Module-II (Part-3)

Database Security

By : Aruna Khubalkar

Topics

- Database Security Requirements
- Reliability and Integrity
- Sensitive Data
- Inference Attacks
- Multilevel Database Security

By : Aruna Khubalkar

Introduction

Concept :

- A database is a collection of data and a set of rules that organize the data by specifying certain relationships among the data.
- The user describes a logical format for the data.
- The precise physical format of the file is of no concern to the user.

Introduction

- A database administrator is a person who defines the rules that organize the data and also controls who should have access to what parts of the data.
- The user interacts with the database through a program called a database manager or a database management system (DBMS), informally known as a front end.

Why Data Security?

- Databases are essential to many business and organizations.
- Database organization and the contents are considered valuable corporate assets that must be carefully protected.
- Protection provided by DBMS had mixed results.
- Several controls have been developed to handle security concerns.
-

Why Data Security?

- Amount of acquired data is increasing
- More sensitive data being exposed
- The advent of the Internet as well as networking capabilities has made the access to data much easier
- Damages and misuses of data affect not only a single user or an application; they may have disastrous consequences on the entire organization

Security Requirements

- Physical database integrity
- Logical database integrity
- Element integrity
- Auditability
- Access control
- User authentication
- Availability

Security Requirements

- Physical database integrity
  - immune to physical problems, such as power failures, media failure
    - physical securing hardware, UPS
    - regular backups
- Logical database integrity
  - structure is maintained i.e., reconstruction ability
    - maintain a log of transactions
    - replay log to restore the systems to a stable point

Security Requirements

- Element integrity
  - data contained in each database elements are accurate
    - field checks
      - allow only acceptable values
    - access controls
      - allow only authorized users to update elements
    - change log
      - used to undo changes made in error
    - referential Integrity (key integrity concerns)
    - two phase locking process

- Auditability
- - To track who or what has accessed (or modified) the elements
  log read/write to database

By : Aruna Khubalkar

Security Requirements

- Access Control (similar to OS)
  - User is allowed to access only authorized data
  - logical separation by user access privileges
- User Authentication
  - Every user is positively identified, both for the audit trail and for permission to access certain data.
- Availability
  - Users can access all the data for which they are authorized.
  - concurrent users
    - granularity of locking
  - Reliability

Integrity

- Integrity of the Database

- Two situations can affect the integrity of a database:
  - when the whole database is damaged
  - when individual data items are unreadable.

- Integrity of the database as a whole is the responsibility of
  - The DBMS
  - The operating system
  - The (human) computing system manager.

Integrity

- Integrity of the Database
- Sometimes it is important to be able to reconstruct the database at the point of a failure.

 The DBMS must maintain a log of transactions.

 The system can obtain accurate account balances by reverting to a backup copy of the database and reprocessing all later transactions from the log.

Element Integrity

The **integrity** of database elements is their correctness or accuracy.

This corrective action can be taken in three ways .

- **Field checks** - activities that test for appropriate values in a position.
- **Access control**
- A **change log** - A change log lists every change made to the database; it contains both original and modified values. Using this log, a database administrator can undo any changes that were made in error.

Auditability

- For some applications it may be desirable to generate an audit record of all access (read or write) to a database.
- Such a record can help to maintain the database's integrity, or at least to discover after the fact who had affected which values and when.

Access Control

- Databases are often separated logically by user access privileges.

User Authentication

 The DBMS can require rigorous user authentication.
 A DBMS might insist that a user pass both specific password and time-of-day checks.
 This authentication supplements the authentication performed by the operating system.

Integrity/Confidentiality/Availability – Computer Security

- Integrity is a major concern in the design of database management systems.
- Confidentiality is a key issue with databases because of the inference problem,
- Availability is important because of the shared access motivation underlying database development.

Reliability and Integrity

Databases amalgamate data from many sources, and users expect a DBMS to provide access to the data in a reliable way.

- **Reliability** - mean that the software runs for very long periods of time without failing.

Reliability and Integrity

Database concerns about reliability and integrity can be viewed from three dimensions:

 Database integrity: concern that the database as a whole is protected against damage
 Element integrity: concern that the value of a specific data element is written or changed only by authorized users.
 Element accuracy: concern that only correct values are written into the elements of a database.

Reliability and Integrity

- Protection Features from the Operating System
- Two-Phase Update
- Redundancy/Internal Consistency
- Error Detection and Correction Codes
- □ Shadow Fields
- Recovery
- Concurrency/Consistency
- Monitors

  Range Comparisons

  Filters or patterns

  State constraints

  Transition constraints

Summary of Data Reliability :

- Reliability, correctness and integrity are three closely related concepts in databases.
- Users trust the DBMS to maintain their data correctly, so integrity issues are very important in database security.

Sensitive Data

- Sensitive data are data that should not be made public.
- There exist cases that some but not all of the elements in the database are sensitive.
- There may be varying degrees of sensitivity.

Sensitive Data

- Several factors can make data sensitive.
- Inherently sensitive.
- The value itself may be so revealing that it is sensitive. Examples are the locations of defensive missiles.
- From a sensitive source.
- The source of the data may indicate a need for confidentiality.
- An example is information from an informer whose identity would be compromised if the information were disclosed.

Sensitive Data

- Several factors can make data sensitive (contd..)
- Declared sensitive.
- The database administrator or the owner of the data may have declared the data to be sensitive.
- Part of a sensitive attribute or a sensitive record.
- In a database, an attribute or record may be classified as sensitive.
- Sensitive in relation to previously disclosed information.
- Some data become sensitive in the presence of other data.

Access Decisions

- DBMS may consider several factors when deciding whether to permit an access.
- Availability of the data
- Acceptability of the access
- Authenticity of the user

Access Decisions

- **Availability of Data :**
- One or more required elements may be inaccessible.
- For example, if a user is updating several fields, other users accesses' to those fields must be blocked temporarily.
- This blocking ensures that users do not receive inaccurate information.

Access Decisions

- **Acceptability of Access :**
- One or more values of the record may be sensitive and not accessible by the general user.
- A DBMS should not release sensitive data to unauthorized individuals.

Access Decisions

- **Assurance of Authenticity :**
- Certain characteristics of the user external to the database may also be considered when permitting access.
- Example, to enhance security, the database administrator may permit someone to access the database only at certain times, such as during
- working hours.

Access Decisions

- **Types of Disclosures**
- Data can be sensitive and also its characteristics.
- Discriptive information about data is a form of disclosure.
- Exact Data - The most serious disclosure is the exact value of a sensitive data item itself
- Bounds - Another exposure is disclosing bounds on a sensitive value; indicating that a sensitive value, y, is between two values, L and H.

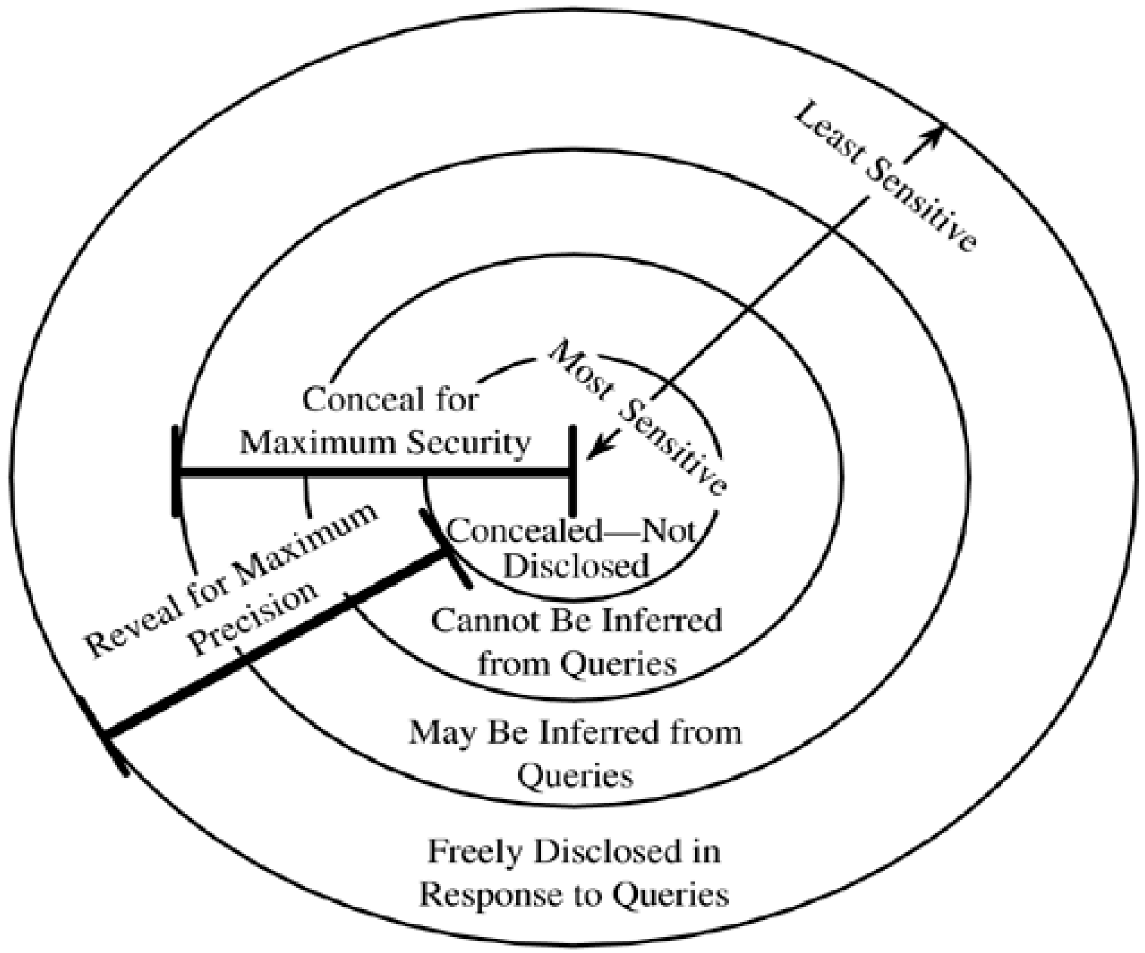
Access Decisions

- **Types of Disclosures**
- Negative Result - Sometimes we can word a query to determine a negative result. That is, we can learn that z is not the value of y.
- Existence - The existence of data is itself a sensitive piece of data.
- Probable Value - it may be possible to determine the probability that a certain element has a certain value

Security vs. Precision

- Precision: revealing as much non sensitive data as possible
  - disclose as much data as possible
  - Issue: User may put together pieces of disclosed data and infer other, more deeply hidden, data
- Security: reveal only those data that are not sensitive
  - rejecting any query that mentions a sensitive field
  - Issue: may reject many reasonable and non disclosing queries
- The ideal combination : perfect confidentiality with maximum precision
  - achieving this goal is not easy !

Security vs. Precision


Least Sensitive
Most Sensitive
Conceal for Maximum Security
Reveal for Maximum Precision
Concealed—Not Disclosed
Cannot Be Inferred from Queries
May Be Inferred from Queries
Freely Disclosed in Response to Queries

By : Aruna Khubalkar

Security vs. Precision

- Precision: revealing as much non sensitive data as possible
  - disclose as much data as possible
  - Issue: User may put together pieces of disclosed data and infer other, more deeply hidden, data
- Security: reveal only those data that are not sensitive
  - rejecting any query that mentions a sensitive field
  - Issue: may reject many reasonable and non disclosing queries

- The ideal combination : perfect confidentiality with maximum precision
  - achieving this goal is not easy !

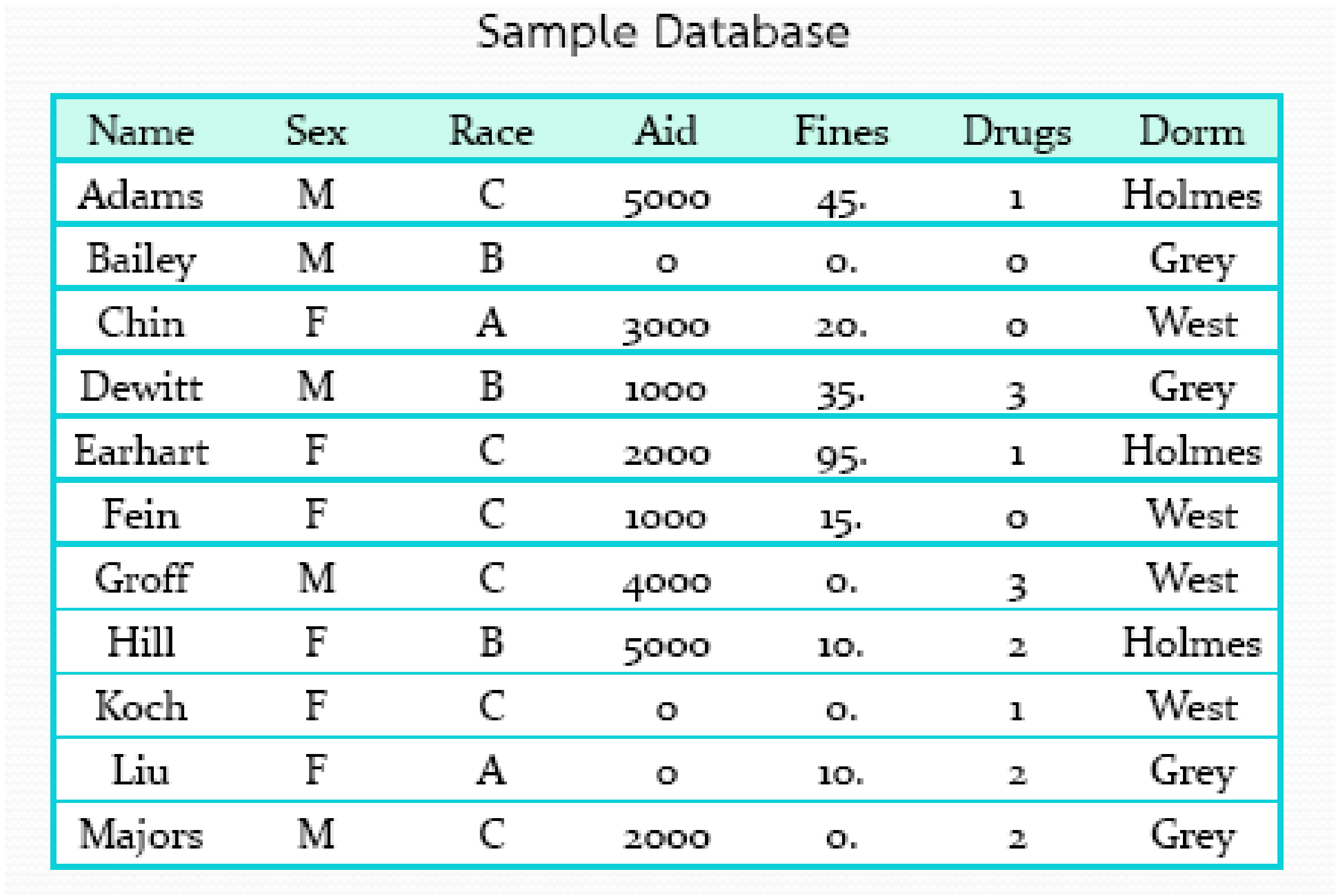
Inference

Inference is a way to infer or derive sensitive data from non-sensitive data

- - attacker combines information from outside the database with database responses

- Security issue with statistical databases

Different Ways for Inference



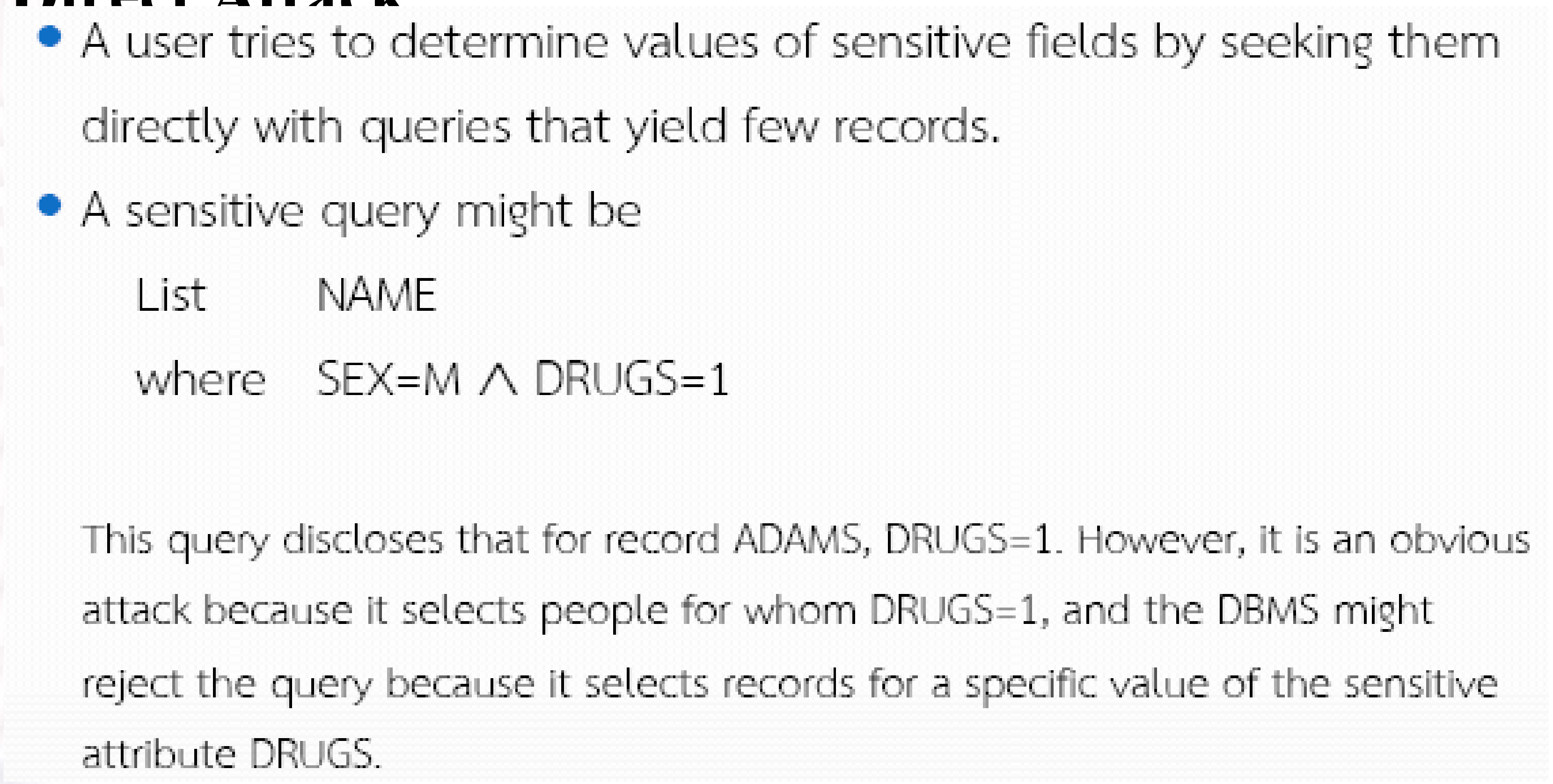
Sample Database

Name	Sex	Race	Aid	Fines	Drugs	Dorm
Adams	M	C	5000	45.	1	Holmes
Bailey	M	B	0	0.	0	Grey
Chin	F	A	3000	20.	0	West
Dewitt	M	B	1000	35.	3	Grey
Earhart	F	C	2000	95.	1	Holmes
Fein	F	C	1000	15.	0	West
Groff	M	C	4000	0.	3	West
Hill	F	B	5000	10.	2	Holmes
Koch	F	C	0	0.	1	West
Liu	F	A	0	10.	2	Grey
Majors	M	C	2000	0.	2	Grey

By : Aruna Khubalkar

Different Ways for Inference

- **Direct Attack**



- A user tries to determine values of sensitive fields by seeking them directly with queries that yield few records.
- A sensitive query might be

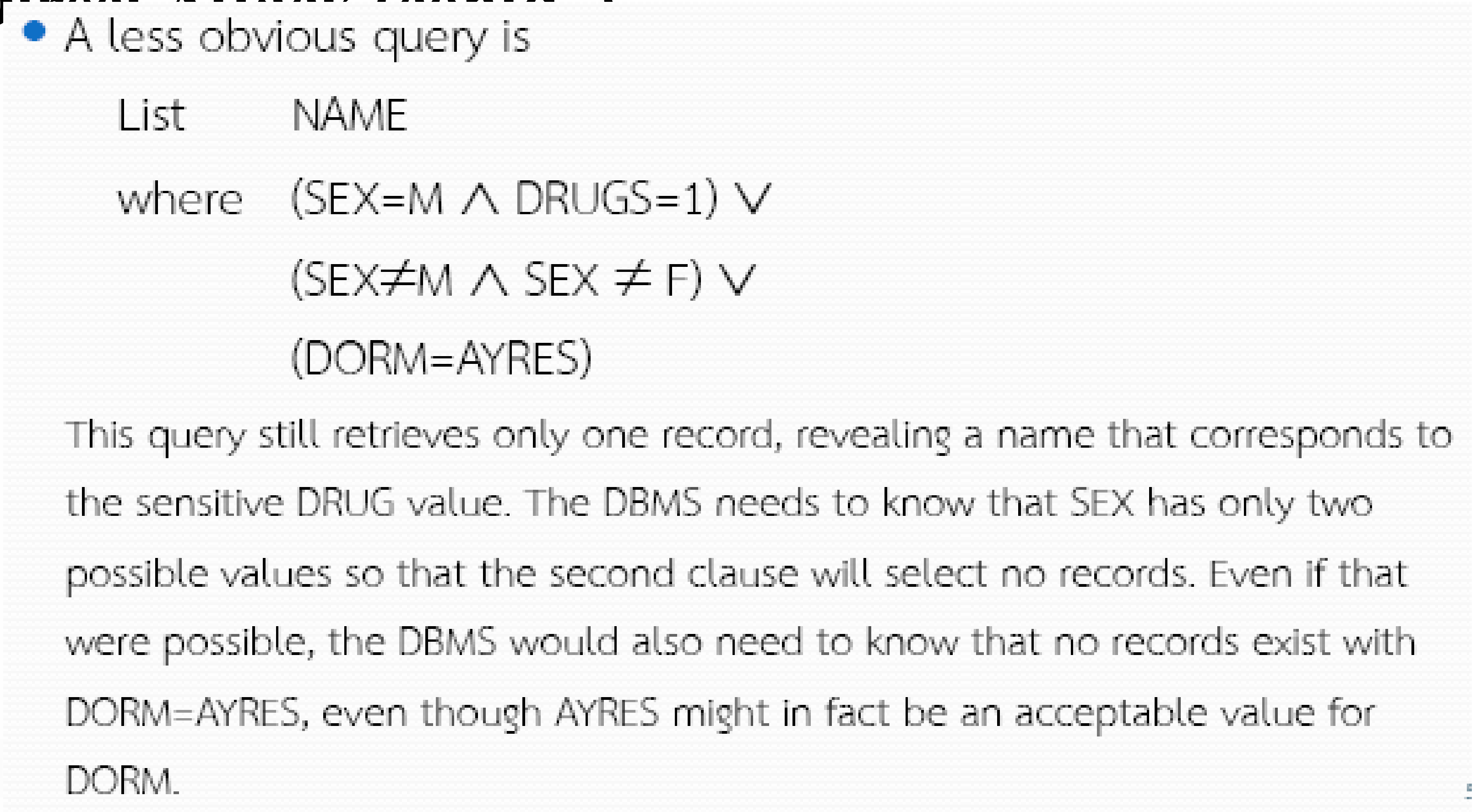
```
List    NAME
where   SEX=M ∧ DRUGS=1
```

This query discloses that for record ADAMS, DRUGS=1. However, it is an obvious attack because it selects people for whom DRUGS=1, and the DBMS might reject the query because it selects records for a specific value of the sensitive attribute DRUGS.

By : Aruna Khubalkar

Different Ways for Inference

- **Direct Attack (contd..)**



- A less obvious query is

List NAME

where $(SEX=M \wedge DRUGS=1) \vee$

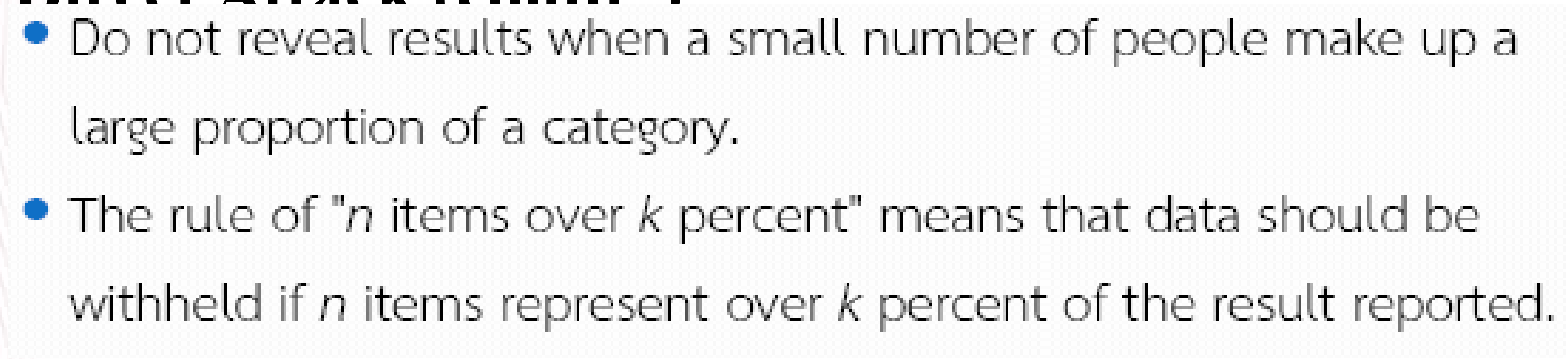
$(SEX \neq M \wedge SEX \neq F) \vee$

$(DORM=AYRES)$

This query still retrieves only one record, revealing a name that corresponds to the sensitive DRUG value. The DBMS needs to know that SEX has only two possible values so that the second clause will select no records. Even if that were possible, the DBMS would also need to know that no records exist with DORM=AYRES, even though AYRES might in fact be an acceptable value for DORM.

Different Ways for Inference

- **Direct Attack (contd.)**

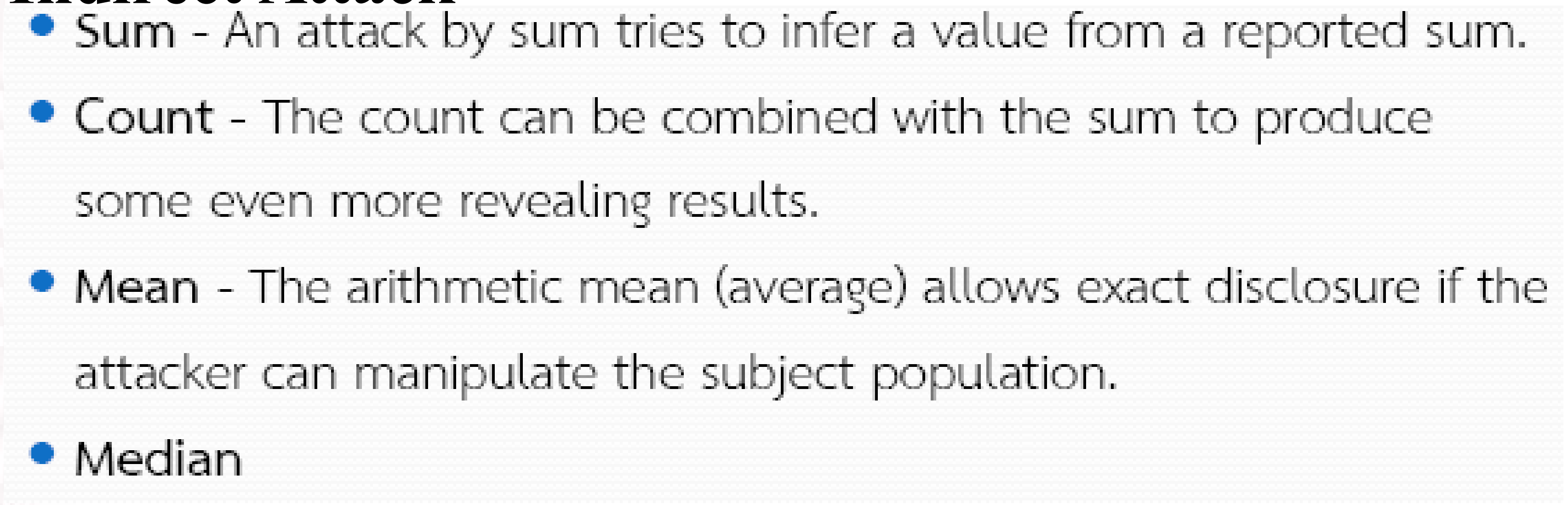


- Do not reveal results when a small number of people make up a large proportion of a category.
- The rule of "*n* items over *k* percent" means that data should be withheld if *n* items represent over *k* percent of the result reported.

By : Aruna Khubalkar

Different Ways for Inference

- **Indirect Attack**

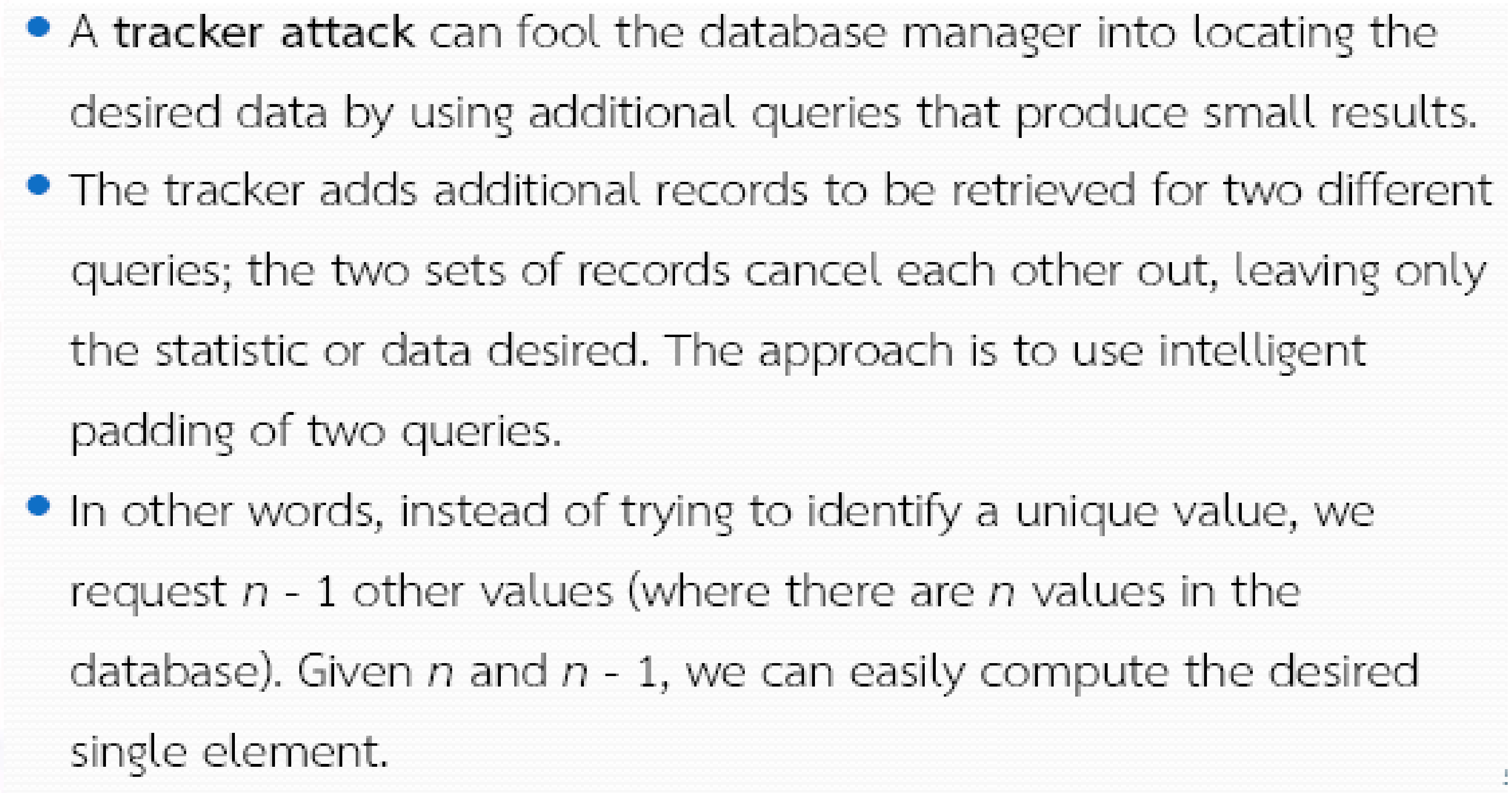


- **Sum** - An attack by sum tries to infer a value from a reported sum.
- **Count** - The count can be combined with the sum to produce some even more revealing results.
- **Mean** - The arithmetic mean (average) allows exact disclosure if the attacker can manipulate the subject population.
- Median

By : Aruna Khubalkar

Different Ways for Inference

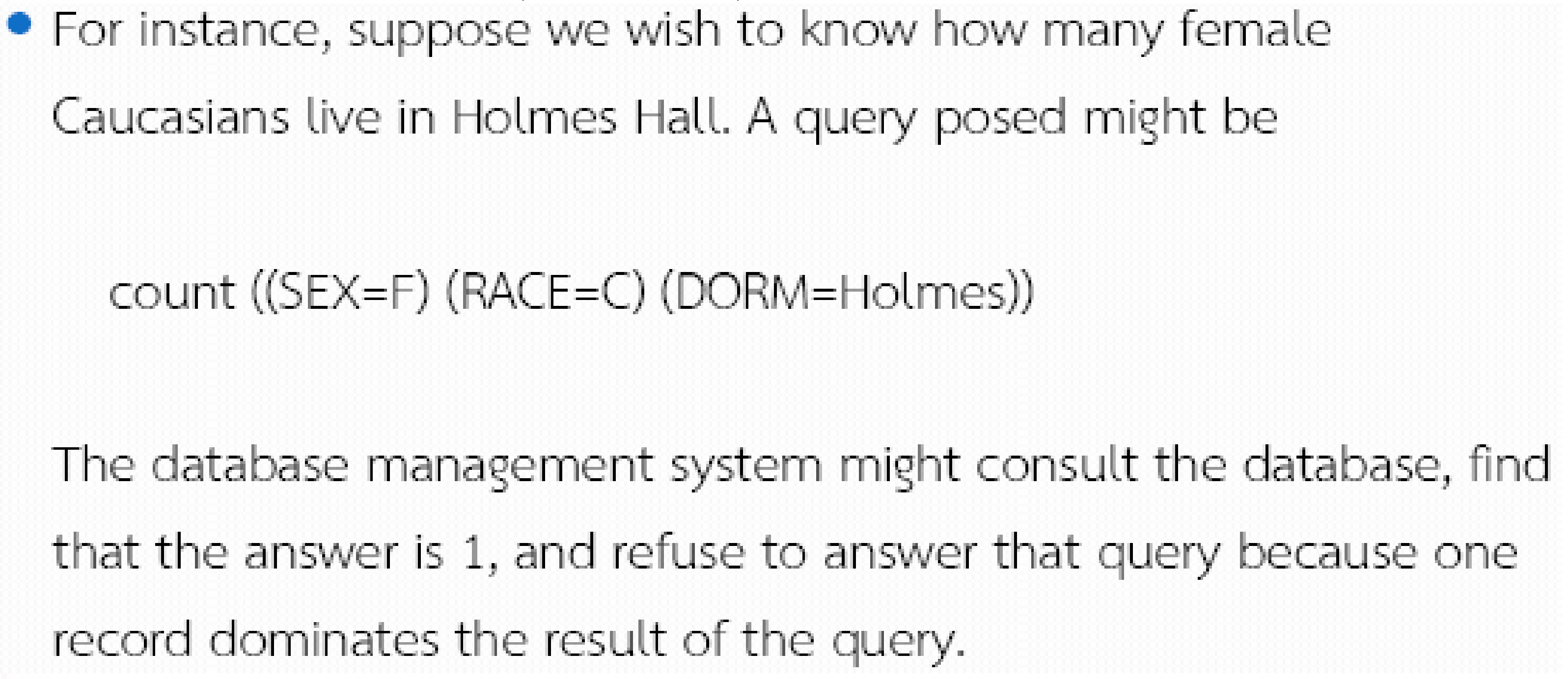
- **Tracker Attacks**



- A **tracker attack** can fool the database manager into locating the desired data by using additional queries that produce small results.
- The tracker adds additional records to be retrieved for two different queries; the two sets of records cancel each other out, leaving only the statistic or data desired. The approach is to use intelligent padding of two queries.
- In other words, instead of trying to identify a unique value, we request *n* - 1 other values (where there are *n* values in the database). Given *n* and *n* - 1, we can easily compute the desired single element.

Different Ways for Inference

- **Tracker Attacks (contd..)**



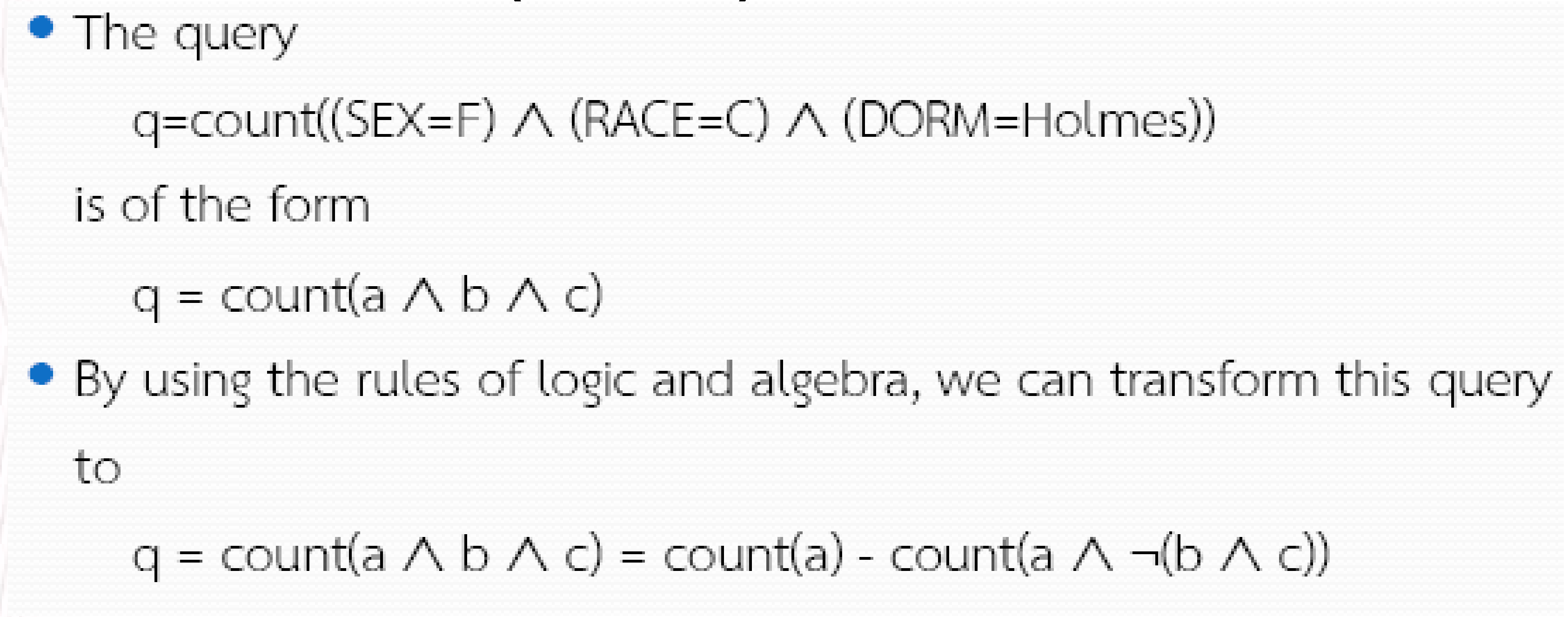
- For instance, suppose we wish to know how many female Caucasians live in Holmes Hall. A query posed might be

  count ((SEX=F) (RACE=C) (DORM=Holmes))

  The database management system might consult the database, find that the answer is 1, and refuse to answer that query because one record dominates the result of the query.

Different Ways for Inference

- **Tracker Attacks (contd..)**



- The query

  q=count((SEX=F) ∧ (RACE=C) ∧ (DORM=Holmes))

  is of the form

  $q = count(a \wedge b \wedge c)$

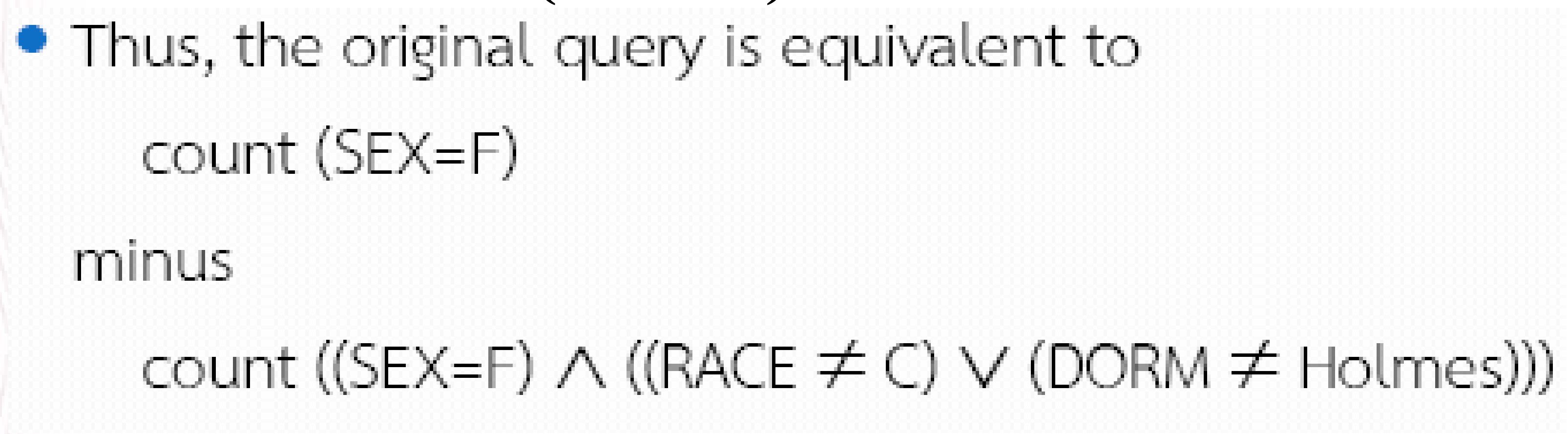
- By using the rules of logic and algebra, we can transform this query to

  $q = count(a \wedge b \wedge c) = count(a) - count(a \wedge \neg(b \wedge c))$

By : Aruna Khubalkar

Different Ways for Inference

- **Tracker Attacks (contd..)**



- Thus, the original query is equivalent to

  count (SEX=F)

  minus

  count ((SEX=F) $\wedge$ ((RACE $\neq$ C) $\vee$ (DORM $\neq$ Holmes)))

Controls for Statistical Inference Attacks

2 ways to protect inference attacks

1)Controls applied to queries

- difficult to determine whether query discloses sensitive data.

- query control are effective against direct attacks

2)Controls applied to data items

- suppression and concealing are two controls applied to data items.

Controls for Statistical Inference Attacks

- **Suppression :**
  sensitive data values are not provided; the query is rejected without response.

- **Concealing :**
  answer provided is *close to but not exactly the* actual value.

Controls for Statistical Inference Attacks

- These two controls reflect the contrast between security and precision.
  - With suppression, any results provided are correct, yet many responses must be withheld to maintain security.
  - With concealing, more results can be provided, but the precision of the results is lower.

  The choice between suppression and concealing depends on the context of the database.

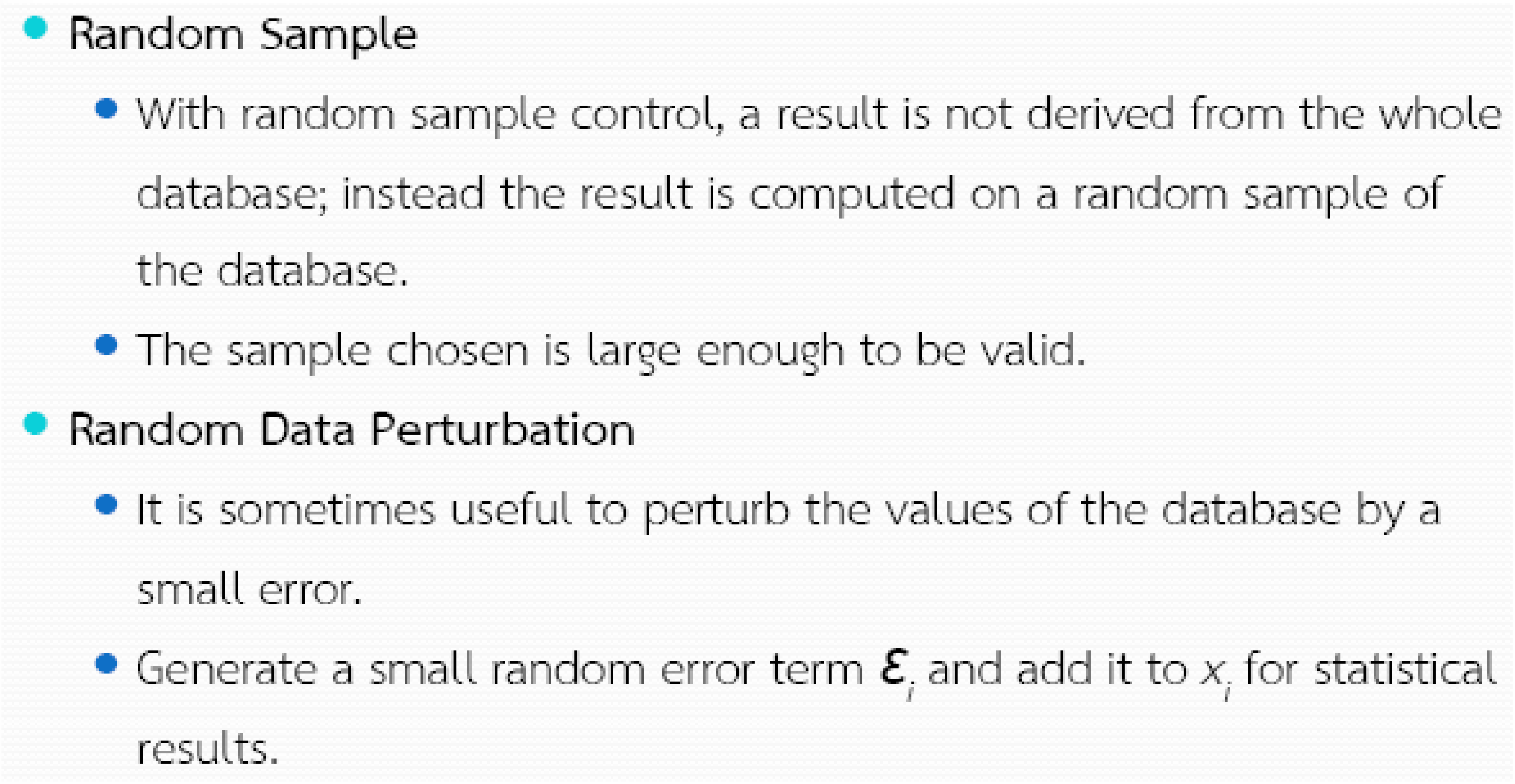
By : Aruna Khubalkar

Controls for Statistical Inference Attacks

Examples of controls - suppression and concealing

- Limited response suppression:
  - Rule of n items over k percent: eliminate low-frequency elements.
  - More sufficient way: suppress additional cells on the same row and column. (Table 8-6 and 8-7)
- combining results:
  - To combine some rows and columns. (Table 8-8 and 8-9)
  - Or to present values in ranges.
  - Or to present values by rounding.

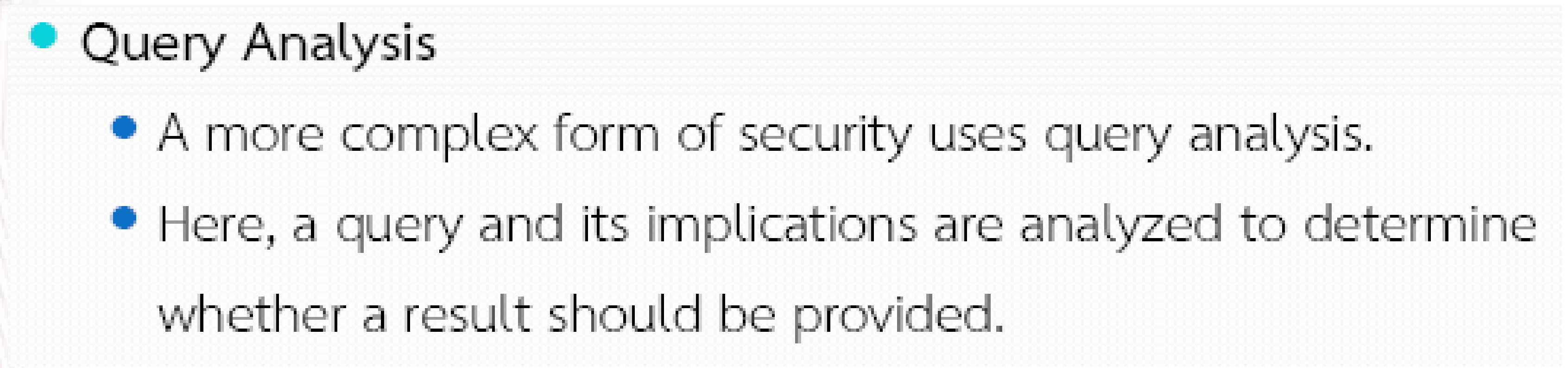
Controls for Statistical Inference Attacks



- Random Sample
  - With random sample control, a result is not derived from the whole database; instead the result is computed on a random sample of the database.
  - The sample chosen is large enough to be valid.
- Random Data Perturbation
  - It is sometimes useful to perturb the values of the database by a small error.
  - Generate a small random error term ε_i and add it to x_i for statistical results.

By : Aruna Khubalkar

Controls for Statistical Inference Attacks



- Query Analysis
  - A more complex form of security uses query analysis.
  - Here, a query and its implications are analyzed to determine whether a result should be provided.

- Requirement:
  - Maintain a query history for each user.
  - Judge each query on the context of previous queries.

Conclusion on Inference Attacks

- No perfect solutions to the inference problem.
- The approaches to controlling it
  - *Suppress obviously sensitive information.*
  - *Track what the user knows.*
  - *Disguise the data.*

Multilevel Databases

- An illusion: Sensitivity was determined just by attribute. ▭ sensitive vs nonsensitive attributes
- The fact: differentiated data security
- Three characteristics of database security:
  - The security of a single element may be different from the others in the same row or column.
  - Several grades of security are needed.
  - The security of an aggregate (a sum, a count, or a group of values) may be different from the security of the individual elements.

Multilevel Databases —— Granularity

- An analogy: Defining the sensitivity of each value in a data base is similar to applying a sensitivity level to each individual word of a document.
- Both element and combination of elements may have a distinct sensitivity.
- In order to keep each value of a database being in its own sensitivity level:
  - An access control policy must define which users can have access to what data.
- **confidentiality** (or **secrecy**)
  - Each value must be guaranteed NOT to be changed by any unauthorized person.
- **integrity**

Multilevel Databases - Security Issues (1)

- Integrity
  - *-property for access control: In multilevel databases a high-level user should not be able to write a lower-level data element.
  - Problem: Sometimes read and write must happen to the same process, like DBMS.
  - Solution: Either the process cleared at a high level cannot write to a lower level, or the process must be a “trusted process”.

Multilevel Databases - Security Issues (2)

- Confidentiality
  - In multilevel databases two different users from different levels of security may get two different answers to the same query.
  - **Unknowing redundancy**, known as **polyinstantiation**, may be introduced; that is, one record can appear many times, with different level of secrecy each time.

Proposals for Multilevel security - Partitioning

- Model: The database is divided into separate databases, each at its own security level.
- Weakness:
  - Destroy basic advantage of a database: elimination of redundancy.
  - Problem with combined usage of separate databases.

Proposals for Multilevel security - Encryption

- Model: Each level of sensitive data is stored in a table encrypted under a key unique to the level of sensitivity.
- Weakness:
  - Chosen plaintext attack may increase.

Controls: different encryption keys, cipher block chaining (Fig. 8-11, p.365)

  - High overhead of processing a query: decrypting required fields.
- Encryption is not often used to implement separation in data bases.

Proposals - Integrity and Sensitivity Locks (1)

- Integrity lock (spray paint): The protection is made with elements, not with tables.

- Each data item includes (Figure 8-12):
  - Data itself: stored in plaintext for efficiency of access.
  - Sensitivity label: defines the sensitivity of the data.

Sensitivity labels are *unforgeable, unique,* and *concealed.*

  - Checksum: computed across both data and sensitivity label. (Cryptographic checksum: Figure 8-13)

Proposals - Integrity and Sensitivity Locks (2)

- **Sensitivity lock**: is a combination of a unique identifier and the sensitivity level. (Figure 8-14)
  - Sensitivity lock = E(Key, Sensitivity label, unique identifier)
- Intention:
  - Use any (untrusted) database manager with a trusted procedure that handles access control. (Figure 8-15)

Proposals - Integrity and Sensitivity Locks (3)

- It's only a short-term solution for multilevel security.
- Weakness:
  - Efficiency of integrity locks is a serious drawback.
  - The space required is expanded.
  - The processing time of decoding sensitivity label is a problem.
  - Trojan horse attack: database manager sees all data.

Proposals – Trusted Front-End

- Model (Figure 8-16):
  - A trusted front-end (known as a *guard*) is added.
  - Takes advantage of existing tools and expertise, enhancing security with minimal change to system.
  - Interaction b/w user, front-end, and DBMS – 9 steps (page368)
  - Inefficient as much data is retrieved and then discarded as inappropriate for the user.

Proposals – Commutative filters

- Simplification of trusted interface to DBMS
- **Commutative filters**:
  - Reformat user's request if necessary and return only data of appropriate sensitivity to the user.
  - Advantage:

• Database manager can do as much as possible, like selection, optimization, subquery …

• Overall efficiency of the system.

Proposals – Distributed/federated Database

- Operation of a trusted front-end:
  - Control access to two database managers with different sensitivity.
  - Take user's query and formulate single-level queries to the databases as appropriate.
  - Return results to user, combining results appropriately if they are obtained from different databases.
- Weakness:
  - The front-end is potentially including most of the functionality of a full database manager.
  - Data must be kept in separate databases according to their different sensitivity degrees.

Proposals – Window/View

- Window
  - A window is a subset of a database, containing exactly the information that a user is entitled to access.
- View
  - A view can represent a single user's subset database, so that all of a user's queries access only that subset database and sensitive data to the user may be filtered.

Proposals – Window/View

- Sea Views project
- Integrates trusted OS to form a trusted database manager
  - A Layered system (TCB, trusted computing base):

- First /lowest layer (reference monitor): file interaction, enforcing BLP access control & user authentication.
- Second layer: basic indexing & computation functions of database.
- Third layer: translate views into the base relations.

- The remaining layers implement normal DBMS functions and UI.

References

- *Security in Computing* by Charles P. Pfleeger, Chapter 8