## Database

- database
  - Collection of inter-related data
  - Collected and stored for some purpose

1

## File

- Programming languages traditionally handle their external data as files (some languages may have a different term for files, like data stream in Java)
- We may divide the files into
  - structured files , and
  - non-formatted files (free form, text file)
- A structured file consists of addressable data elements like records and fields.

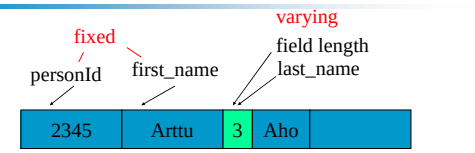
2

## Structured record

- In structured files the data elements may be accessed either using the identifier or the location of the element, for example:
  - records accessed by their relative sequence number
    - 5th record in file
  - fields accessed by their name
    - field in bytes 10-15 within record
    - field named DATE_OF_BIRTH
- in fixed format records a field is in same location in each record and the relation between element's identifier and its location is determined by the compiler of the programming language used in writing the program to construct the file.
- In varying format records to location of fields is not fixed, Fields may also vary in length.

3

## Structured record




```
<person>
  <personId>2345</personId>
  <first_name>Arttu</first_name>
  <last_name>Aho</last_name>
</person>
```

XML-format: fields may be addressed by their name

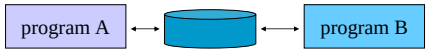
4

## How databases differ from files

- Our definition of database was very general
- Term database may be used in the general meaning but usually there are some additional expectations
  - structured data
  - data independency
  - parallel use
  - good security and privacy
  - transaction integrity
  - data integrity
  - efficient and versatile search facilities

5

## Database – data independency




- When a file is shared by many programs each of the programs must re-define the file
  - How do we guarantee that all programs defile the file in the same way – if they don't the system fails
    - If the programs are written in the same programming language we may include the same separate piece of code in each of them
  - If the programs must be written in different languages
    - Sharing of files may not be possible.
- Databases usually isolate the description of data into a database schema that is separate from programs -> the structure of data is independent of programs

6

## Database – data independency

- When some program needs to append new fields into records
  - When traditional files are used each program that uses the file must be changed and re-compiled and the file must be re-structured to reflect the change.
- Databases may provide the programs independency of data:
  - A program must be recompiled only when such structured are changed that it really uses
  - There may be different simultaneous views of the data – each program may have its own view

7

## Parallel use

- The use of files is normally exclusionary.
  - Only one user at a time
- Databases allow many simultaneous users
  - Users are hidden from each other
  - Minimal interference

8

## Search capabilities

- File:
  - Sequential processing = access the records in the order that they reside in the file
  - Direct access by address (for example by the sequence number) may be possible
- Database
  - Get data that satisfies the gives search criteria
    - Employee records where the value in salary field exceeds 2000
  - Search based on information contents not on the location of data

9

## Protection of data

- Operating system protects files and folders
  - Protection is coarse
  - read, write and execution privileges for the entire files
  - In UNIX owner, one group or anybody may get the privilege
- Databases provide focused protection
  - even on record and field basis (access to salary fields is restricted to the manager of human resources dept.)
  - Role based protection (teachers, students, …)
  - Each operation may be logged

10

## Transaction processing and recovery

- Files
  - User is responsible on backups.
  - Programmers are responsible on handling error situations, completing operations and providing facilities for canceling them
- Databases
  - Guarantee that the changes made by a transaction are persistent
  - Support canceling operations
  - Provide automatic recovery in some situations

11

## Large quantities of data

- The amount of data tends to increase
- They must anyhow be accesses efficiently. This may assume re-organization of the structures of data
- Databases typically support many levels of abstraction and the changes needed for tuning the database are not visible for programs. When dealing with files the level of abstraction is lower.
- Databases provide efficient access methods for data
  - When the database gets bigger the access time should not increase accordingly.

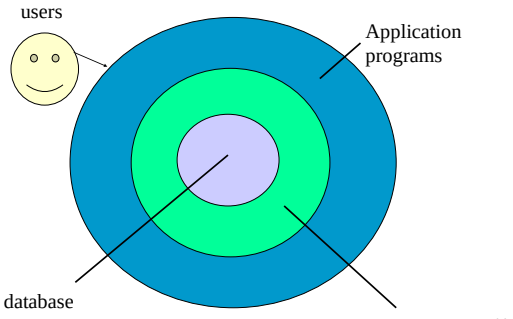
12

## Data integrity

- A database portrays some phenomena in the real world
- There are rules and regulations in the real world (for example a person may have only one spouse at a time). The database must also respect these rules and regulations.
  - If the rules are checked in application programs, they must be checked in each program that modifies the database.
- Databases may provide a separate facility for checking the rules and retaining integrity.

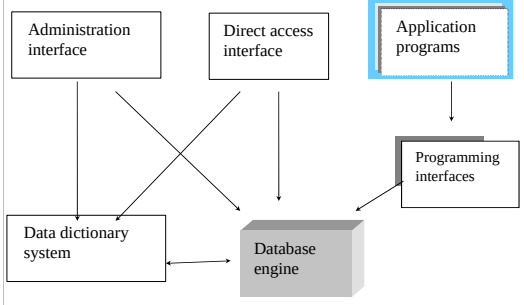
13

## Database system



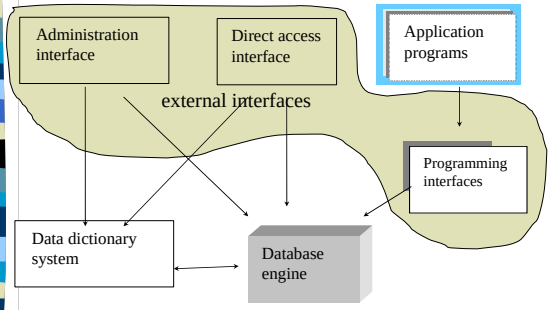

14

## Parts of a database managements system



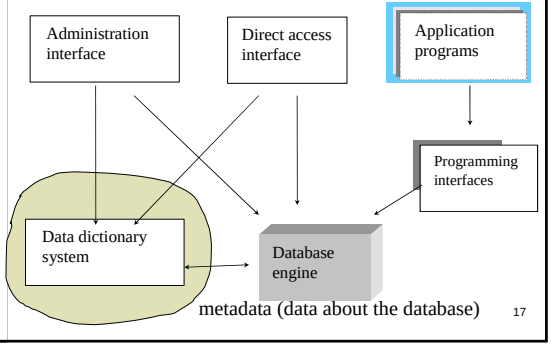

15

## Parts of a database managements system




16

## Parts of a database managements system

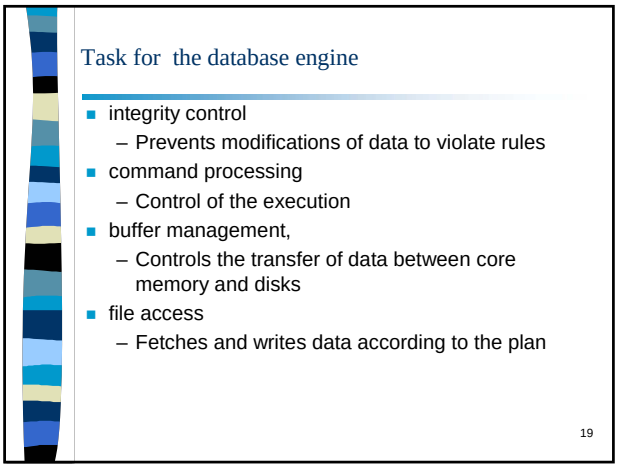



17

## Task for the database engine

- authorization control
  - Checks that the user is entitled to carry out the operation
- query optimizing
  - Queries are expressed in a high level language. The optimizer decides how the query is actually executed.
- transaction management
  - Control of simultaneous operations
  - Allocation and release of resources,
  - Finalizing operations and takes care of cancels.

18


Task for the database engine
integrity control
– Prevents modifications of data to violate rules
command processing
– Control of the execution
buffer management,
– Controls the transfer of data between core memory and disks
file access
– Fetches and writes data according to the plan
19