

Hotellapp – 3NF

Vad jag har gjort för att säkerställa att mina tabeller följer 3NF:s krav. Huvudanledningen till varför man ska följa 3NF, datanormalisering är för att datan ska vara organiserad och lättare att hantera. Det reducerar dubbel arbete, gör att allt är på sin plats och reducerar potentiella fel!

1NF: First Normal Form

I mina tabeller ser jag till att alla värden är enkla och atomära, det vill säga att varje ruta innehåller endast ett värde, vara unik och en tabell måste innehålla en "primary key". Till exempel har jag separerat "RoomSize" och "TypeOfRoom" i olika kolumner istället för att ha dem i samma. Samma är för "Address" där allt är uppdelat istället för att ha en stor lång adress med en massa olika data i.

2NF: Second Normal Form

För att uppfylla 2NF måste 1NF vara uppfylld.

2NF handlar om att ta bort "partial dependencies", alltså att ingen kolumn ska vara beroende av en del av "primary key". Jag har delat upp informationen i en separat "Address"-tabell, där exempelvis "Street", "City" och "PostalCode" är beroende av "primary key" "Id".

3NF: Third Normal Form

För att uppfylla 3NF måste både 1NF och 2NF vara uppfyllda.

3NF handlar om att ta bort "transitive functional dependencies", där en kolumn är beroende av en annan icke-nyckelkolumn. Till exempel, tidigare fanns adressinformationen i **Customer**-tabellen, men jag har nu flyttat den till en separat **Address**-tabell. På så sätt slipper vi duplicera adressen varje gång vi pratar om en kund.