Database Systems
Introduction (Lecture-02)

Dr. Rudra Pratap Deb Nath

Associate Professor
Department of Computer Science & Engineering
University of Chittagong

rudra@cu.ac.bd

4th Semester 2025

Overview

Main questions to be addressed

Questions:

- How to model and design data?
- How to store data reliably and durably?
- How to process data efficiently?
- How to handle concurrent access of multiple users?

Course contents

Contents:

- Relational Model
- Relational Algebra
- Entity Relationship Model (ER model)
- Database Design (Normalization)
- (Advanced) SQL
- Transactions
- Recovery
- Query processing and optimization
- Introduction to current and advanced database technologies

Entrance to the problem solving world

Model some part of the real world (mini-world)

- What do we need to model?
- What level of details?
- Identify of entities
- Identify their roles
- Identify how they are connected to each other

Requirement gathering and analysis

Mini world: CSE Dept $\rightarrow$ requirement analysis
Some facts

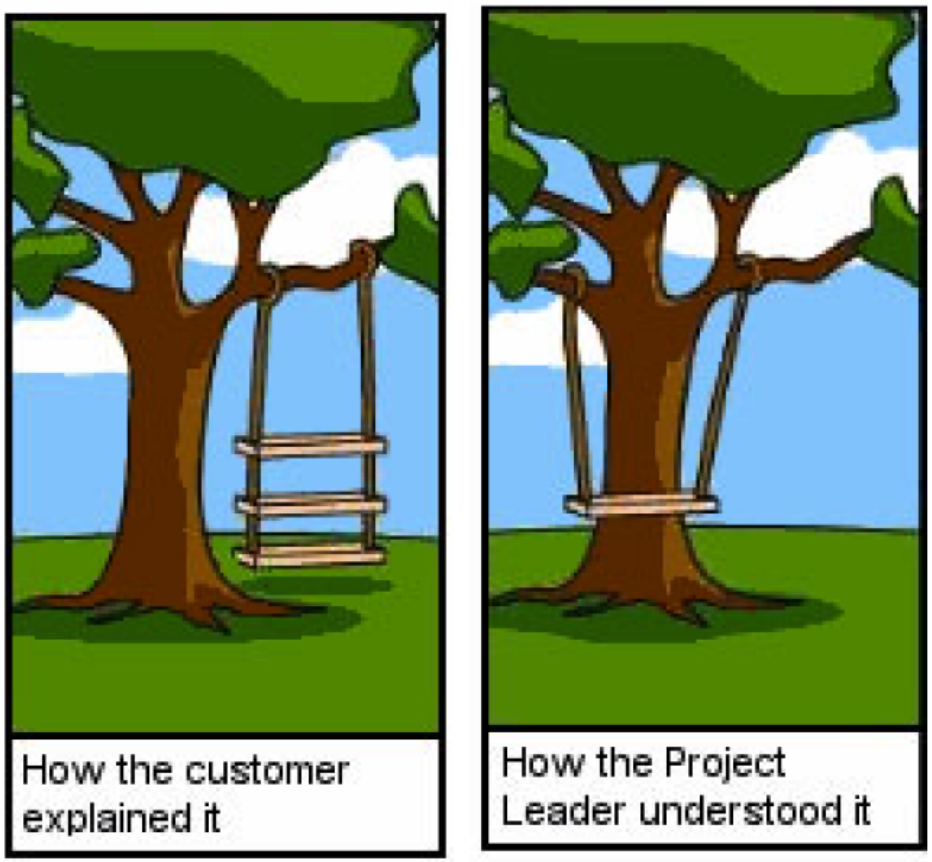
- Department offers lectures
- Instructors conduct the courses
- Students attend courses
- .....

Requirement gathering and analysis

Mini world: CSE Dept $\rightarrow$ requirement analysis
Main constructs

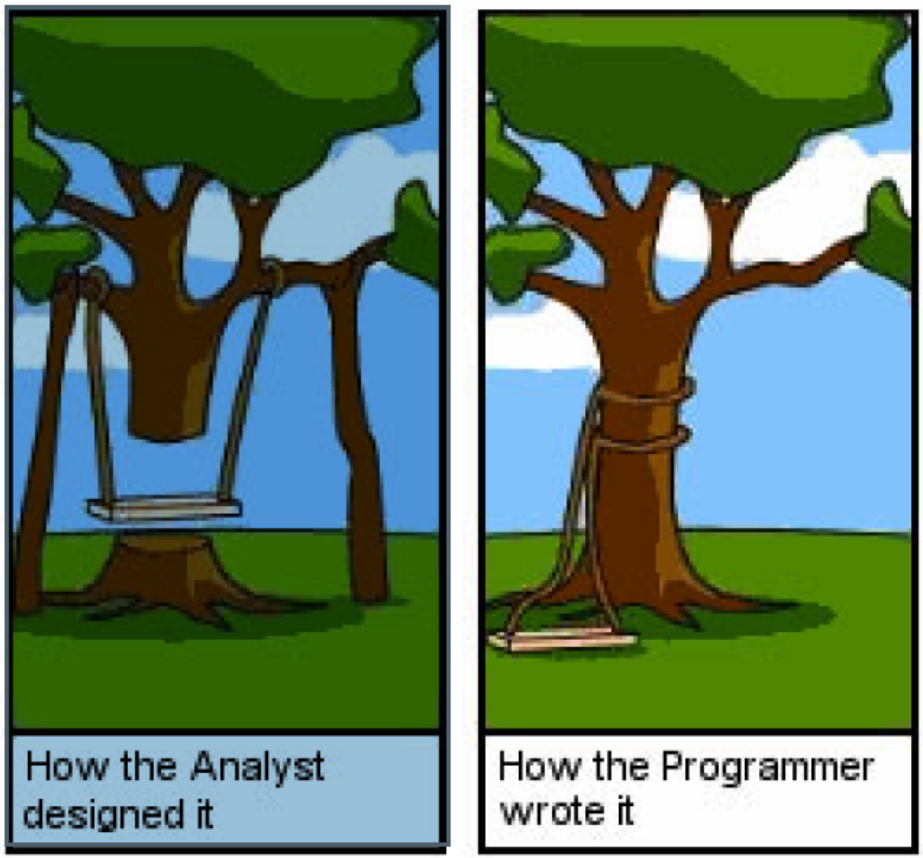
- Identification of organizational units
- Identification of relationships
- Identification of processes
- Formalization
- Requirement specifications

Entity relationship modeling



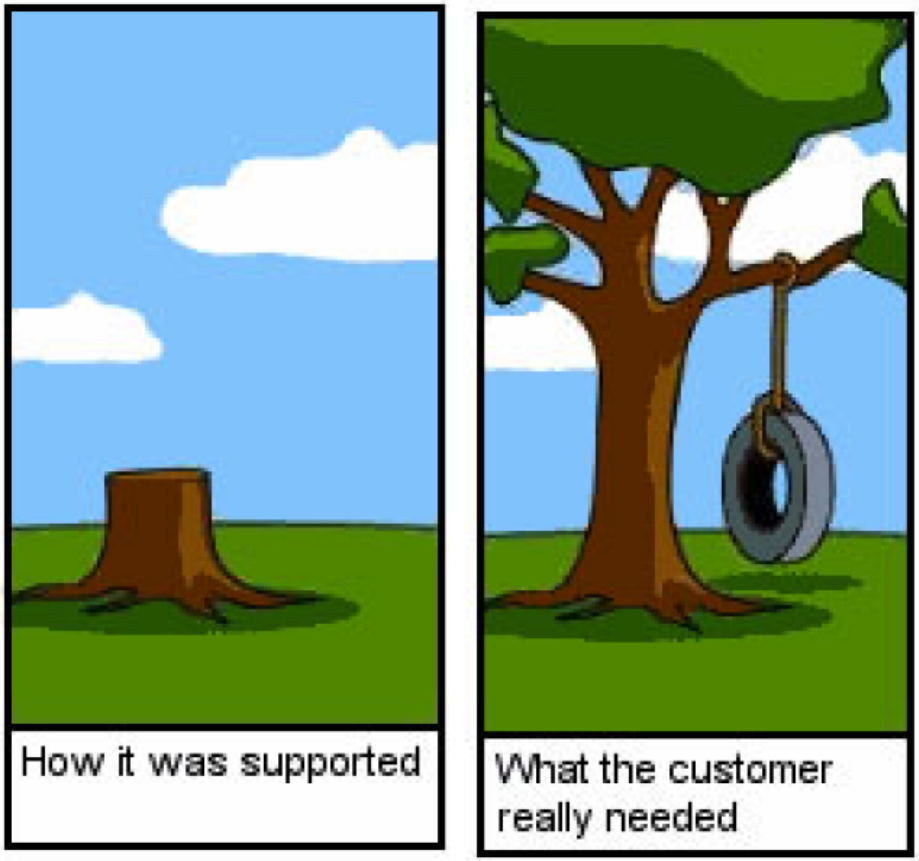
http://www.fuki.ch/

Entity relationship modeling



http://www.fuki.ch/

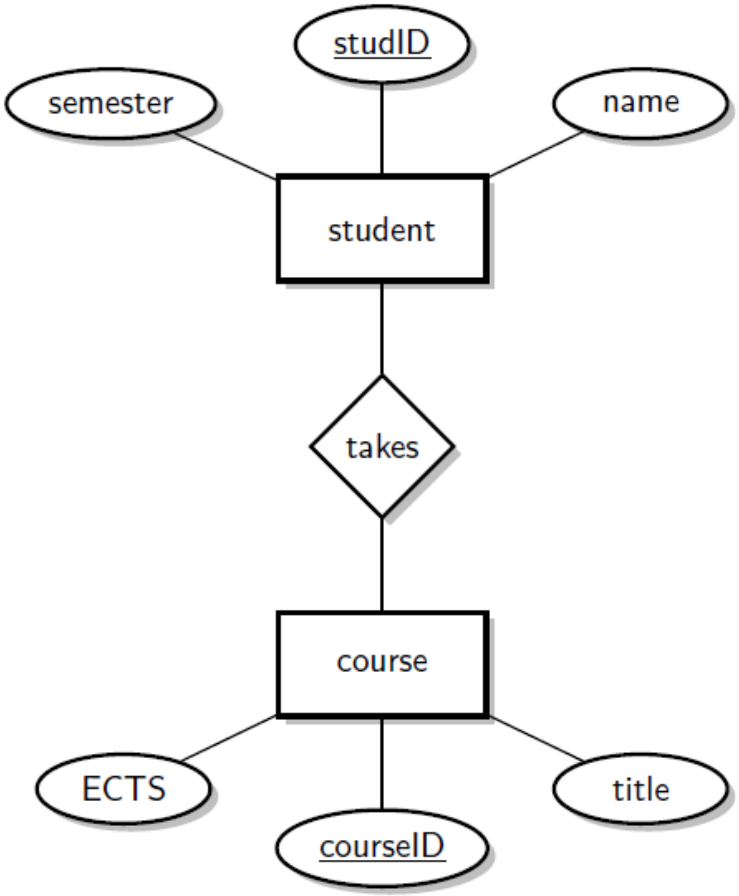
Entity relationship modeling



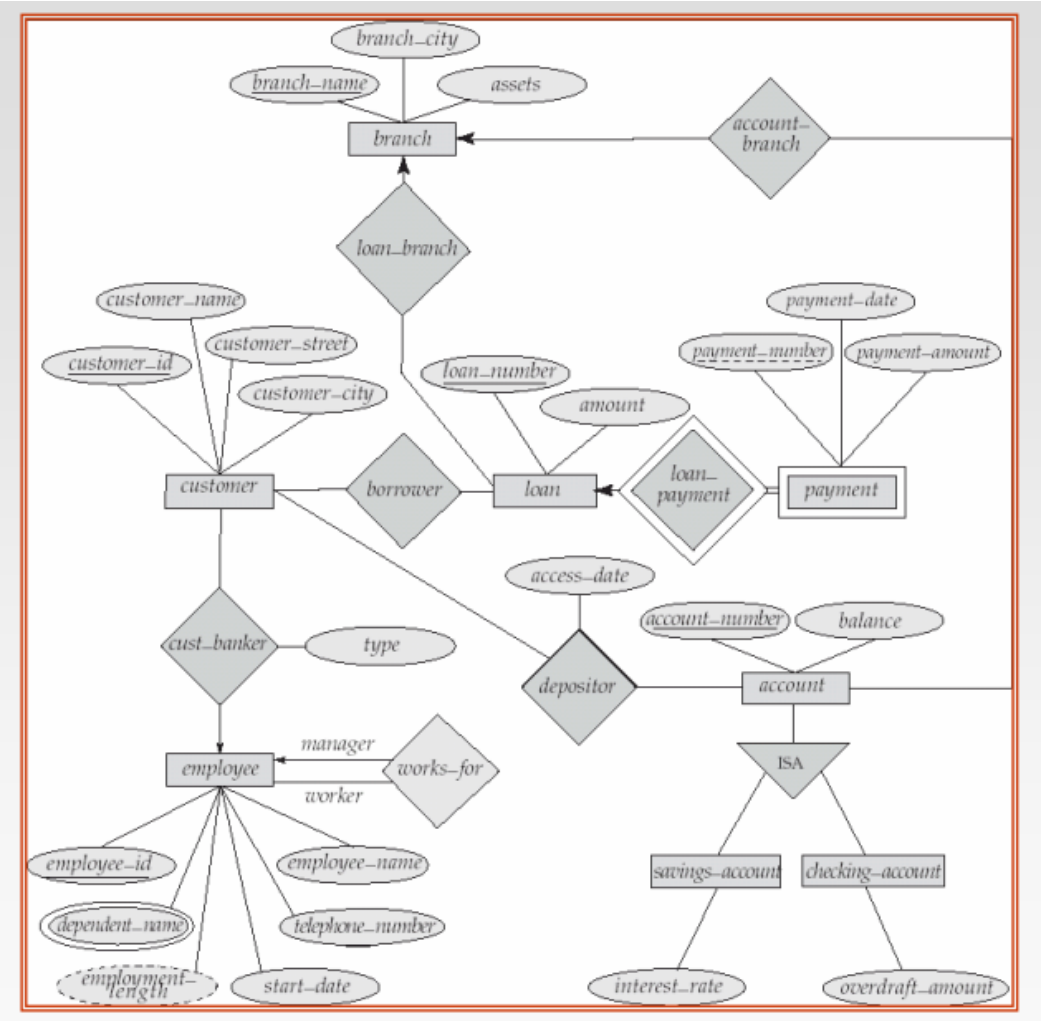
http://www.fuki.ch/

Entity relationship modeling

1. Entity → Entity type
2. Relationship → Relationship type
3. Attribute
4. Primary key ....



Another ER example: Banking enterprise



Relational model

ER $\rightarrow$ relations

For instance

- student(<u>stuID</u>, name, semester)
- course(<u>courseID</u>, credit, title)

Relational Algebra

Type of operations

- selection σ
- projection π
- cross product X
- join $\bowtie$
- rename ρ
- difference $-$
- ...

Examples

$\sigma_{student.stuID=s197010110}(student)$

Relational Algebra

Type of operations

- selection σ
- projection π
- cross product X
- join $\bowtie$
- rename ρ
- difference $-$
- ...

Examples

$\sigma_{student.stuID=s197010110}(student)$

$\pi_{customer_{name}}(\sigma_{branch_{name}="Downtown"}(depositor \bowtie account)) \cap \pi_{customer_{name}}(\sigma_{branch_{name}="Uptown"}(depositor \bowtie account))$

Tables and SQL

student	
studID	name
26120	K. Pedersen
25403	T. Jensen
...	...

attend	
studID	courseID
25403	5022
26120	5001
...	...

course	
courseID	title
5001	DBS
5022	Robotics
...	...

```
SELECT name
FROM student, attend, course
WHERE student.studID = attend.studID AND
      attend.courseID = course.courseID AND
      course.title = 'DBS'
```

Tables and SQL

student	
studID	name
26120	K. Pedersen
25403	T. Jensen
...	...

attend	
studID	courseID
25403	5022
26120	5001
...	...

course	
courseID	title
5001	DBS
5022	Robotics
...	...

UPDATE course
SET title = 'Database Systems'
WHERE courseID = 5001

Integrity constraints

- Uniqueness
- avoid data inconsistencies

CREATE TABLE student
(studID INTEGER PRIMARY KEY,
name VARCHAR(30) NOT NULL,
semester INTEGER **CHECK semester BETWEEN 1 AND 13**)

Normalizations

Goal:avoiding redundancy

empid	name	rank	office	courseId	title	ects
2125	Socrates	C4	226	5041	Ethics	4
2125	Socrates	C4	226	5049	DBS	2
2125	Socrates	C4	226	4052	Logics	4
2137	Kant	C4	7	5001	Basics	4
2126	Russel	C4	232	5043	Theory of Cognition	3
2126	Russel	C4	232	5052	Theory of Science	3
2128	Russel	C2	230	5216	Bioethics	2
2133	Popper	C3	52	5259	Advanced Algorithms	2
2134	Augustinus	C3	309	5022	Belief and Knowledge	2
2137	Kant	C4	7	4630	Constructive Criticism	4
-	-	-	-	4000	Data Structures	4

- Insert a new course
- Update the office to 338 who teaches Ethics
- What happen if constructive criticism course is deleted from the syllabus?

Transactions

How to handle concurrent transactions

	transaction 1	transaction 2
1.	x:=balance	
2.	x:=x+2000 // *salary*	y:=balance
3.	balance:=x	y=y-100 // *withdrawal*
4.		balance:=y

Transactions

How to handle concurrent transactions

	transaction 1	transaction 2
1.	x:=balance	
2.	x:=x+2000 // *salary*	y:=balance
3.	balance:=x	y=y-100 // *withdrawal*
4.		balance:=y

This is a lost update

Transactions

Synchronization

- Resolve problems arisen when multiple users concurrently read/write data
- The database system allows concurrent access but prevents conflicts

Recovery

Failure at time t

- Transactions committed successfully prior to t must not be lost
- All aborted transaction should be completely deleted

Indexing

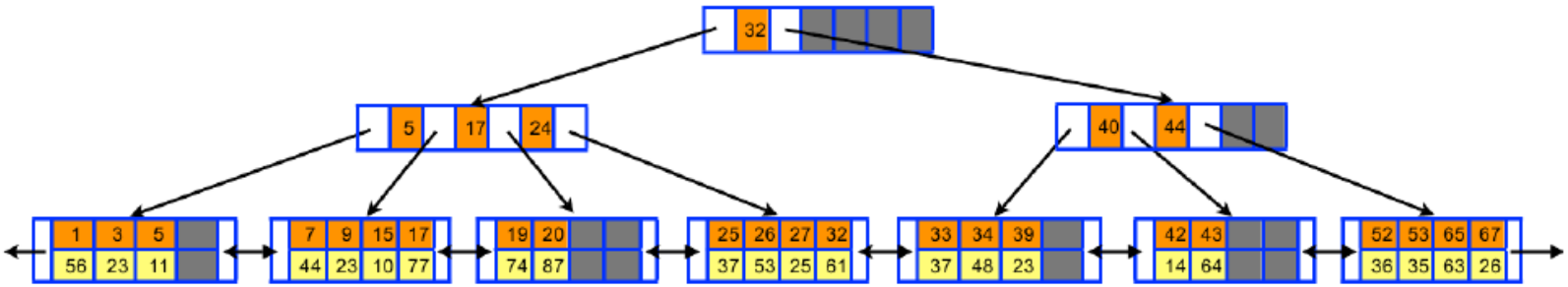
Increase performance

- Problem: Efficient search in big database
- Mapping: Keys $\rightarrow$ set of entries
- Main memory too small for all data
- Hard disk access is expensive

B+ tree

Reading in blocks instead of bit by bit
Better than a binary tree

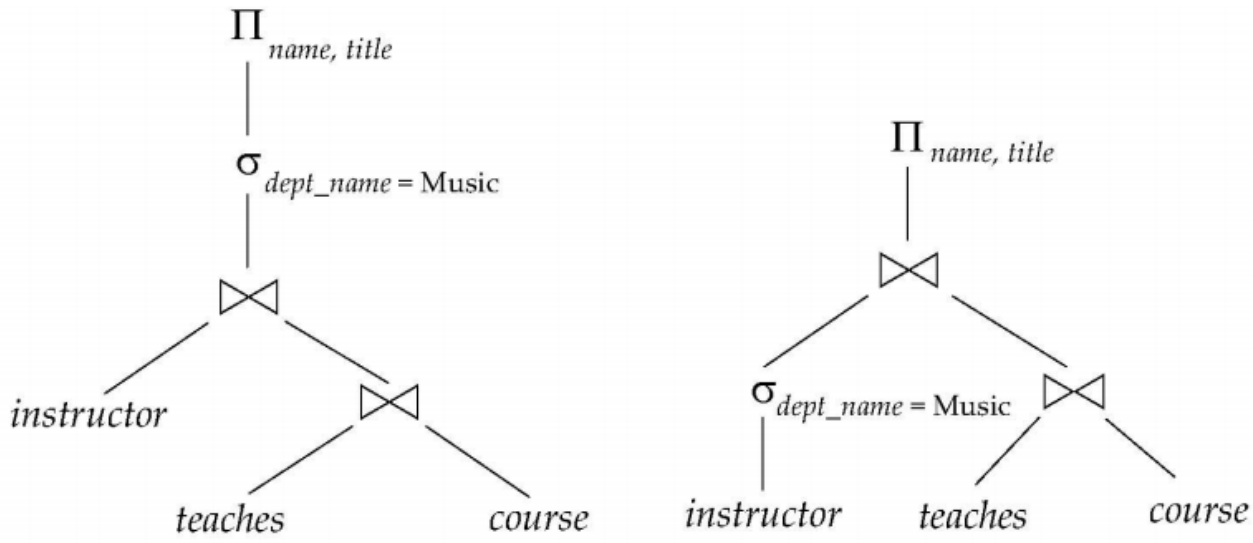
- n-ary tree
- One page on disk = one node/leave
- Data are stored in leaves



Query processing and optimization

Efficient execution of queries

- a query $\rightarrow$ an executable query plan
- Cost model: find the cheapest execution plan.



Acknowledgement

- Katja Hose, Aalborg University
- slides of Database System concept book

Thank you

Self-studies

- Review the concepts of “Relations” you studied in your Discrete Math course.
- Review the concepts of class diagram you studied in your OOP course.