When to not use a DB

8/31

high overheads:

- high investment (hardware, software, training)
- generality (data processing & defining)
- security, concurrency control

Avoid:

- ~~real~~ real time system w/ DB causing enough overhead to not meet time constraints
- embedded systems w/ limited program mem and very limited storage
- no multi-user access ← Wrong

Data Models, Schemas, and Instances

data abstraction: supress of details of storage org

data model: describe structure of data

Conceptual data model

entity: represent a real world object (STUDENT)

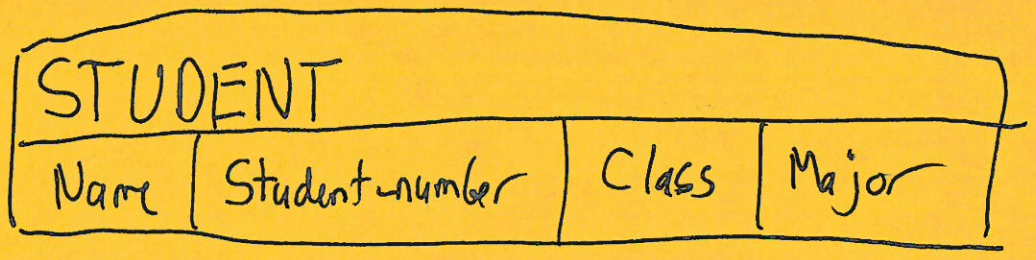
attributes: properties of interest about an entity (e.g. name & student_id)

relationships: associations between entities

Schema

Schema: description of data

schema diagrams: visual rep of schema



~~Snapshot~~

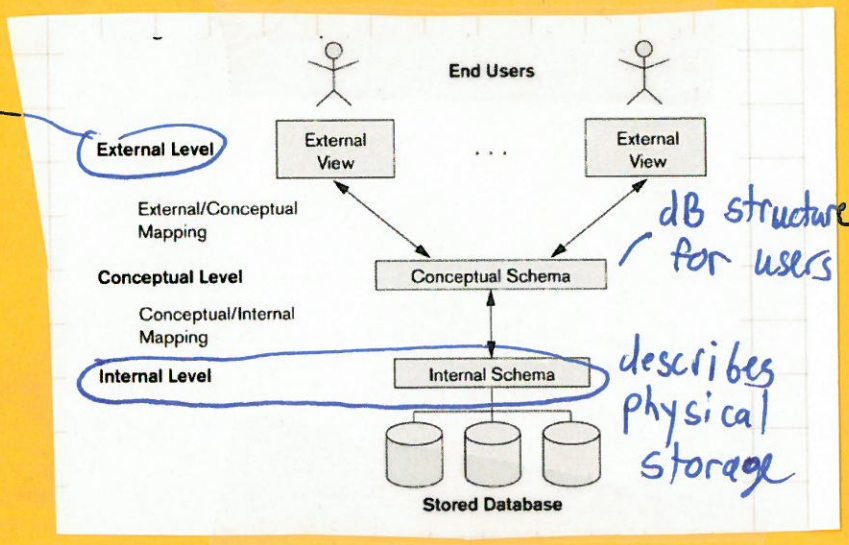
Snapshot: database at particular moment

Schema vs. State

- define DB schema: only schema, state is empty
- initalize the state (w/ data)
- update operation produces another state

3 schema arch for data independence

External level
describes how particular groups see particular data



data independence: ability to change a layer's schema and the change doesn't propagate

Physical ~~Logical~~ data independence:
changes to internal schema don't change conceptual schema

Logical independence
- change to conceptual doesn't change external

TRANSCRIPT

Student_name	Student_transcript				
	Course_number	Grade	Semester	Year	Section_id
Smith	CS1310	C	Fall	08	119
	MATH2410	B	Fall	08	112
Brown	MATH2410	A	Fall	07	85
	CS1310	A	Fall	07	92
	CS3320	B	Spring	08	102
	CS3380	A	Fall	08	135

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

GRADE_REPORT

Student_number	Student_name	Section_identifier	Course_number	Grade
17	Smith	112	MATH2410	B
17	Smith	119	CS1310	C
8	Brown	85	MATH2410	A
8	Brown	92	CS1310	A
8	Brown	102	CS3320	B
8	Brown	135	CS3380	A