En introduktionskurs i databassystem

http://user.it.uu.se/~udbl/dbt-sommar05/
alt. http://www.it.uu.se/edu/course/homepage/dbdesign/st05/

Kjell Orsborn
Uppsala Database Laboratory
Department of Information Technology, Uppsala University,
Uppsala, Sweden
Database Security and Authorization
(Elmasri/Navathe ch. 23)

Kjell Orsborn

Department of Information Technology
Uppsala University, Uppsala, Sweden
Database security and authorization

• A DBMS normally includes a subsystem for security and authorization that is responsible for security against unauthorized access to the database.

• The reasons behind the introduction of restrictions on the availability of data varies a lot but e.g.:
  – legal or ethical reasons (e.g. person databases).
  – policy reasons within governmental, public, or industrial organisations (e.g. credit validation and medical information).
  – system-related reasons to prevent unauthorized access to database systems.
  – security levels within organisations (secret vs. free info)
Hence, one would like to protect the database ...

- Against who and/or what?
  - Corrupt, disloyal, naughty, evil, wily, malicious, spiteful, hateful malevolent, vicious and maybe simply hostile users.
  - Erroneuous data and program errors.
  - Failure in hardware/software that cause corrupted data.

- How?
  - Introduce integrity constraints in the database.
  - Introduce recovery system.
  - Introduce security mechanisms.
Security mechanisms

• Several of the security questions that exist in database systems are not unique for the database field but also exist in other types of systems.
  – e.g. in the design of operative systems

• Security mechanisms:
  – Discretionary access control (or privilege-based security mechanisms) issuing privileges to users for access rights to certain data.
  – Mandatory access control (or multi-level security mechanisms) using security classes.
  – Access control - user accounts and passwords to prevent access to the system itself.
  – Statistical database security - säkerhetsmekanismer mot missbruk av statistiska databaser.
Security mechanisms . . .

- Security mechanisms cont’d...
  - **Data encryption** - e.g. for data transported over communication networks.
  - **Physical protection** - e.g. secure procedures for storage and handling of hard disks and backup copies.
  - Mechanisms (e.g. fire walls and virus prevention/repair software) for providing protection against **data virus**.
Security administration

• The database administrator is responsible for the management of the database security:
  – Create accounts and passwords
  – Grant privileges
  – Revoke privileges
  – Assign security levels

• Logging of user activities
  – The database log must contain user data
  – The **audit trail** - a database log used mainly for security handling and subsequent analysis.
Privilige-based mechanisms

• A common method for **discretionary access control** in database systems is to **grant** and **revoke** privileges.

• Two types of privilege levels exists:
  – the account level, general privileges for single users (not in SQL92) (create schema, create table, create view, alter, drop, modify, select)
  – the relation level, privileges for specific relations and views. Even privileges on attribute level exists. (supported in SQL92)
Privilege-based mechanisms ...

- The access matrix model
  - Access matrix: $M(s,o) \rightarrow p$
    - where $s$, *subject*, are rows in the matrix (users, accounts, program),
    - and $o$, *object*, are columns in the matrix (relation, tuple, column, view, operations), and $p$ is the *privilege type* (read, update)

- Every relation is owned by an account
  - e.g. account that created the relation.

- The owner has complete access rights

- The owner can delegate access rights to other subjects
Priviliges in SQL92

• In SQL92 the following priviliges exist on the relation level:
  – SELECT
  – MODIFY (divided further into UPDATE, DELETE, INSERT)
    • INSERT and UPDATE also on attribute level
  – REFERENCES
    • also on attribute level

• Priviliges can be retracted by ...
  – REVOKE
Priviliges in SQL92 . . .

• Example:
  • DBA:
    CREATE SCHEMA EXAMPLE AUTHORIZATION A1;
  • A1:
    CREATE TABLE EMPLOYEE(...)
    CREATE TABLE DEPARTMENT(...)
• GRANT: Delegate priviliges to subject (i.e. set element in the access matrix)
• Syntax:
  GRANT privilige types ON object TO subject
Priviliges in SQL92 . . .

• Example:

• A1:

  GRANT INSERT,DELETE ON EMPLOYEE,DEPARTMENT TO A2;

• OBS: A2 can not forward priviliges

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

• => A3 can forward priviliges to other accounts.
Priviliges in SQL92 . . .

• A3:
  GRANT SELECT ON EMPLOYEE TO A4
• A2:
  REVOKE SELECT ON EMPLOYEE FROM A3
• => A4 can not either access EMPLOYEE!
• GRANT and REVOKE can also be applied on views.
• One can be granted priviliges from more than one source
• Actual priviliges = the union of all priviliges recieved
Multi-level mechanisms

- Security mechanisms based on classification of data and users into security classes are called **multi-level security control** or **mandatory access control**.
- Not supported in commercial system.
- There is demand within, military, and intelligence organizations as well as in industrial and service enterprises.
- Usually, a combination of privileges and multi-level control is used.
Multi-level mechanisms ...

- One classifies subject and object into security classes such as: TS (top secret), S (secret), C (confidential), U (unclassified), incorporating an order TS > S > C > U.

- An extended access matrix: $M(s,o) \rightarrow <p,c>$,
  - where $s$, *subject*, are rows in the matrix (users, accounts, program),
    and $o$, *object*, are columns in the matrix (relation, tuple, column, view, operations),
    and $p$ is *privilege type* (read, update),
    and $c$ is security class.
Multi-level mechanisms ...

- Classification of subject - object are denoted by:
  - \texttt{class}(s) and \texttt{class}(o) respectively.

- Two restrictions are forced upon data at access based on subject/object classification:
  - A subject S is not allowed to have read access for an object O if not \texttt{class}(s) \geq \texttt{class}(o) holds. This is called \textit{simple security property}.
  - A subject S is not allowed to have write access for an object O if not \texttt{class}(s) \leq \texttt{class}(o) holds. This is called \textit{*-property} or \textit{star property}.
Authorization using views

- **Views** can also be used as a security mechanism.
- Transformation of DML queries for certain users.
  - e.g. add a selection and projection to each query that WALMART employees asks. The DBA provide:

  ```sql
  USE DATABASE WALMART;
  CREATE TABLE SUPPLIES (STORE CHAR, ITEM CHAR, PRICE DECIMAL(10,2), PRIMARY KEY(STORE, ITEM))
  CREATE VIEW WMSUPPLIES AS
  SELECT STORE, ITEM, PRICE FROM SUPPLIES
  WHERE STORE = 'WALMART'
  ```
Authorization using views . . .

• Priviliges are granted:
  – GRANT SELECT, INSERT, DELETE ON WMSUPPLIES TO WALLIES

• WALLIES can not access SUPPLIES only WMSUPPLIES
  – SELECT PRICE
    FROM WMSUPPLIES S
    WHERE S.ITEM = ‘TOMATOES’

• Translated to:
  – SELECT PRICE
    FROM SUPPLIES S
    WHERE S.ITEM = ‘TOMATOES’ AND S.STORE = ‘WALMART’
Authorization using views . . .

- Advanced security policies can be accomplished with views
- OBS! views are not always updatable
- The key (and other ”not null” attributes) in the base table must be included in the view definition for the view to be updatable.
Statistical database security

- Statistical databases often include sensitive information about single individuals that must be protected from unallowed use.
- However, statistical information should be extractable from the database.
- Statistical database security must prohibit access of individual data elements.
- Three main security mechanisms: conceptual, restriction-based, and perturbation-based. Examples:
  - prohibit queries on attribute level
  - only queries for statistical aggregation (statistical queries)
  - statistical queries are prohibited when the selection from the population is too small.
  - prohibit repeated statistical queries on the same tuples.
  - introduce distortion into data.