Database Security

Security requirements:
- Physical database integrity
- Logical database integrity (internal consistency)
- Element integrity (external consistency)
- Auditability
- Access Control Functionality
- Authentication Functionality
- Availability

1.) Database Integrity Controls:
   
a.) Semantic Integrity Controls

Monitor:
Unit of DBMS that checks value being entered to ensure their consistency with
  - rest of the database
  - characteristics of the particular field

Forms of Monitor checks:

Range comparison: check that values are within acceptable range
  - days in January: 1 – 31
  - salary of employees < 50000

State constraints: describe conditions for entire database
  - all employees have different employee numbers
  - only one employee is “president”

Transition constraints: conditions necessary before changes to be applied
  - employee who is “married” cannot become “single”

b.) Operational Integrity Controls:

Concurrency manager as a DBMS subsystem has to ensure serializability of transactions

→ Two-Phases Locking Protocol
### c.) Recovery

Each transaction has to be atomic:
- terminate correctly, modifying the accessed data
- terminate unsuccessfully without modifying the accessed data

**Logging system records:**
- Operations performed on data
- Transaction control operations (begin transaction, commit, abort, end transaction)
- Old value (“before image”)
- New value (“after image”)

**Transaction errors, system errors** → rollback using “before images”

**Storage error** → reconstruction with dump file and “after image”

### 2.) Access Control

**Special problems:**
- access by inference
- database objects have fine granularity → effects on efficiency

**Access control policies:**
- discretionary
- mandatory

**Authorisation rules:**
- **name dependent** (based on object name/id)
- **content-dependent** (based on object content)
  → content-based views, query modification
- **context-dependent** (based on system variables such as data, time, query source)
  → context-based views

### a.) Multi-Level Secure Databases:

- Implement Bell LaPadula’s Mandatory (“Multi-Level”) Security policy in a relational database
- First prototype in the Seaview (Secure dAta VIEW) project (1988)
- Major database vendors have DBMS versions with multi-level database security support
Labeling Objects:

R: multi-level relation with n attributes

A tuple in R is of the form

\((v_1, c_1, \ldots, v_n, c_n, tc)\)

where

- \(v_i\): ith attribute value
- \(c_i\): security level of the ith field (not visible to users)
- \(tc\): security level of the tuple

Example:

- Employee name C name
- Department
- Profession

John Bob S Dept-1 S Cryptology expert TS TS
Mary Doe U Dept-2 S IT Security specialist S S
Rita Hanks U Dept-2 U Secretary U U

U: unclassified
S: Secret
TS: Top Secret

C-Instance of a relation:

Information in relation accessible by users at class C. Values not accessible are replaced by null values.

Examples:

<table>
<thead>
<tr>
<th>Employee name</th>
<th>Department</th>
<th>Profession</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Bob</td>
<td>Dept-1</td>
<td>-</td>
</tr>
<tr>
<td>Mary Doe</td>
<td>Dept-2</td>
<td>IT Security specialist</td>
</tr>
<tr>
<td>Rita Hanks</td>
<td>Dept-2</td>
<td>Secretary</td>
</tr>
</tbody>
</table>

U-Instance:

<table>
<thead>
<tr>
<th>Employee name</th>
<th>Department</th>
<th>Profession</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary Doe</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Rita Hanks</td>
<td>Dept-2</td>
<td>Secretary</td>
</tr>
</tbody>
</table>

Consistent Addressing:

In order to address a data item, you have to specify
- a database D
- a relation R within D
- a primary key for a tuple r within D
- the attribute i, identifying element r_i within r

\[ f_O(D) \leq f_O(R) \leq f_O(r_i) \]

(fO: object security level)

A user can only access element r_i if

- Since a user who has access to a tuple r has also access to all its elements

\[ f_O(r_i) \leq f_O(r) \]

is required
**Multi-level entity integrity:**
- No tuple in an instance of R have null values for any of the primary key attributes
- All components of a primary key of a relation R have the same security level, which is dominated by the security levels of all non-key attributes

**Polyinstantiation:**
Several tuples might exist for the same primary key

**Polyinstantiated elements:**
Elements of an attribute which have different security levels, but are associated with the same primary key and key class

**Problem:**
Tradeoff between confidentiality (covert channel protection) and integrity

**Examples:**
How do polyinstantiated elements arise?

A subject updates what appears a null element in a tuple, but which actually hides data with a higher (or incomparable) security level

**Problem:**
- Subject cannot be informed about existence of higher class data (-> covert channel)
- Overwriting the old value allows “low” users to unwittingly destroy “high” data
  - Insertion must be accepted

**U-subject requires the following operation:**
Update employee
SET profession = "Programmer"
WHERE name = "Mary Doe"

<table>
<thead>
<tr>
<th>Employee name</th>
<th>Cname</th>
<th>Department</th>
<th>CDept</th>
<th>Profession</th>
<th>Cprof</th>
<th>TC</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Bob</td>
<td>S</td>
<td>Dept-1</td>
<td>S</td>
<td>Cryptology expert</td>
<td>TS</td>
<td>TS</td>
</tr>
<tr>
<td>Mary Doe</td>
<td>U</td>
<td>Dept-2</td>
<td>S</td>
<td>IT Security specialist</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>Rita Hanks</td>
<td>U</td>
<td>Dept-2</td>
<td>U</td>
<td>Secretary</td>
<td>U</td>
<td>U</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Employee name</th>
<th>Cname</th>
<th>Department</th>
<th>CDept</th>
<th>Profession</th>
<th>Cprof</th>
<th>TC</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Bob</td>
<td>S</td>
<td>Dept-1</td>
<td>S</td>
<td>Cryptology expert</td>
<td>TS</td>
<td>TS</td>
</tr>
<tr>
<td>Mary Doe</td>
<td>U</td>
<td>Dept-2</td>
<td>S</td>
<td>IT Security specialist</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>Mary Doe</td>
<td>U</td>
<td>Dept-2</td>
<td>S</td>
<td>Programmer</td>
<td>U</td>
<td>S</td>
</tr>
<tr>
<td>Rita Hanks</td>
<td>U</td>
<td>Dept-2</td>
<td>U</td>
<td>Secretary</td>
<td>U</td>
<td>U</td>
</tr>
</tbody>
</table>
Polyinstantiation Integrity:

A multilevel relation \( R \) satisfies polyinstantiation integrity =>
\[ \text{AK, C}_{\text{AK}}, C_i \rightarrow A_i \]
(\( \text{AK} \): key attributes, \( \text{C}_{\text{AK}} \): classifications of key attributes, \( A_i \): non-key attribute, \( C_i \): classification of \( A_i \)).

The primary key of a multi-level relation is formed of \( \text{AK} \cup \text{C}_{\text{AK}} \cup \text{C}_R \)
\( \text{C}_R \): vector of classification attributes for non-key data attributes

Example: Trusted Oracle
- stores security level for each database row
- Bell LaPadula MAC policy can be
  a.) Operating system enforced (OS MAC mode)
    ♦ relies on OS to enforce MAC
    ♦ database server stores DB information at the same security level in single-level files
  b.) DBMS / trusted subject enforced (DBMS MAC mode)
    ♦ Trusted Oracle runs as trusted subject, privileged to bypass MAC policy
    ♦ requests policy decisions from underlying OS

b.) Role-Based Access Control Features in Commercial Database Management Systems [Ramaswamy / Sandhu 1998]:

<table>
<thead>
<tr>
<th>Item</th>
<th>Feature</th>
<th>Informix</th>
<th>Sybase</th>
<th>Oracle</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ability for a role grantee to grant that role to other users</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>2</td>
<td>Multiple active roles for a user session</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>3</td>
<td>Specify a default active role set for a user session</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>4</td>
<td>Build a role hierarchy</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>5</td>
<td>Specify static separation of duty constraints on roles</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>6</td>
<td>Specify dynamic separation of duty constraints on roles</td>
<td>(YES)</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>7</td>
<td>Specify maximum or minimum cardinality for role membership</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>8</td>
<td>Grant DBMS System Privileges to a role</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>9</td>
<td>GRANT DBMS Object Privileges to a role</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>

c.) Inference Controls for statistical Databases:

Statistical Database:
- Information is retrieved by means of statistical queries on an attribute (column) of a table
- Attributes directly identifying persons (e.g., names, personal numbers) are not allowed for statistical queries
**Example:**

<table>
<thead>
<tr>
<th>Record No.</th>
<th>Name</th>
<th>Sex</th>
<th>Age</th>
<th>Major</th>
<th>GP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Mayer</td>
<td>m</td>
<td>20</td>
<td>CS</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Smith</td>
<td>f</td>
<td>18</td>
<td>CS</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Sneyer</td>
<td>m</td>
<td>21</td>
<td>Math</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>Hall</td>
<td>m</td>
<td>21</td>
<td>Math</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>Bob</td>
<td>f</td>
<td>20</td>
<td>Math</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>Fisher</td>
<td>m</td>
<td>21</td>
<td>Math</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>Knuth</td>
<td>m</td>
<td>21</td>
<td>Math</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>Silver</td>
<td>m</td>
<td>21</td>
<td>CS</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>Cohn</td>
<td>m</td>
<td>19</td>
<td>CS</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>Sveniek</td>
<td>m</td>
<td>18</td>
<td>CS</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>Olofsson</td>
<td>m</td>
<td>19</td>
<td>CS</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td>Marian</td>
<td>f</td>
<td>19</td>
<td>Math</td>
<td>4</td>
</tr>
<tr>
<td>13</td>
<td>Anders</td>
<td>m</td>
<td>23</td>
<td>CS</td>
<td>4</td>
</tr>
<tr>
<td>14</td>
<td>Frank</td>
<td>m</td>
<td>22</td>
<td>Math</td>
<td>4</td>
</tr>
</tbody>
</table>

Name: identity data (identifying the persons)
Sex, Age, Major: demographic data (generally known to many people)
GP(student grades): analysis data (not publicly known, of interest for attackers)

**Statistical queries:**

- \( q(C,U) \) (or simply: \( q(C) \))
- \( q \): statistical function
- \( C \): characteristic formula, logical formula over the values of attributes using the operators OR, AND, NOT
- \( U \): subset of attributes

**Example:**

- \( \text{COUNT} \left( \left( \text{SEX} = \text{MALE} \right) \land \left( \text{MAJOR} = \text{CS} \right) \right) \) =>
  - Smith is the only female CS student.
- \( \text{SUM}(\left( \text{SEX} = \text{MALE} \right) \land \left( \text{MAJOR} = \text{CS} \right), \text{GP}) = \) Smith’s GP = 4

It is not sufficient to suppress only small query sets! The same statistics can be calculated by:

- \( \text{COUNT(ALL)} - \text{COUNT(\neg (\text{SEX} = \text{FEMALE}) \land \text{MAJOR} = \text{CS}))} \)
- \( \text{SUM(ALL, GP)} - \text{SUM(\neg (\text{SEX} = \text{FEMALE}) \land \text{MAJOR} = \text{CS}), GP)} \)

**Small and Large Query Set Attacks:**

Attacker knows that Smith is a female CS student:

\( \text{COUNT(\left( \text{SEX} = \text{FEMALE} \right) \land \left( \text{MAJOR} = \text{CS} \right))} \) =>

Smith is the only female CS student.

**Protection: Query Set Size Control:**

A statistic \( q(C) \) is permitted only if

\[ n \leq |C| \leq N - n \]

for parameter \( n \geq 2 \),

\( N \): size (No. of tuples) of database

\( q(\text{ALL}) \) can be computed from:

\[ q(\text{ALL}) = q(C) + q(\text{NOT C}) \]

for \( C \) such that \( n \leq |C| \leq N - n \)

However: Tracker attacks can still compromise security!
Individual Tracker:
Suppose: q(C) is rejected, because |(C)| = 1
C = C1 AND C2,   n ≤ |(C1)| ≤ N - n
n ≤ |(C1 AND NOT C2)| ≤ N - n
Individual Tracker: { C1, C1 AND NOT C2}

Individual Tracker Compromise: (for q : SUM or COUNT))
q(C) = q(C1 AND C2) = q(C1) - q(C1 AND NOT C2)

Example:
n = 3,
Individual Tracker = { (Major = CS), (Major = CS) AND NOT (SEX = f))
SUM((Major = CS) AND (Sex = f), GP)
= SUM ((Major = CS) AND NOT (Sex = f), GP)
= 20 - 16 = 4

General Tracker:
Characteristic Formula T such that
2*n ≤ |(T)| ≤ N - 2*n , n ≤ N/4
General Tracker Compromise:
q(ALL) = q(T) + q(not T)
If |(C)| < n:
q(C) = q(C or T) + q(C or not T) - q(ALL)

Example: n = 3,  T = (Sex= Male)
SUM ((SEX = FEMALE) AND (MAJOR = CS), GP) =
SUM ((SEX = FEMALE) AND (MAJOR = CS)) OR (SEX = MALE), GP) +
SUM (((SEX = FEMALE) AND (MAJOR = CS)) OR (NOT (SEX = MALE)), GP) -
SUM (ALL, GP)
= 31 + 10 - 37 = 4
SUM (ALL, GP) = SUM (SEX = MALE, GP) + SUM (NOT (SEX = MALE), GP)
Security Mechanisms for Statistical Databases:

Output Controls

Data Perturbation
(adds noise to data values)

Output Selection:
(rejects “sensitive” or “critical” statistics)
+ Query set size control
+ Maximum Order Control
+ Minimal frequency control

Output Modification:
(modifies statistics, adds small relative errors)
+ Rounding
+ Adding random numbers
+ Random sample queries