data
distributed
database
DBMS
parallel
shared
architecture
processing
network
system
systems
DBMSs
Distributed
DDBMS
access
processors
performance
memory
centralized
multiple
sites
site
single
disk
computer
applications
application
technology
nothing
processor
may
fragments
typically
major
require
physically
within
users
machines
computers
control
scalability
approach
improve
distinction
logical
Database
like
Parallel
also
make
provide
replicated
discuss
many
reliability
eliminates
fact
across
communication
distribution
sometimes
computing
one
split
order
scalable
architectures
referred
operational
decentralized
directly
com
Date
disks
resources
islands
large
remote
defined
controlled
based
local
often
Management
number
dis
user
makes
per
logically
stored
way
global
provides
incompatible
structure
collection
associated
consists
partitioned
used
least
must
System
bottleneck
interconnection
administered
organizational
mode
Figure
capable
nodes
definition
development
Shared
databases
share
management
transparent

Database System Concept (CSE 3213)

Lecture 01-Day 02

Nazmus Sakib, Assistant Professor, Dept. of CSE, AUST.

University Database Example

- Application program examples
  - Add new students, instructors, and courses
  - Register students for courses, and generate class rosters
  - Assign grades to students, compute grade point averages (GPA) and generate transcripts
- In the early days, database applications were built directly on top of file systems

Levels of Abstraction

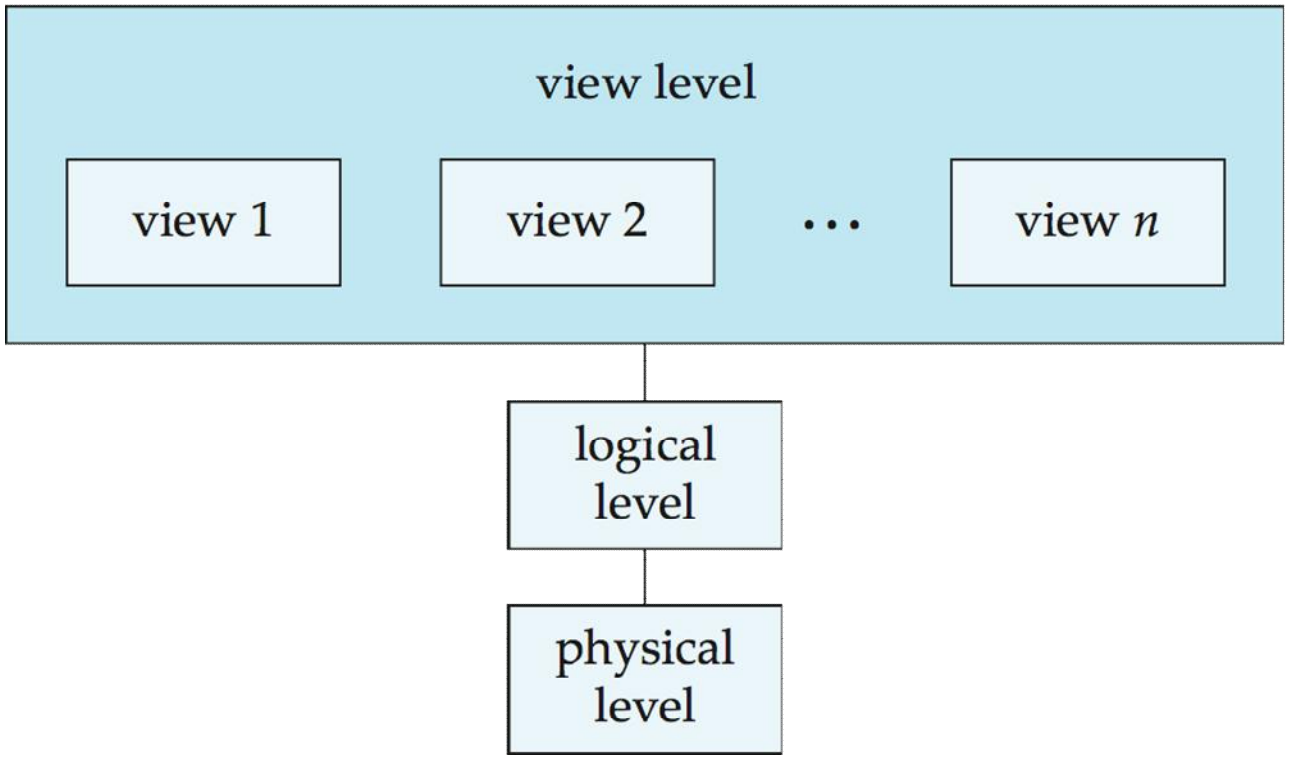
- **Physical level:** describes how a record (e.g., instructor) is stored.
- **Logical level:** describes data stored in database, and the relationships among the data.

```
type instructor = record
        ID : string;
        name : string;
        dept_name : string;
        salary : integer;
        end;
```

- **View level:** application programs hide details of data types. Views can also hide information (such as an employee's salary) for security purposes.

View of Data

An architecture for a database system



Instances and Schemas

- **Logical Schema** – the overall logical structure of the database
  - Example: The database consists of information about a set of customers and accounts in a bank and the relationship between them
    - Analogous to type information of a variable in a program
- **Physical schema**– the overall physical structure of the database
- **Schema**– describe the overall design of the database
- **Sub Schema**– describe the different views of the database
- **Instance** – the actual content of the database at a particular point in time.
  - Analogous to the value of a variable
- **Physical Data Independence** – the ability to modify the physical schema without changing the logical schema
  - Applications depend on the logical schema
  - In general, the interfaces between the various levels and components should be well defined so that changes in some parts do not seriously influence others.
- **Logical Data Independence** – the ability to modify the logical schema without changing the View Level

Database Users & DBA Activities

- **Database User Classification**
  - **Application Programmer**
  - **Sophisticated User**
  - **Specialized User**
  - **Naïve User**
- **Functions of DBA**
  - **Storage Structure**
  - **Schema Defination**
  - **Authorization**

Data Models

- Relational model
- Entity-Relationship data model (mainly for database design)
- Object-based data models (Object-oriented and Object-relational)
- Semi-Structured data model (XML)
- Other older models:
  - Network model
  - Hierarchical model

Data Definition Language (DDL)

- Specification notation for defining the database schema

  Example: **create table** *instructor* (
  ID **char**(5),
  name **varchar**(20)**,**
  dept_name **varchar**(20)*,*
  salary **numeric**(8,2))

- DDL compiler generates a set of table templates stored in a ***data dictionary***
- Data dictionary contains metadata (i.e., data about data)
  - Database schema
  - Integrity constraints
    - Primary key (ID uniquely identifies instructors)
  - Authorization
    - Who can access what

Data Manipulation Language (DML)

- Language for accessing and manipulating the data organized by the appropriate data model
  - DML also known as query language (Procedural &Declarative)
- Two classes of languages
  - **Pure** – used for proving properties about computational power and for optimization
    - Relational Algebra
    - Tuple relational calculus
    - Domain relational calculus
  - **Commercial** – used in commercial systems
    - SQL is the most widely used commercial language


data
distributed
database
DBMS
parallel
shared
system
architecture
processing
network
DBMSs
Distributed
systems
processors
access
centralized
performance
multiple
memory
DDBMS
sites
site
single
nothing
applications
computers
computer
disk
stored
technology
processor
fragments
may
require
physically
Parallel
sometimes
number
provide
local
large
one
decentralized
logically
Shared
application
bottleneck
Figure
interconnection
organizational
management
transparent
development
definition
nodes
logical
distinction
control
scalability
approach
improve
Database
within
users
machines
like
typically
major
across
fact
eliminates
reliability
communication
distribution
also
processor
replicated
make
discuss
many
often
sites
user
dis
makes
per
Management
consists
administered
mode
capable
partitioned
least
used
global
way
associated
incompatible
structure
collection
provides
must
System
directly
com
Date
based
controlled
defined
remote
islands
resources
disks
operational
referred
computing
split
order
scalable
architectures
databases
share
typically