Chương 4:
Điều khiển truy cập tùy quyền
Discretionary Access Controls (DAC)

Khoa Khoa học và Kỹ thuật Máy tính
Đại học Bách Khoa Tp.HCM

Nội dung

1. Giới thiệu về điều khiển truy cập tùy quyền
2. Mô hình điều khiển truy cập tùy quyền
3. Điều khiển dữ liệu với SQL
4. DAC và điều khiển dòng thông tin

Giới thiệu DAC

- ***Điều khiển truy cập tùy quyền (Discretionary Access Control -DAC)***:
  - Người dùng có thể bảo vệ những gì thuộc về mình
  - Chủ của dữ liệu sẽ có toàn quyền trên dữ liệu đó
  - Chủ của dữ liệu có quyền định nghĩa các loại truy cập đọc/ghi/thực thi (read/write/execute/…) và gán những quyền đó cho những người dùng khác.

Giới thiệu DAC

- Cách thức cơ bản điều khiển truy cập của DAC trong một hệ cơ sở dữ liệu (HCSDL) là dựa vào 2 thao tác cơ bản:
  - ***Gán quyền (granting privileges)***: cho phép người dùng khác được quyền truy cập lên đối tượng do mình làm chủ
  - ***Thu hồi quyền (revoking privileges)***: thu hồi lại quyền đã gán cho người dùng khác

Các loại quyền trong DAC

- ***Quyền ở cấp tài khoản/hệ thống (account/system level)***: là những quyền này độc lập với các đối tượng trong HCSDL. Những quyền này do người quản trị hệ thống định nghĩa và gán cho mỗi người dùng
- ***Quyền ở cấp đối tượng (object level)***: là những quyền trên mỗi đối tượng trong hệ CSDL. Người dùng tạo ra đối tượng nào thì sẽ có tất cả các quyền trên đối tượng đó.

Các loại quyền trong DAC

- Quyền ở cấp tài khoản/hệ thống: gồm có các quyền
  - **CREATE SCHEMA**: tạo lược đồ CSDL
  - **CREATE TABLE**: tạo bảng dữ liệu/ quan hệ (relation)
  - **CREATE VIEW**: tạo view
  - **ALTER**: chỉnh sửa các schema/relation
  - **DROP**: xóa relation/view
  - **MODIFY**: quyền thêm/ xóa/ sửa các hàng dữ liệu (record/ tuple)
  - **SELECT**: quyền thực hiện câu truy vấn thông tin trong CSDL

Các loại quyền trong DAC

- Quyền ở cấp đối tượng: gồm các đối tượng dữ liệu và các loại truy cập mà người dùng được phép thực hiện trên đối tượng đó.
  - Các đối tượng dữ liệu này gồm: các **relation** hoặc **view**
  - Các thao tác gồm:
    - **INSERT**: thêm dữ liệu vào relation
    - **UPDATE**: cập nhật /chỉnh sửa dữ liệu trong relation
    - **DELETE**: xóa dữ liệu trong relation
    - **REFERENCE**: tham khảo đến dữ liệu trong relation

Nội dung

1. Giới thiệu điều khiển truy cập tùy quyền
2. Mô hình điều khiển truy cập tùy quyền
3. Điều khiển dữ liệu với SQL
4. DAC và điều khiển dòng thông tin

Mô hình bảo mật

- Mô hình bảo mật (Security model)
- Mô hình ma trận truy cập (Access matrix model)
- Mô hình Take-Grant (Take-Grant model)

Mô hình bảo mật

- **Mô hình bảo mật (Security model)** cung cấp một cách biểu diễn giàu ngữ nghĩa cho các thuộc tính cấu trúc và thuộc tính chức năng (functional and structural properties) của một hệ thống bảo mật.
- Mô hình bảo mật giúp biểu diễn được các đặc tả yêu cầu về bảo mật cho một hệ thống.
- Mô hình bảo mật là mô hình ý niệm cấp cao (high-level conceptual model) và độc lập với các phần mềm.
- Mô hình bảo mật có thể dùng để chứng minh các tính chất cần có của bảo mật hệ thống thông tin.

Mô hình bảo mật

- Có 2 loại mô hình bảo mật:
  - Mô hình bảo mật tùy quyền (discretionary model)
  - Mô hình bảo mật bắt buộc (non-discretionary model or mandatory model)

Mô hình bảo mật tùy quyền

- **Mô hình bảo mật tùy quyền**, hoặc mô hình điều khiển truy cập tùy quyền (DAC model), quản lý và điều khiển các truy cập của người dùng đến các thông tin dựa vào danh định của người dùng và tập các luật điều khiển truy cập. Luật điều khiển truy lập định nghĩa với mỗi người dùng và đối tượng (object), sẽ có quy định các loại truy cập mà người dùng được phép làm trên đối tượng đó.
- Khi người dùng yêu cầu truy cập đến một đối tượng, một bộ phận định quyền (authorization module) sẽ kiểm tra xem người dùng đó có được phép truy cập không. Nếu có thì cho phép, còn không thì từ chối

Mô hình bảo mật tùy quyền

- Mô hình ma trận truy cập (Access matrix model)
- Mô hình Take-Grant (Take-Grant model)
- Mô hình Acten (Action-Entity model)
- Mô hình Wood và cộng sự (Wood et al. model)

Mô hình ma trận truy cập

- Là mô hình bảo mật được dùng cho cả cấp hệ điều hành và cấp cơ sở dữ liệu.
- Mô hình được đề nghị bởi Lampson (1971), và được Graham và Denning mở rộng (1972).
- 1976, Harrison và các cộng sự đã trình bày mô hình ma trận truy cập một cách có hệ thống.

Mô hình ma trận truy cập

- Ma trận truy cập là ma trận giữa các chủ thể (subject), các đối tượng (object) và các quyền tương ứng giữa của chủ thể với đối tượng.
- Trạng thái định quyền (Authorization state)

$$Q = (S, O, A)$$

- *S* (**S**ubjects): là tập các chủ thể - các thực thể chủ động (active entity) sử dụng các nguồn tài nguyên của hệ thống.

  Ví dụ: người dùng, nhóm các người dùng (group), quá trình (process)

Mô hình ma trận truy cập

- Trạng thái định quyền

$$Q = (S, O, A)$$

- *O* (**O**bjects): là tập các đối tượng - các thực thể cần được bảo vệ, bao gồm các thực thể bị động (passive object) như tài nguyên hệ thống và các chủ thể
  - Ví dụ: ở cấp hệ điều hành: file, bộ nhớ, segments, quá trình

    ở cấp CSDL: CSDL, quan hệ, thuộc tính, hàng, trường dữ liệu của hàng

Mô hình ma trận truy cập

- Trạng thái định quyền

  $Q = (S, O, A)$

- *A* (Access matrix): là ma trận truy cập.
  - Hàng: các chủ thể
  - Cột: các đối tượng
  - Mỗi ô *A[s,o]* chứa các chế độ truy cập mà chủ thể *s* được quyền làm trên đối tượng *o*

	O_1	…	O_i	…	O_m
S_1	$A[s_1,o_1]$		$A[s_1,o_i]$		$A[s_1,o_m]$
…					
S_i	$A[s_i,o_1]$		$A[s_i,o_i]$		$A[s_i,o_m]$
…					
S_n	$A[s_n,o_1]$		$A[s_n,o_i]$		$A[s_n,o_m]$

Mô hình ma trận truy cập

- Trong hệ CSDL, *A[s,o]* còn chứa các điều kiện cần thỏa để chủ thể *s* có thể truy cập đối tượng *o*
  - Phụ thuộc dữ liệu (data-dependent): chỉ xem được thông tin của các nhân viên có *salary < 1000* trong bảng *Employee*
  - Phụ thuộc thời gian (time-dependent): chỉ được truy cập bảng Employee từ 8:00 sáng đến 5:00 chiều
  - Phụ thuộc ngữ cảnh: có thể truy cập riêng từng thuộc tính *name* và *salary* trong bảng *Employee*, nhưng không thể truy cập cả 2 thuộc tính này cùng lúc.
  - Phụ thuộc lịch sử: chỉ xem được thuộc tính *salary* của các nhân viên nếu như trước đó chưa xem thuộc tính *name* của nhân viên.

Một ví dụ khác của ma trận truy cập

	Asset 1	Asset 2	file	device
Role 1	read, write, execute, own	execute	read	write
Role 2	read	read, write, execute, own		

Cách hiện thực mô hình ma trận truy cập

- Cách hiện thực mô hình:
  - *S → {(O,A)}:* danh sách khả năng (capability list - CL)

    Alice → {(file X, {read, delete}), (file Y, {update})}
  - *O → {(S,A)}:* danh sách điều khiển truy cập (access control list - ACL)

    File X → {(Alice, {read, delete}), (Bob, {read})}
- Ưu điểm và khuyết điểm:
  - CL: khi cần tìm các chủ thể có thể truy cập đến một đối tượng *o* → duyệt tất cả danh sách
  - ACL: ngược lại

Mô hình ma trận truy cập

a)

S	O	A[s,o]
P_1	F_1	Own, R, W
P_1	F_2	E, R^+
P_1	M_1	R, W, E
P_1	P_2	Ctrl
P_2	F_1	R, W
P_2	M_1	W
P_3	F_1	R, W, E^+
P_3	F_2	Own, E
P_3	M_2	W, E

(a) <S_i, O_j, A[S_i, O_j]>

(b)

P_1	
P_2	
P_3	

F_1	Own, R, W
F_2	E, R^+
M_1	R, W, E
P_2	Ctrl

F_1	R, W
M_1	W

F_1	R, W, E^+
F_2	Own, E
M_2	W, E

(b) CL

(c)

F_1	
F_2	
M_1	
M_2	
P_1	
P_2	
P_3	

P_1	Own, R, W
P_2	R, W
P_3	R, W, E^+

P_1	E, R^+
P_2	Own, E

P_1	R, W, E
P_2	W

P_3	W, E

P_1	Ctrl

(c) ACL

Ví dụ:

- Hệ thống có 3 người dùng:
  - Alice: tạo ra file 1
  - John: tạo ra file 2
  - Sally: tạo ra file 3
- Một file có ba quyền là Đọc (R), Ghi (W), Thực thi (E)
- Các người dùng cấp quyền trên các file cho các người dùng khác:
  - Alice cấp quyền đọc, ghi cho John trên file 1 và chỉ quyền đọc trên file này cho Sally.
  - John cấp quyền đọc, thực thi trên file 2 cho Alice.
  - Sally cấp quyền đọc, thực thi trên file 3 cho Alice và quyền đọc, thực thi trên file này.
- Vẽ ma trận truy cập, danh sách khả năng, danh sách điều khiển truy cập.

Mô hình bảo mật tùy quyền

- Mô hình ma trận truy cập (Access matrix model)
- **Mô hình Take-Grant (Take-Grant model)**
- Mô hình Acten (Action-Entity model)
- Mô hình Wood và cộng sự (Wool et al. model)

Mô hình Take-Grant

- Johns và các cộng sự đề nghị mô hình Take-Grant năm 1976
- Sử dụng các cấu trúc hình học để biểu diễn mối quan hệ về quyền giữa các chủ thể với đối tượng, giữa chủ thể với chủ thể và giữa đối tượng với đối tượng.
- Có thể được xem là một dạng mở rộng của mô hình ma trận truy cập

Mô hình Take-Grant

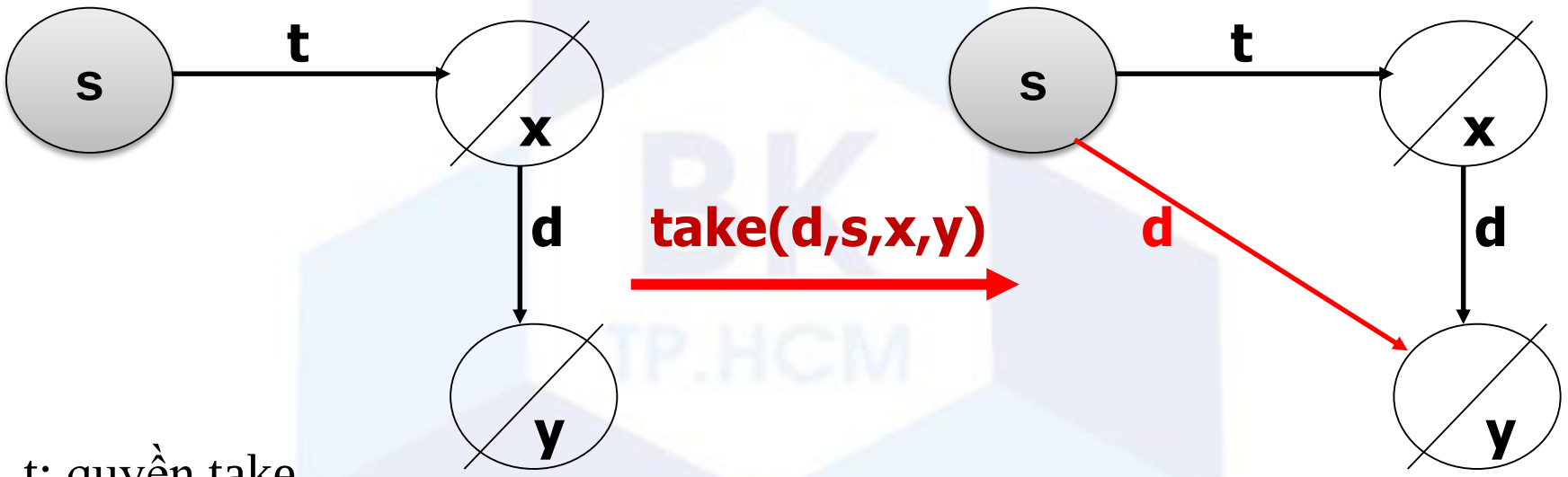
- Trạng thái định quyền:

  $$G = (S, O, E)$$

  - *S:* tập các chủ thể (người dùng, quá trình, chương trình)
  - *O:* tập các đối tượng bị động (file, bộ nhớ, CSDL, bảng, hàng, trường dữ liệu)
  - $V = S \cup O$: tập các đỉnh, $S \cap O = \Phi$
  - *E*: tập các cung được đánh nhãn

Thao tác Take và Grant

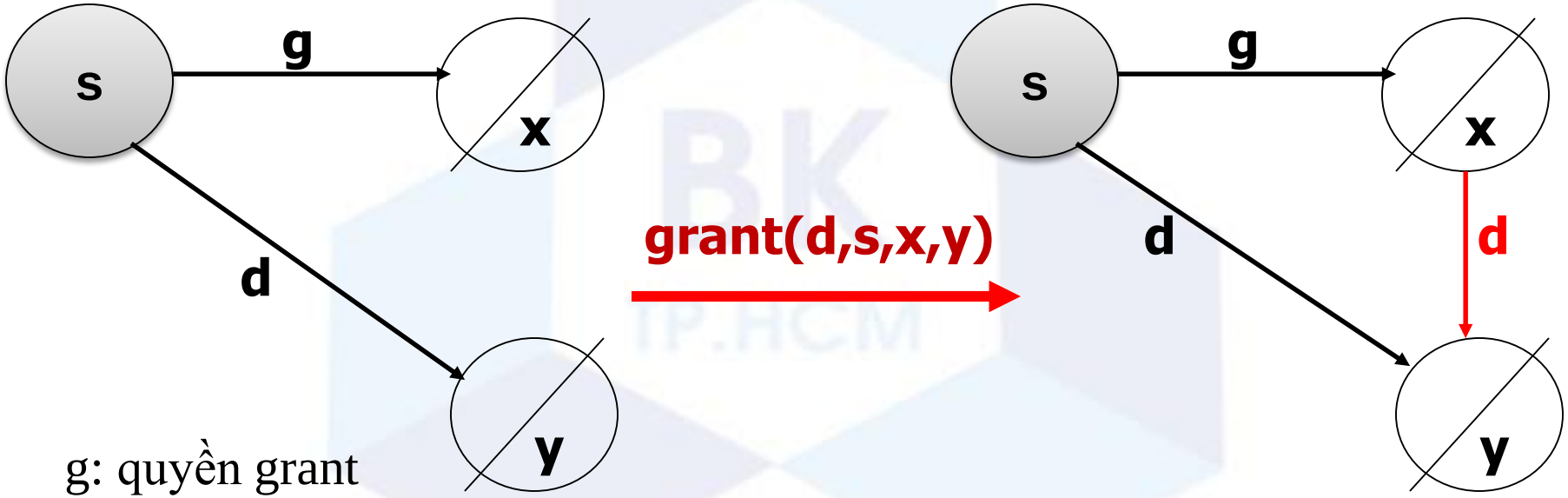
- **take(d, s, x, y)**: chủ thể *s* lấy quyền *d* trên đối tượng/chủ thể *y* từ đối tượng/chủ thể *x*



t: quyền take

Thao tác Take và Grant

- **grant(d, s, x, y)**: chủ thể *s* gán quyền *d* trên đối tượng/chủ thể *y* cho đối tượng/chủ thể *x*



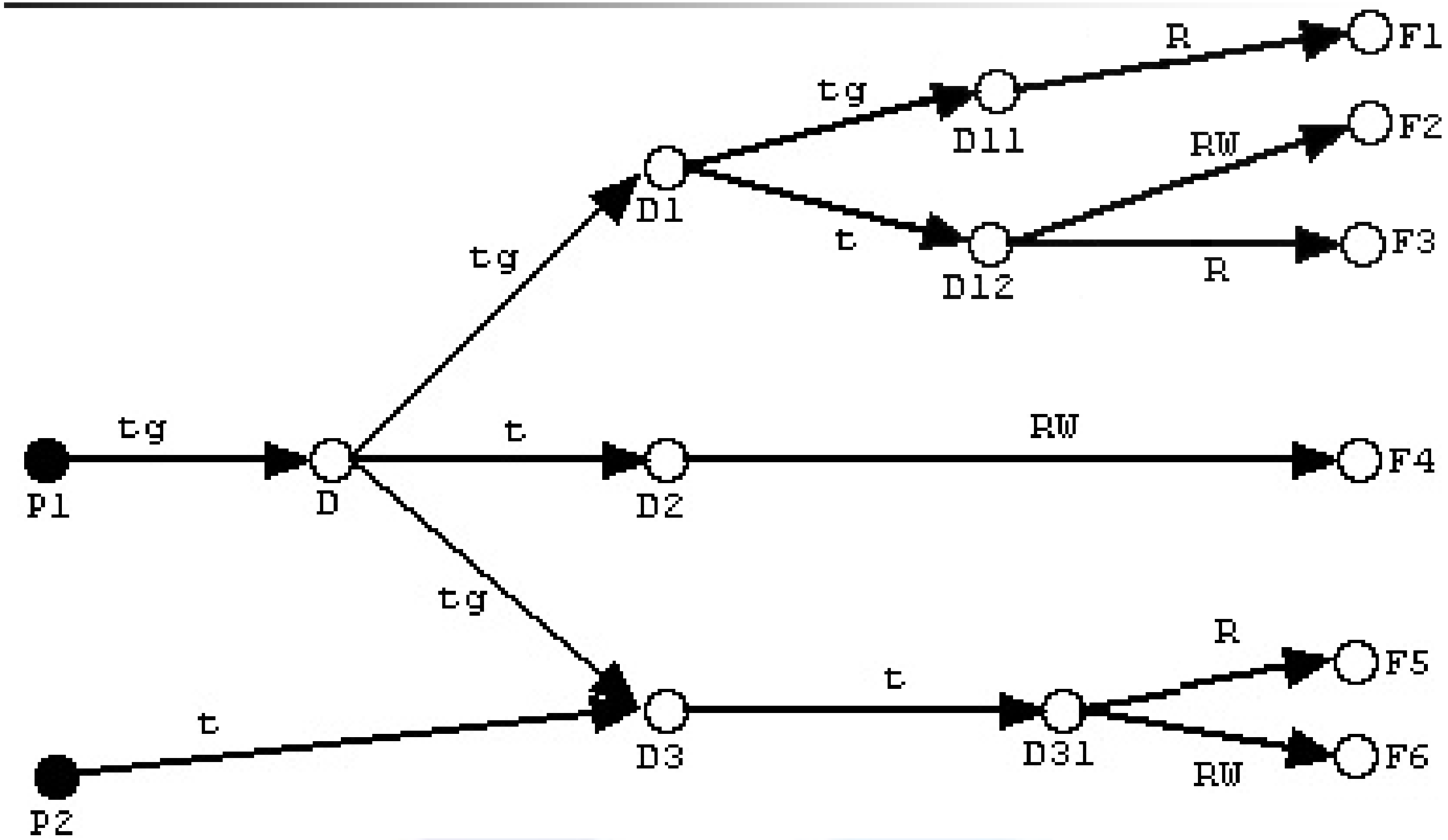
Mô hình Take-Grant

- Các loại quyền truy cập: read, write, take, grant
  - read, write: không làm thay đổi trạng thái định quyền
  - take, grant: làm thay đổi trạng thái định quyền
- Các loại thao tác truyền quyền: take, grant, create, remove
  - take, grant: lấy và gán quyền
  - create(s, x): chủ thể *s* tạo đối tượng/chủ thể *x*. Khi đó cung nối giữa *s* và *x* sẽ được đánh nhãn *p (possess: sở hữu)*
  - $remove_p$(s, x): chủ thể *s* bị thu hồi quyền *p* trên đối tượng/chủ thể *x*

Mô hình Take-Grant

- Khuyết điểm của mô hình Take-Grant:
  - Không có tính chọn lọc của các quyền quản lý:
    - Tất cả các quyền của *s* đều có thể bị truyền đi nếu *s* sở hữu quyền GRANT
    - Tất cả các quyền của *o/s* đều có thể bị lấy đi (truyền đi) nếu có một quyền TAKE trên nó.
  - Không quản lý được sự lan truyền quyền
  - Tính không cục bộ: nếu *s* có quyền GRANT trên *o* thì *s* có thể truyền bất kỳ quyền gì của mình cho *o*. Như vậy không kiểm soát được tập quyền có thể có trên *o*.
  - Khả năng lan truyền ngược của dòng di chuyển quyền

Take-grant representation of directory structure



Mô hình bảo mật tùy quyền

- Mô hình ma trận truy cập (Access matrix model)
- Mô hình Take-Grant (Take-Grant model)
- **Mô hình Acten (Action-Entity model)**
- **Mô hình Wood và cộng sự (Wool et al. model)**
  - Tham khảo: S. Castano, M. Fugini, G. Martella, and P. Samarati (1995). ***Database Security***, ACM Press & Addison-Wesley, ISBN 0-201-59375-0.

Nội dung

1. Giới thiệu về điều khiển truy cập tùy quyền
2. Mô hình điều khiển truy cập tùy quyền
3. Điều khiển dữ liệu với SQL
4. DAC và điều khiển dòng thông tin

Điều khiển dữ liệu với SQL

- Hai câu lệnh cơ bản:
  - GRANT
  - REVOKE
- Dựa trên 3 đối tượng chính trong CSDL:
  - người dùng
  - Các đối tượng CSDL
  - Các quyền: lấy dữ liệu (SELECT), chỉnh sửa (INSERT, UPDATE, DELETE), và tham khảo (REFERENCE)

Điều khiển dữ liệu với SQL

- **GRANT:** truyền những quyền trên các đối tượng dữ liệu của mình cho những người dùng khác

  GRANT <danh sách các quyền>

  ON <các đối tượng dữ liệu>

  TO <danh sách các người dùng>

- **REVOKE:** lấy lại (hủy bỏ) những quyền trên các đối tượng dữ liệu của mình từ những người dùng khác

  REVOKE <danh sách các quyền>

  ON <các đối tượng dữ liệu>

  FROM <danh sách các người dùng>

Điều khiển dữ liệu với SQL

- Trong SQL, những quyền sau đây có thể được gán cho một quan hệ R
  - **SELECT:** truy xuất/đọc dữ liệu trong R.
  - **MODIFY:** chỉnh sửa dữ liệu trong R
    - **INSERT, UPDATE, DELETE**
    - INSERT và UPDATE có thể bị giới hạn trên một số thuộc tính nào đó (thay vì nguyên cả hàng)
  - **REFERENCE:** tham khảo đến R (khóa ngoại)
    - REFERENCE cũng có thể bị giới hạn trên một số thuộc tính nào đó

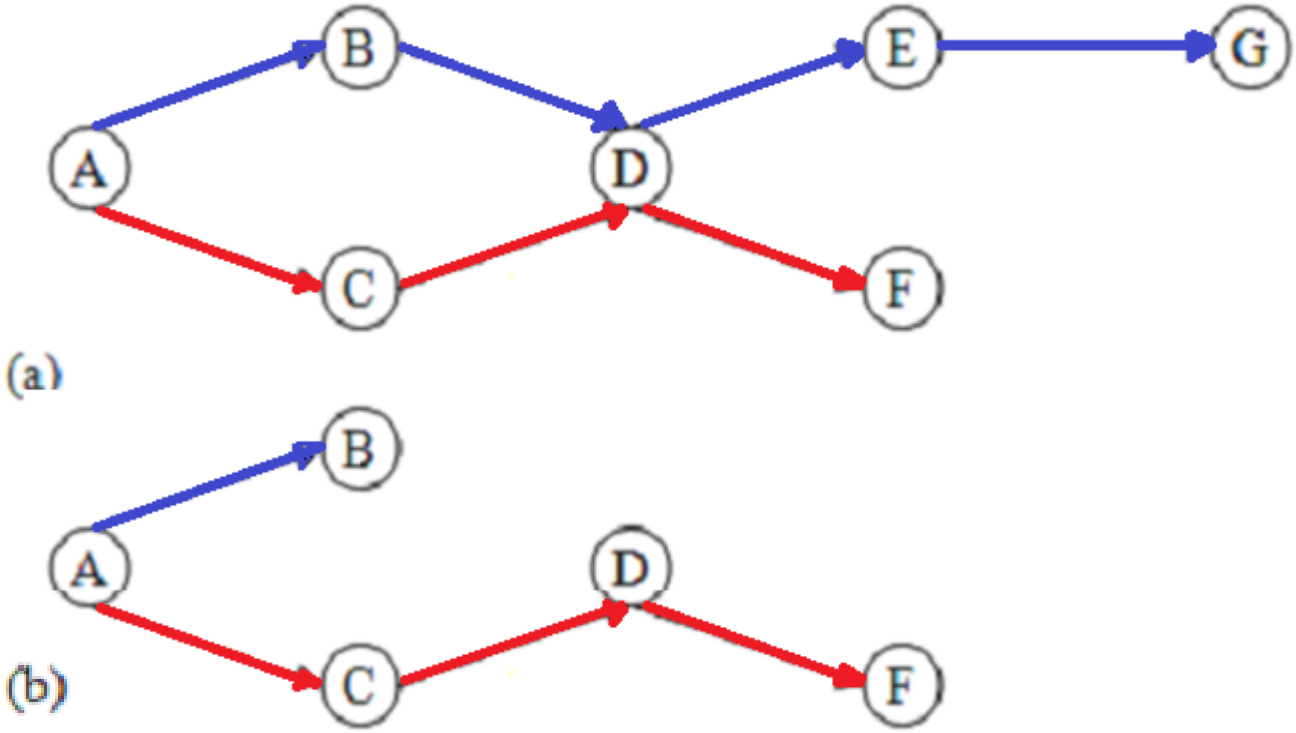
Sự lan truyền quyền với GRANT OPTION

- người dùng *A* là chủ của quan hệ *R*. người dùng *A* có thể gán một quyền *D* trên *R* cho người dùng *B* với **GRANT OPTION** hoặc không.
- Nếu trong câu lệnh gán có **GRANT OPTION**, điều này có nghĩa là *B* được phép gán quyền *D* trên *R* cho những người dùng khác.
- *B* lại gán quyền *D* trên *R* cho một người dùng khác là *C* và cũng với **GRANT OPTION**

  → Chủ của quan hệ *R* (người dùng *A*) có thể **không biết** về sự lan truyền của quyền *D* trên *R* cho những người dùng khác

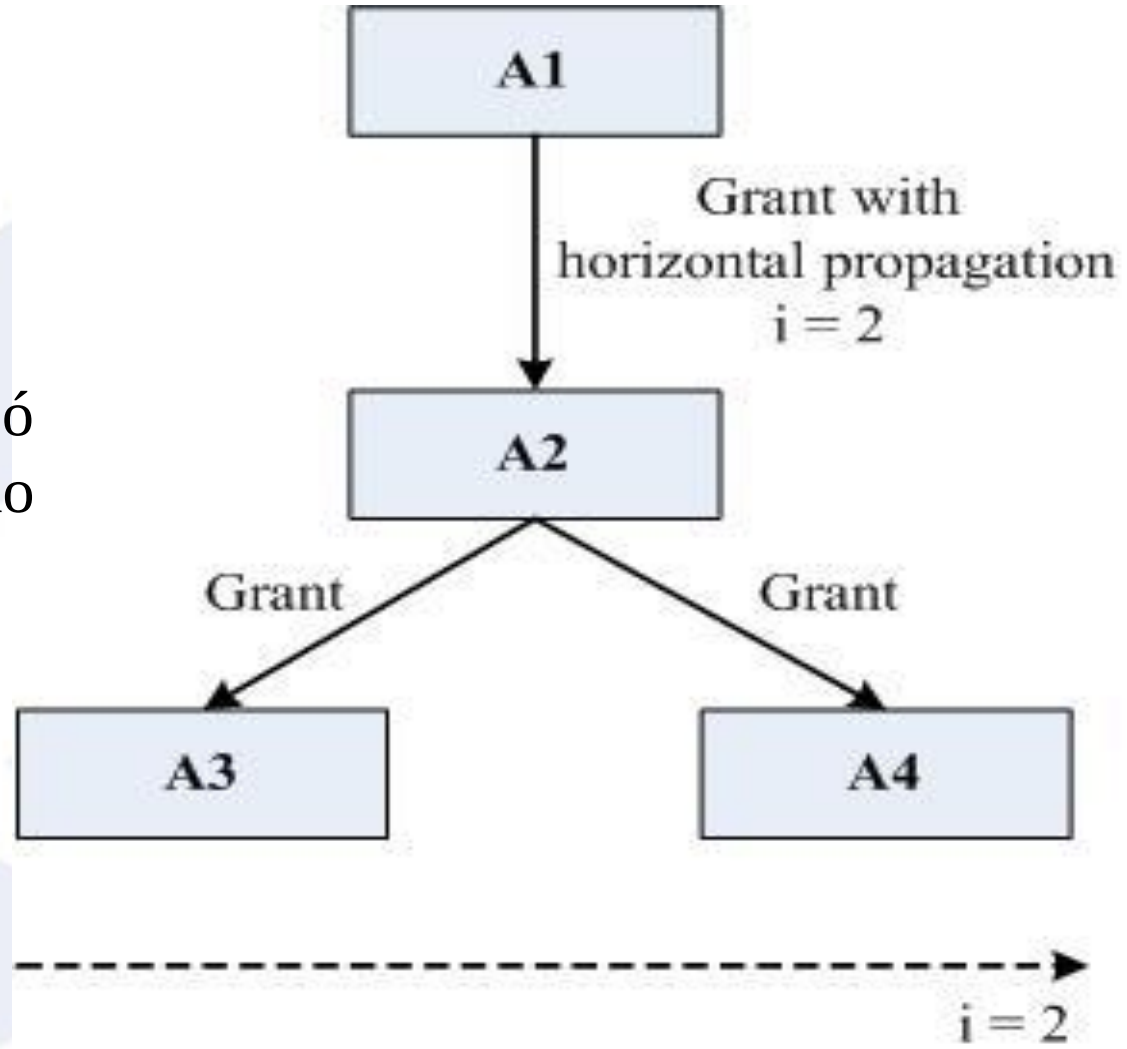
Sự lan truyền quyền với GRANT OPTION

- Nếu người dùng *A* thu hồi lại quyền *D* đã gán cho người dùng *B* thì tất cả những quyền *D* lan truyền bắt đầu từ *B* phải bị hệ thống **tự động thu hồi lại.**



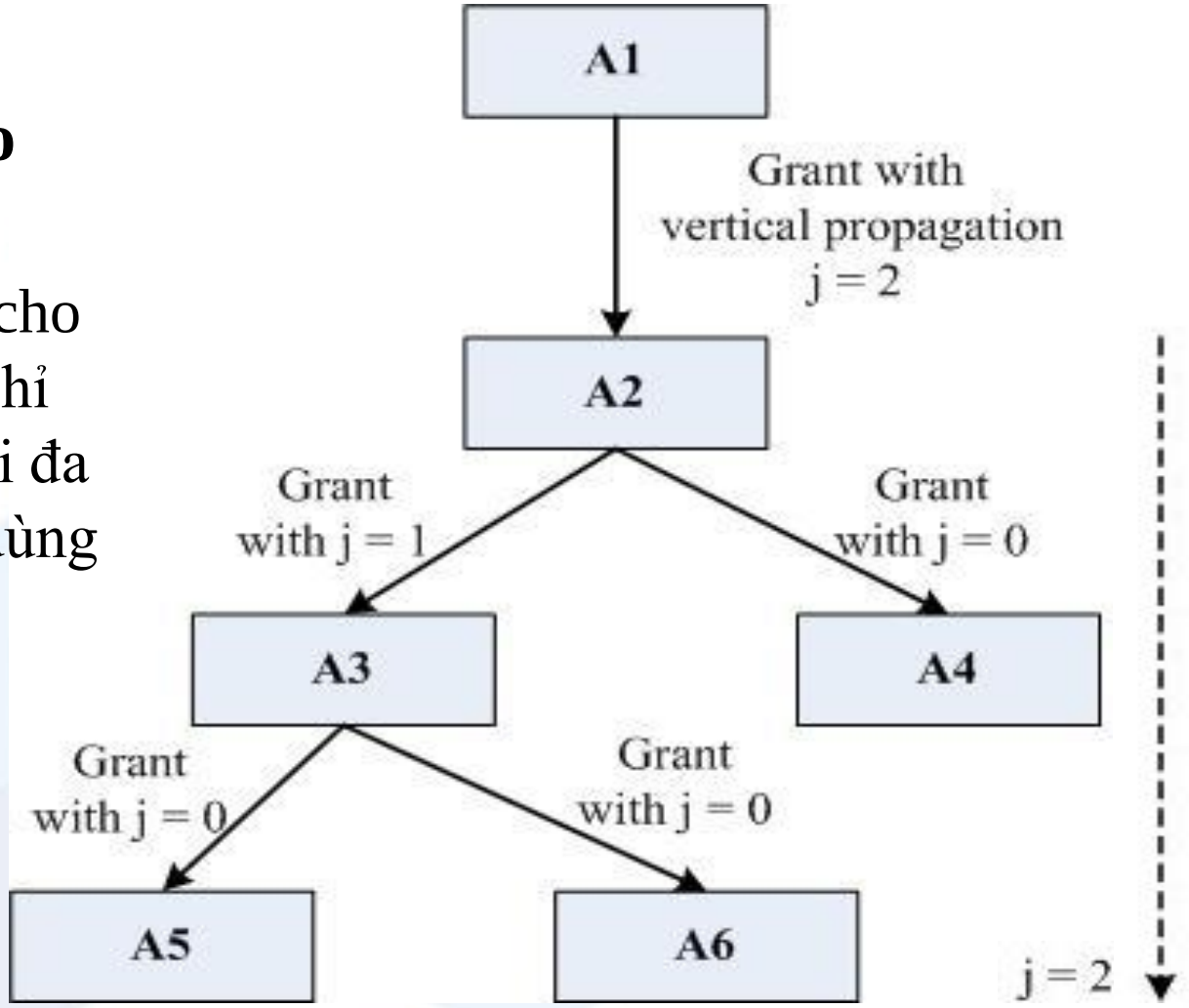
Giới hạn sự lan truyền quyền

- **Giới hạn sự lan truyền quyền theo chiều ngang:**
  - *A1* gán quyền *D* cho *A2* và muốn *A2* chỉ có thể gán (trực tiếp) cho tối đa *i* = *2* người dùng khác



Giới hạn sự lan truyền quyền

- **Giới hạn sự lan truyền quyền theo chiều dọc:**
  - *A1* gán quyền *D* cho *A2* và muốn *A2* chỉ có thể gán cho tối đa *i = 2* cấp người dùng khác



Điều khiển dữ liệu với SQL

- Điều khiển truy cập tùy quyền bằng VIEW (quan hệ ảo)
  - người dùng *A* là chủ của quan hệ *R*. *A* muốn gán cho người dùng *B* quyền truy xuất trên *R* nhưng chỉ muốn cho *B* xem **một số thuộc tính** nhất định. *A* có thể **tạo view *V*** của *R* chỉ chứa những thuộc tính đó và sau đó gán cho *B* quyền SELECT trên *V*
  - Tương tự, nếu *A* muốn giới hạn *B* chỉ được xem **một số hàng** trong quan hệ *R* thì *A* có thể **tạo view *V'*** chỉ chứa những hàng đó trong *R* và sau đó gán quyền truy xuất trên *V'* cho *B*

Ví dụ

- Nhà quản trị CSDL (Database administrator - DBA) tạo 4 người dùng: A1, A2, A3, A4
- DBA gán cho A1 quyền tạo các quan hệ

```
GRANT CREATETAB TO A1;
```

- DBA tạo ra schema EXAMPLE và cho phép A1 các quyền thao tác trên đó.

```
CREATE SCHEMA EXAMPLE AUTHORIZATION A1;
```

Ví dụ

- A1 có thể tạo ra các quan hệ trong schema EXAMPLE.
- A1 tạo ra 2 quan hệ EMPLOYEE và DEPARTMENT
  - A1 là chủ của 2 quan hệ EMPLOYEE và DEPARTMENT nên A1 có tất cả các quyền trên 2 quan hệ này

EMPLOYEE

Name	Ssn	Bdate	Address	Sex	Salary	Dno

DEPARTMENT

Dnumber	Dname	Mgr_ssn

Ví dụ

- A1 muốn gán cho A2 quyền INSERT và DELETE trên cả hai quan hệ trên, nhưng không cho phép A2 có thể truyền các quyền này cho những người dùng khác

```
GRANT INSERT, DELETE ON
    EMPLOYEE, DEPARTMENT TO A2;
```

- Và A1 muốn cho phép A3 truy vấn thông tin từ cả 2 quan hệ, đồng thời có thể truyền quyền SELECT cho những người dùng khác

```
GRANT SELECT ON EMPLOYEE, DEPARTMENT
    TO A3 WITH GRANT OPTION;
```

Ví dụ

- A3 có thể gán quyền SELECT trên quan hệ EMPLOYEE cho A4:

```
GRANT SELECT ON EMPLOYEE TO A4;
```

- A4 không thể truyền quyền SELECT cho người dùng khác vì không có GRANT OPTION

Ví dụ

- Sau đó, A1 quyết định thu hồi lại quyền SELECT trên quan hệ EMPLOYEE của A3

```
REVOKE SELECT ON EMPLOYEE FROM A3;
```

- Hệ quản trị CSDL (Database Management System – DBMS) cũng phải tự động thu hồi lại quyền SELECT trên quan hệ EMPLOYEE của A4. Bởi vì A3 gán cho A4 quyền này mà bây giờ A3 không còn quyền này nữa.

Ví dụ

- Giả sử A1 muốn gán lại cho A3 quyền truy vấn có giới hạn trên quan hệ EMPLOYEE và cho phép A3 lan truyền quyền này.
  - A3 chỉ có thể xem các thuộc tính NAME, BDATE và ADDRESS của những hàng có giá trị DNO = 5.
- A1 tạo view A3EMPLOYEE như sau:

```
CREATE VIEW A3EMPLOYEE AS
    SELECT NAME, BDATE, ADDRESS
    FROM EMPLOYEE
    WHERE DNO = 5;
```

Ví dụ

- Sau khi tạo view, A1 có thể gán quyền SELECT trên view A3EMPLOYEE cho A3:

```
GRANT SELECT ON A3EMPLOYEE TO A3
    WITH GRANT OPTION;
```

- Sau cùng, A1 muốn cho phép A4 chỉ cập nhật thuộc tính SALARY của quan hệ EMPLOYEE;

```
GRANT UPDATE ON EMPLOYEE (SALARY) TO A4;
```

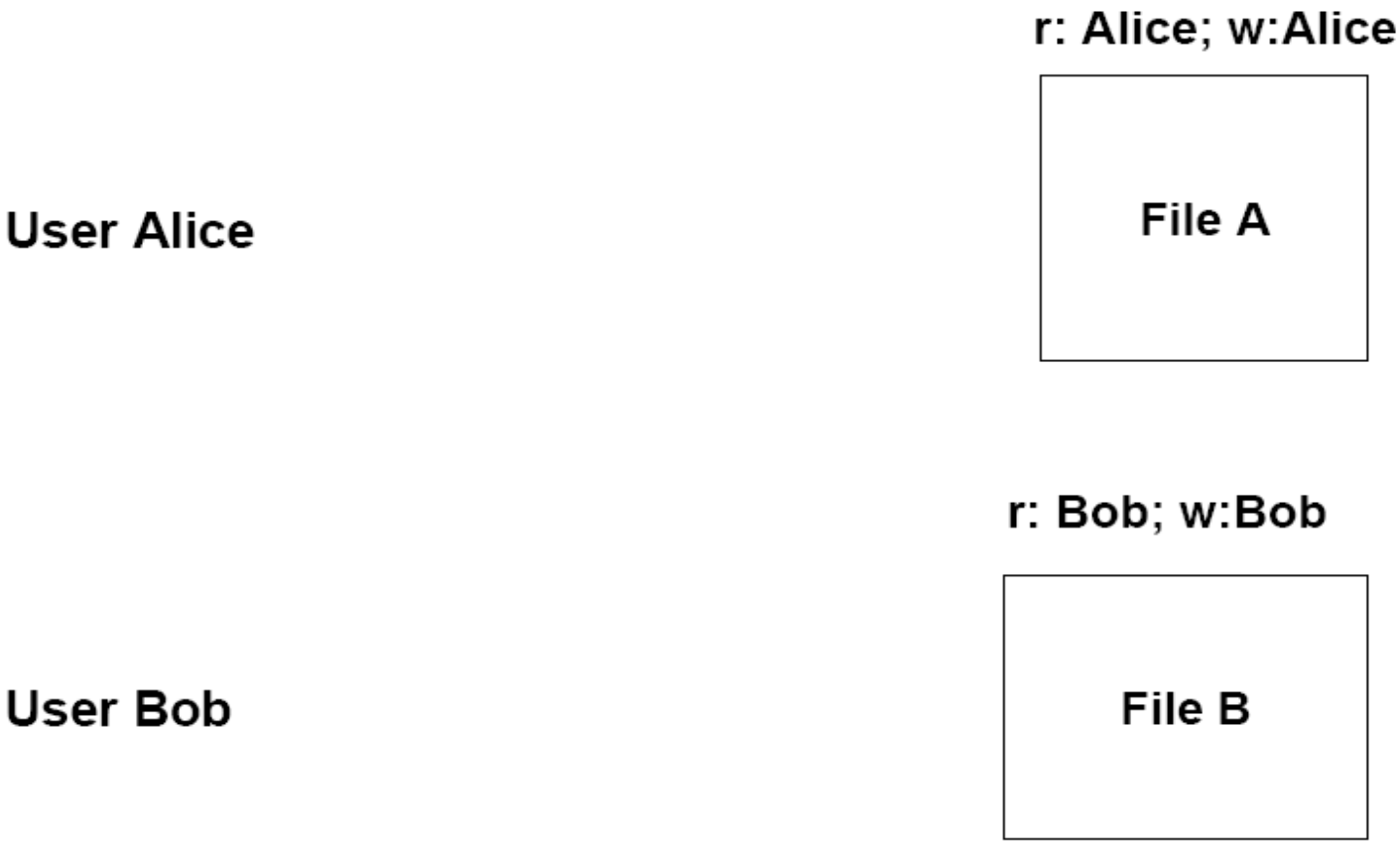
Nội dung

1. Giới thiệu về điều khiển truy cập tùy quyền
2. Mô hình điều khiển truy cập tùy quyền
3. Điều khiển dữ liệu với SQL
4. DAC và điều khiển dòng thông tin

DAC và Điều khiển dòng thông tin

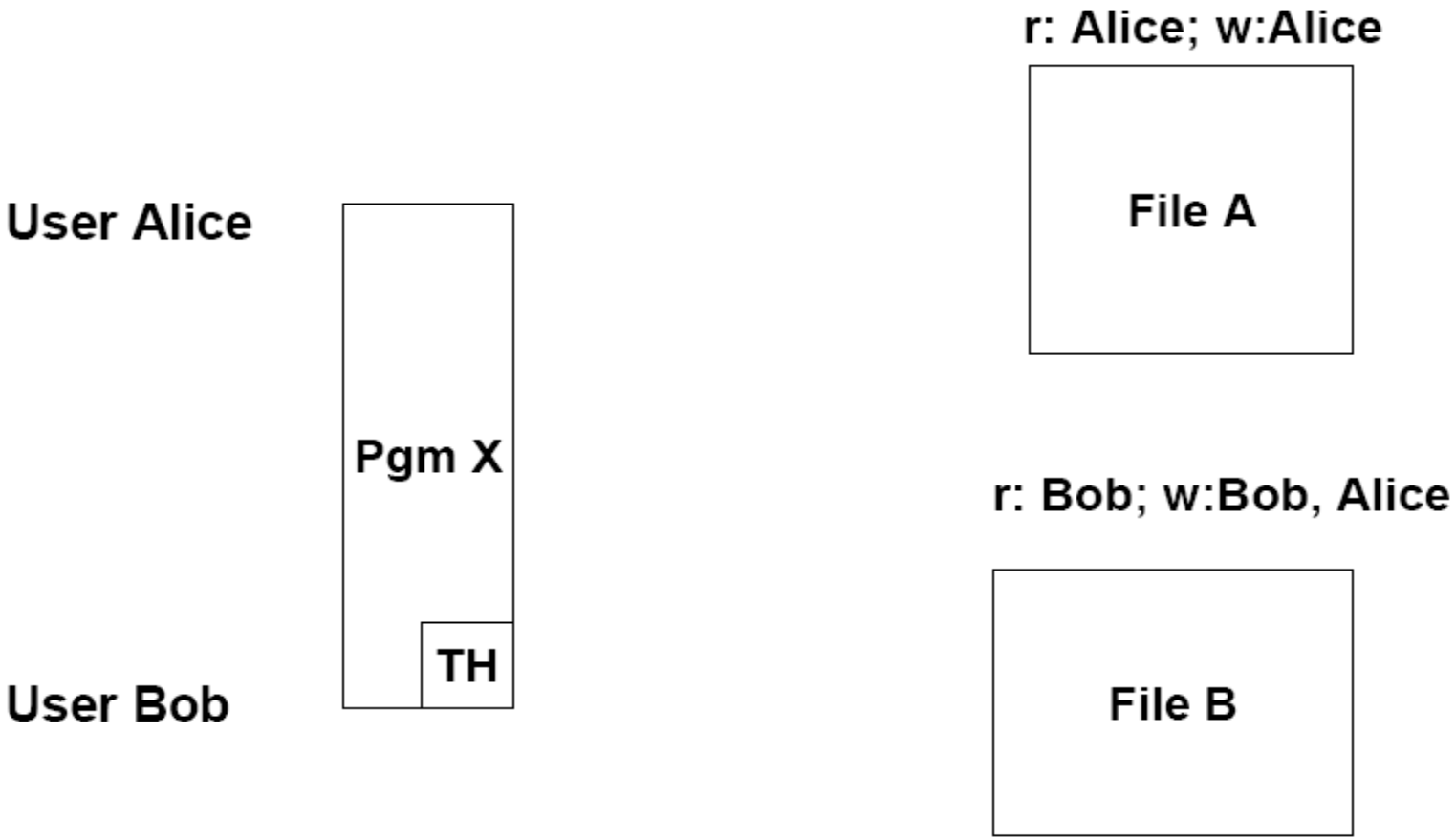
- Khuyết điểm của DAC: cho phép dòng thông tin từ đối tượng này truyền sang đối tượng khác bằng cách đọc thông tin lên từ một đối tượng rồi ghi thông tin đó xuống đối tượng khác
- Ví dụ: Bob không được phép xem file A, nên anh ta nhờ Alice (đồng lõa với Bob) copy nội dung của file A sang file B (Bob có thể xem được file B)
- Giả sử các người dùng đều đáng tin cậy và không làm việc như trên thì cũng có thể một Trojan Horses sẽ làm việc sao chép thông tin từ đối tượng này sang đối tượng khác.

Ví dụ về Trojan horse

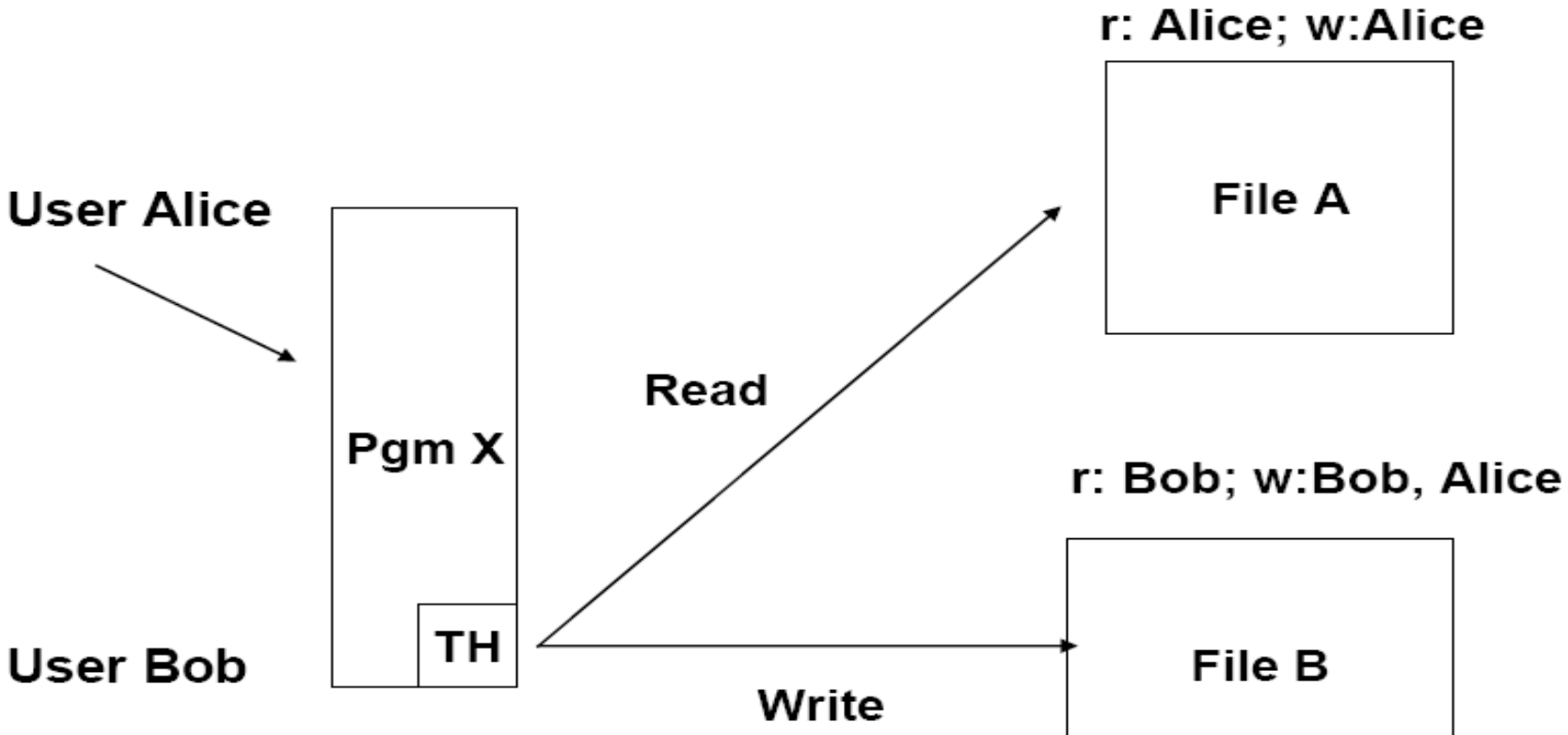


Bob không thể đọc được nội dung của file A

Ví dụ về Trojan horse



Trojan horse Example



Bob có thể đọc được nội dung của file A sau khi nó được sao chép sang file B

Nội dung

1. Giới thiệu về điều khiển truy cập tùy quyền
2. Mô hình điều khiển truy cập tùy quyền
3. Điều khiển dữ liệu với SQL
4. DAC và điều khiển dòng thông tin