

Mapping Self (Recursive) Relationships in Databases

This guide explains how to map **Self-Relationships (Recursive Relationships)** in relational databases for two types:

- ✓ One-to-One Self Relationship
- ✓ Many-to-Many Self Relationship

1) One-to-One Self Relationship – Mapping

Principle

A **1:1 self-relationship** means:

Each record can be related to **exactly one other record** in the same table, and vice versa.

Real-life examples:

- ✓ A person with a **spouse**
- ✓ An employee with a **backup contact**

Relational Mapping

We use a **foreign key (FK)** column in the same table referencing the **primary key (PK)** of the same table.

Example: Person table

Column	Type
PersonID	PK
Name	TEXT
PartnerID	FK → Person(PersonID)
D	Person(PersonID)

- **PartnerID** references the same table.
- This ensures a **1:1 relationship** because each person can have only **one partner**.
- If a person has no partner → PartnerID is **NULL**.

2) Many-to-Many Self Relationship – Mapping

Principle

A **M:N self-relationship** means:

A record can be related to **many other records** in the same table, and vice versa.

Real-life examples:

- ✓ Social networks → A person can have many friends
- ✓ Students collaborating on multiple projects

Relational Mapping

We need a **junction/bridge table** containing **pairs of foreign keys** referencing the same table.

Example: Student table

| StudentID | Name |

Junction Table: Friendship

StudentAID	FK → Student(StudentID)
StudentBID	FK → Student(StudentID)
Primary Key	(StudentAID, StudentBID)

- Each row in Friendship represents a relationship between two records in the same table.

Comparison Table

Relationship Type	Mapping Requirement
One-to-One Self	FK in same table + UNIQUE
Many-to-Many Self	Junction table (Bridge) with 2 FKS

Quick Concept Diagram

- ✓ **Self Relationship** → The same table plays two roles (e.g., Parent/Child or Person/Partner).
- ✓ **One-to-One** → Single FK with UNIQUE constraint.
- ✓ **Many-to-Many** → Use a junction (bridge) table.