BITS Pilani
Pilani Campus

Database Systems (CS F212)- Lecture 3

Recapitulation of Lecture 2

- Problems in file processing systems, Program data dependence
- Data Abstraction: Logical data independence, Physical data dependence
- Instance and Schemas

Data Models

Data model: Shows the logical structure of a database, including the relationships and constraints that determine how data can be stored and accessed

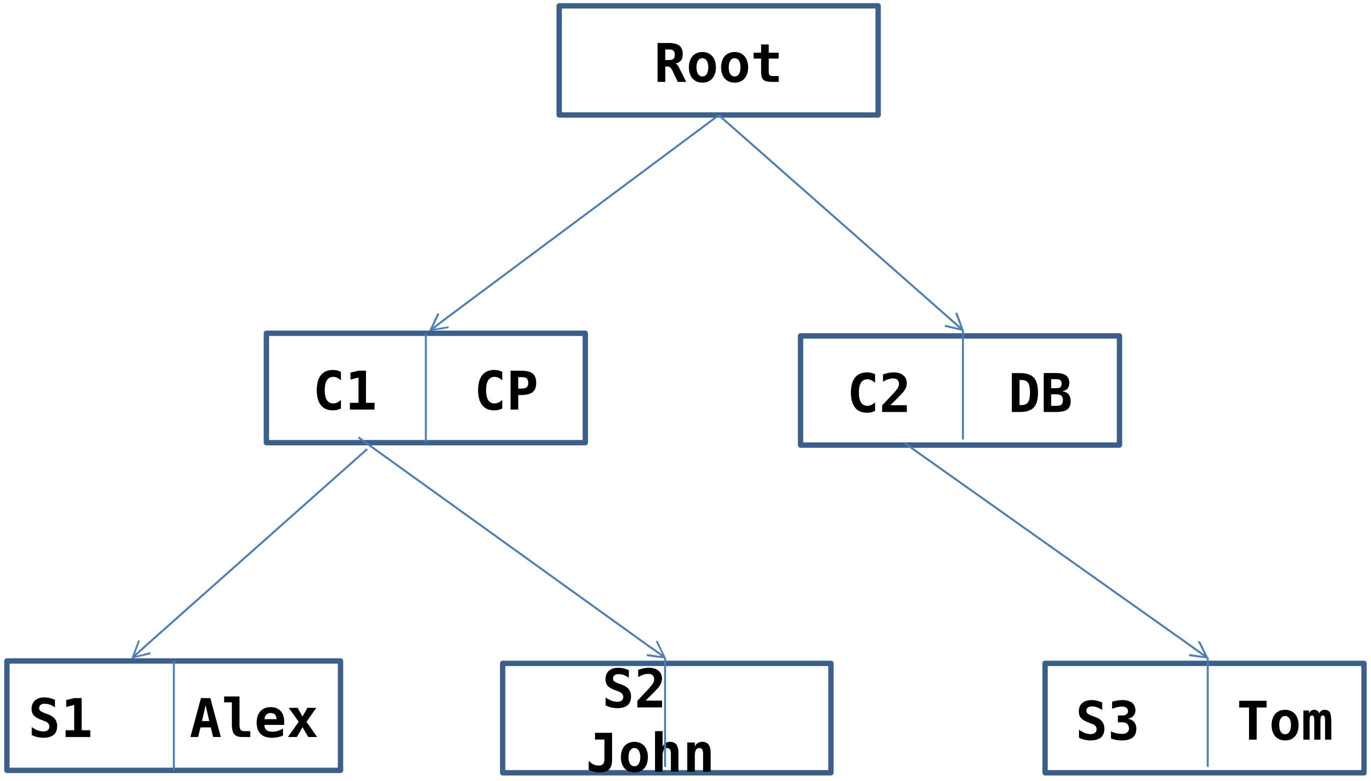
Need for data modelling:

- Accurate representation of data objects
- Helps in designing physical database
- Identify missing and redundant data.
- Cheaper and faster upgrade and maintenance of IT infrastructure

Hierarchical data model

- Data is organized into a tree like structure
- Data is stored in the form of records
- Each record can have at-most one parent record and one or many children record
- Sibling records are sorted in a particular order, which is used as the physical order for storing the database.

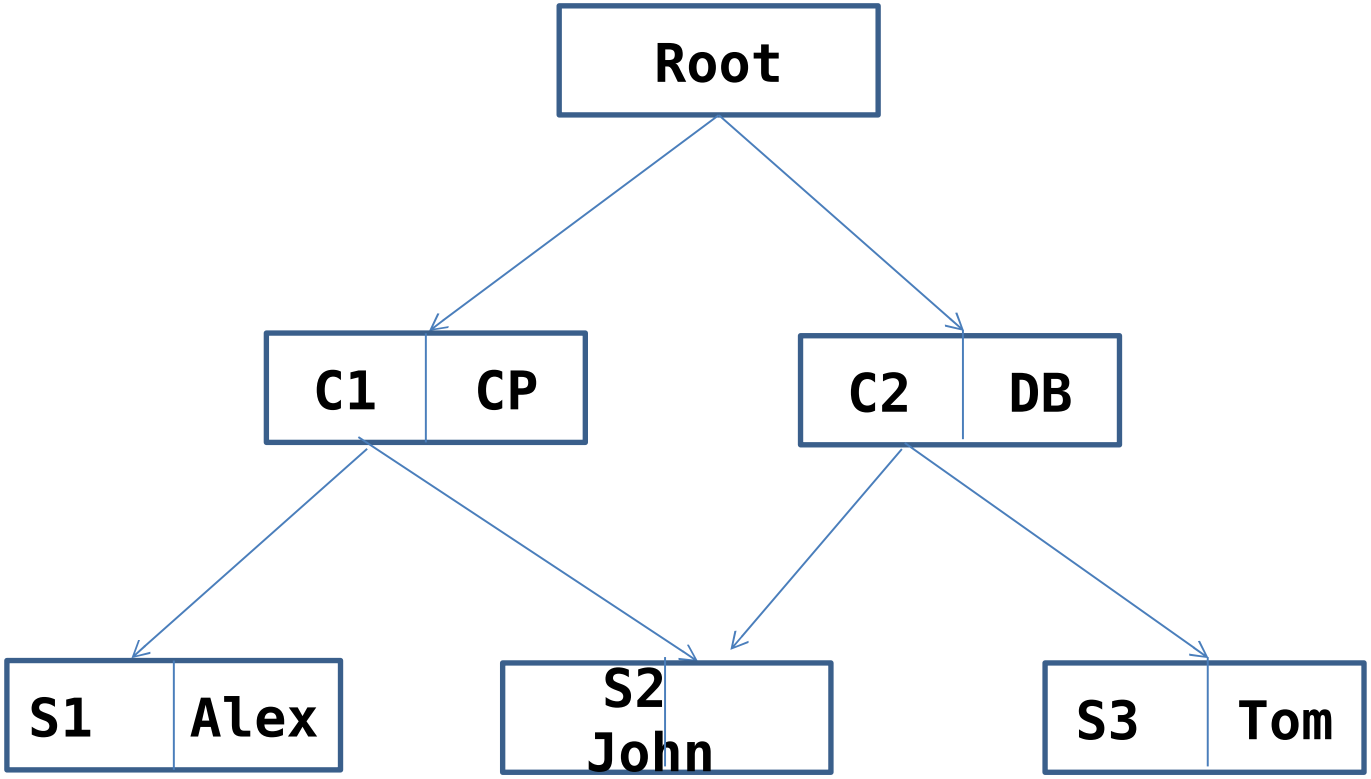
Hierarchical data model



Network data model

- Extension of the Hierarchical model
- Data is organised more like a graph, and are allowed to have more than one parent node
- A network structure thus allows 1:1, l: M, M: M relationships among entities.
- Most widely used model before relational model

Network data model



Relational data model

- Represents the database as a collection of relations in form of tables
- Every row in the table represents a collection of related data values
- These rows in the table denote a real-world entity or relationship
- Most widely used data model

Relational data model

Student

S_ID	S_Name	CID
S1	Alex	C1
S2	John	C1
S3	Tom	C2

Course

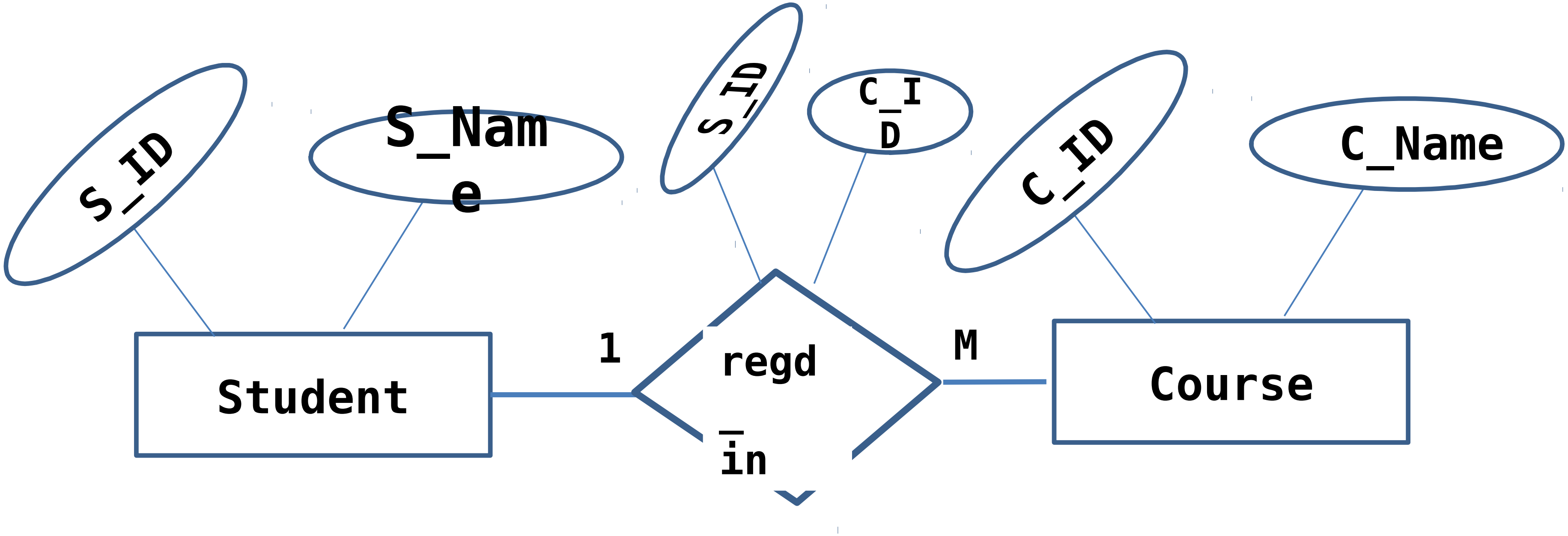
CID	C_Name
C1	CP
C2	DB

Entity-Relational Model

- Meant for high-level designs
- Represents the database as a collection of entities
- Entities are associated with a set of attributes
- Association among two entities is represented as relationship
- Relationship can have their own attributes
- These rows in the table denote a real-world entity or relationship

Entity-Relational data model

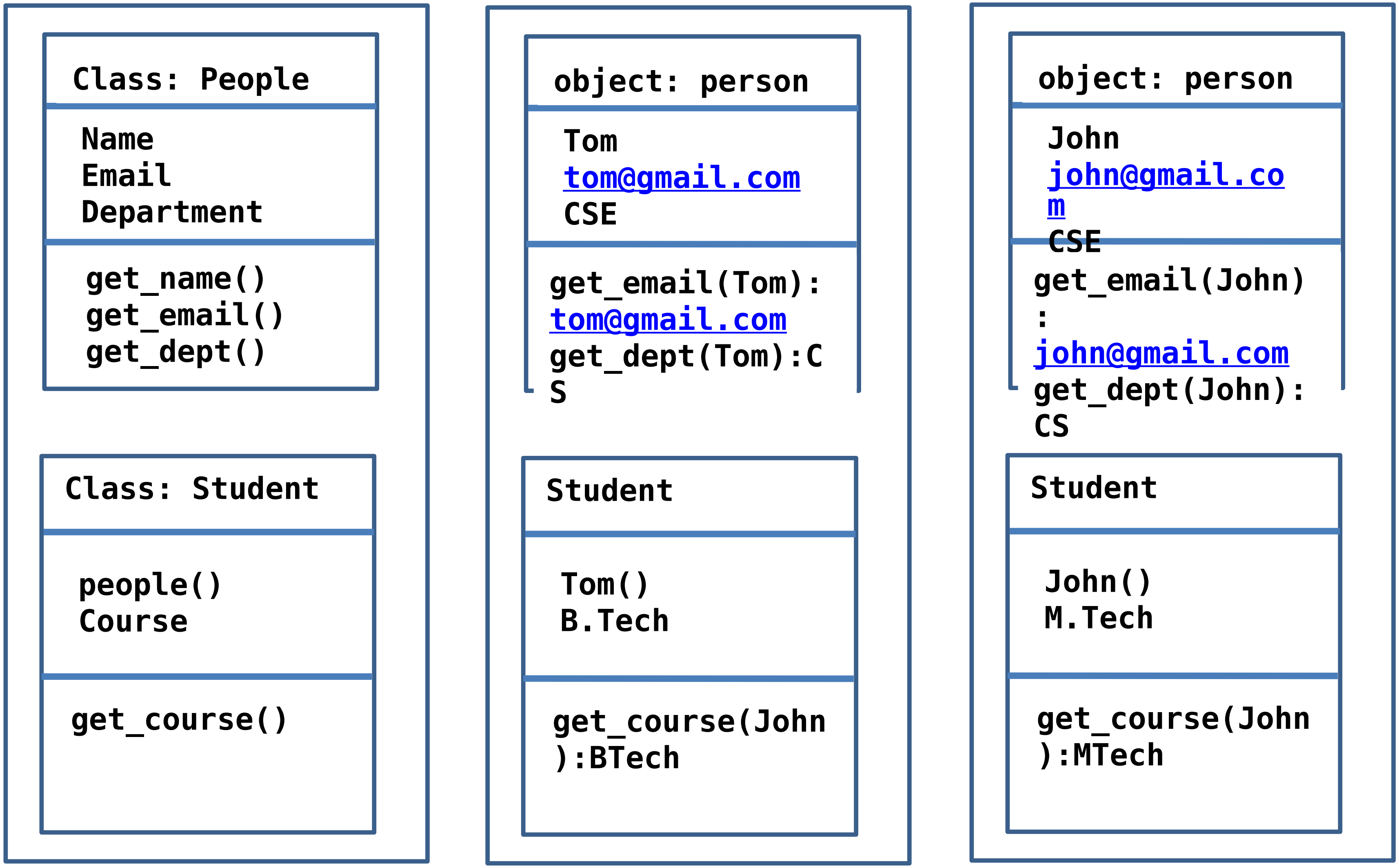
- Cardinality of entities in a association is written explicitly



Object oriented data model

- Entities are represented as objects
- Characteristics of an object are represented using Attributes.
- The behaviour of the objects is represented using Methods
- Similar attributes and methods are grouped together in a class
- A new class can be derived from the original class.

Object oriented data model



Class: People

Name
Email
Department

get_name()
get_email()
get_dept()

Class: Student

people()
Course

get_course()

object: person

Tom
tom@gmail.com
CSE

get_email(Tom):
tom@gmail.com
get_dept(Tom):C
S

Student

Tom()
B.Tech

get_course(John
):BTech

object: person

John
john@gmail.co
m
CSE

get_email(John)
:
john@gmail.com
get_dept(John):
CS

Student

John()
M.Tech

get_course(John
):MTech

Object oriented data model

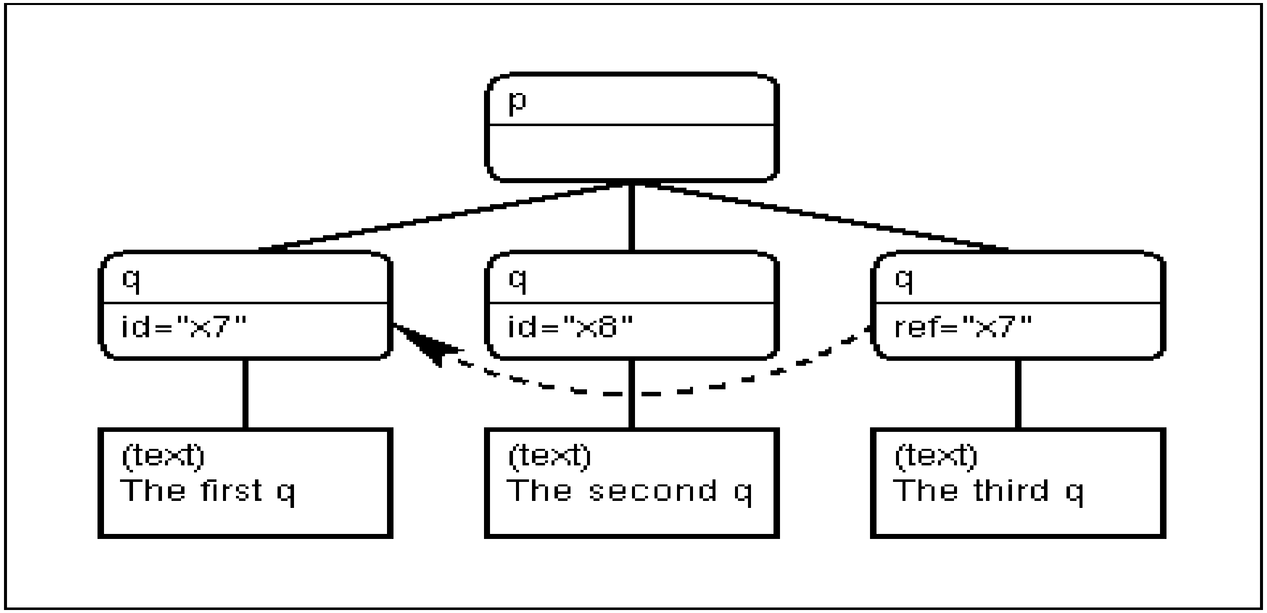
- Still in development stage
- Approach for solving the requirement, not a technology

Semi structured data model

- Data is not constrained by a fixed schema
- Individual data items of the same type may have different sets of attributes
- New technology, not very much explored
- Elements of a record can be represented using graph

Semi structured data model

```
<p>
<q id="x7">The first q</q>
<q id="x8">The second q</q>
<q href="#x7">The third q</q>
 </p>
```



Evolution of Databases

1950s and early 1960s:

- Data processing using magnetic tapes for storage
    - Tapes provided only sequential access
- Punched cards for input

Late 1960s and 1970s:

- Hard disks allowed direct access to data
- Network and hierarchical data models in widespread use
- Ted Codd defines the relational data model
    - Would win the ACM Turing Award for this work
    - IBM Research begins System R prototype
    - UC Berkeley begins Ingres prototype
- High-performance (for the era) transaction processing

Evolution of Databases

1980s:

- Research relational prototypes evolve into commercial systems
    - SQL becomes industrial standard
- Parallel and distributed database systems
- Object-oriented database systems

1990s:

- Large decision support and data-mining applications
- Large multi-terabyte data warehouses
- Emergence of Web commerce

Early 2000s:

- XML and XQuery standards
- Automated database administration

Later 2000s:

- Giant data storage systems
    - Google BigTable, Yahoo PNuts, Amazon