

Min databasdesign uppnår 3NF genom att:

1. Första Normalform (1NF):

- Primärnycklar: Varje tabell har en unik identifierare (PK).
- Atomära värden: Inga kolumner innehåller listor eller sammansatta värden.
- Inga upprepade grupper: Varje attribut lagras i en egen kolumn.

Exempel: Istället för att lagra kontaktuppgifter direkt i person, separeras de till person_details för en renare struktur.

2. Andra Normalform (2NF):

- Fullständigt funktionellt beroende: Alla tabeller uppfyller 1NF och alla icke-nyckelkolumner beror på hela primärnyckeln.

Exempel: I kopplingstabellen student_enrollment är kolumnen grade (betyg) beroende av både student_id och assignment_id. Man kan inte veta ett betyg utan att veta både vilken student och vilket kurstillfälle det gäller.

3. Tredje Normalform (3NF):

- Inga transitiva beroenden: Tabellerna uppfyller 2NF och inga icke-nyckelkolumner beror på andra icke-nyckelkolumner. Allt data beror endast på primärnyckeln.

Exempel: I tabellen educator är hourly_rate (timlös) direkt beroende av educator_id. Vi har inte lagt lönen i person eller course_assignment, vilket elimineras risken för inkonsekvent data om en person har flera roller eller uppdrag.

Andra viktiga designbeslut för 3NF:

a) Person-hierarkin:

- Genom att dela upp personer i person, student, educator och education_leader elimineras redundans. Gemensam information (namn) lagras en gång, medan rollspezifika data endast lagras där den hör hemma.

b) Kurs- och klass-struktur:

- Vi separerar kursdefinitionen (course) från själva utförandet (course_assignment). Detta gör att vi inte behöver upprepa kursnamn eller poäng varje gång en ny klass läser kursen.

c) Eliminering av transitiva platsberoenden:

- Tabellen facility innehåller information om stad och adress. Tabellen class refererar endast till facility_id. Om en skola byter adress behöver vi bara uppdatera det på ett ställe, inte på varje enskild klass.

c) Separation av betyg och studentstatus:

- Betyg lagras i student_enrollment för ett specifikt kurstillfälle, vilket skiljer sig från studentens generella status i student-tabellen.

Fördelar med 3NF-design

1. Minimal redundans: Varje faktum lagras endast på ett ställe.
2. Enkelt underhåll: Uppdateringar (t.ex. ändring av en kurskod) behöver bara göras i en tabell.
3. Datakonsekvens: Risken för att samma information visas olika på olika ställen är elimineras.