CTT201

An toàn và Bảo mật Dữ liệu trong HTTT

Chương 2: Điều khiển truy cập (Access Control – AC)

Phần 1 - DAC

TS. Phạm Thị Bạch Huệ
Ths. Hoàng Anh Tú

Khoa Công nghệ thông tin – Đại học Khoa học tự nhiên

Có 3 kiểu điều khiển truy cập

- DAC (Discretionary Access Control)
  - Cho biết chủ thể nào có thể truy cập kiểu gì đến các đối tượng CSDL.
  - Có những nguyên tắc để 1 chủ thể có thể tùy ý cấp quyền hay lấy lại quyền cho/ từ 1 chủ thể khác.
- MAC (Mandatory Access Control)
  - Định trước các nguyên tắc để chủ thể (thuộc 1 lớp) truy cập trực tiếp hoặc gián tiếp đến các lớp dữ liệu.
- RBAC (Role-based Access Control
  - Vai trò là 1 tập các quyền. Không thực hiện cấp quyền cho từng chủ thể mà gán cho chủ thể 1 vai trò, khi đó chủ thể sẽ có tất cả các quyền thuộc vai trò đó.

Nội dung DAC

- Định nghĩa & Mô hình System R
- Câu lệnh GRANT
- Câu lệnh Revoke
- View & phân quyền trên nội dung
- Quyền khẳng định & Phủ định
- DAC - Ràng buộc ngữ cảnh
- Nhận xét DAC

ĐỊNH NGHĨA & MÔ HÌNH SYSTEM R

Định nghĩa

- DAC (Discretionary Access Control)

- **Điều khiển truy cập nhiệm ý** là điều khiển việc truy cập của chủ thể vào đối tượng thông qua định danh của chủ thể và các luật định trước.
- Cơ chế này được gọi là **nhiệm ý** là do cơ chế cho phép chủ thể có thể cấp quyền cho chủ thể khác truy cập các đối tượng của nó.

Điều khiển truy cập ở các HQT CSDL thương mại

- Tất cả các HQT CSDL thương mại đều cài đặt DAC.
- Các mô hình cấp quyền theo cơ chế DAC hiện tại đều dựa trên mô hình System R.

- **System R** (*Griffiths và Wade – 1976*), là một trong những mô hình ra đời đầu tiên cho hệ quản trị cơ sở dữ liệu quan hệ.
  System R: dựa trên nguyên lý *ủy quyền quản trị cho người sở hữu*.

Mô hình System R

- Các đối tượng được quản lý trong mô hình là table và view.
- Các phương thức truy cập (privilege): `select`, `insert`, `update`, `delete`, `drop`, `index` (*chỉ cho table*), `alter` (*chỉ cho table*).

- Hỗ trợ làm việc trên group, không hỗ trợ role.
- Dùng câu lệnh GRANT để cấp quyền, có tùy chọn GRANT OPTION.

Mô hình System R

- Sự ủy quyền được thể hiện thông qua `GRANT OPTION`, nếu cấp quyền cho 1 user bằng câu lệnh có `GRANT OPTION` thì user đó ngoài việc có thể thực hiện quyền được cấp, còn có thể cấp quyền đó cho các user khác.

- 1 user chỉ có thể cấp quyền trên 1 quan hệ cho trước nếu user đó là người sở hữu quan hệ hoặc user đó được người khác cấp quyền với câu lệnh có tuỳ chọn là `GRANT OPTION`.

CÂU LỆNH GRANT

Câu lệnh GRANT

- **GRANT** *PrivilegeList* | ALL[PRIVILEGES]
  ON *Relation* | *View*
  TO *UserList* | PUBLIC
  [**WITH GRANT OPTION**]

- Có thể cấp quyền (privilege) trên table hoặc view.
- Privilege áp dụng trên cả table (hoặc view).
- Đối với quyền cập nhật (update privilege), cần phải chỉ định quyền này áp dụng trên cột dữ liệu nào.

Câu lệnh GRANT – Ví dụ

- A: `GRANT select, insert ON NHANVIEN TO B WITH GRANT OPTION;`
  A: `GRANT select ON NHANVIEN TO C WITH GRANT OPTION;`
  B: `GRANT select, insert ON NHANVIEN TO C;`

- C có quyền select (từ A và B) và quyền insert (từ B).
- C có thể cấp quyền select cho các user khác, nhưng C không thể cấp quyền insert.

Câu lệnh GRANT

- Với từng user, hệ thống ghi nhận lại:
  - Tập A: tập hợp các quyền mà user sở hữu
  - Tập B: tập hợp các quyền mà user có thể ủy quyền cho người khác.

- Khi 1 user thi hành câu lệnh GRANT, hệ thống sẽ:
  - Tìm phần giao của tập B với tập quyền trong câu lệnh.
  - Nếu phần giao là rỗng, thì câu lệnh không được thi hành

Câu lệnh GRANT – Ví dụ

1) A: `GRANT select, insert ON NHANVIEN TO C WITH GRANT OPTION;`
2) A: `GRANT select ON NHANVIEN TO B WITH GRANT OPTION;`
3) A: `GRANT insert ON NHANVIEN TO B;`
4) C: `GRANT update ON NHANVIEN TO D WITH GRANT OPTION;`
5) B: `GRANT select, insert ON NHANVIEN TO D;`

Câu lệnh GRANT – Ví dụ

- Trong ví dụ trên, hãy cho biết:
  1) Câu lệnh nào được thực thi **hoàn toàn**?
  2) Câu lệnh nào **không** được thực thi?
  3) Câu lệnh nào thực thi **một phần**?

- **TRẢ LỜI:**
  1) 3 câu lệnh đầu tiên thực thi hoàn toàn vì A là người sở hữu table.
  2) Câu lệnh thứ 4 không thực thi vì C không có quyền update trên table.
  3) Câu lệnh thứ 5 thực thi một phần, B có quyền select, insert nhưng B không có grant option cho quyền insert nên D chỉ nhận được quyền select.

CÂU LỆNH REVOKE

Câu lệnh Revoke

```
REVOKE PrivilegeList| ALL[PRIVILEGES]
      ON Relation | View
      FROM UserList | PUBLIC
```

- Câu lệnh này dùng để thu hồi quyền đã cấp.
- User chỉ có thể thu hồi quyền mà user đã cấp.
- User không thể chỉ thu hồi grant option.
- Một người chỉ có thể bị thu hồi quyền truy xuất p khi tất cả những người cấp quyền p cho họ đều thu hồi quyền p lại.

Câu lệnh Revoke – Ví dụ

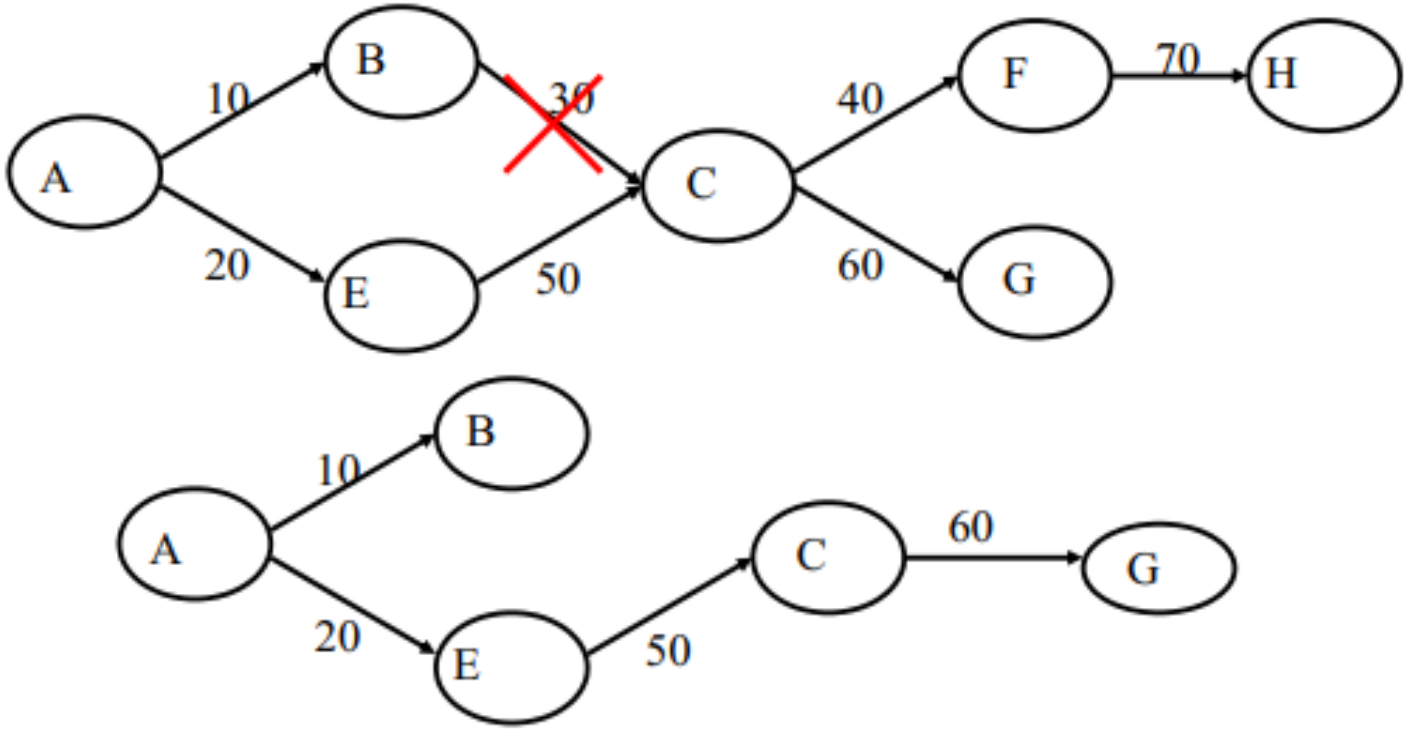
- A: `GRANT select ON NHANVIEN TO C WITH GRANT OPTION;`
- A: `GRANT select ON NHANVIEN TO B WITH GRANT OPTION;`
- C: `GRANT select ON NHANVIEN TO D;`
- B: `GRANT select ON NHANVIEN TO D;`
- C: **`REVOKE`** `select ON NHANVIEN FROM D;`

- Sau câu lệnh này thì D vẫn có quyền select trên bảng NHANVIEN vì B vẫn chưa thu hồi quyền select của D.

Câu lệnh Revoke – Đệ quy

- Thu hồi quyền đệ quy (recursive revocation): khi người dùng A thu hồi quyền truy xuất trên bảng của một người B thì tất cả các quyền mà B đã gán cho người khác đều bị thu hồi

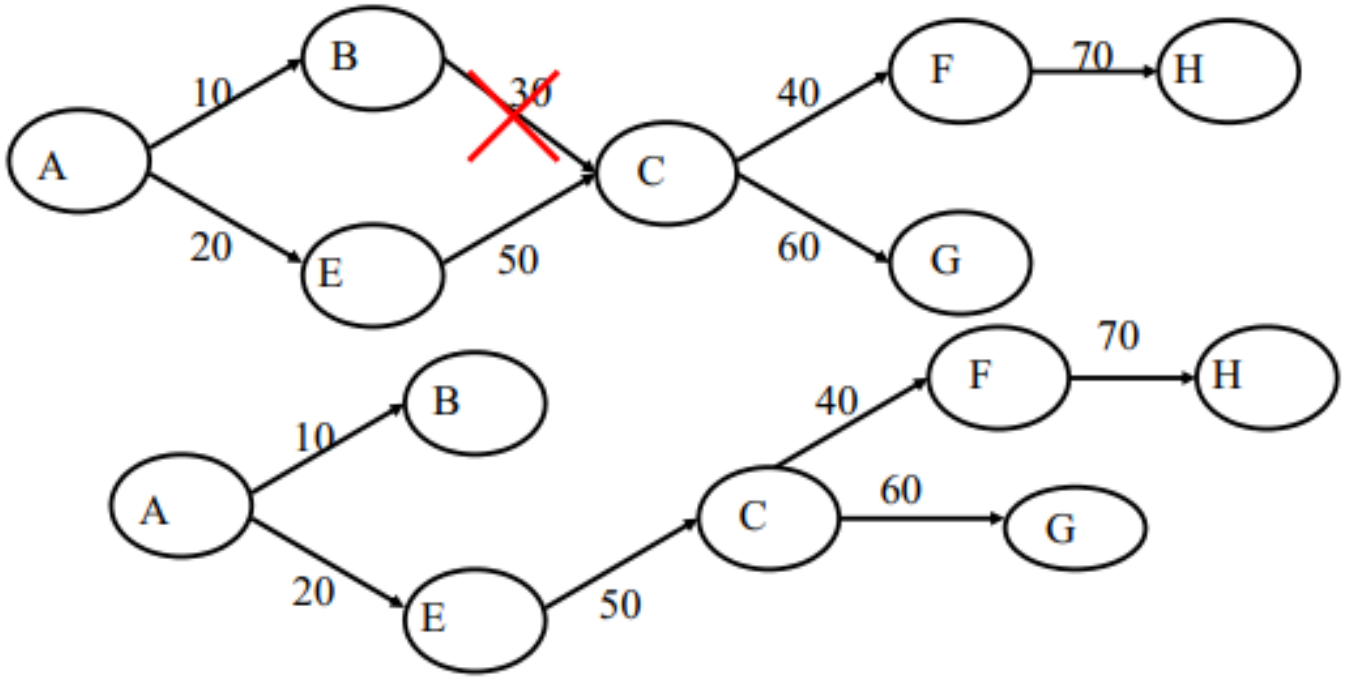
Thu hồi quyền đệ quy



Thu hồi quyền đệ quy

- Sẽ phá vỡ quyền truy xuất mà đối tượng bị thu hồi quyền đã cấp.
- Thực tế, khi 1 user A thay đổi công việc hay vị trí thì chỉ muốn lấy lại quyền truy xuất của A mà không muốn lấy lại các quyền truy xuất mà A đã cấp cho người khác.
- Thu hồi quyền đệ quy trong System R dựa vào nhãn thời gian mỗi lần cấp quyền truy xuất cho người dùng.
- Một biến thể của cách tiếp cận này là không dựa vào nhãn thời gian, mục đích là để tránh thu hồi quyền dây chuyền. Khi đó, nếu C bị B thu hồi quyền và C lại có quyền tương đương do người khác cấp (mặc dù sau đó) thì quyền mà C cấp cho người khác vẫn được giữ.

Một biến thể của thu hồi quyền đệ quy

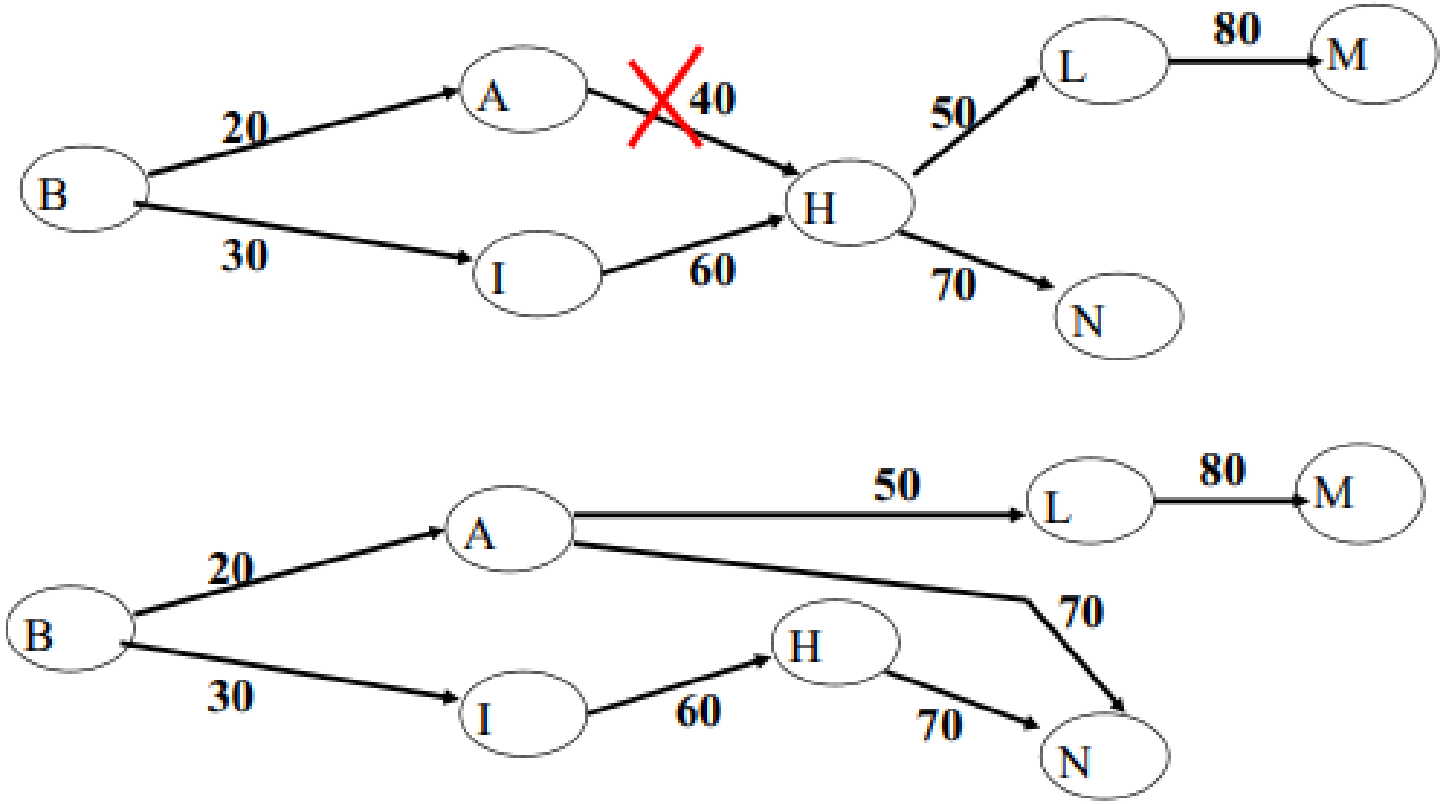


Thu hồi quyền không dây chuyền (Noncascading revoke)

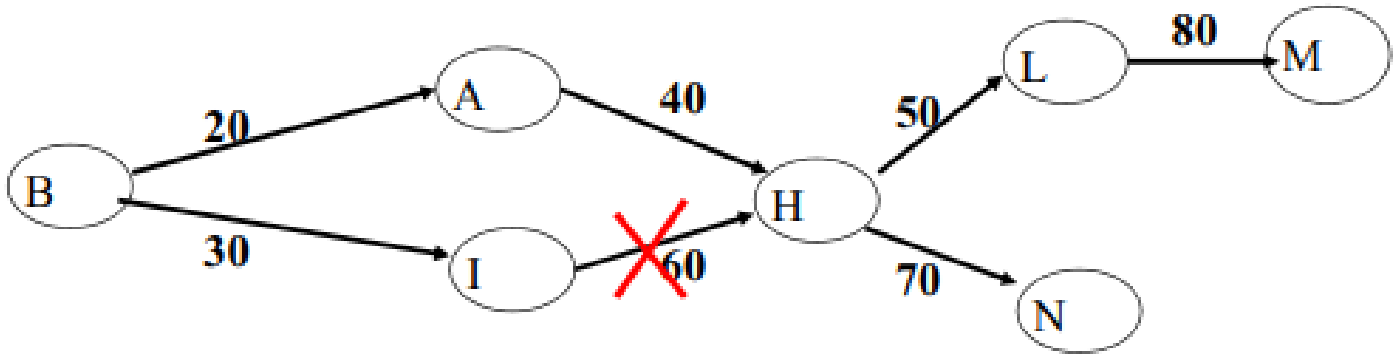
- Khi A thu hồi quyền truy xuất trên B thì tất cả quyền truy xuất mà B đã cấp cho chủ thể khác được thay bằng A đã cấp cho những chủ thể này.

- **Chú ý:**
  - Bởi vì B được cấp quyền truy xuất trên đối tượng từ nhiều chủ thể (khác A), nên không phải tất cả các quyền mà B cấp đều thay bằng A cấp. Và A được xem là người cấp thay cho B khi B sử dụng quyền A đã cấp cho mình cấp cho những chủ thể khác.
  - A sẽ là người cấp các quyền mà B đã cấp sau khi nhận quyền đó từ A có chỉ định `WITH GRANT OPTION`. Với những quyền B đã được cấp bởi C (≠ A), đến lượt B cấp cho người khác thì B vẫn là người cấp các quyền này.

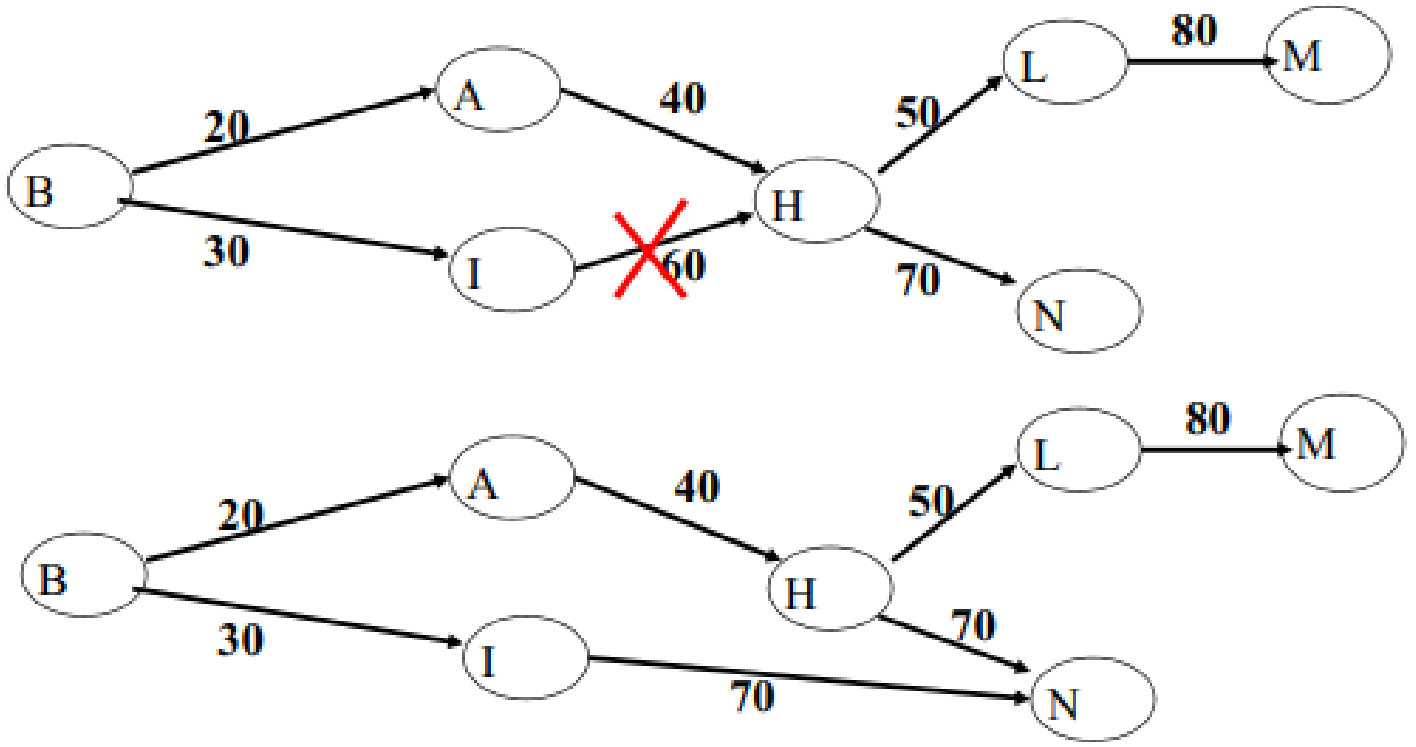
Thu hồi quyền không dây chuyền



Thu hồi quyền không dây chuyền



Thu hồi quyền không dây chuyền



Lưu ý, với quyền mà H cấp cho L, sau khi thu hồi quyền, không được thay I như là người cấp vì quyền này được cấp trước khi I cấp quyền cho H.

VIEW & PHÂN QUYỀN TRÊN NỘI DUNG

View và sự phân quyền dựa trên nội dung

- Trong các RDBMS, view là một cơ chế thường được dùng để hỗ trợ việc điều khiển truy cập dựa trên nội dung
- Dùng các vị từ (predicate) để giới hạn nội dung dữ liệu cần cấp quyền.
- Chỉ những bộ của quan hệ thỏa mãn vị từ được xem là các đối tượng để cấp quyền.

- Việc điều khiển truy cập dựa trên nội dung trong các RDBMS được thực hiện như sau:
  - Định nghĩa 1 view dùng các vị từ để chọn ra các dòng dữ liệu muốn cấp cho chủ thể S.
  - Cấp cho S các quyền trên view.

Ví dụ

- Ví dụ: giả sử ta muốn cấp quyền cho user B truy cập chỉ những nhân viên có lương ít hơn 20000:

- ```
  CREATE VIEW V_NHANVIEN AS
      SELECT * FROM NHANVIEN
      WHERE LUONG < 20000;
  ```

- ```
  GRANT Select ON V_NHANVIEN TO B;
  ```

Các bước xử lý truy vấn

- Parsing
- Catalog lookup
- Authorization checking

- View Composition
  - ✓ Truy vấn trên view sẽ được chuyển thành truy vấn trên bảng cơ sở thông qua bước này.
  - ✓ Kết quả sẽ dựa trên vị từ của câu truy vấn và vị từ định nghĩa nên view.
  - ✓ B:
```
SELECT * FROM V_NHANVIEN
    WHERE CONGVIEC = 'Lap trinh vien';
```
  - ✓ Câu truy vấn sau bước view composition:
```
SELECT * FROM NHANVIEN
WHERE LUONG < 20000 AND CONGVIEC = 'Lap trinh vien';
```

- Query optimization

Nhận xét

- Vì việc kiểm tra quyền được thực hiện trước bước view composition nên quyền được kiểm tra sẽ dựa trên view chứ không dựa trên các bảng cơ sở dùng để định nghĩa view.
- View hữu ích khi cấp quyền trên:
  - Các cột – chỉ cần tạo view gồm các cột muốn cấp quyền.
  - Dữ liệu thống kê (dữ liệu sinh ra từ các hàm AVG, SUM, …)

- Chủ thể truy cập có thể cấp quyền truy xuất hay thu hồi trên view tương tự như trên bảng dữ liệu.
- Người dùng muốn tạo View thì phải có quyền Select trên bảng dữ liệu.
- Nếu người tạo View bị thu hồi quyền (hay cấp quyền) trên bảng thì cũng bị thu hồi quyền (hay cấp quyền) trên View, và thu hồi những người dùng khác được người này cấp quyền.

View và cấp quyền dựa trên nội dung

- Người tạo view: view definer.

- Quyền mà view definer có trên view phụ thuộc vào:
  - Ngữ nghĩa của view hay các quan hệ cơ sở dùng để tạo nên view.
  - Quyền mà view definer có trên các bảng cơ sở.

- Quyền alter và index không thể áp dụng cho view, nên view definer không bao giờ có quyền này trên view.

A:
```
CREATE VIEW V1 (MANV, TONGTIEN)
    AS SELECT MANV, LUONG+THUONG
        FROM NHANVIEN WHERE CONGVIEC = 'Lap trinh vien'
```

Thao tác update không được định nghĩa trên trường TONGTIEN của view nên A sẽ không thể có quyền update trên trường này.

Phân quyền trên view

- Để xác định quyền mà view definer có trên view, hệ thống phải:
  - Tìm giao tập quyền mà view definer có trên các quan hệ cơ sở với tập quyền ứng với các thao tác có thể thực hiện trên view.

Ví dụ

- Xét quan hệ NHANVIEN và giả sử A là người tạo nên quan hệ NHANVIEN
  A: `GRANT Select, Insert, Update ON NHANVIEN to D;`
  D: `CREATE VIEW V1 AS SELECT MANV, LUONG FROM NHANVIEN;`
  D: `CREATE VIEW V2 (MANV, LUONG_NAM) AS SELECT MANV, LUONG*12 FROM NHANVIEN;`

- D có thể thực hiện tất cả các quyền trên V1 như là các quyền mà D có trên quan hệ NHANVIEN, đó là Select, Insert, Update
- Tuy nhiên, D chỉ có thể thực hiện trên V2 quyền Select và Update trên cột MANV.

Quyền trên view – ví dụ

- Hoàn toàn có thể cấp quyền trên view.
  - Quyền mà user có thể cấp là những quyền mà user có `with grant option` trên các quan hệ cơ sở.

- Ví dụ trên: user D không thể cấp bất cứ quyền gì trên view V1 và view V2 mà D đã định nghĩa vì D không được chỉ định `with grant option` khi D được cấp quyền.

Ví dụ

- Xét các câu lệnh sau:
  - A: `GRANT Select ON NHANVIEN TO D WITH GRANT OPTION;`
  - A: `GRANT Update, Insert ON NHANVIEN TO D;`
  - D: `CREATE VIEW V4 AS SELECT MANV, LUONG FROM NHANVIEN;`

- Quyền của D trên V4 sẽ là:
  - Select with Grant Option;
  - Update, Insert without Grant Option;

QUYỀN KHẲNG ĐỊNH & PHỦ ĐỊNH

DAC – Quyền khẳng định & Phủ định

- System R và hầu hết các HQT dùng chính sách đóng.
  - Với chính sách đóng, thiếu quyền truy xuất ≡ không có quyền truy xuất.

- Khi chủ thể truy xuất đến 1 đối tượng dữ liệu, HT kiểm tra trong danh sách quyền mà chủ thể được truy xuất, nếu không có thì truy xuất bị từ chối.
  - Hạn chế: Việc thiếu quyền truy xuất *không ngăn cấm* chủ thể sẽ nhận quyền này từ chủ thể khác.
  - **Ví dụ:** ***x*** không được quyền truy xuất trên đối tượng ***o***, nhưng trong trường hợp hệ thống sử dụng chính sách phân chia quyền quản trị thì chủ thể có quyền cấp quyền truy xuất trên ***o*** vô tình cấp quyền cho ***x***.

- Người ta đã đưa ra ***quyền phủ định*** để giải quyết vấn đề rang buộc này.

DAC – Quyền khẳng định & Phủ định

- *Quyền khẳng định*: danh sách quyền truy xuất được sử dụng.
- *Quyền phủ định*: danh sách quyền truy xuất không được sử dụng.
- Tuy nhiên sử dụng Quyền khẳng định và Quyền phủ định thì gây nên xung đột.

- *Ví dụ:*
  A có quyền WRITE trên bảng NHANVIEN.
  A không được READ trên PHONGBAN.
  A không được WRITE trên thuộc tính LUONG của NHANVIEN.

- Thường người ta giải quyết xung đột bằng cách ưu tiên quyền phủ định.

DAC – Quyền khẳng định & Phủ định

- *Quyền phủ định* được thực hiện như là chặn quyền.
- Khi chủ thể bị gán quyền phủ định trên đối tượng thì quyền khẳng định trên đối tượng mà họ có trước đó bị chặn lại.

- Nếu sau này chủ thể được rút quyền phủ định thì họ có thể sử dụng lại quyền khẳng định của mình trước đó.
  - Ưu điểm:
    - Thứ nhất nếu vô tình gán quyền phủ định cho người dùng thì có thể thu hồi lại.
    - Thứ hai là có thể chặn quyền truy cập của chủ thể trong một thời gian bằng cách gán quyền phủ định và sau đó thu hồi lại.

Câu lệnh DENY

- **DENY** {ALL [PRIVILEGES] | *permission*[,…*n*]}
  { [(*column*[,…*n*])] ON { *table* | *view*} |
  ON {*table* | *view*} [(*column*[,…*n*])] |
  {*procedure* | *extended_procedure*} }
  TO *security_account*

- Ví dụ:

```
DENY SELECT, INSERT, UPDATE
ON NHANVIEN
TO A, B
```

Thu hồi quyền đã cấp hoặc quyền đã cấm (Revoking Granted and Denied Permissions)

- **Dùng câu lệnh REVOKE:**
  - **Thể thu hồi lại quyền khẳng định (quyền đã cấp với lệnh GRANT)**
  - **Thu hồi lại quyền phủ định (quyền đã cấm với câu lệnh DENY)**

- Câu lệnh REVOKE giống lệnh DENY ở chỗ không cho thực hiện điều gì đó.
- Câu lệnh REVOKE khác lệnh DENY ở chỗ REVOKE sẽ thu lại quyền đã cấp, còn DENY sẽ cấm một chủ thể (hoặc 1 vai trò) thực hiện một quyền gì đó trong tương lai.

DAC - RÀNG BUỘC NGỮ CẢNH

DAC - Ràng buộc ngữ cảnh

- Thực tế, người dùng chỉ được phép truy cập dữ liệu trong 1 khoảng thời gian nhất định.
- Cần phải có một cơ chế hỗ trợ việc truy xuất trong khoảng thời gian cho trước.
  Ví dụ: cơ chế chỉ cho phép những người làm việc bán thời gian chỉ được phép truy cập dữ liệu vào khoảng từ 9am đến 1pm từ ngày 1/1/98.

- Trong hầu hết các hệ quản trị cơ sở dữ liệu chính sách này thường triển khai ở chương trình ứng dụng.
  Hạn chế: khi xác nhận và thay đổi chính sách điều khiển truy cập, không bảo đảm rằng chính sách này thực thi.

- Mô hình điều khiển truy cập dựa vào thời gian được đề xuất để giải quyết vấn đề này.

DAC - Ràng buộc ngữ cảnh

- **Thời gian hiệu lực:**
  - ✓ Mỗi quyền truy xuất đều có khoảng thời gian hiệu lực
  - ✓ Khi hết thời gian hiệu lực thì quyền truy xuất tự động bị thu hồi tự động (không cần người quản trị thu hồi).

- **Chu kỳ sử dụng quyền truy xuất:**
  - ✓ Quyền truy xuất theo chu kỳ có thể là quyền khẳng định hay quyền phủ định.
  - ✓ Nếu trong cùng một khoảng thời gian mà người dùng vừa có quyền khẳng định vừa có quyền phủ định trên cùng một đối tượng và cùng phương thức truy cập thì ưu tiên cho quyền phủ định.

DAC - Ràng buộc ngữ cảnh

- **Cơ chế suy diễn dựa vào quy tắc suy diễn**
  - ✓ Quy tắc suy diễn biểu thị ràng buộc của các quyền truy xuất theo thời gian.
  - ✓ Quy tắc cho phép suy ra quyền truy xuất mới dựa sự tồn tại hay không tồn tại của quyền truy xuất khác trong khoảng thời gian xác định.
  - ✓ Bằng cách sử dụng quy tắc suy diễn đã đáp ứng được yêu cầu bảo vệ dữ liệu một cách ngắn gọn và rõ ràng.

- Ví dụ: Nếu hai người dùng làm chung một dự án thì phải cùng quyền truy xuất trên các đối tượng.

DAC - Ràng buộc ngữ cảnh

- Quyền truy xuất được định nghĩa là một bộ gồm 5 thuộc tính auth = (s, o, m, pn, g).
  Trong đó:
  – **s** (chủ thể), **g** (người gán) U (danh sách người dùng).
  – **m** ∈ M (phương thức truy cập).
  – **o** ∈ O (đối tượng).
  – **pn** ∈ {+ , -} (quyền khẳng định, quyền phủ định).

- *Ví dụ:*
  `(B, o1, read, +, C)` : C gán quyền cho B có thể read đối tượng o1.
  `(B, o1, write, -, C)` : C không cho phép B được quyền write trên đối tượng o1.

DAC - Ràng buộc ngữ cảnh

- • Quyền truy xuất theo chu kỳ là bộ ba ([begin,end], P, auth). Trong đó:
  - **begin** là ngày bắt đầu.
  - **end** là ngày kết thúc và lớn hơn ngày bắt đầu hay có thể là .
  - **P** là biểu thức chu kỳ.
  - **auth** là quyền truy xuất.
- Quyền truy xuất theo chu kỳ thể hiện quyền truy xuất có hiệu lực trong chu kỳ P với ngày sử dụng quyền lớn hơn hay bằng tb (ngày bắt đầu) và nhỏ hơn hay bằng te (ngày kết thúc).

- *Ví dụ 2:* `A1= ([1/1/94, ], Mondays, (A, o1, read, +, B))` quyền truy xuất này được B gán, thể hiện A có quyền read trên đối tượng o1 vào các ngày thứ hai bắt đầu từ ngày 1/1/94.

DAC - Ràng buộc ngữ cảnh

- Khi sử dụng quyền phủ định thì có thể dẫn đến hiện tượng xung đột .

- Ví dụ: Cho 2 quyền truy xuất được gán tuần tự như sau:
  - `A2 = ([1/1/95, ], Working-days, (A, o1, read, -, B))`
  - `A1= ([1/1/94, ], Mondays, (A, o1, read, +, B)).`
  - Lúc này bắt đầu từ ngày 1/1/95 thì A vừa có quyền phủ định vừa có quyền khẳng định trên cùng đối tượng o1 và cùng phương thức read.
  - Hiện tượng xung đột được giải quyết bằng cách ưu tiên quyền phủ định.
  - Do đó trong, khoảng thời gian [1/1/94, 12/31/94] thì A vẫn có quyền read trên đối tượng o1 vào ngày thứ hai, tuy nhiên từ ngày 1/1/95 thì A không được quyền read trên đối tượng này vào các ngày làm việc kể cả thứ hai.

DAC - Ràng buộc ngữ cảnh

- Quy tắc suy diễn được định nghĩa là bộ ba `([tb, te], P, A <OP> β )`. Trong đó:
  – **tb** là ngày bắt đầu.
  – **te** là ngày kết thúc và lớn hơn ngày bắt đầu hay có thể là .
  – **P** là biểu thức chu kỳ,
  – **A** là quyền truy xuất.
  – β là biểu thức Bool của các quyền truy xuất.
  – **OP** là một trong các toán tử WHENEVER, ASLONGAS, UPON.

- Ngữ nghĩa của từng toán tử trong quy tắc suy diễn:
  `([tb, te], P, A WHENEVER β )`: quyền truy xuất A có hiệu lực vào thời điểm t ∈ chu kỳ P và t ∈ [tb, te] khi A có hiệu lực.

DAC - Ràng buộc ngữ cảnh

Ví dụ:

- A1 = ([1/1/95,1/1/96], Working-days, (M, o1, read, B))
- R1= ([1/1/95,], Summer-time, (S, o1, read, +, Bob) WHENEVER (M, o1, read, +, B)).

→ S chỉ được read trên đối tượng o1 vào thời điểm mùa hè, từ ngày 1/1/95 khi M được read trên đối tượng này.

NHẬN XÉT DAC

Nhận xét DAC

- **Ưu điểm:**
  – DAC linh động trong chính sách nên được hầu hết các HQT CSDL ứng dụng.

- **Khuyết điểm:**
  – Không thể điều khiển được dòng thông tin (information flow control) để có thể chống lại tấn công dạng Trojan Horse.

Câu hỏi

TS. Phạm Thị Bạch Huệ - ptbhue@fit.hcmus.edu.vn

Ths. Hoàng Anh Tú – hatu@fit.hcmus.edu.vn