>>

Data Models

- Data Modelling
- Some Design Ideas
- Exercise: GMail Data Model
- Quality of Designs

∧   >>

❖ Data Modelling

Aims of data modelling:

- describe what information is contained in the database
  (e.g., entities: students, courses, accounts, branches, patients, ...)

- describe relationships between data items
  (e.g., John is enrolled in COMP3311, Tom's account is held at Coogee)

- describe constraints on data
  (e.g., 7-digit IDs, students can enrol in no more than 3 courses per term)
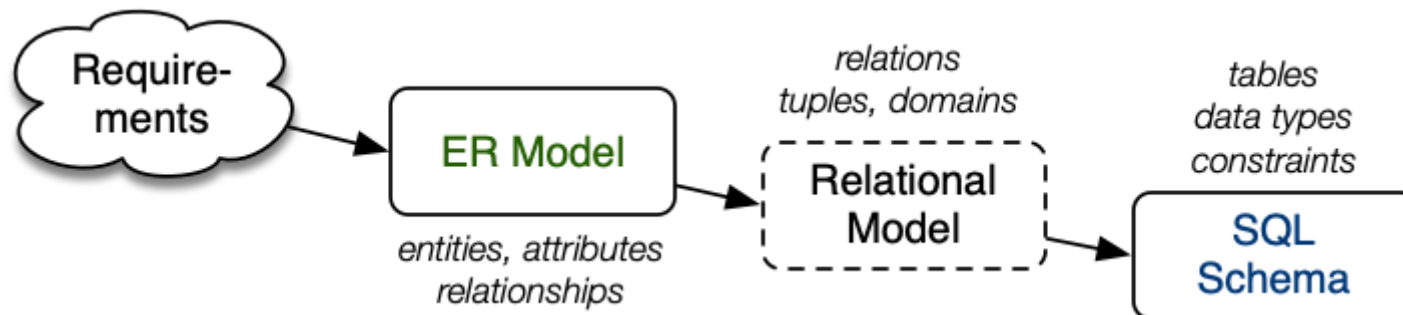
Data modelling is a design process

- converts requirements into a data model

COMP3311 21T1 ◇ Data Models ◇ [1/6]

<< ∧ >>

❖ Data Modelling (cont)

Kinds of data models:

- logical: abstract, for conceptual design, e.g., ER, ODL, UML

- physical: record-based, for implementation, e.g., relational, SQL

Strategy: design using abstract model; map to physical model

<<   ∧   >>

❖ Some Design Ideas

Consider the following while working through exercises:

- start simple ... evolve design as problem better understood

- identify objects (and their properties), then relationships

- most designs involve kinds (classes) of people

- keywords in requirements suggest data/relationships
  (rule-of-thumb: nouns →data, verbs →relationships)

- don't confuse operations with relationships
  (operation: he **buys** a book; relationship: the book **is owned** by him)

- consider all possible data, not just what is available

COMP3311 21T1 ◇ Data Models ◇ [3/6]

<<   ∧   >>

❖ Exercise: GMail Data Model

Consider the GMail system (or any other modern mail client)

Develop an informal data model for it by identifying:

- the data items involved (objects and their attributes)
- relationships between these data items
- constraints on the data and relationships

<< ∧ >>

❖ Exercise: GMail Data Model (cont)

Objects in GMail data model:

```
  users
            gmail-address, name, password, ...

  messages
            timestamp, sender*, title, content, ...

  tags
            owner, name, colour parent*

  settings
            name, value, user*
```

Relationships in GMail data model:

```
  recipients
            user - message
```

```
sent
        user - message

tag-hierarchy
        child-tag - parent-tag

settings
        user - setting
```

Constraints in GMail data model:

```
gmail-address values are unique

users must have a password (strong?)

every message has a sender

every message has a non-empty title and content

values for each setting are valid for that setting
```

<< ∧

❖ Quality of Designs

There is no single "best" design for a given application.

Most important aspects of a design (data model):

- correctness   (satisfies requirements accurately)

- completeness   (all reqs covered, all assumptions explicit)

- consistency   (no contradictory statements)

Potential inadequacies in a design:

- omits information that needs to be included

- contains redundant information ($\Rightarrow$ inconsistency)

- leads to an inefficient implementation

- violates syntactic or semantic rules of data model

Produced: 10 Feb 2021