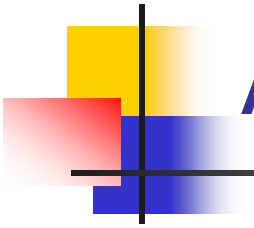
CS590U

Access Control: Theory and Practice

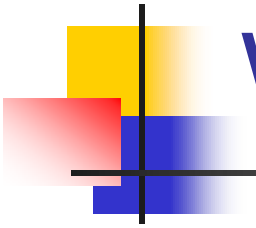
The Griffith-Wade Scheme



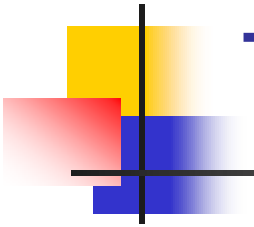
Access Control in SQL

- Griggiths & Wade. “An Authorization Mechanism for a Relational Database System”. [TODS, 1976]
  - Discusses access control mechanism in System R
  - The Grant/Revoke Mechanism remained essentially unchanged in today’s database systems
- Rosenthal & Sciore. “Bringing Relational Access Control into the

The Goal of [Griggiths & Wade]



- Permits users to selectively share data while retaining the ability to restrict data access in a multi-user database system



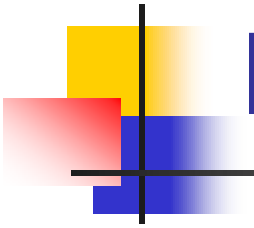
The Approach:

- The creator of a table is fully authorized to perform any actions on the table.
- The creator may explicitly grant to any other user any or all of his privileges.
- The grantor may specify that the user is authorized to further grant his privileges.
- A grantor may revoke the granted privileges.
- Views are used for granting access to row and column subsets.

Two Types of Relations

- Base relations (physically stored)
  - e.g., EMPLOYEE(NAME, SALARY, MANAGER, DEPARTMENT)
- Views (a virtual, dynamic window on the data base)
  - e.g., DEFINE VIEW AS

```
SELECT   NAME, SALARY
FROM     EMPLOYEE
WHERE    DEPARTMENT = 'TOY'
```



Privileges on a Relation

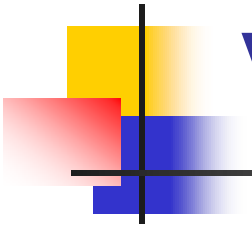
- READ: use the relation in a query, e.g., to read tuples, or define views
- INSERT: insert rows
- DELETE: delete rows
- UPDATE: modify existing data
  - may be restricted to a subset of the columns of the table
  - some views may not be updatable
- DROP: delete the entire table

The Syntax for Granting Permissions

- A user executes the following grant command:
- GRANT

ALL RIGHTS
| ⟨privileges⟩
| ALL BUT ⟨privileges⟩
ON ⟨table⟩
TO ⟨user-list⟩
[WITH GRANT OPTION]

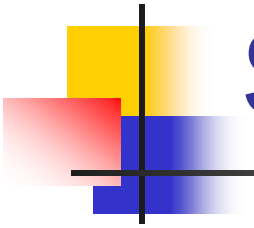
The Implementation: Basic Version



- SYSAUTH
  - USERID: the user being authorized
  - TNAME: name of the table
  - TYPE: ‘R’ if a base relation, `V’ if a view
  - A column for each of the privileges READ, INSERT ... (excluding UPDATE), containing a ‘Y’ to indicate an authorization has the privilege
  - UPDATE: `ALL’ (all columns), `NONE’ (no update), or `SOME’
  - GRANTOPT: whether can be further granted

The Implementation: Basic Version (continued)

- For each table, a user has at most two tuples in SYSAUTH: one for grantable privileges, and one for nongrantable privileges
- SYSCOLAUTH
  - used if UPDATE is `SOME'
  - for each updatable column, a (user, table, column, grantor, grantopt) tuple is inserted into SYSCOLAUTH



Semantics of GRANT

- When a user issues a GRANT command, the set of privileges actually granted is the intersection of
  - the set of grantable privileges possessed by the grantor
  - and the set of privileges in the grant
- The effect of a GRANT is
  - to insert a new tuple
  - or to appropriately modify an existing one

An Example

- Let A be the creator of the table EMPLOEE, after
  1. A: GRANT READ, INSERT ON EMPLOYEE TO B WITH GRANT OPTION
  2. B: GRANT READ, DELETE ON EMPLOYEE TO X
- X has READ privilege on EMPLOYEE
- X has no privilege if 1 and 2 are switched

Syntax for Revocation

- REVOKE

  ALL RIGHTS

  | ⟨privileges⟩

  ON ⟨table⟩

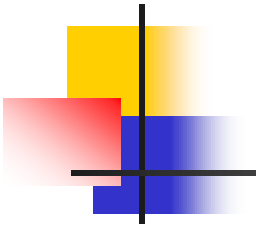
  FROM ⟨user-list⟩

Semantics of Revocation

- Let the sequence of grant commands of a specific privilege on a given table by any user before any REVOKE commands be

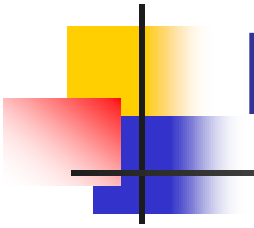
  $G_1, G_2, \ldots, G_{i-1}, G_i, G_{i+1}, \ldots, G_n$

  - Grants of several privileges are represented as a sequence of individual grants
- If a revocation R occurs, and G_i is the only one affected (same grantor, same user, same privilege), then the state of the authorization should be identical to the state after the sequence $G_1, G_2, \ldots, G_{i-1}, G_{i+1}, \ldots, G_n$



Implications

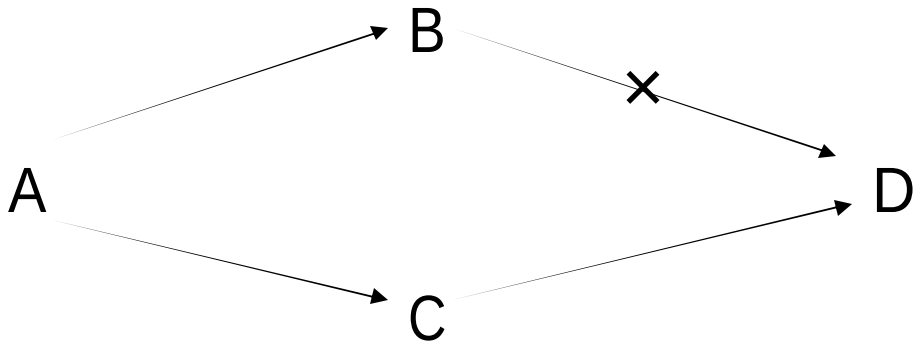
- One may make the same grant multiple times, one revoke statement revokes all of them.
- If a revokee possesses other grants of the revoked privilege from an *independent* source, then he retains these privileges.

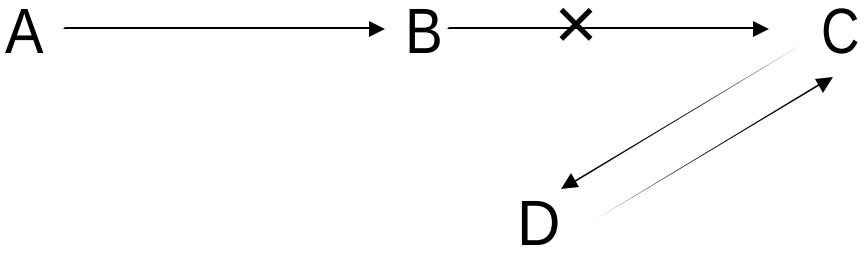


Recursive Revocation

- Consider the following sequence:
  - A grants ALL RIGHTS to X with GRANT OPTION
  - X grants ALL RIGHTS to Y
  - A revokes ALL RIGHTS from X
- This should be equivalent to
  - X grants ALL RIGHTS to Y

  which has no effect
- Need to do recursive revocation


B
A
D
C


A
B
C
D

What permissions are required to run a query Q?

- Some examples
  - Is S is a query, then OPS(S) contains (SELECT, A) for all columns A mentioned in S
  - If S is an update command