Min databasdesign uppnår 3NF genom att:

1. Första Normalform (1NF):

- Varje tabell har en primärnyckel
- Alla kolumner innehåller atomära värden (inte listor eller arrays)
- Ingen kolumn har upprepande grupper

Exempel: person_details separerar personnummer och email från person-tabellen

2. Andra Normalform (2NF):

- Alla tabeller är i 1NF
- Alla icke-nyckelkolumner är fullständigt beroende av hela primärnyckeln

Exempel: student_enrollment är både grade och status helt beroende av kombinationen (student_id, assignment_id)

3. Tredje Normalform (3NF):

- Alla tabeller är i 2NF
- Inga transitiva beroenden - inga icke-nyckelkolumner är beroende av andra icke-nyckelkolumner

Exempel: consultant är hourly_rate direkt beroende av consultant_id, inte av company_id

Andra viktiga designbeslut för 3NF:

a) Person-hierarkin:

- Eliminerar redundans av personuppgifter
- Varje specialisering har endast kolumner relevanta för respektive roll

b) Kurs-struktur:

- Separerar kursdefinition från kursutförande
- Eliminerar upprepning av kursinfo för varje klass

c) Studentinskrivning:

- Separerar klass-tillhörighet från kurs-inskrivning
- Tillåter att studenter kan vara inskrivna i olika kurser

- Transitiva beroenden som eliminerats
- Stad/Adress beroende
- facility innehåller stad och adress
- class refererar bara till facility_id, inte upprepar stad/adress

Kontraktsinformation:

- consultant har kontraktsdatum
- consultant_company har företagsinfo
- Separerade för att undvika redundans

Betyg/Status:

- student_enrollment har betyg och status för en specifik kurs
- Separerat från generell student-status i student tabellen

Fördelar med 3NF-design:

- Minimal dataredundans. Varje faktum lagras på ett ställe
- Enkel underhåll. Uppdateringar görs på ett ställe
- Konsekvent data. Ingen risk för inkonsekventa uppdateringar
- Flexibel utbyggnad. Lätt att lägga till nya funktioner