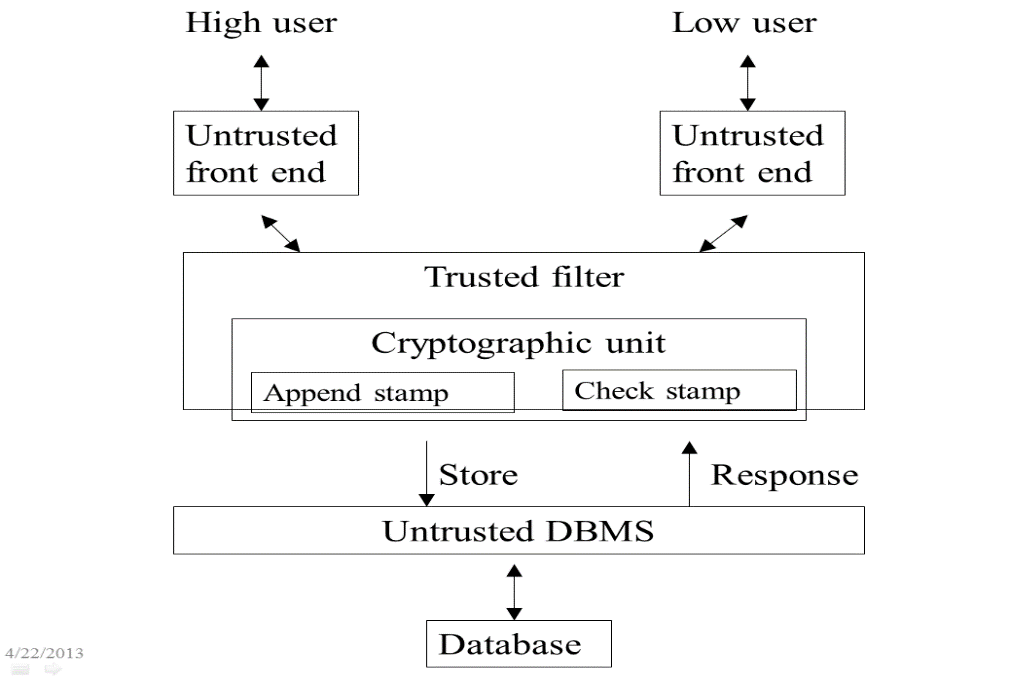
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **HoTen** | **Label** | **Tuoi** | **Label** | **Nghề nghiệp** | **Label** | **Thu nhập** | **Label** |
| Bình | C | 45 | C | Cảnh sát | S | 10000 | S |
| Trung | S | 51 | C | Giáo viên | C | 44000 | S |
| Hiếu | TS | 33 | C | Bộ đội | TS | 74500 | TS |

- Sử dụng kiến trúc Integrity Lock hãy gắn tem cho cơ sở dữ liệu trên, với 4 khóa bí mậ*t k1, k2, k3, k4*, mỗi khóa cho một cột.

- Một người dùngcó nhãn *S*, thì có thể đọc được những gì trong bảng trên sau khi đã gắn tem.

- Trình bày tính toàn vẹn khi các người dùng truy xuất dữ liệu vào bảng trên.

**Kiến trúc khóa toàn vẹn:**

****

Sử dụng cơ chế gán tem dùng mật mã khóa bí mật k1,k2,k3,k4 mỗi khóa cho một cột:

*Stamp = Ek(Dữ liệu + độ nhậy cảm user)*

Mã hoá (dữ liệu + độ nhạy cảm user) bằng một khoá bí mật K, tạo ra *tem,* và lưu vào trong CSDL cùng với mục dữ liệu đó (gắn với mục dữ liệu).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **HoTen** | **Stamp** | **Tuoi** | **Stamp** | **Nghề nghiệp** | **Stamp** | **Thu nhập** | **Stamp** |
| Bình(C) | Ek1(Bình+C) | 45(C ) | Ek2(45+C) | Cảnhsát(S) | Ek3(cảnh sat+ S) | 10000(S) | Ek4(10000+ S) |
| Trung(S) | Ek1(Trung+S) | 51(C ) | Ek2(51+C) | Giáoviên(C) | Ek3(giáo viên+ C) | 44000(S) | Ek4(44000+ S) |
| Hiếu(TS) | Ek1(Hiếu+TS) | 33(C ) | Ek2(33+C) | Bộ đội(TS) | Ek3(bộ đội+TS) | 74500(TS) | Ek4(74000+ S) |

**Một người dùngcó nhãn *S*, thì có thể đọc được những gì trong bảng trên sau khi đã gắn tem?**

Một người dùng có nhãn S thì đọc được đối tượng có mức nhạy cảm S và C.

**Kiểm tra tính toàn vẹn khi các người dùng truy xuất dữ liệu vào bảng trên:**

Khi người dùng lấy dữ liệu từ Database , TFE check stamp để kiểm tra tính toàn vẹn bằng cách giải mã tem và so sánh với dữ liệu nhận được. Nếu không trùng thì dữ liệu đã bị thay đổi.

## Khóa toàn vẹn

- Khóa toàn vẹn được đề xuất lần đầu tiên tại Viện nghiên cứu của Lực lượng Không quân về An toàn cơ sở dữ liệu [AF83], được dùng để kiểm soát tính toàn vẹn và sự truy nhập cho CSDL.

- Kiến trúc khóa toàn vẹn đã có trong hệ quản trị thương mại TRUDATA

### Đặc điểm

- TFE (Trusted Front End)thực thi bảo vệ nhiều mức bằng cách gắn các nhãn an toàn vào các đối tượng CSDL dưới dạng các tem – Stamps.

- Một tem là một trường đặc biệt của một đối tượng, lưu thông tin về nhãn an toàn và các dữ liệu điều khiển liên quan khác.

- Tem là dạng mã hóa của các thông tin trên, sử dụng một kỹ thuật niêm phong mật mã gọi là **Integrity Lock.**

- Một mô hình về khóa toàn vẹn cơ bản được chỉ ra như trên hình vẽ

|  |  |  |
| --- | --- | --- |
| Nhân viên an toàn | TS | 10FB |

Tổng kiểm tra

Tính nhạy cảm

Dữ liệu

- TFE gồm có

* Tem
* Mật mã

- Checksum = f (DL + Nhãn)

Tem = EK (Checksum)

**VD1:**

|  |  |  |
| --- | --- | --- |
| TenDuAn | NganSach | Level |
| DA1 | 100.000.000 | TS |
| DA2 | 10.000.000 | S |

=> Khóa toàn vẹn

|  |  |  |
| --- | --- | --- |
| TenDuAn | NganSach | Level |
| DA1 + EK1(DA1+TS) | 100.000.000 + EK1(100.000.000+TS) | TS |
| DA2 + EK2(DA2+S) | 10.000.000 + EK2(10.000.000+S) | S |

VD2:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MaSV | Label | Hoten | Label | GT | Label |
| 01 | S | Tú | TS | M | S |
| 02 | C | Giang | S | F | C |
| 03 | C | Minh | TS | F | S |

Gán tem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MaSV | Label | Hoten | Label | GT | Label |
| 01 + EK1 (01+S) | S | Tu + EK1 (Tu + TS) | TS | M + EK1(M+S) | S |
| 02 + EK2 (02+C) | C | Giang + EK2 (Giang + S) | S | F + EK2(F+C) | C |
| 03 + EK3 (03+C) | C | Minh + EK3 (Minh + TS) | TS | F + EK3(F+S) | S |

User Linh = S

|  |  |  |
| --- | --- | --- |
| MaSV | Hoten | GT |
| R, W | W | R, W |
| R | R, W | R |
| R | W | R, W |

Phòng Khoa(mã phòngkhoa, tên phòngkhoa)

Cán bộ(mã CB, tên CB, địa chỉ, mã phânviện, mã đềtài)

Đề tài ( mã đềtài, tên đềtài, mã tàiliệu)

Phân viện( mã phânviện, tên phânviện, mã đềtài)

Tài liệu( mã tàiliêu, tên tàiliệu)

create database hvktmm

create table phongkhoa

(

maphongkhoa char(5) primary key,

tenphongkhoa varchar(50) not null,

)

create table canbo

(

macanbo char(5) primary key,

ten char(10) not null,

diachi varchar(50) not null,

maphanvien char(5) not null,

madetai char(5) not null

)

create table detai

(

madetai char(5) Primary key,

tendetai char(50) not null,

matailieu char(50) not null,

)

create table phanvien

(

maphanvien char(5) Primary key,

tenphanvien char(50) not null,

madetai char(50) not null,

)

create table tailieu

(

matailieu char(5) primary key,

tentailieu varchar(50) not null,

)

alter table canbo add constraint fk\_maphanvien foreign key (maphanvien) references phanvien(maphanvien)

alter table canbo add constraint fk\_madetai foreign key (madetai) references detai(madetai)

alter table detai add constraint fk\_matailieu foreign key (matailieu) references tailieu(matailieu)

alter table phanvien add constraint fk\_madetai foreign key (madetai) references detai(madetai)

**Thực hiện các thiết kế an toàn:**

* Xem xét các yêu cầu an toàn bảo đảm tính sẵn sàng, bí mật và toàn vẹn cho hệ thống
* Xem xét xem cần bảo vệ bảng dữ liệu nào, những bảng nào có chứa dữ liệu nhạy cảm. Ví dụ như bảng điểm có trường điểm thi.
* Lựa chọn hệ quản trị csdl phù hợp có tính bảo mật cao như Oracle hay SQL
* Lựa chọn dùng các phương pháp để đảm bảo an toàn như:

+Dùng chính sách cấp quyền DAC, MAC

+Dùng cơ chế sao lưu phục hồi dữ liệu

+Dùng cơ chế mã hoá dữ liệu

BÀI TẬP KỸ THUẬT GIẤU Ô

Cho cơ sở dữ liệu thống kê về nhân viên của một công ty như sau:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Ten | ChucVu | Phong | Tuoi | GioiTinh | Luong |
| 01 | Nam | Nhân viên | Maketing | 29 | F | 3500 |
| 02 | Lan | Trưởng phòng | Maketing | 33 | F | 6200 |
| 03 | Huệ | Nhân viên | Kế hoạch | 27 | M | 4000 |
| 04 | Minh | Giám sát viên | Maketing | 24 | F | 3600 |
| 05 | Bình | Nhân viên | Tài vụ | 23 | F | 2000 |
| 06 | Hải | Nhân viên | Kế hoạch | 25 | M | 1500 |
| 07 | Hiền | Nhân viên | Tài vụ | 21 | F | 1700 |
| 08 | Thành | Nhân viên | Kế hoạch | 20 | M | 3000 |
| 09 | Trưởng | Phó phòng | Kế hoạch | 27 | M | 5000 |
| 10 | Bích | Nhân viên | Tài vụ | 33 | F | 1000 |
| 11 | Hoàng | Phó phòng | Kế hoạch | 35 | M | 2500 |
| 12 | Phượng | Nhân viên | Kế hoạch | 52 | F | 4500 |
| 13 | Cường | Trưởng phòng | Tài vụ | 34 | F | 6500 |
| 14 | Việt | Nhân viên | Maketing | 57 | F | 5000 |

Từ cơ sở dữ liệu quan hệ trên, ta có cơ sở dữ liệu thống kê tổng lương của các công nhân theo Phòng và theo độ tuổi như sau:

TONGLUONG

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tuổi | Phòng | | | Tổng lương |
|  | Kế hoạch | Maketing | Tài vụ |  |
| <27  27-30  >30 | 4500(2) | 3600(1) | 3700(2) | 11800 |
| 9000(2) | 3500(1) | 0(0) | 12500 |
| 7000(2) | 11200(2) | 7500(2) | 27200 |
| Tổng lương | 20500 | 18300 | 12700 | 51500 |

1. Theo kỹ thuật kiểm soát Giấu ô, bạn hãy giải thích quy tắc giấu ô đối với thống kê Count, và quy tắc với thống kê Sum “đáp ứng n, trội k%”. Cho ví dụ về quy tắc này.

2. Áp dụng quy tắc trên với n = 1, k = 90% để tìm các ô cần giấu trong bảng **TONGLUONG**

LG

\* Thống kê Count: Tiêu chuẩn nhạy cảm là kích cỡ tối thiểu của tập truy vấn. Các ô (Các thống kê) có giá trị ≤ kích cỡ tối thiểu này đều được giấu đi. Trường hợp các ô có kích cỡ = 1, người ta đã đề xuất 1 kỹ thuật để đối phó như sau: Kết hợp 1 ô có kích cỡ = 1 với 1 ô có kích cỡ > 1 gây mất mát thông tin 🡪 Kém hiệu quả.

\* Thống kê Sum: Tiêu chuẩn nhạy cảm được sử dụng là quy tắc đáp ứng n trội k%. Theo tiêu chuẩn này, một thống kê nhạy cảm nếu các giá trị thuộc tính của n hoặc ít hơn n (thực thể) bản ghi tạo thành k% hoặc > k% trong toàn bộ thống kê sum đó thì ô này cần được giấu đi.

n = 1

k = 90%

d = aj1 + … + ajn

q = q(C)

d > (k/100)q

Count(Phong=”Marketing” ˄ Tuoi<27) =1

=> Ô (1,2) là ô nhạy cảm cần giấu đi

Count(Phong=”Marketing” ˄ Tuoi=27-30) =1

=> Ô (2,2) là ô nhạy cảm cần giấu đi

Count(Phong=”Marketing” ˄ Tuoi>30) =1

=> Ô (2,3) là ô nhạy cảm cần giấu đi

Bảng SDB vĩ mô về nhân viên được giấu ô

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tuổi | Phòng | | | Tổng lương |
|  | Kế hoạch | Maketing | Tài vụ |  |
| <27  27-30  >30 | 4500(2) | - | 3700(2) (Dấu bổ sung) | 11800 |
| 9000(2) | - | - | 12500 |
| 7000(2) | 11200(2) | 7500(2) | 27200 |
| Tổng lương | 20500 | 18300 | 12700 | 51500 |

\* Dấu bổ sung

Vì các ô bị giấu vẫn có thể bị suy diễn ra bởi các ô khác trong bảng