



Normalisering

Vad är Normalisering?

Normalisering handlar om att strukturera sin databas smart - ungefär som att organisera sin kod i mindre, återanvändbara funktioner istället för att ha all kod i en enda stor funktion.

Varför Normaliserar vi?

För att slippa upprepa samma data på flera ställen (precis som DRY-principen i kod)

För att undvika buggar när data uppdateras

För att göra det enklare att ändra i databasen senare

Hur Ser Onormaliserad Data Ut?

Orders:

OrderID	Kund_Namn	Kund_Email	Produkt	Produkt_Prisk
1	Anna Ek	anna@mail.com	T-shirt	299
2	Anna Ek	anna@mail.com	Jeans	899
3	Bengt Skog	bengt@mail.com	T-shirt	299

Annas information upprepas

Om Anna byter email måste vi uppdatera på flera ställen

Produktpriset upprepas för varje t-shirt

Normaliserad Version

Kunder:

KundID	Namn	Email
1	Anna Ek	anna@mail.com
2	Bengt Skog	bengt@mail.com

Produkter:

ProduktID	Namn	Pris
1	T-shirt	299
2	Jeans	899

Orders:

OrderID	KundID	ProduktID
1	1	1
2	1	2
3	2	1

Fördelar med Normalisering

Uppdatera kundens email på ETT ställe

Ändra produktpris på ETT ställe

Mindre risk för fel

Sparar lagringsutrymme

När Behövs Normalisering?

När samma information upprepas på flera ställen

När data behöver kunna uppdateras enkelt

När du vill undvika att data blir inkonsekvent

Första Normalformen (1NF)

Varje cell ska bara innehålla ETT värde

Exempel på FEL:

UserID	Namn	Telefonnummer
1	Anna	070-123, 073-456

Exempel på RÄTT:

UserID	Namn	Telefon1	Telefon2
1	Anna	070-123	073-456
2	Bengt	070-789	null

Andra Normalformen (2NF)

All information ska vara kopplad till rätt sak. Exempel på FEL:

OrderID	Produkt	Pris	Kund_Namn	Kund_Adress
1	T-shirt	299	Anna	Storgatan 1
2	Jeans	899	Anna	Storgatan 1

Exempel på RÄTT:

Orders:

OrderID	ProduktID	KundID
1	1	1
2	2	1

Produkter:

ProduktID	Namn	Pris
1	T-shirt	299
2	Jeans	899

Tredje Normalformen (3NF)

Information som hör ihop ska ligga i samma tabell. Exempel på FEL:

OrderID	ProduktID	Land_Kod	Land_Namn
1	1	SE	Sverige
2	2	SE	Sverige

Exempel på RÄTT: Orders:

OrderID	ProduktID	Land_Kod
1	1	SE
2	2	SE

Länder:

Land_Kod	Land_Namn
SE	Sverige

Reglerna summerade

1NF: Ett värde per cell

2NF: Relaterad information i egna tabeller

3NF: Information som hör ihop ska vara i samma tabell

Tips för JavaScript-utvecklare

Tänk på normalisering när du designar din databas

Du behöver inte vara expert - fokusera på att undvika upprepningar

Använd JOIN i dina queries för att hämta relaterad data

Sammanfattning

Normalisering = Organisera data smart

Undvik upprepningar

Dela upp i logiska tabeller

Använd ID:n för att koppla ihop data