# DATABASEOPGAVE

Database model:

Et billede, der indeholder tekst, skærmbillede, diagram, Font/skrifttype

Automatisk genereret beskrivelse

Relationernes Begrundelse:

**Hotel:**

* HotelID: Unikt ID for hvert hotel (auto-generate)
* Navn og Adresse: Information om hotellet.
* Bruges i Værelse og Hotelfacilitet for at forbinde med specifikke hoteller.

**Facilitet:**

* FacilitetID: Unikt ID for hver facilitet (auto-generate).
* Beskrivelse: Detaljer om faciliteten.
* Bruges i Hotelfacilitet for at identificere faciliteter på hoteller.

**Kunde:**

* KundeID: Unikt ID for hver kunde (auto-generate).
* Navn og Email: Information om kunden.
* Bruges i Reservation for at forbinde reservationer med kunder.

**Værelse:**

* VærelseID: Unikt ID for hvert værelse (auto-generate).
* HotelID: Refererer til det hotel, værelset tilhører.
* Nummer, Type, Pris: Detaljer om værelset.
* Bruges i Reservation for at forbinde reservationer med specifikke værelser.

**Reservation:**

* ReservationID: Unikt ID for hver reservation (auto-generate).
* KundeID og VærelseID: Forbinder reservationer med specifikke kunder og værelser.
* StartDato, SlutDato: Varighed af reservationen.

**Hotelfacilitet:**

* En tabel der forbindes Hotel og Facilitet.
* HotelID og FacilitetID: Viser hvilke faciliteter der findes på hvilke hoteller.

Redegør for 1-2/3je Normalform:

# Hvad er 3je NormalForm:

Den 3. normalform handler om at sikre, at alle data i en database er organiseret korrekt. Kort sagt, en tabel er i 3. normalform, hvis den allerede er i 2. normalform, og ydermere sikrer, at alle data i tabellen er direkte afhængige af hovednøglen (den unikke ”identifikator” for hver række). Det betyder, at der ikke er nogen data, som indirekte afhænger af nøglen gennem et andet felt. Dette hjælper med at forhindre unødige gentagelser og holder dataene rene og præcise

# Redegørelse for Databasen i 3je Normalform:

Databasen opfylder de tre normalformer, som kan ses ud fra følgende:

1. **Normalform (1NF):**

Databasen opfylder 1. normalform, da hver tabel har en unik primær nøgle (som HotelID i Hotel, FacilitetID i Facilitet), og hvert felt indeholder kun én slags information

**2. Normalform (2NF):**

Databasen er i 2. normalform, fordi den er i 1. normalform, og alle felter, der ikke er nøgler, relaterer direkte til den primære nøgle. I Kunde-tabellen, for eksempel, er både Navn og Email fuldt afhængige af KundeID.

**3. Normalform (3NF):**

Databasen opfylder 3. normalform, da den allerede overholder både 1. og 2. normalform, og alle ikke-nøgleattributter i hver tabel er afhængige udelukkende af deres respektive primære nøgler. For eksempel, i Hotel-tabellen, er Navn og Adresse direkte afhængige af HotelID, uden nogen transitiv afhængighed.

Samlet set sikrer overholdelse af disse normalformer, at databasen er velstruktureret, reducerer dataredundans og forbedrer dataintegriteten."

# Sekvensdiagram over ”tilføj hotel” metoden

Her vil jeg gerne vise en del af min kode der demonstrerer, hvordan metoden interagerer med databasen for at tilføje nye hotelinformationer

Sekvensdiagrammet:

Et billede, der indeholder tekst, skærmbillede, diagram, Font/skrifttype

Automatisk genereret beskrivelse

**Dette sekvensdiagram viser trinnene i processen, når en bruger tilføjer et nyt hotel via programmet:**

**Bruger Vælger Handling:**

* Brugeren starter processen ved at vælge at tilføje et nyt hotel.
* Programmet Sender Anmodning: Hovedprogrammet modtager brugerens valg og anmoder HotelDbContext om at tilføje det nye hotel.

**Database Behandling:**

* HotelDbContext håndterer anmodningen og tilføjer hotellet til databasen.

**Resultat Returneres til Bruger:**

\* Efter at hotellet er tilføjet, informerer programmet brugeren om, at handlingen er udført.

Dette diagram illustrerer interaktionen mellem brugeren, programmet og databasen i tilføjelsesprocessen.

**Eksempel på metodenEt billede, der indeholder tekst, skærmbillede, software

Automatisk genereret beskrivelse**