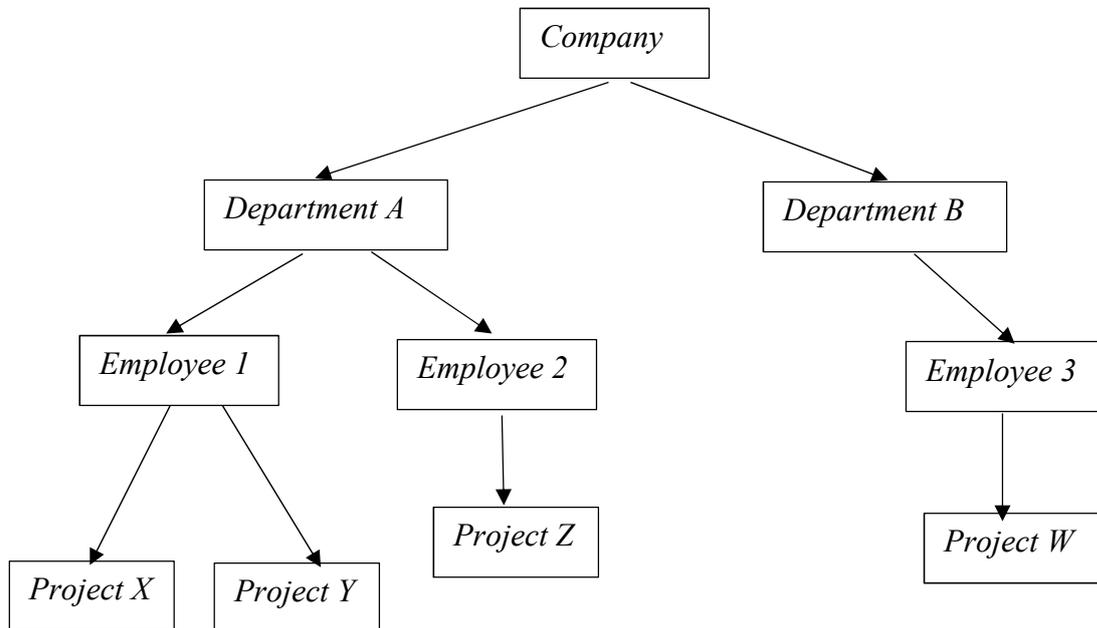


Solution Tutorial N°1 :

Exercise 01 :

1. Hierarchical tree



2. Relationship Types

Relationship	Type
Company → Department	One-to-many
Department → Employee	One-to-many
Employee → Project	One-to-many

(No many-to-many supported)

3. Limitation

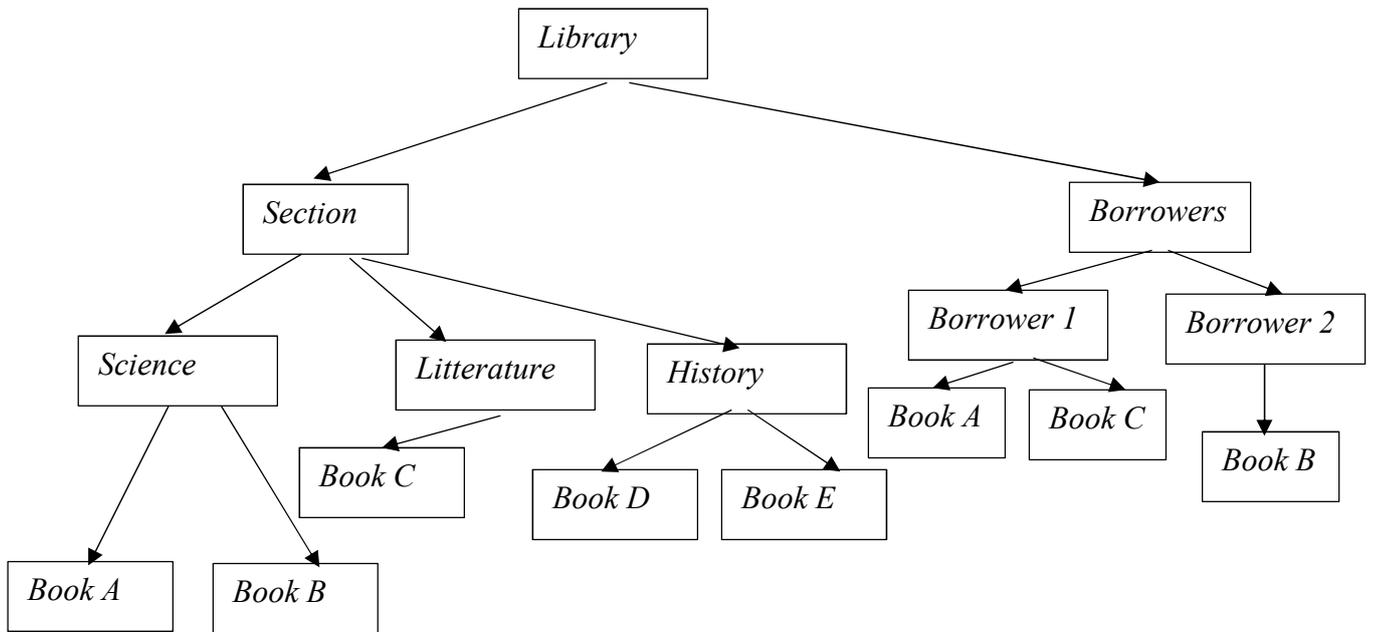
Projects cannot be shared between employees because:

Each child can only have one parent

To assign the same project to two employees → it must be duplicated → causes redundancy and inconsistency.

Exercise 02 :

1. Hierarchical tree:



2. Relationship types:

Relationship	Type
Library → Section	One-to-many
Section → Book	One-to-many
Borrower → Book	One-to-many (at a time)

Many-to-many is **not naturally supported**.

3. Limitation:

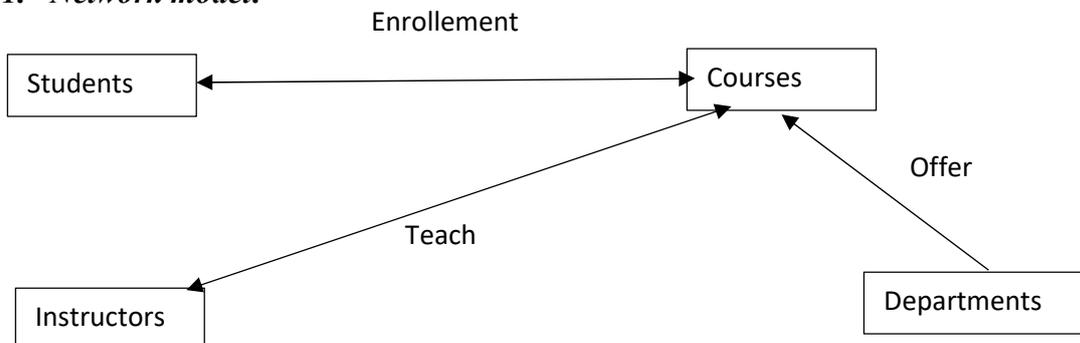
If the same book is borrowed again later:

→ It must be **duplicated as another record** which causes **redundancy + inconsistency**

Because in the hierarchical model: A child can have only one parent at a time.

Exercise 03:

1. Network model:



2. Relationship types:

Relationship	Type
Student ↔ Course	Many-to-many
Department ↔ Course	One-to-many
Instructor ↔ Course	Many-to-many

3. Limitation:

Main limitations:

✗ Structural complexity

- Many owner–member sets create complicated schemas
- Hard to understand and maintain

✗ High dependency on physical structure

- Applications must know navigation paths
- Data access is procedural (navigate pointer-to-pointer)

✗ Poor data independence

- Any structural change requires modifying programs

✗ Less flexible than relational model

- Queries are not declarative
- No ad-hoc querying like SQL

✗ Hard to represent very large or evolving systems

These limitations are why relational databases eventually replaced network DBMSs