

The background is a solid red color. In the upper left, there is a rectangular area containing a grid of small white dots. On the right side, there are two overlapping circles: a larger, lighter red circle and a smaller, darker red circle inside it.

Relationer

Varför är relationer viktiga?

Hjälper oss modellera verkliga kopplingar mellan olika typer av data

Säkerställer att data förblir konsistent och förhindrar fel

Möjliggör effektiv organisering utan duplicering av data

Tillåter komplexa sökningar över flera relaterade tabeller

Hur fungerar relationer?

Varje tabell har en primärnyckel som unikt identifierar varje rad

Tabeller kopplas ihop genom främmande nycklar

En främmande nyckel är en kolumn som refererar till en annan tabells primärnyckel

Exempel: En order har en `customer_id` som pekar på en specifik kund i `customers`-tabellen

One-to-one

En rad i en tabell kopplas till exakt en rad i en annan tabell

Används när vi vill dela upp information som hör till samma sak

Exempel från e-handel:

- En produkt har en detaljerad beskrivning
- products-tabellen kopplas till product_details-tabellen via product_id
- Används när viss information inte behöver hämtas varje gång

One-to-Many

Vanligaste typen av relation i databaser

En rad i första tabellen kan kopplas till flera rader i andra tabellen

Exempel från e-handel:

- En kund kan ha många ordrar
- customers-tabellen kopplas till orders-tabellen via customer_id
- Varje order tillhör exakt en kund

Many-to-many

Båda tabellerna kan ha flera kopplingar till varandra

Kräver en kopplingstabell (junction table) mellan huvudtabellerna

Exempel från e-handel:

- En produkt kan tillhöra flera kategorier
- En kategori kan innehålla flera produkter
- Kopplingstabell `product_categories` med både `product_id` och `category_id`

Praktiskt exempel från e-handel

Kunder och ordrar:

- En kund kan ha många ordrar (1:N)
- Kopplas via `customer_id` i `orders`-tabellen

Ordrar och produkter:

- En order innehåller många produkter (N:M)
- Kopplas via `order_items`-tabellen

Produkter och kategorier:

- Produkter kan tillhöra flera kategorier (N:M)
- Kopplas via `product_categories`-tabellen

Cascade - Automatisk uppdatering av relaterad data

Cascade är regler som styr vad som händer med relaterad data vid ändringar

Kan konfigureras separat för UPDATE och DELETE

Vanliga cascade-alternativ:

- CASCADE: Ändringar fortplantar sig automatiskt till relaterad data
- RESTRICT: Förhindrar ändringar om det finns relaterad data
- SET NULL: Sätter främmande nyckeln till NULL vid ändringar
- NO ACTION: Standardbeteende, liknar RESTRICT

Cascade - Exempel från e-handel

När en kund raderas:

- CASCADE: Alla kundens ordrar raderas automatiskt
- RESTRICT: Kan inte radera kunden om den har ordrar
- SET NULL: Orderna finns kvar men kopplingen till kunden försvinner

Cascade - Exempel från e-handel

En kund byter id-nummer i customers-tabellen:

- Kundens gamla id: 5
- Kundens nya id: 10
- Med UPDATE CASCADE: Alla kundens ordrar uppdateras automatiskt från `customer_id = 5` till `customer_id = 10`
- Utan CASCADE: Orderna skulle fortfarande peka på det gamla id:t och kopplingen skulle gå sönder

Fördelar med väldesignade relationer

Garanterar dataintegritet och konsistens

Förenklar underhåll och uppdateringar

Möjliggör effektiva sökningar över flera tabeller

Förhindrar att data blir föråldrad eller inkonsistent

Databasen kan växa utan att grundstrukturen behöver ändras

Enklare att återanvända data i olika sammanhang