IS53012B/A Computer Security

Dr Ida Pu

Room 10, 29 St James
Goldsmiths, University of London

2019-20 (since 2007)

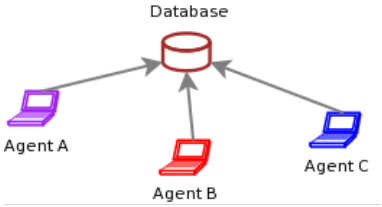
Part I

Database security

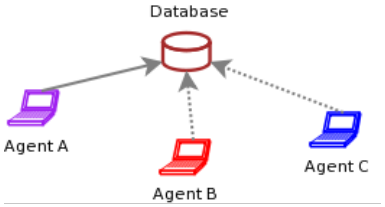
Example

Airplane ticket booking system

Three agents A, B and C are connected to a central server.



B and C are idle, or B and C compete the connection.



Databases

database A collection of data in files, and rules used to organise the data, and constantly running software program to provide services

roles administrators, users, database management system (DBMS)

administrator Persons who define the rules and control the access to databases

users People who use (and own sometimes) the resources of the databases 24 hours everyday; local or remote

DBMS Software that help database maintenance and enable interactions between the users and databases also referred to as *database manager*, or informally *front end*

Data and organisation

Records of mixed types of data (attributes, fields, tables) and management software

schema Logical structure of a database

subschema Schema for part of a database; interested and queried records that form a subtable of the original database

attribute The name of each column

relation A set of columns.

Examples I

Schema tables, records, attributes, data types.

Example

ID	FNAME	GName	Date of Birth	Course	Health

Records, attributes, and data types

Example

ID	FNAME	GName	Date of Birth	Course	Health
00010090	ALLEN	Alex	01 07 1989	IC	1
00010100	ADAMS	Ben	18 08 1986	CC	2
00100101	CARTER	Michael	10 10 1977	CS	2

Example II: Subschema

The part for which a particular user has the access to
Consider different personnel, such as a system administrator, head of the department, lecturers, students, each has different access permissions.

Lecturers

ID	FNAME	GName	Date of Birth	Course	Health
00010090	ALLEN	Alex	01 07 1989	IC	1
00010100	ADAMS	Ben	18 08 1986	CC	2
00100101	CARTER	Michael	10 10 1977	CS	2

Students

ID	FNAME	GName	Date of Birth	Course	Health
00010090	ALLEN	Alex	01 07 1989	IC	1
00010100	ADAMS	Ben	18 08 1986	CC	2
00100101	CARTER	Michael	10 10 1977	CS	2

Examples III - Queries

Lecturers

ID	FNAME	GName	Course	Health
00010090	ALLEN	Alex	IC	1
00010100	ADAMS	Ben	CC	2
00100101	CARTER	Michael	CS	2

Students

ID	FNAME	GName	Course
00010090	ALLEN	Alex	IC
00010100	ADAMS	Ben	CC
00100101	CARTER	Michael	CS

Example

```
SELECT FNAME = 'ALLEN'
SELECT (Course = 'CS') AND (FNAME = 'AL')
```

Advantages of using databases

shared access Available for many users
minimal redundancy Users do not need to collect/maintain their own data.
data consistency A change to one value applies to all users.
data integrity Data are protected against accidental or malicious attacks.
controlled access Allows only valid users.

Problems

Security interests may have conflicts with performance.

Security requirements

- Integrity
  - physical database integrity no physical problems such as power failures, object missing or damaged
  - logical database integrity preserve the structure and restrict the modification to data fields and keep them normalised
    - records accurate and consistent
    - auditability detailed system logs and be possible to trace the past activities
- Confidentiality
  - access control allow accesses only to authorised subjects
  - user authentication every user is correctly identified.
- Availability
  - access control Users access to all the authorised data.
  - anywhere, any time local or remote, 24 hours

Integrity of databases

Correctness and accuracy

- updates are only performed by authorised individuals
- data are protected from corruption.

Responsibilities of the integrity of databases:

- DBMS
- operating system
- human computing system manager.

3D view to Reliability and Integrity

Database integrity database as a whole is protected against damages and accidents.
Element integrity values of specific data elements are written only by authorised individual.
Element accuracy only allow correct values to be written into the elements of databases

Record integrity

- Data correctness
  - field checks Monitoring the input data to databases;
  - access control Restrict certain users' access or block certain queries
  - change logs A full record of every change made to the database
- Update integrity
  - timely Frequently and in time

Two-phase update

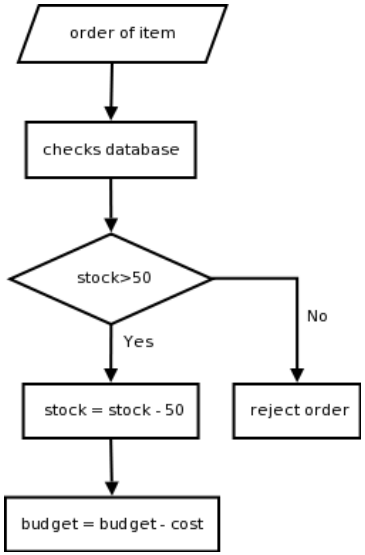
Intent phase DBMS gathers the resources required to perform an update
Committing phase writing a **commit flag** and make a permanent changes

(These must be repeatable).

Commitment of making permanent changes

Any failed operation in Committing Phase may cause incomplete data in the database, but DBMS should be able to repair it.

Example - i



- Central administration office
- Inventory in database
- Each department has a budget to cover the cost of its order

Example - ii

Intent:

1. Check the value of COMMIT-FLAG, halt if it is on
2. Compare stock to order_amount, halt if not enough stock
3. tmp_stock = STOCK − order
4. tmp_budget = BUDGET − cost
5. Check if it requires top-up the stock, update tmp_reorder.

Commit:

1. Set COMMIT-FLAG in database
2. Copy tmp_stock → STOCK
3. Copy tmp_budget → BUDGET
4. Copy tmp_reorder → REORDER
5. Clear COMMIT-FLAG in database

Note: It is still possible for the system to fail after 'transaction complete' and before clearing the COMMIT-FLAG.

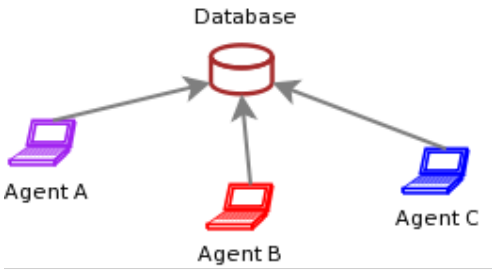
Concurrency

Problems for multiuser systems. Certain operations are more problematic than others.

- Read: concurrent read would cause no problem
- Write: concurrent write conflicts
- Delete: ?
- Insert: ?

- Build Hadrian's Wall (a structure 120km in length and 2 meters high to separate England and Scotland)
- Classroom booking
- Airplane (or concert) seat reservation
- World cup
- Duplex communication.

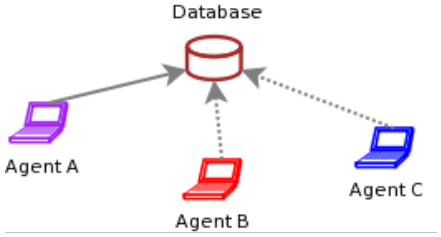
Flight seat reservation I

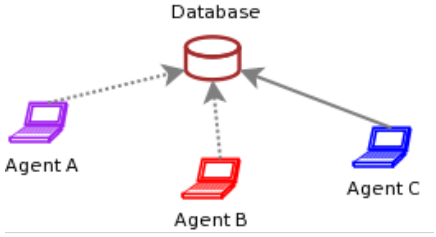



```
Agent A: SELECT (SEAT.NO='12A')
         ASSIGN 'H.Smith' TO PASSENGER.NAME

Agent B: SELECT (SEAT_NO='12A')
         ASSIGN 'J.Manwood' TO PASSENGER_NAME
```


Flight seat reservation II





Flight seat reservation III

Example

Seat booking Let the state be represented by a flag variable UNASSIGNED.

```
Agent A: SELECT (SEAT.NO='12A')
         IF UNASSIGNED THEN
            UNASSIGNED='FALSE'
            ASSIGN 'H.Smith' TO PASSENGER.NAME

Agent C: SELECT (SEAT_NO='12A')
         IF UNASSIGNED THEN
            UNASSIGNED='FALSE'
            ASSIGN 'M.Brown' TO PASSENGER_NAME
```

Monitors

Prerequisites

Range comparisons set and enforce an acceptable range of values and reject those attempted writing out-ranged values into databases

State constraints describe the conditions of the entire database

Transition constraints describe conditions necessary before (and/or after) changes can be applied to a database.

Example

Name	Sex	Salary	Role	Fines	Drugs	Accommodation
Adams	M	5000	President	45	1	Dean House
Allen	F	3000	Cleaner	25	0	Loring
Peter	M	113000	**Cleaner**	25	0	Loring

Sensitive data

Sensitive data are data that should be remained hidden from the public.

Easy cases:
- nothing sensitive
- everything is sensitive, e.g.

Difficult cases: some but not all of the records are sensitive in the database.

It can be very subjective!

Examples

ID	FNAME	GName	Course	Award
00010090	ALLEN	Alex	IC	
00010100	ADAMS	Ben	CC	best-student
00100101	CARTER	Michael	CS	

ID	FNAME	GName	Crime	Personal Experience
00010090	ALLEN	Alex	Plagiarism	Hated
00010100	ADAMS	Ben	Rumour	Loved
00100101	CARTER	Michael	Prison	Harmed

Name	Sex	Race	Aid	Fines	Drugs	Accommodation
Adams	M	C	5000	45	1	Dean House
Allen	F	A	3000	25	0	Loring

Types of sensitive data

inherently sensitive value itself is sensitive

part of a sensitive attribute or record an entire attribute or record may be classified as sensitive

sensitive source the source of the data is confidential

declared sensitive administrators or owners declare certain data are sensitive

jointly sensitive data that are sensitive in relation to previously disclosed information.

Types of sensitive data

Example

- locations of defensive missiles
- median income of doctors with only one doctor in the town
- salary fields in a personnel database or a record describing a secret space mission
- classified military data, an informer when information is released
- military data
- name of the anonymous donor of art painting
- longitude and latitude coordinates of a secret gold mine.

Example

Name	Sex	Race	Loans (£)	Fines (£)	Drugs (#0–3)	Address
Alice	F	C	5000	45	1	Green, London
Bob	M	B	0	0	0	Grey, Essex
Bailey	M	C	1000	20	2	Coventry
Charlie	M	A	1000	35	3	101, Oxford
David	M	A	8000	10	1	101, Manchester

Types of disclosures

Exact data Sensitive data are released to unauthorised users.

Bounds indicating a sensitive value is within the range between of two values: *low* and *high*, [low, high].

Negative result learning some data is of uncertain value.

Existence Sometimes knowing certain things exist itself is sufficient to cause harm, regardless of actual value.

probable value knowing certain information with some degree of uncertainty

partial disclosure Certain characteristics of sensitive data are released.

Inference

A way to derive sensitive data from insensitive data.

Inference problem is a subtle vulnerability in database security.

Name	Sex	Race	Loans (£)	Fines (£)	Drugs (#0–3)	Address
Alice	F	C	5000	45	1	Green, London
Bob	M	B	0	0	0	Grey, Essex
Bailey	M	C	1000	20	2	Coventry
Charlie	M	A	1000	35	3	101, Oxford
David	M	A	8000	10	1	101, Manchester

Consider the queries:

```
1. list NAME where DRUGS=1
2. list NAME where SEX=F and DRUGS=1
3. list NAME where SEX=F
4. list NAME where (SEX!=M and DRUGS=1) or
       (SEX!=M and SEX!=F) or (Address=Glasgow)
```

Answers

1. (Alice, David)
2. Alice
3. Alice
4. Alice

Direct attacks vs. indirect attacks

An attack by

- **Sum** may infer a value from a reported sum.
- **Count** combined with sum.
- **Mean** average allows exact disclosure if information about the population is known.
- **Median** an individual value can be determined from medians.

Examples: What are the results?

Name	Sex	Race	Loans (£)	Fines (£)	Drugs (#0–3)	Course
Alice	F	C	5000	35	1	CC
Bobi	F	B	0	20	0	CIS
Bailey	M	C	1000	35	2	IT
Charlie	M	A	2000	20	3	CS
David	M	A	8000	10	1	CS

1. list NAME where total(LOANS)>0
2. list NAME where total(LOANS)=0
3. list NAME where (total(LOANS)>0) & (SEX=F)
4. list NAME where (total(LOANS)>0) & (count(SEX=F)<2)
5. y <- median(FINES),
   x <- median(LOANS where SEX=M),
   list NAME where x & y

Answers

1. Alice, Bailey, Charlie, David
2. Bobi
3. Alice, Bobi
4. [] (empty)
5. y is 20, x is 2000,
   x is (Bobi, Charlie), y is (Bailey, Charlie, David).
   so list NAME where x & y is Charlie.

Tracker attacks

Try to 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, to let the two sets of records cancel each other out, leaving only the statistic or data desired.
Consider

```
1. count ((SEX=F) & (RACE=C)) will be rejected by
      a 'secured' DBMS as there is only 1 record is found.
2. count (SEX=F) -
      count (SEX=F) & ((RACE!=C) & (COURSE!=CC))
This is equivalent to
    count (SEX=F) & (RACE=C) & (COURSE=CC)
```


Linear system vulnerability

It is possible to construct a series of queries that returns results relating to several different sets.
For example, let $q_1, \cdots, q_5$ be 5 queries that do not reveal any single value of $c1, \cdots, c5$ from a database. However, if the following relationships is found, then the values of $c1, \cdots, c5$ can be devised.

$$q_1 = c_1 + c_2 + c_3 + c_4 + c_5,$$

$$q_2 = c_1 + c_2 + c_4,$$

$$q_3 = c_3 + c_4,$$

$$q_4 = c_4 + c_5,$$

$$q_5 = c_2 + c_5$$

Data and information I

- A way to build sensitive results from less sensitive input data
- A related problem to inference problem
- Data mining is the process of shifting through multiple databases and correlating multiple data elements to find useful information.

Data and information II

Example

Iceland protects privacy against inference

- distinctive family features disclose individuals' anonymity due to partially published birth and death records
- life's history of medical events may identify an individual, broken legs, car accidents etc
- small sample set restrictions on queries may fail to protect against algebraic attacks
- data mining techniques may lead to arbitrary selection of combinations of the results.

Separation

Partitioning A database is divided into separate databases, each at its own level of sensitivity.
Advantages: Disadvantages:

Encryption Sensitive data are encrypted.
Advantages: Disadvantages:

1. mount a plaintext attack: user can easily decrypt an encrypted field by creating a new field of his own data, encrypting it and making a comparison to the encrypted unknown data.
2. impossible to use a different encryption key for each record: too time consuming for normal database queries.

Integrity lock A way to provide both integrity and limited access for a database

Integrity lock

- It was first proposed at the U.S. Air Force Summer Study on Databases. Security 1983
- Add extra fields Sensitivity Mark and Checksum
- Sensitivity Mark defines the degree of the sensitivity.
- Checksum can be computed to prevent unauthorised data modification.
- One serious drawback of integrity locks is the efficiency problem (space and time efficiency).

Cryptographic checksum

An error-detecting code

- must be unique
- data and links to certain location

Trusted front end (guard)

Interaction between a user, a trusted front end, and a DBMS involves the following steps:

1. Alice (the user) identifies herself to the front end (FE); the front end authenticates her identity
2. Alice issues a query to FE
3. FE verifies Alice's authorisation to data
4. Bob (the database manager) performs I/O access, interacting with low-level access control to achieve access to actual data
5. Bob returns the result of the query to the trusted FE
6. FE analyses the sensitivity levels of the data items in the result and selects those items consistent with the user's security level
7. FE transmits selected data to the untrusted FE for formatting
8. Untrusted FE transmits formatted data to the user.

Data correctness and integrity

1. correcting mistakes in data
2. using comparable data
3. eliminating false matches