

Cascading i SQL

Hantering av referensintegritet

Vad är cascading?

Cascading innebär att en ändring i en tabell påverkar relaterade tabeller automatiskt.

Två huvudtyper:

- ▶ **ON DELETE CASCADE** → Raderar relaterade rader automatiskt.
- ▶ **ON UPDATE CASCADE** → Uppdaterar relaterade rader automatiskt.

Problem utan cascading

Vad händer om vi inte har cascading?

Exempel:

- ▶ Om vi raderar en kund i en databas kan dess ordrar bli "föräldralösa".
- ▶ Om vi ändrar ett ID i huvudtabellen måste vi manuellt uppdatera alla relaterade rader.

Lösning: Använd cascading för att hantera detta automatiskt.

Syntax

```
CREATE TABLE customers (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    name VARCHAR(100) NOT NULL );
```

```
CREATE TABLE orders (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    customer_id INTEGER,  
    FOREIGN KEY (customer_id) REFERENCES customers(id)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE );
```

Övning

Skapa tabeller för customers och orders (utan cascading).

Lägg till en kund och några ordrar.

Radera en kund - vad händer?

Ändra en kunds ID - vad händer?

Skapa nya tabeller, denna gång med Cascading

Lägg till en kund och några ordrar.

Radera en kund - vad händer?

Ändra en kunds ID - vad händer?

Alternativ till cascading

- ▶ **ON DELETE CASCADE** → Använd om relaterade data ska tas bort.
- ▶ **ON DELETE SET NULL** → Använd om vi vill behålla relaterade rader, men ta bort referensen.
- ▶ **ON DELETE RESTRICT** → Förhindrar att en rad raderas om den har relaterade poster.

Testa dessa på föregående övning

Risker med cascading

- ▶ **Oavsiktlig dataförlust** - Om en rad tas bort, kan mycket mer data raderas.
- ▶ **Prestandaproblem** - Stora databaser kan bli långsamma vid många kaskadändringar.
- ▶ **Cirkulära beroenden** - Två tabeller som refererar varandra kan skapa problem.

Hur används cascading i stora databaser?

- ▶ Används i CRM-system, e-handel, bokningssystem etc.
- ▶ Vanligt i API-databaser där data ska synkas automatiskt.
- ▶ Ibland ersatt av **soft delete** (*istället för att radera, markera data som borttagen*).

Övning

Bygg en skoldatabas med cascading

Tabeller: students, courses, enrollments

Regler:

- ▶ Om en student tas bort, tas dess kursregistreringar bort.
- ▶ Om en kurs tas bort, tas alla registreringar bort.