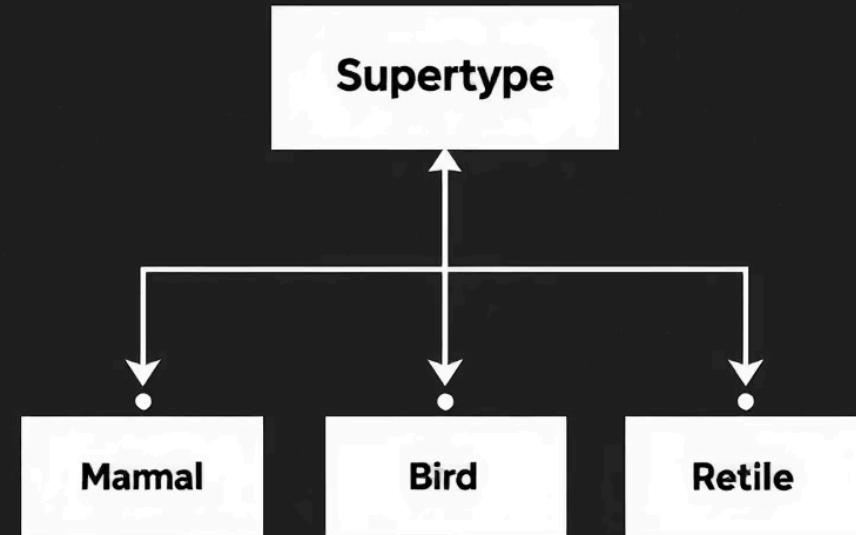


Specifying Constraints in Supertype/Subtype Relationships

This presentation explores how to effectively specify constraints in supertype/subtype relationships within the Entity-Relationship (E-R) model. These constraints are crucial for maintaining data integrity and accurately reflecting real-world business rules. Let's dive in.



Supertype/Subtype Example: Employee

Employee Supertype

Consider an `Employee` supertype with attributes like employeeID, name, and address. This represents the general concept of an employee.

Salaried vs. Hourly Subtypes

Specialized subtypes, such as `Salaried_Employee` (salary) and `Hourly_Employee` (hourly_rate), inherit these attributes but add their own.



Total
Specialization

The diagram consists of two large circles, each with a white center and a dark gray outer ring. The left circle contains the text 'Total Specialization' and the right circle contains 'Partial Specialization'. A black arrow points from the right side of the left circle to the left side of the right circle, indicating a transition or relationship between the two concepts.

Partial
Specialization

Constraint Types: Completeness Constraint

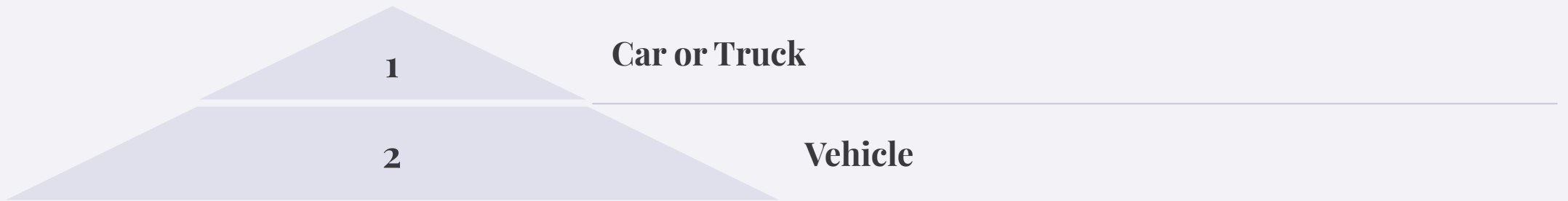
Total Specialization

Specifies that every supertype instance **must** be a member of at least one subtype.

Partial Specialization

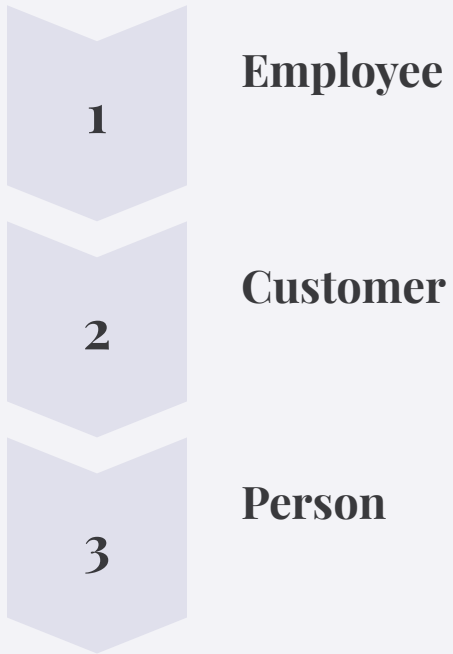
Specifies that a supertype instance **can** exist without being a member of any subtype.

Total Specialization Example



A `Vehicle` supertype must be either a `Car` or a `Truck`. No vehicle can exist as *only* a generic `Vehicle`.

Partial Specialization Example



A `Person` supertype can be an `Employee`, a `Customer`, or neither. A person can exist without being associated with either subtype.



Constraint Types: Disjointness Constraint

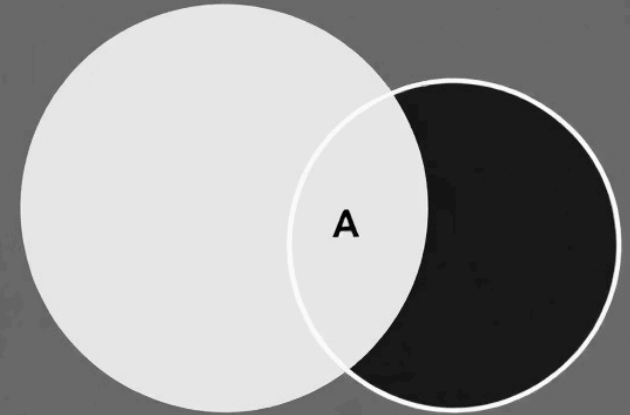
Disjoint Rule

A supertype instance can belong to **only one** subtype.

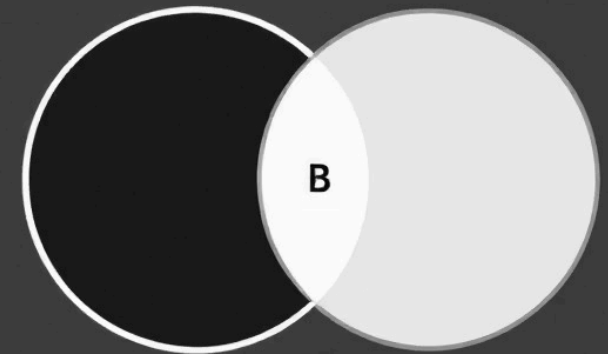
Overlap Rule

A supertype instance can belong to **multiple** subtypes simultaneously.

Disjoinens of Disjoprojung



An Oecolping two Constraint



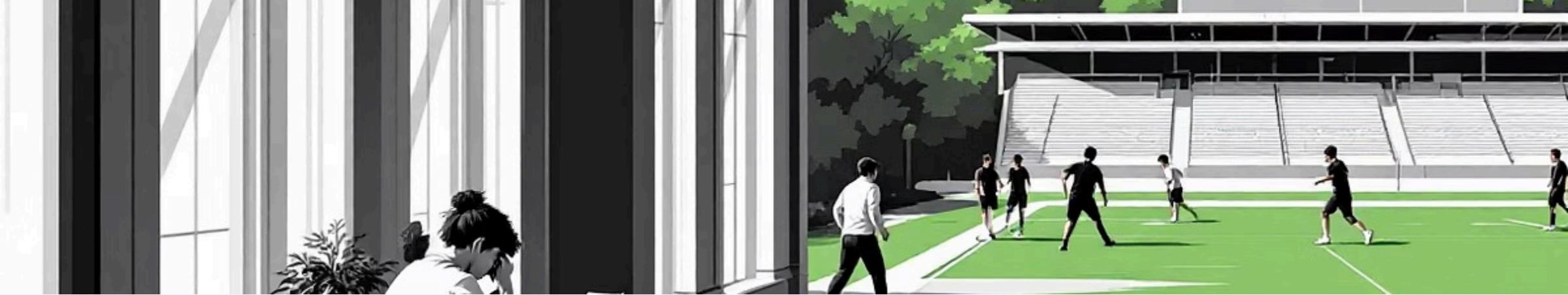
Disjoint Rule Example

Inpatient

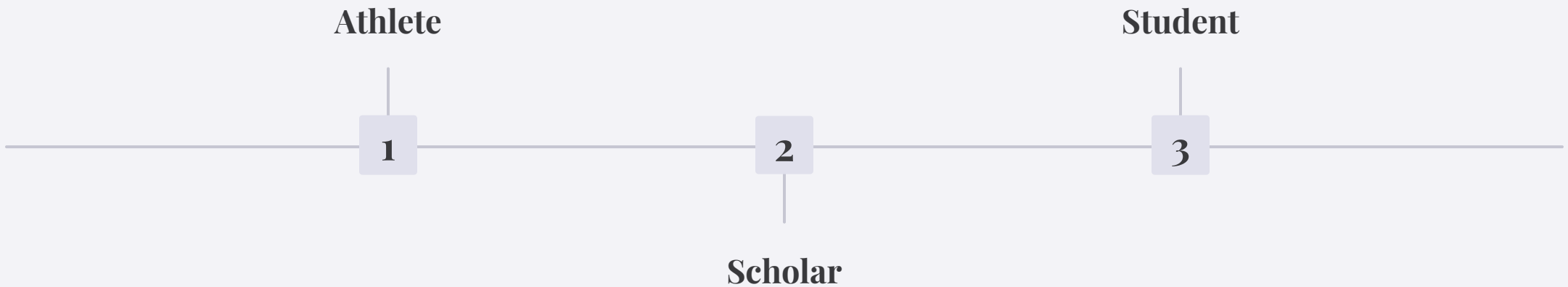
Outpatient



A `Patient` supertype can be either an `Inpatient` OR an `Outpatient`, but not both at the same time. It is disjoint.

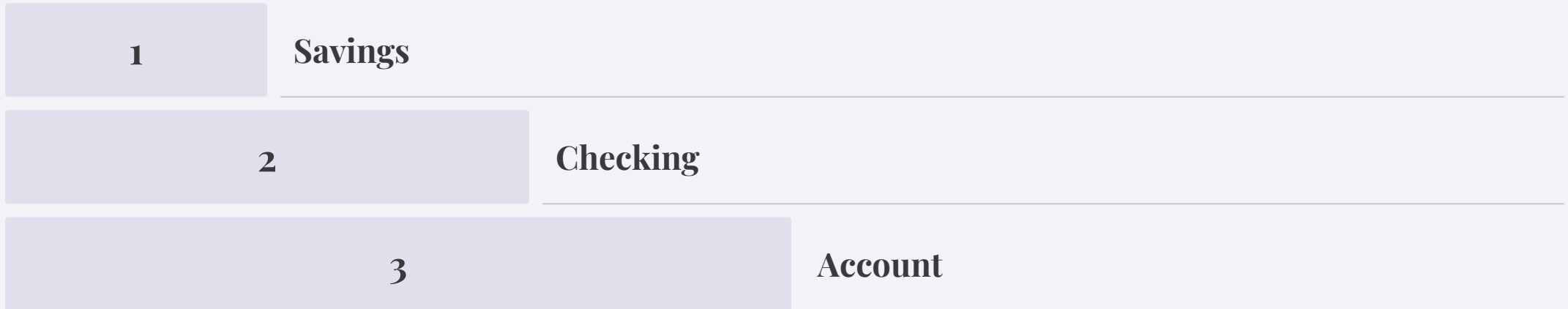


Overlap Rule Example



A `Student` supertype can be both an `Athlete` AND a `Scholar`. There is overlap because students can participate in both athletics and academics.

Combining Constraints: Example



An `Account` (Supertype) uses Total Specialization for (`Checking`, `Savings`) and a Disjoint Constraint. Accounts must be one or the other, but not both.

Conclusion: Importance of Constraints



Enforce Rules



Data Quality



Design

Constraints are essential. They enforce business rules, improve data quality, and help make better decisions during database design. They make sure data stays consistent and accurate.