Database Design Methodology

Learning Outcomes

- Understand Database Design Methodology, including:
  - Conceptual database design
  - Logical database design
  - Physical database design
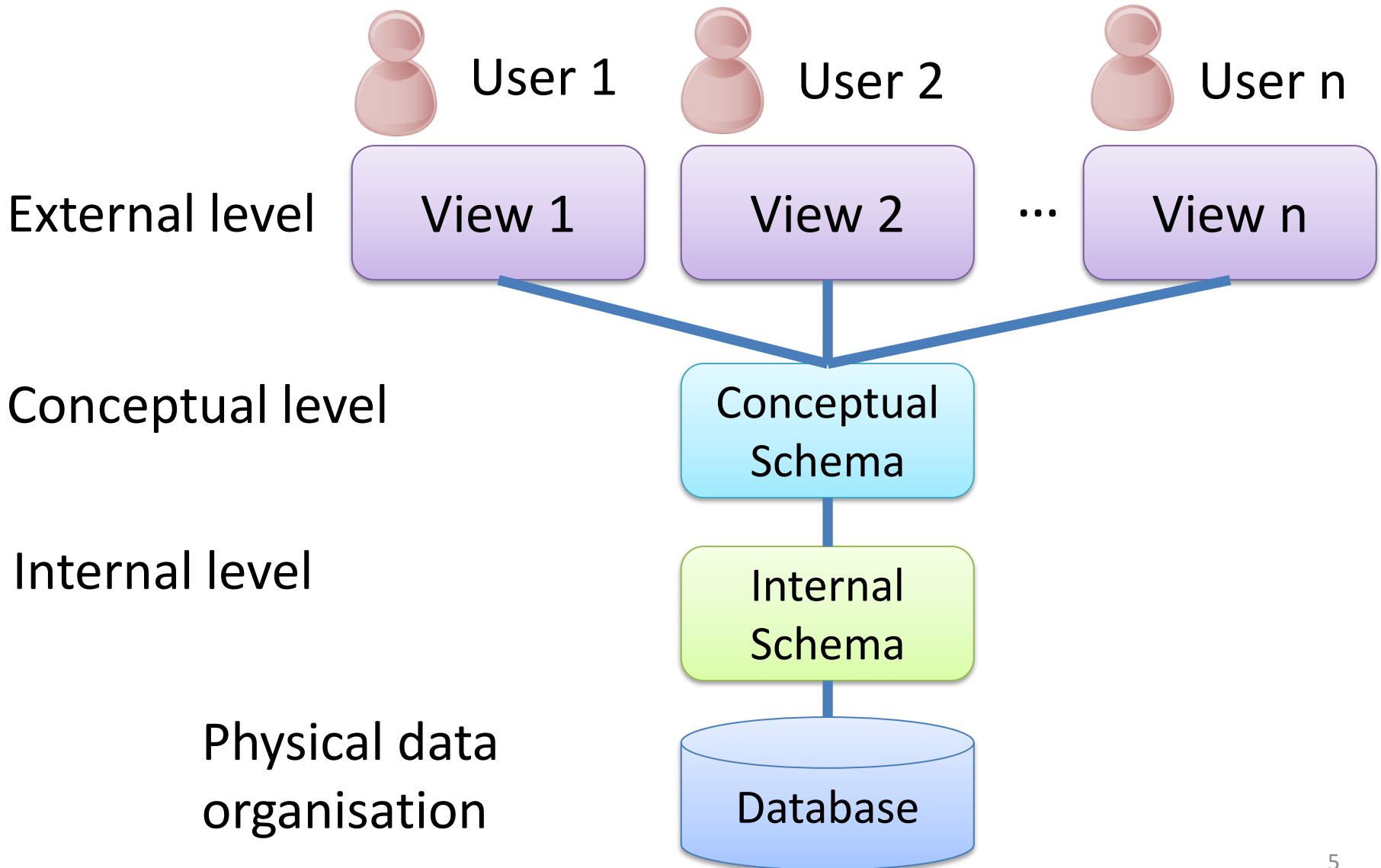- Be able to explain the general steps for Database Design Methodology.

Database Design Methodology

- A structured approach that uses procedures, techniques, tools, and documentation aids to support and facilitate the process of design.

- A design methodology consists of phases each containing a number of steps

Database Design Methodology

- Main phases
  - Gather requirements
  - Conceptual database design
  - Logical database design
  - Physical database design

Three level ANSI-SPARC architecture

External level — User 1 — View 1

User 2 — View 2

... — User n — View n

Conceptual level — Conceptual Schema

Internal level — Internal Schema

Physical data organisation — Database

- # Conceptual database design

  The process of constructing a model of the data used in an enterprise, independent of *all* physical considerations.

- # Logical database design

  Maps the conceptual data model on to a logical model (e.g. relational), but independent of a particular DBMS and other physical considerations.

- # Physical database design

  The process of producing a description of the implementation of the database (tailored to specific DBMS); it describes the base relations, file organizations, and indexes design, and any associated integrity constraints and security measures.

Overview Conceptual database design

- Step 1   Build conceptual data model
  - Step 1.1 Identify entity types
  - Step 1.2 Identify relationship types
  - Step 1.3 Identify and associate attributes with entity or relationship types
  - Step 1.4 Determine attribute domains
  - Step 1.5 Determine candidate, primary, and alternate key attributes

Overview Conceptual database design

- Step 1   Build conceptual data model (continued)
  - Step 1.6 Consider use of enhanced modeling concepts (optional step)
  - Step 1.7   Check model for redundancy
  - Step 1.8   Validate conceptual model against user transactions
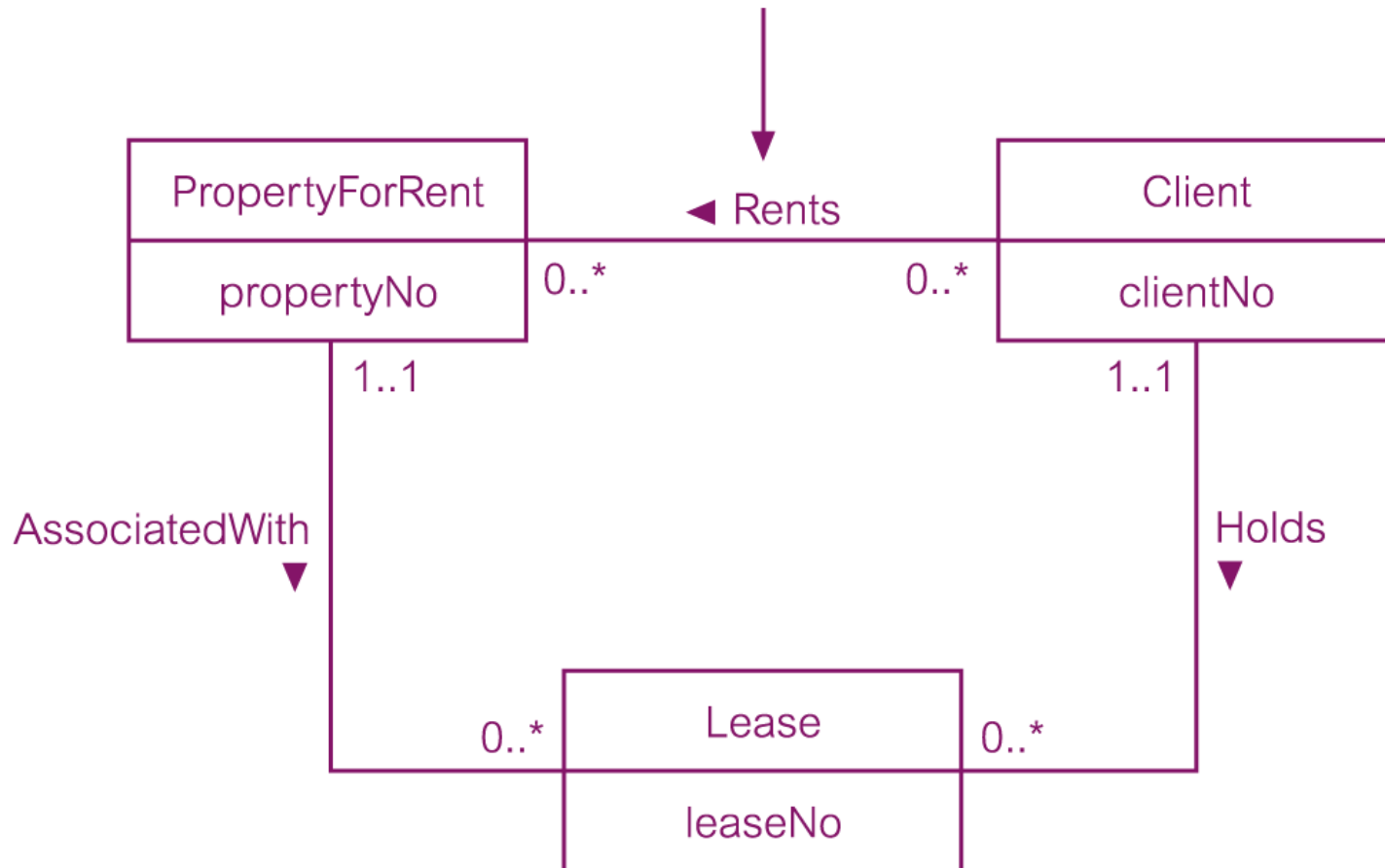 - Step 1.9   Review conceptual data model with user

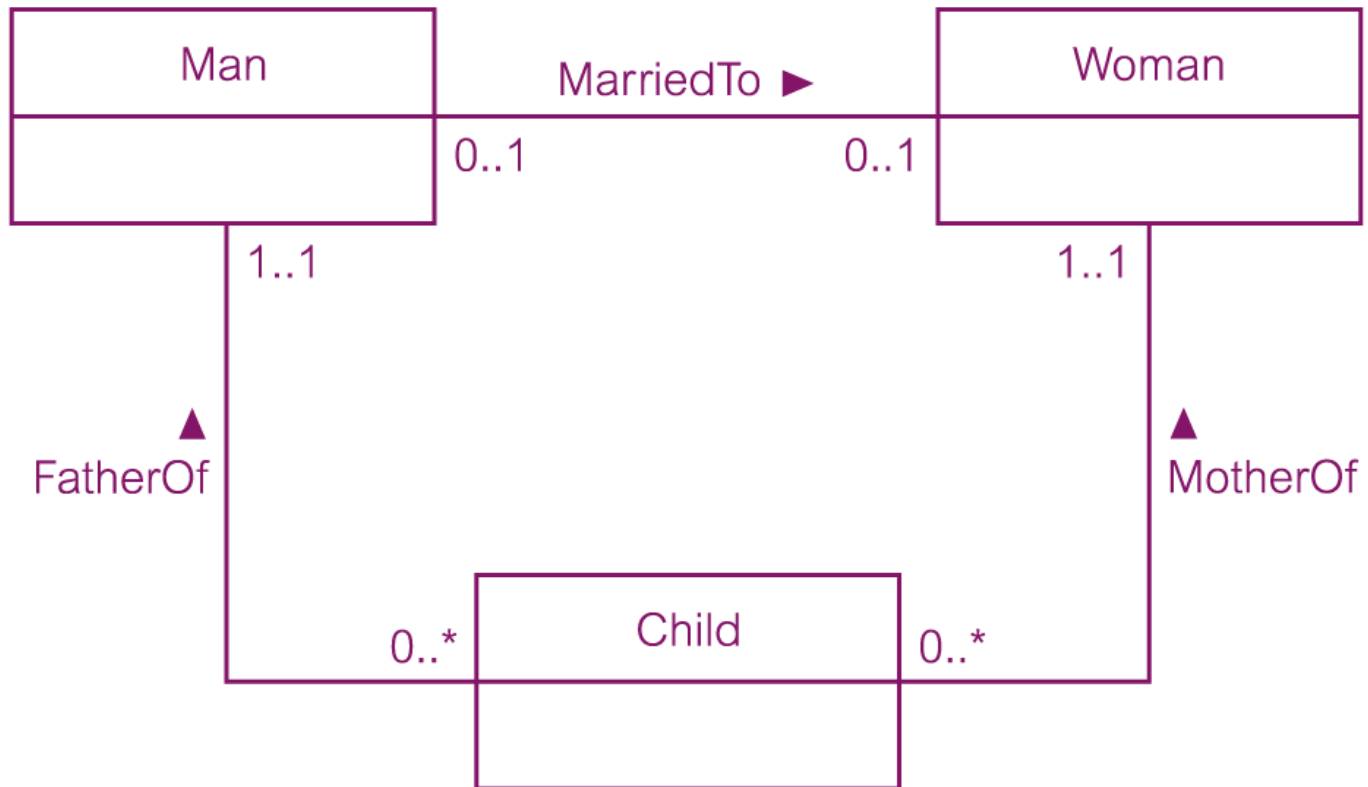# Step 1.5 Determine candidate, primary, alternate key attributes

- Guidelines for choosing candidate, primary, and alternate key attributes:
  - the candidate key with the minimal set of attributes;
  - the candidate key that is least likely to have its values changed;
  - the candidate key with fewest characters (for those with textual attribute(s));
  - the candidate key with smallest maximum value (for those with numerical attribute(s));
  - the candidate key that is easiest to use from the users' point of view.

Step 1.7 Check model for redundancy

- Re-examine one-to-one (1:1) relationships; Client, Renter
- Remove redundant relationships;
- Consider time dimension.

Remove the redundant
relationship called *Rents*

PropertyForRent
propertyNo

◄ Rents

0..*   0..*

Client
clientNo

1..1   1..1

AssociatedWith
▼

Holds
▼

0..*

Lease
leaseNo

0..*

Man | MarriedTo ► | Woman
0..1 | | 0..1
1..1 | | 1..1

FatherOf ▲

MotherOf ▲

0..* | Child | 0..*

Overview Logical database design

- Step 2   Build and validate logical data model
  - Step 2.1 Derive relations for logical data model
  - Step 2.2 Validate relations using normalization
  - Step 2.3 Validate relations against user transactions
  - Step 2.4 Check integrity constraints
  - Step 2.5 Review logical data model with user
  - Step 2.6 Merge logical data models into global model (optional step)
  - Step 2.7 Check for future growth

Step 2.4 Check integrity constraints

- To check integrity constraints are represented in the logical data model. This includes identifying:
    - Required data (value not allowed to be null)
    - Attribute domain constraints
    - Multiplicity (*:* relationships)
    - Entity integrity (primary key can not be null)
    - Referential integrity (foreign keys)
    - General constraints

Referential integrity

- Parent relation:

Staff (<u>staffNo</u>, fName, lName, position, sex, DOB, salary)

- Child relation:

PropertyForRent(propertyNo, street, city, postcode, type, rooms, rent, ownerNo, staffNo)

Case 1: insert tuple into child relation (PropertyForRent)

Case 2: Delete tuple from child relation(PropertyForRent)

Case 3: update foreign key of child tuple (PropertyForRent)

Case 4: Insert tuple into parent relation (Staff)

Case 5: Delete tuple from parent relation (Staff)

NO ACTION, CASCADE, SET NULL, SET DEFAULT

Case 6: Update primary key of parent tuple (Staff)

PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo, staffNo)

Primary Key propertyNo

Foreign Key ownerNo **references** PrivateOwner(ownerNo) and BusinessOwner(ownerNo)

ON UPDATE CASCADE ON DELETE NO ACTION

Foreign Key staffNo **references** Staff(staffNo)

ON UPDATE CASCADE ON DELETE SET NULL

Overview Physical database Design

- Step 3 Translate logical data model for target DBMS
  - Step 3.1 Design base relations
  - Step 3.2 Design representation of derived data
  - Step 3.3 Design general constraints
- Step 4 Design file organizations and indexes
  - Step 4.1 Analyze transactions
  - Step 4.2 Choose file organization
  - Step 4.3 Choose indexes
  - Step 4.4 Estimate disk space requirements

Overview Physical database Design

- Step 5 Design user views

- Step 6 Design security mechanisms

- Step 7 Consider the introduction of controlled redundancy

- Step 8 Monitor and tune the operational system

What have we learned?

- Database design methodology:
    - Gather requirements
    - Conceptual database design
    - Logical database design
    - Physical database design

Extra reading

- Database Systems: a practical approach to design, implementation, and management

  Chapter 16, 17, 18