Database Systems

Concepts and Architecture

Chapter 1

Contents

Contents

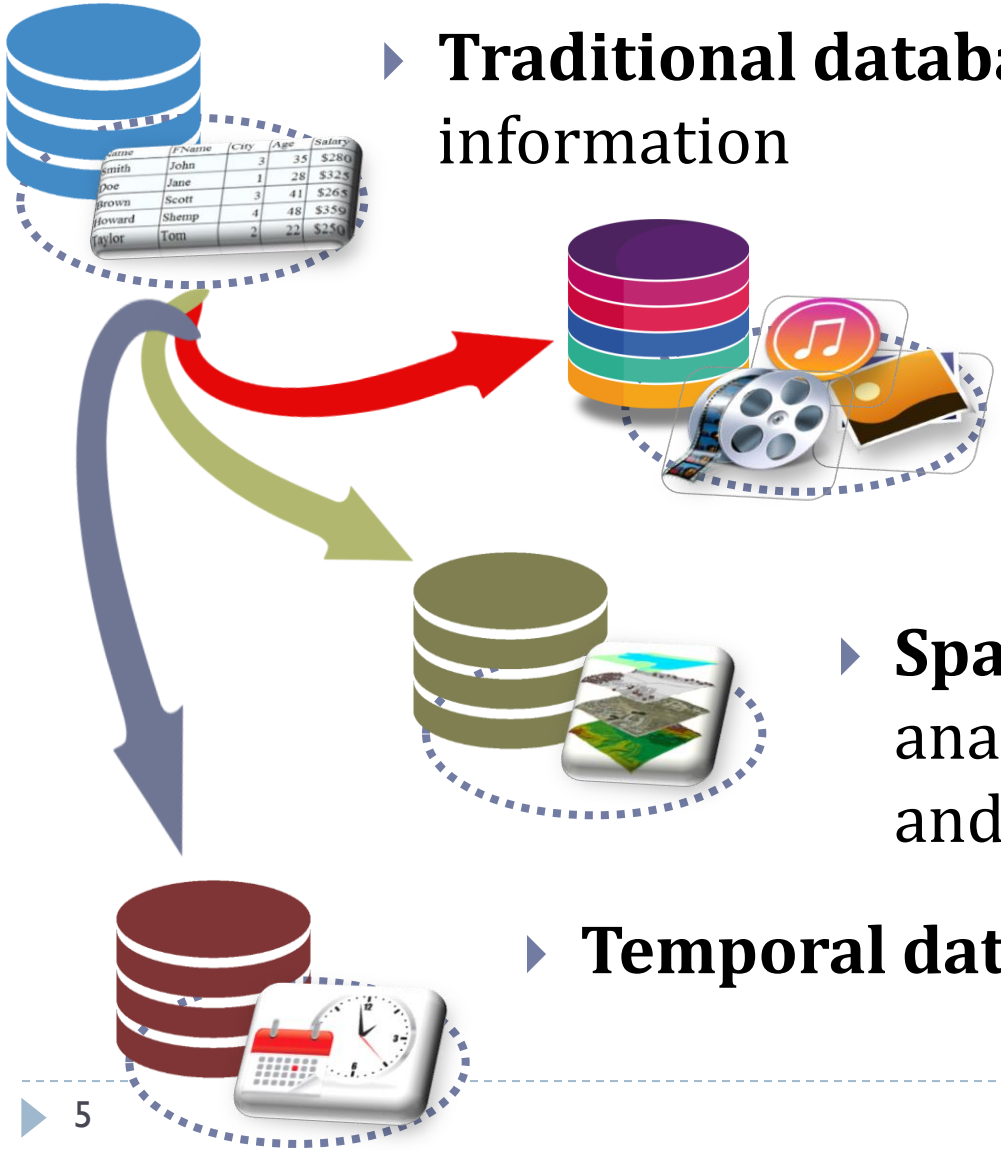
Introduction

- Collection of *related data* with an implicit meaning
- Represents some aspect of the real world
- Designed, built, and populated with data for a *specific purpose*.

- **UNIVERSITY** database
  - Information concerning students, courses, and grades in a university environment

→ Data records: STUDENT, COURSE, SECTION, GRADE_REPORT, LECTURER

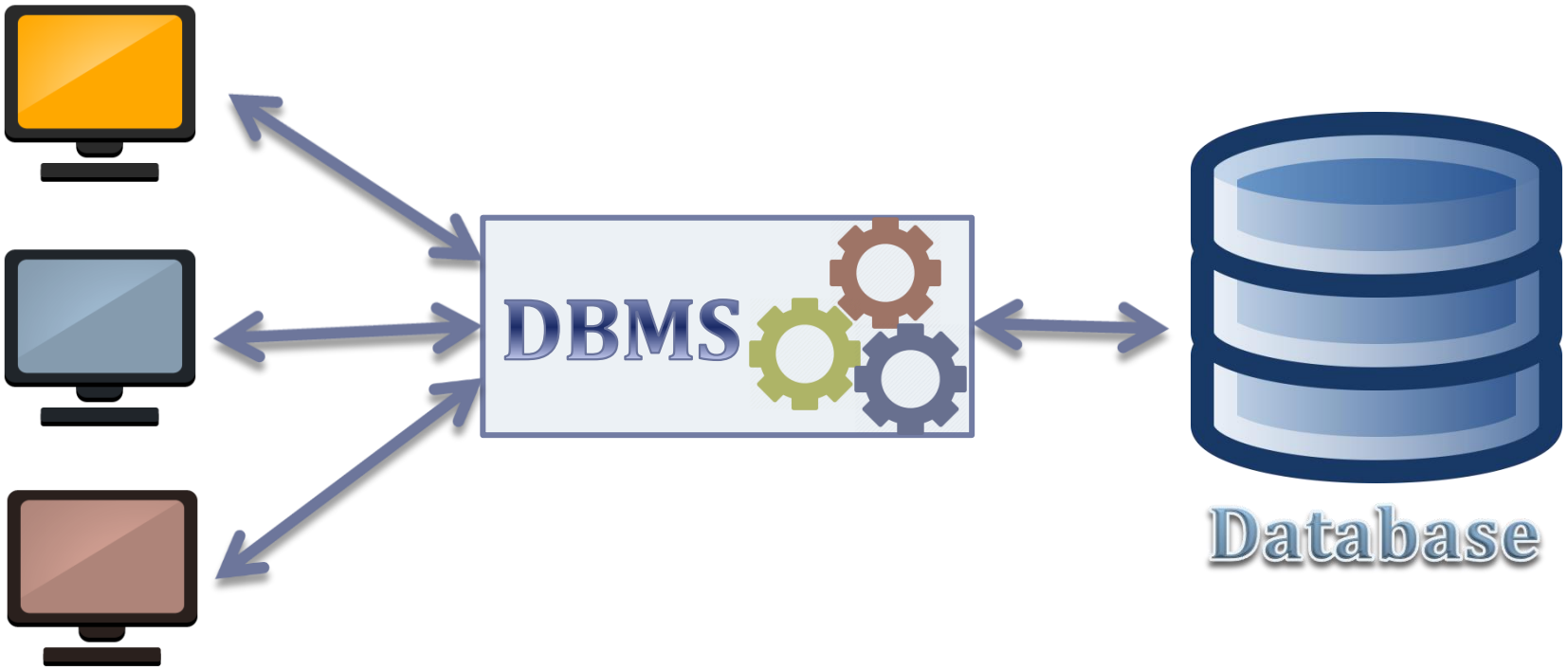
Introduction



- **Traditional database:** textual or numeric information
- **Multimedia database:** images, audio clips, and video streams
- **Spatial database:** Store and analyze maps, weather data, and satellite images
- **Temporal database:** historical data

Introduction

- **Database management system (DBMS)**
  - Collection of programs to create and maintain a database



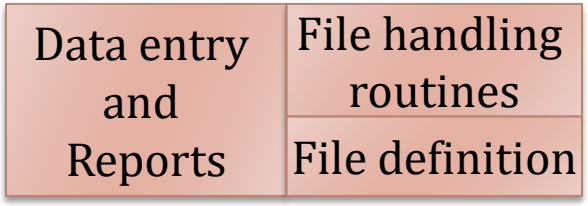
Contents

File-based Approach

- Data is stored in one or more separate computer files
- Data is then processed by computer programs - applications

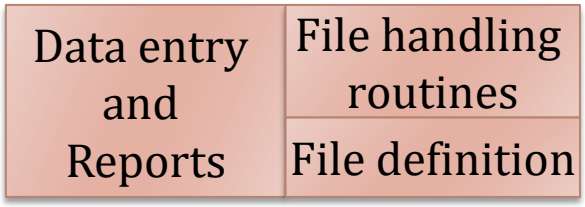
File-based Approach

Sales Department



Sales Application Programs

Contract Department



Contract Application Programs

Sales files

PrivateOwner (ownerNo, fName, lName, address, telNo)
PropertyForRent (propertyNo, street, postcode, rooms, ownerNo)
Client (clientNo, fName, lName, address, telNo, prefType, maxRent)

Lease (leaseNo, propertyNo, clientNo, deposit, paid, Start, Finish)
PropertyForRent (propertyNo, street, city, postcode, rent)
Client (clientNo, fName, lName, address, telNo)

Contract files

File-based Approach

- **Problems:**
  - Data Redundancy
    - The same information being kept in several different places (files)
    - Wastes storage space and duplicates effort
  - Data Inconsistency
    - Various copies of the same data are conflicting
    - Inconsistency in data format

File-based Approach

Sales Department

Data entry and Reports

File handling routines

File definition

Sales Application Programs

PropertyForRent File

PrivateOwner File

Client File

Contract Department

Data entry and Reports

File handling routines

File definition

Contract Application Programs

Lease File

PropertyForRent File

Client File

Shared File Approach

Sales Department

Contract Department

Data entry and Reports

File handling routines

File definition

Sales Application Programs

Data entry and Reports

File handling routines

File definition

Contract Application Programs

PrivateOwner File

PropertyForRent File

Client File

Lease File

File-based Approach

- **Shared File Approach**
  - Data (files) is shared between different applications
  - Data redundancy problem is alleviated
  - Data inconsistency problem across different versions of the same file is solved

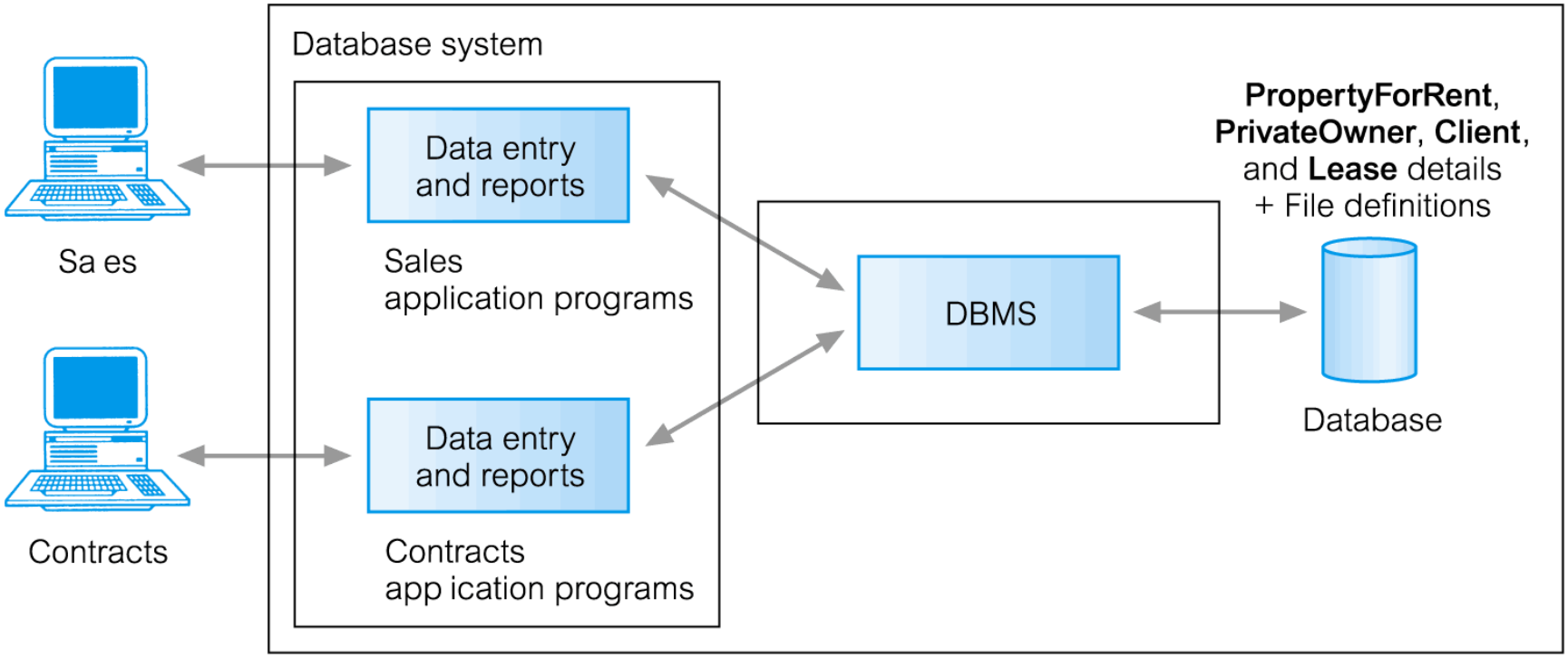
File-based Approach

- **Shared File Approach**
  - Other problems:
    - Rigid data structure: If applications have to share files, the file structure that suits one application might not suit another
    - Physical data dependency: If the structure of the data file needs to be changed in some way, this alteration will need to be reflected in all application programs that use that data file
    - No support of concurrency control: While a data file is being processed by one application, the file will not be available for other applications or for ad hoc queries

Database Approach

- Arose because:
  - Definition of data was embedded in application programs, rather than being stored separately and independently
  - No control over access and manipulation of data beyond that imposed by application programs
- Result:
  - The Database and Database Management System (DBMS).

Database Approach



PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)
PrivateOwner (ownerNo, fName, lName, address, telNo)
Client (clientNo, fName, lName, address, telNo, prefType, maxRent)
Lease (leaseNo, propertyNo, clientNo, paymentMethod, deposit, paid, rentStart, rentF nish)

Database Approach

- **Data**
  - Known facts that can be recorded and that have implicit meaning
  - Information? Knowledge? Wisdom?
  - More: www.whatis.com
- **Database**: Shared collection of logically related data and a description of this data, designed to meet the information needs of an organization

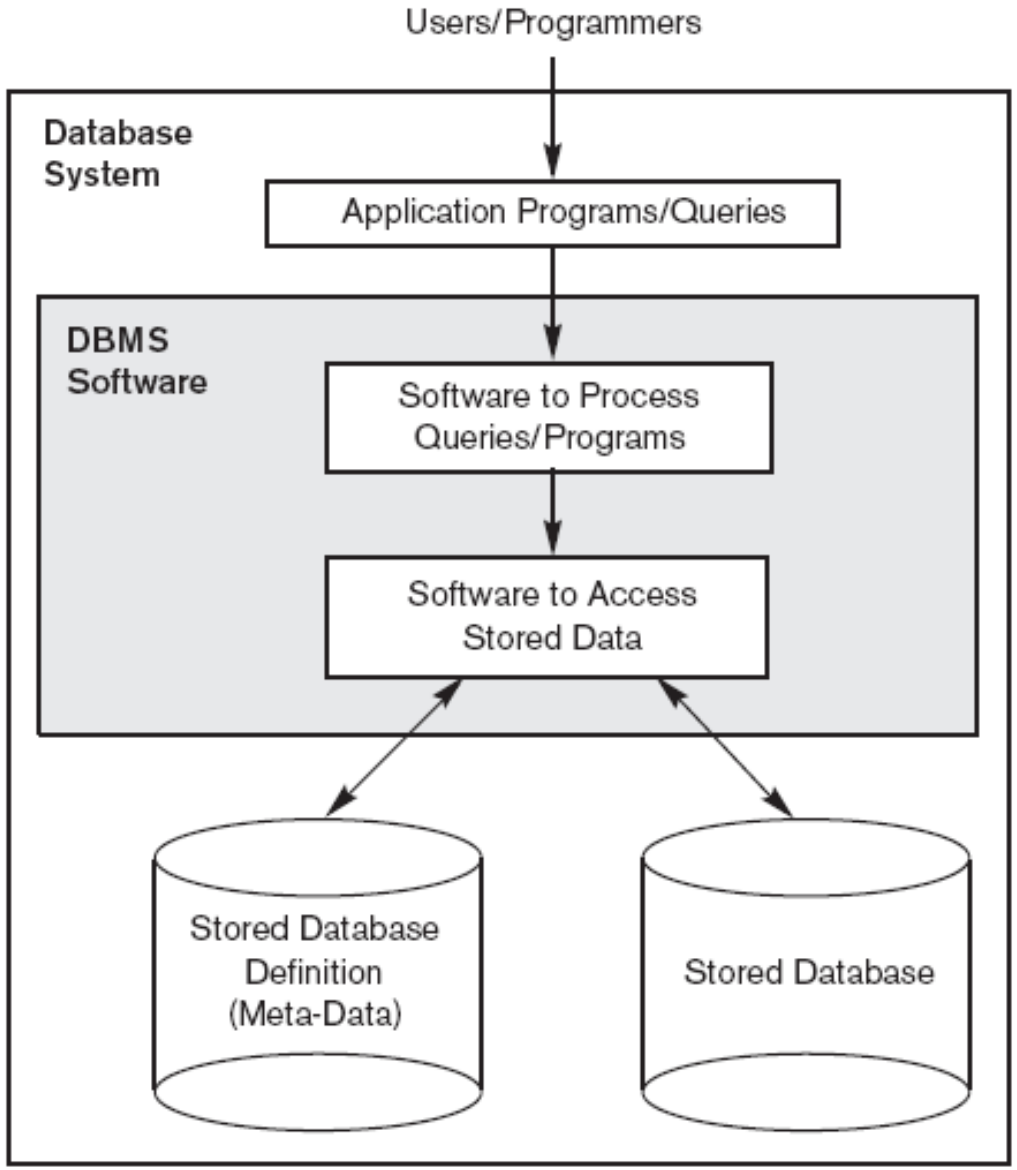
Database Approach

- System catalog (metadata) provides description of data to enable program–data independence
- Logically related data comprises entities, attributes, and relationships of an organization's information
- **D**ata**B**ase **M**anagement **S**ystem (DBMS): a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications

Database Approach

- Database System = Database + DBMS software
- Database approach allows user:
  - Specify data types, structures and any data constraints to be stored in the database. All specifications are stored in the database
  - Query data: retrieve (query), update (insert, delete, modify)
  - Control access to database:
    - a security system
    - an integrity system
    - a concurrency control system
    - a recovery control system
    - a user-accessible catalog

A Simplified Database System Environment



Database Approach

- **Roles in the Database Environment**
  - Database Administrator (DBA)
  - Database Designers
  - Application Programmers
  - End Users

Database Approach

- **Database administrators (DBA)** are responsible for:
  - Authorizing access to the database
  - Coordinating and monitoring its use
  - Acquiring software and hardware resources
- **Database designers** are responsible for:
  - Identifying the data to be stored
  - Choosing appropriate structures to represent and store this data

Database Approach

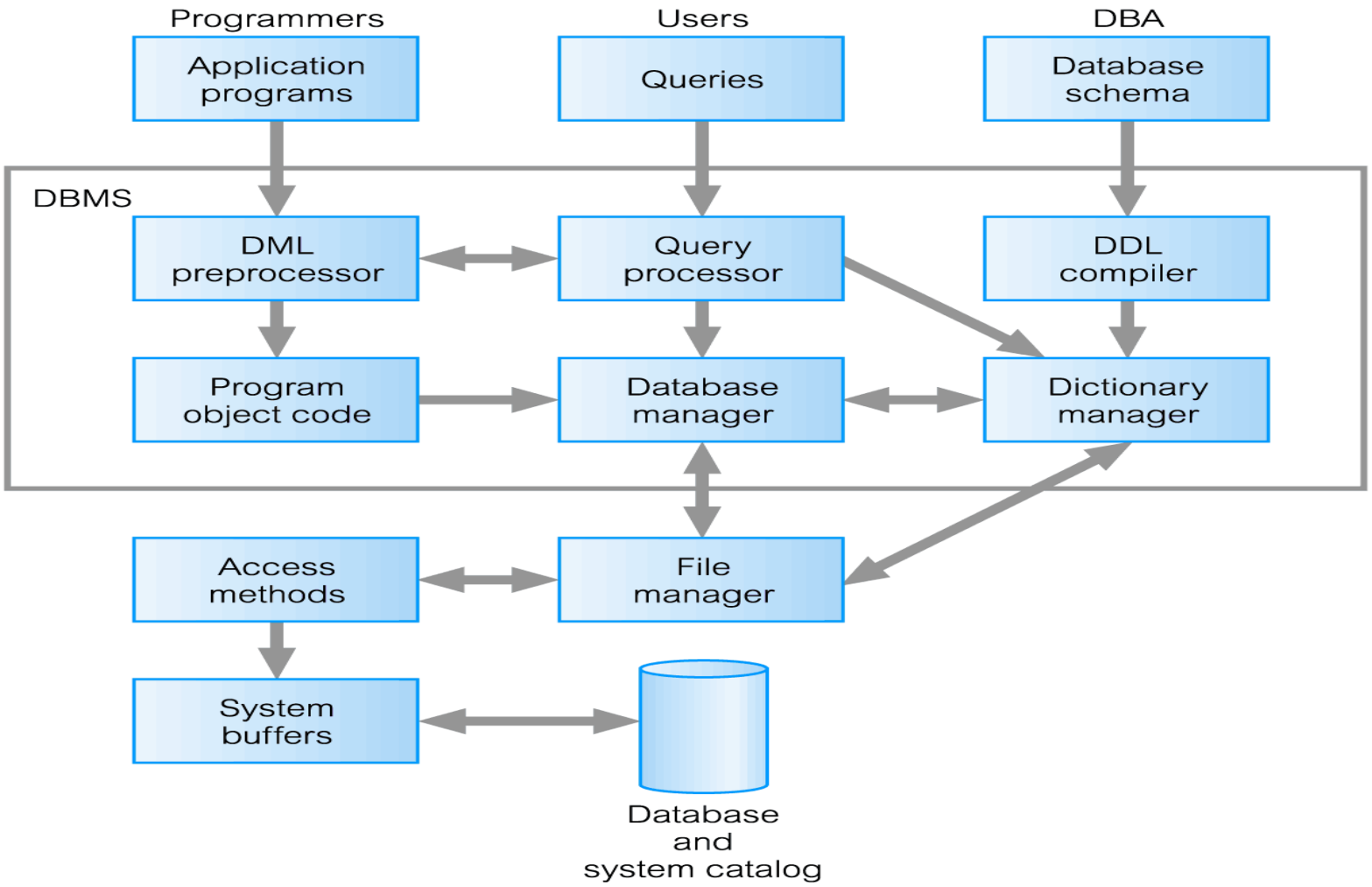
- **Application programmers**
  - Implement these specifications as programs
- **End users**
  - People whose jobs require access to the database

Database Approach

- **Workers behind the Scene**
  - **DBMS system designers and implementers**
    - Design and implement the DBMS modules and interfaces as a software package
  - **Tool developers**
    - Design and implement **tools**
  - **Operators and maintenance personnel**
    - Responsible for running and maintenance of hardware and software environment for database system

Database Approach

- DBMS components:



Database Approach

- Characteristics of the Database Approach
  - Self-describing nature of a database system
  - Insulation between programs and data, and data abstraction
    - Program-data independence + Program-operation independence = Data abstraction
    - A data model is a type of data abstraction
  - Support of multiple views of the data
  - Sharing of data and multi-user transaction processing

When Not to Use a DBMS

- More desirable to use regular files for:
  - Simple, well-defined database applications not expected to change at all
  - Stringent, real-time requirements that may not be met because of DBMS overhead
  - Embedded systems with limited storage capacity
  - No multiple-user access to data

Database Approach

- History of database systems
  - First generation: Hierarchical and Network
  - Second generation: Relational
  - Third generation: Object-Relational, Object-Oriented

Example of Network Model Schema

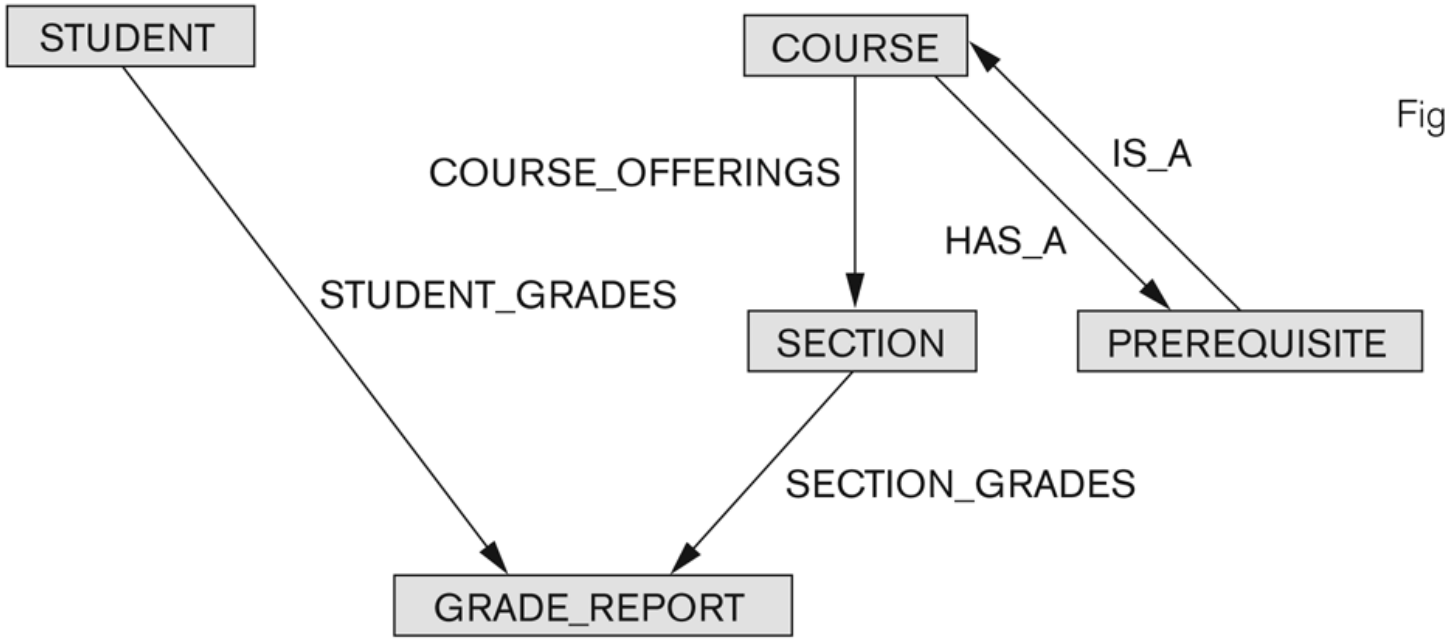


Figure 2.8
The schema of Figure 2.1 in network model notation.

Example of Relational Model Schema

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	04	King
92	CS1310	Fall	04	Anderson
102	CS3320	Spring	05	Knuth
112	MATH2410	Fall	05	Chang
119	CS1310	Fall	05	Anderson
135	CS3380	Fall	05	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

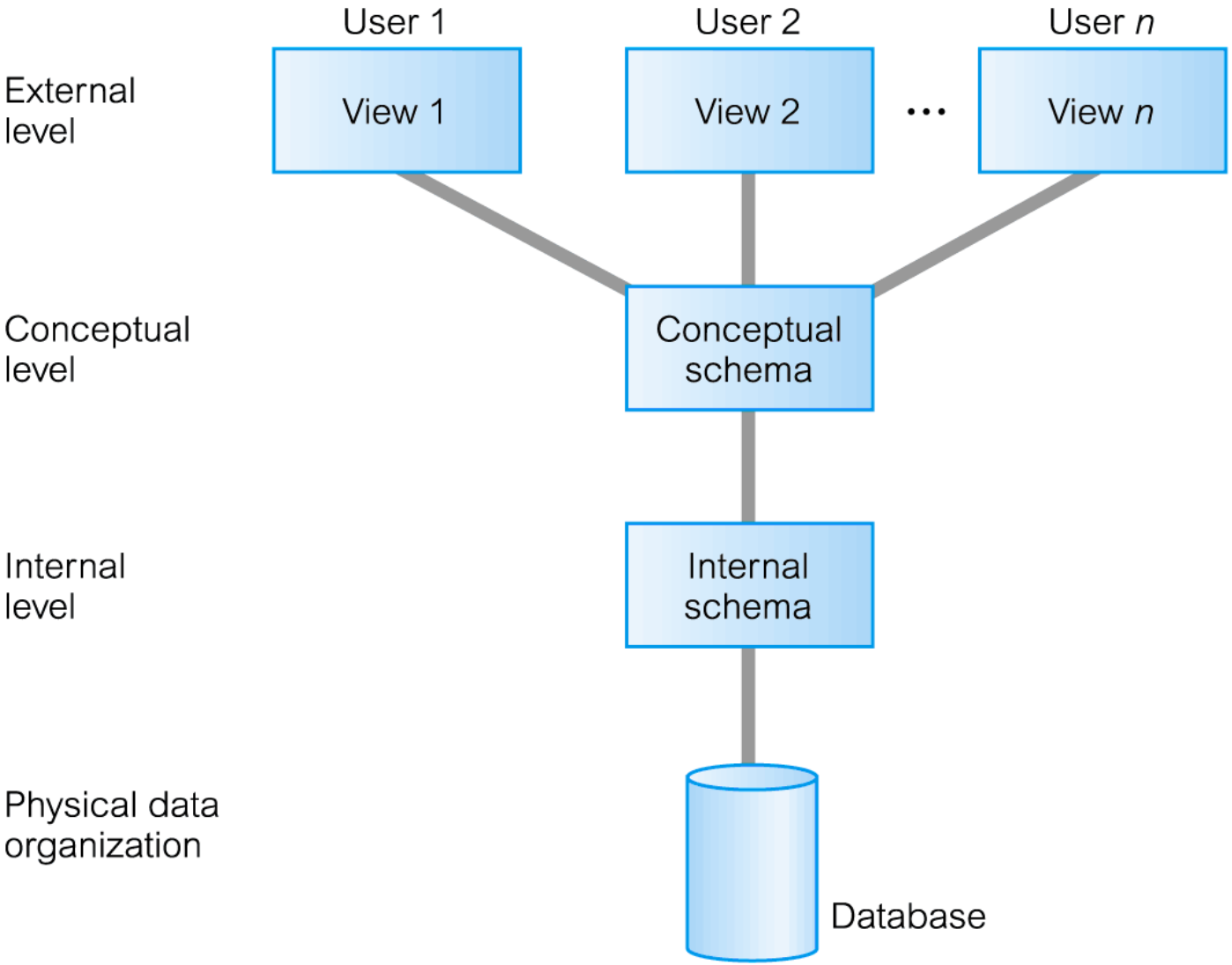
Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Contents

Three-Schema Architecture and Data Independence

- Objectives of Three-Schema Architecture
  - All users should be able to access same data
  - Users should not need to know physical database storage details
  - DBA should be able to change database storage structures without affecting the users' views
  - Internal structure of database should be unaffected by changes to physical aspects of storage
  - DBA should be able to change conceptual structure of database without affecting all users

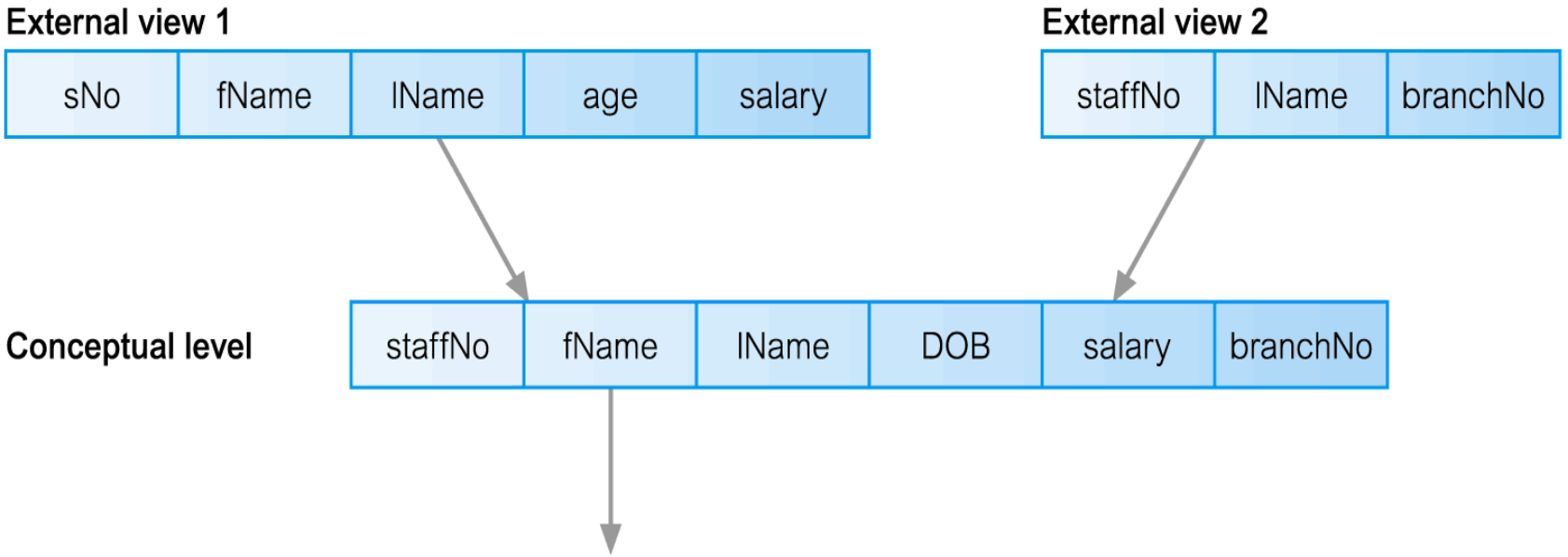
Three-Schema Architecture and Data Independence



Three-Schema Architecture and Data Independence

- **External Level**
  - Users' view of the database
  - Describes that part of database that is relevant to a particular user
- **Conceptual Level**
  - Community view of the database
  - Describes what data is stored in database and relationships among the data
- **Internal Level**
  - Physical representation of the database on the computer.
  - Describes how the data is stored in the database

Three-Schema Architecture and Data Independence



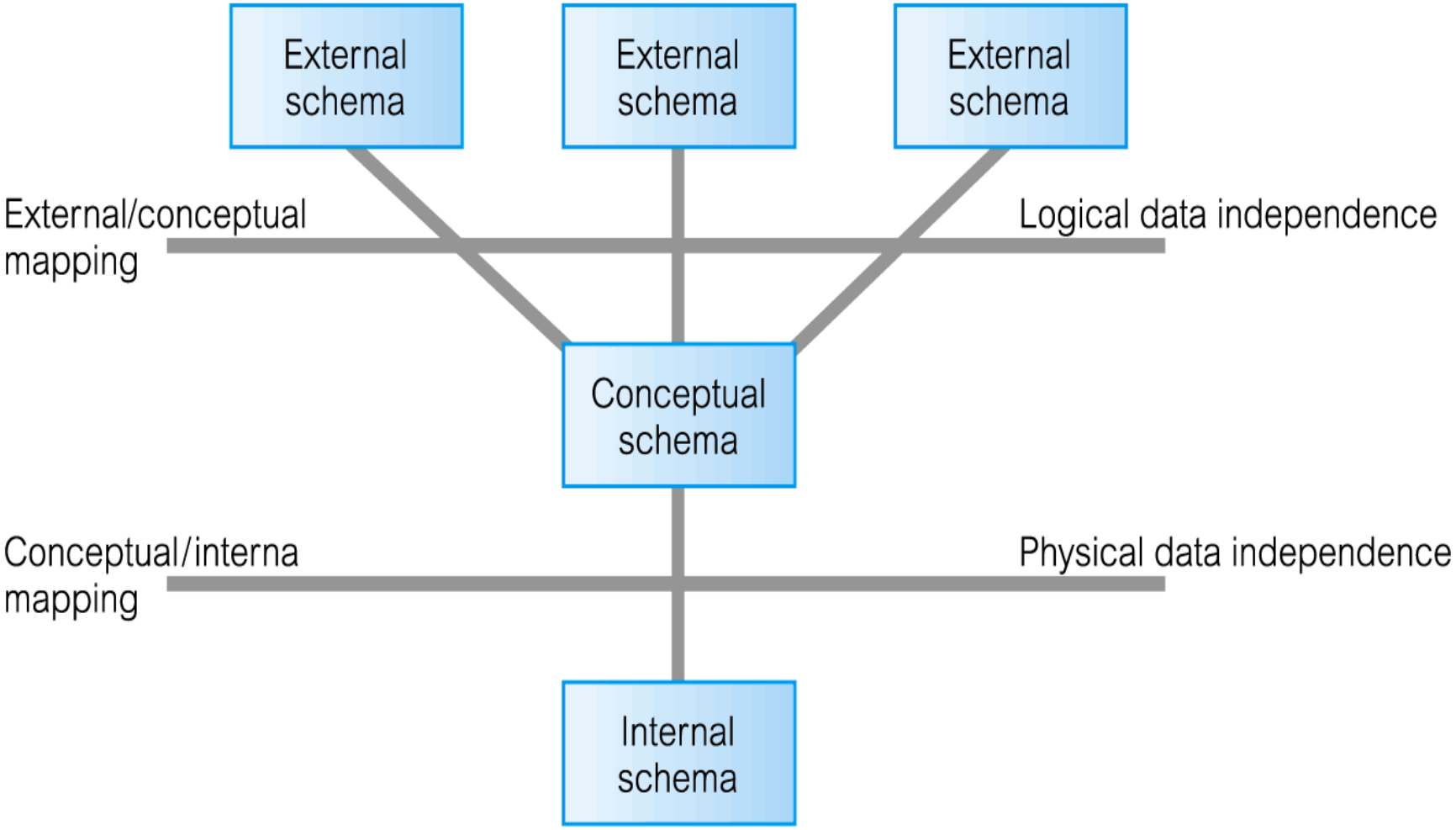
Internal level

```
struct STAFF {
      int staffNo;
      int branchNo;
      char fName [15];
      char lName [15];
      struct date dateOfBirth;
      float salary;
      struct STAFF *next;                    /* pointer to next Staff record */
};
index staffNo; index branchNo;               /* define indexes for staff */
```

Three-Schema Architecture and Data Independence

- **Data Independence:** is the capacity to change the schema at one level of a database system without having to change the schema at the next higher levels
- Logical Data Independence:
  - Conceptual schema changes (e.g. addition/removal of entities) should not require changes to external schema or rewrites of application programs
- Physical Data Independence:
  - Internal schema changes (e.g. using different file organizations, storage structures/devices) should not require changes to conceptual or external schemas

Three-Schema Architecture and Data Independence



Contents

Database Languages

- **Data Definition Language (DDL)** allows the DBA or user to describe and name entities, attributes, and relationships required for the application plus any associated integrity and security constraints
- **Data Manipulation Language (DML)** provides basic data manipulation operations (select, insert, update, delete) on data held in the database
- **Data Control Language (DCL)** defines activities that are not in the categories of those for the DDL and DML, such as granting privileges to users, and defining when proposed changes to a databases should be irrevocably made

Database Languages

- Procedural DML allows user to tell system exactly how to manipulate data (e.g., Network and hierarchical DMLs)
- Non-Procedural DML (declarative language) allows user to state what data is needed rather than how it is to be retrieved (e.g., **SQL**, QBE)
- Fourth Generation Languages (4GLs)
  - Non-procedural languages: SQL, QBE, etc.
  - Application generators, report generators, etc. (see [2])

Contents

Data Models, Database Schema and Database State

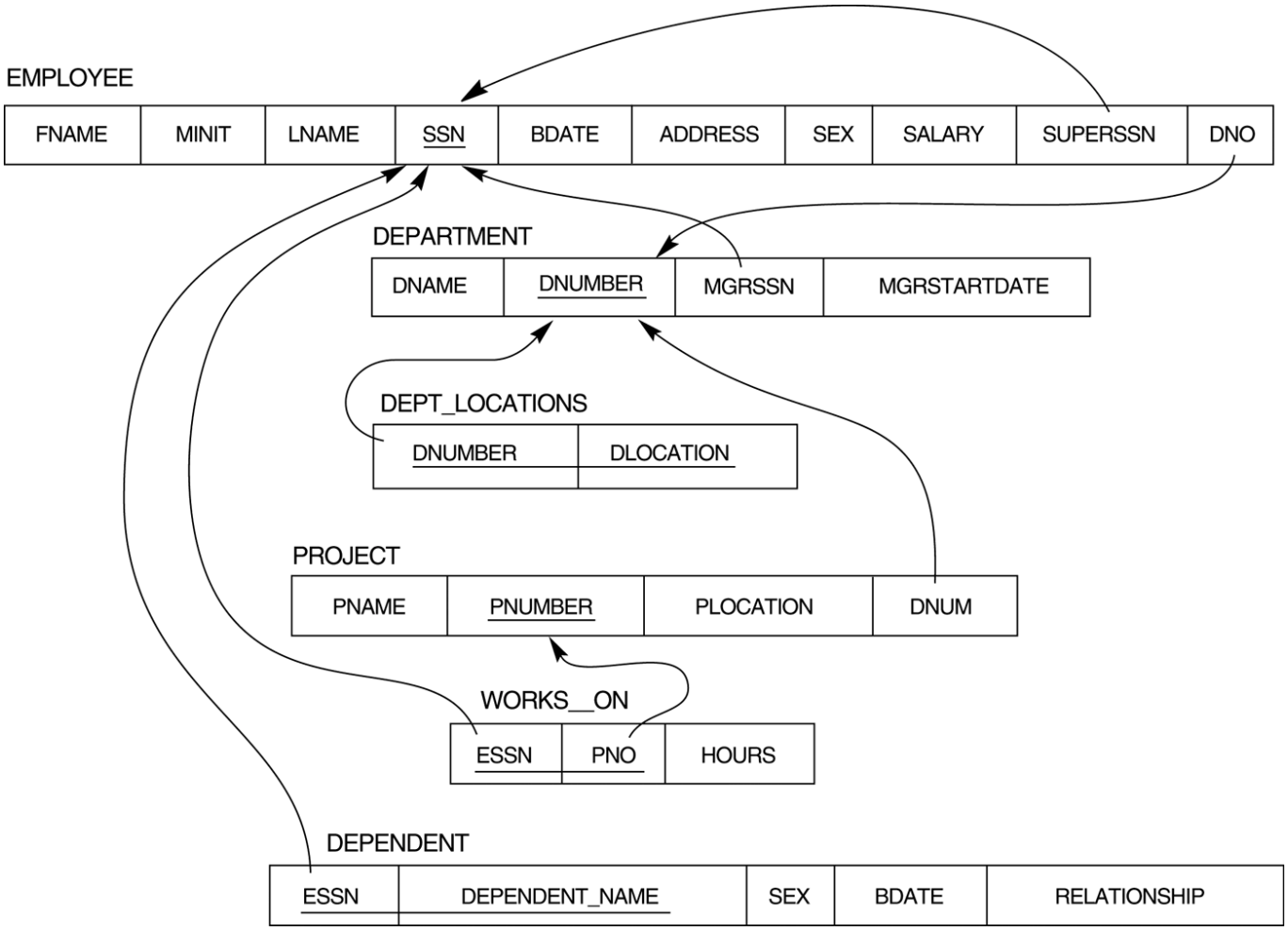
- Data Model: An integrated collection of concepts for describing data, relationships between data, and constraints on the data in an organization
- Categories of data models include:
  - Object-based (Conceptual)
    - ER, Object-Oriented, …
  - Record-based (Representational)
    - **Relational**, Network, Hierarchical
  - Physical: used to describe data at the internal level

Describe data at the conceptual & external levels

Data Models, Database Schema and Database State

- Database Schema: the description of a database, which is specified during database design and is not expected to change frequently
- Schema Diagram: a displayed schema
- Database State (Snapshot): the data in the database at a particular moment in time

Relational Database Schema



Database state

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

Contents

Data Management Systems Framework

Application Layer	Visualization, Collaborative Computing, Mobile Computing, Knowledge-based Systems
Data Management Layer	**Layer 3: information extraction & sharing** Data Warehousing, Data Mining, Internet DBs, Collaborative, P2P & Grid Data Management
	Layer 2: interoperability & migration Heterogeneous DB Systems, Client/Server DBs, Multimedia DB Systems, Migrating Legacy DBs
	Layer 1: DB technologies **DB Systems**, Distributed DB Systems
Supporting Layer	Networking, Mass Storage, Agents, Grid Computing Infrastructure, Parallel & Distributed Processing, Distributed Object Management

Data Management Systems Framework

- Extending database capabilities for new applications
  - Example applications: storage and retrieval of images, videos, data mining (large amounts of data need to be stored and analyzed), spatial databases, time series applications, …
  - More complex data structures than relational representation
  - New data types except for the basic numeric and character string types
  - New operations and query languages for new data types
  - New storage and retrieval methods
  - New security mechanisms
  - …

Contents


Q
A