Security Issues

- Defining authorized access to data / privileges
  *(who may access/change what?)*

- Enforcing authorized access (efficiently)
  *Both with internal logical controls (intra system)*
  *and against unauthorized access to system itself*

- Privilege hierarchy

- Access logs – audit trail

- Encryption

- Statistical databases
Privileges

• Ability to:
  • Access
  • Add
  • Change
  • Reference

  objects in the database

• May be:
  • Granted
  • Revoked
  • Inherited (recursively)
SQL Syntax for Privilege Control

Data Control Language (DCL):

**GRANT**  <Privilege_List>
**ON**  <Object>  Table name (or domain)
**TO**  <User_List>  List of login ID’s or **PUBLIC**

[ **WITH GRANT OPTION** ]  Ability to grant to others
(default is no secondary granting)
**SQL Syntax for Privilege Control**

Data Control Language (DCL):

<table>
<thead>
<tr>
<th>Command</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRANT</td>
<td><code>&lt;Privilege_List&gt;</code></td>
<td>Privilege list</td>
</tr>
<tr>
<td>ON</td>
<td><code>&lt;Object&gt;</code></td>
<td>Table name (or domain)</td>
</tr>
<tr>
<td>TO</td>
<td><code>&lt;User_List&gt;</code></td>
<td>List of login ID’s or PUBLIC</td>
</tr>
<tr>
<td>[ WITH GRANT OPTION ]</td>
<td></td>
<td>Ability to grant to others</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(default is no secondary granting)</td>
</tr>
</tbody>
</table>

For Example:

- **GRANT** SELECT
  **ON** PRODUCT
  **TO** PUBLIC

- **GRANT** UPDATE (BonusPct)
  **ON** SALES
  **TO** SALES_MGR

- **GRANT** DELETE
  **ON** EMPLOYEE
  **TO** PERSONNEL_MGR
Additional Privileges

CREATE INDEX  Why should this be a separate privilege?
CREATE TABLE
CREATE VIEW
CREATE TABLESPACE  Space allocation (DBA)
CREATE USER
CREATE PROCEDURE  Restricted separately because others can use with privileges of owner

ALSO ALTER, DROP, ...
Granting Power to Grant Privileges

GRANT UPDATE (Bonus_Pct)
ON SALES
TO SALES_MGR
WITH GRANT OPTION

Executed by database administrator (DBA)

GRANT UPDATE (Bonus_Pct)
ON SALES
TO ASST_SALES_MGR

Executed by SALES_MGR
(Example) Privilege Hierarchy

Database Administrator

Table Owner

Grantee with Grant Option

Grantee  Grantee

Table Owner

Grantee with Grant Option

Grantee  Grantee  Grantee
Revoking Privileges

Optionally revokes just grant option

\texttt{REVOKE}\ [ \texttt{GRANT OPTION FOR} ] \<\texttt{Privilege\_List}\>

\texttt{ON} \ <\texttt{Object}> 
\texttt{FROM} \ <\texttt{User\_List}> \ [ \texttt{CASCADE} ]

CREATE / INSERT / UPDATE / DELETE etc.

Also revoke privileges granted to user by others (recursively)
Use of Views in Access Control

CREATE VIEW Directory AS

SELECT lname, fname, address, phone
FROM Employee
WHERE unlisted = 'F'

GRANT SELECT ON DIRECTORY TO PUBLIC

Note previous discussion on Updates with views (anomalies, null values, etc.)
Use of Views in Access Control

Assume GRADES relation with attributes Fname, Lname, SSN, AS1, AS2 etc.

CREATE VIEW Cucerzan_Grades AS
SELECT *, SUM(AS1 * .07 + AS2 * .07 + MIDTRM * .15) AVG
FROM GRADES
WHERE Lname = 'Cucerzan'

GRANT SELECT ON Cucerzan_Grades TO Cucerzan

GRANT UPDATE ON Grades TO Kalowsky
Encryption

• Logical privilege mechanisms may not be enough (especially against external intruders)

• Selection/Deletion/Projection etc. may work transparently without modification of database internals

• Problem:
Encryption

• Logical privilege mechanisms may not be enough (especially against external intruders)

• Selection/Deletion/Projection etc. on individually encrypted attributes may work transparently without modification of database internals

• Problem:

  sorting / indexing
Statistical Databases

- Protect confidentiality by only allowing access to statistical / aggregate information on Averages, Counts, Sums, Std Deviations, etc.
  - Implementable with views

Statistical Queries

- Problem:
  - Multiple queries can be formulated on aggregate values that enable deduction of information about an individual.
Deduction of Individual Info from Statistical DBs

Person (relation):

<table>
<thead>
<tr>
<th>Name</th>
<th>SSN</th>
<th>Income</th>
<th>Address</th>
<th>City</th>
<th>State</th>
<th>Zip</th>
<th>Sex</th>
<th>Last_Degree</th>
</tr>
</thead>
</table>

- **not included in statistical DB**
- **Aggregate values for these attributes are queryable**

Example Query:

```
SELECT Average(Income)
FROM Person
WHERE Sex='F' AND LAST_DEGREE='PHD'
```

Problem: ???
## Deduction of Individual Info from Statistical DBs

### Person (relation):

<table>
<thead>
<tr>
<th>Name</th>
<th>SSN</th>
<th>Income</th>
<th>Address</th>
<th>City</th>
<th>State</th>
<th>Zip</th>
<th>Sex</th>
<th>Last_Degree</th>
</tr>
</thead>
</table>

- not included in statistical DB
- Aggregate values for these attributes are queryable

```sql
SELECT Average(Income)
FROM Person
WHERE Sex='F' AND LAST_DEGREE='PHD'
```

### Problem:

- As selectional constraints become more specific, statistics may refer only to a few or one individual
Deduction of Individual Info from Statistical DBs

Person (relation):

<table>
<thead>
<tr>
<th>Name</th>
<th>SSN</th>
<th>Income</th>
<th>Address</th>
<th>City</th>
<th>State</th>
<th>Zip</th>
<th>Sex</th>
<th>Last_Degree</th>
</tr>
</thead>
</table>

- As selectional constraints become more specific, statistics may refer only to a few or one individual.

- **SELECT** Average(Income)
  **FROM** Person
  **WHERE** Sex='F' AND LAST_DEGREE='PHD'

- **SELECT** Count(*)
  **FROM** Person
  **WHERE** Sex='F' AND ZIP='21238' AND LAST_DEGREE='PHD'

- If count=1 then average is equal to individual.
Deduction of Individual Info from Statistical DBs

• Don’t return answer if population on which the result is based is less than a threshold

• Don’t allow multiple queries on the same tuple population if (Previous result \(\cap\) Current result) > \(m\) values
  
  • Q1: Sum(income) where Sex=‘F’ and Degree=‘PhD’ and age<30 and (Zip=21238 OR State=‘CA’)
  
  • Q2: Sum(income) where Sex=‘F’ and Degree=‘PhD’ and age<30 and (State=‘CA’)
  
  • Q1 \(\cap\) Q2 = 1 tuple (zip=21238)

• Introduce minor noise in data to complicate solution by simultaneous equations