

En opsamling af artefakter for Hotel Databasen som REST-service

Bygger på Hotel opgaven i 8 trin

Trin 1: Lav en Domain model

Opgave beskrivelse - Scandic hotel – kæde

Lav en domain model af Hotel-kæden.

Der er flere hoteller. Hvert hotel har flere værelser.

Hotellets kunder kan reservere et eller flere værelser.

Angiv attributter for klasserne og associeringer mellem klasserne.

En Løsning



Trin 2: Lav en database model

Opgave beskrivelse Scandic hotel – kæde – database

Beskriv klasser, attributter og associering som tabeller med primær nøgler, søjler og fremmed nøgler

En løsning

Hotel

<u>Hotel_No (PK)</u>	Navn	Adresse
----------------------	------	---------

Guest

<u>Guest_No (PK)</u>	Navn	Adresse
----------------------	------	---------

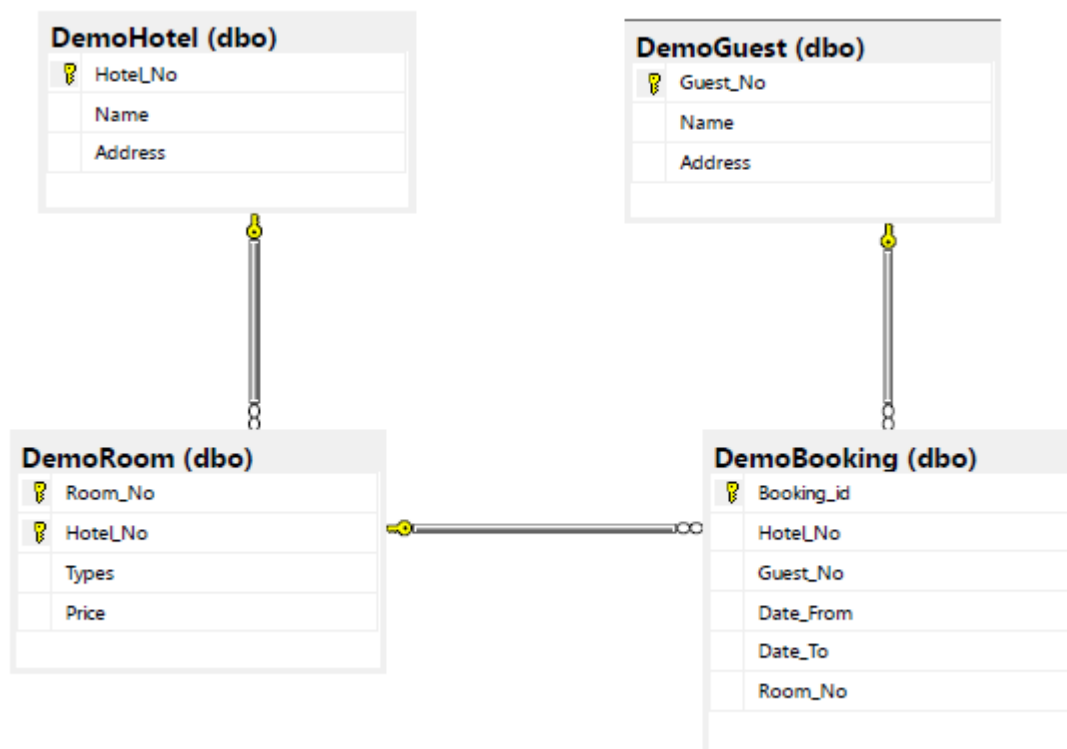
Room

<u>Room_No (PK)</u>	<u>Hotel_No (PK)(FK)</u>	Types	Price
---------------------	--------------------------	-------	-------

Booking

<u>Booking_id (PK)</u>	Hotel_No (FK)	Guest_No (FK)	Date_From	Date_To	Room_No (FK)
------------------------	---------------	---------------	-----------	---------	--------------

Eller som diagram (slags ER-Diagram)



Trin 3: Undersøg tabeller for om de er på 3NF

Opgave Beskrivelse Scandic hotel – kæde – Normalformer

Undersøg om tabellerne er på 3NF

En Løsning

1NF: tabellerne har en primær nøgle, samt kun én værdi i hver par (række, søjle)

Alle fire tabeller har en primær nøgle, og ingen tabel har flere værdier i samme celle (række, søjle)

2NF: Ved sammensatte primære nøgler må kun hele den primære nøgle kunne udpege de andre søjler.

Det er kun tabellen 'Room' der har en sammensat primær nøgle, de øvrige tre tabeller er derfor på 2NF. Room tabellen har Hotel_No og Room_No som sammensat primær nøgle, men hverken types eller Pris kan udpeges af Hotel_No eller Room_No alene, så 'Room' er også på 2NF

3NF: Ingen andre søjler end den primære nøgle må kunne udpege andre søjler.

Ved Hotel kan man ikke ud af navnet (Navn) udpege adressen (Adresse).

Ved Guest kan man ikke ud af navnet (Navn) udpege adressen (Adresse).

Ved Room kan man ikke ud af typen (Types) udpege prisen (Pris), det kunne have været sådan at alle Single-room havde samme pris ligesom double-room og Family-room og så ville typen have udpeget prisen, hvorved der skulle have været en speciel tabel til typer og priser for et specielt hotel.

Ved Booking hverken Hotel_no, Guest_No, Room_No kan udpege hinanden eller Date_From eller Date_To.

Alle fire tabeller er derfor på 3NF.

Trin 4: Opret tabellerne i en Database

Opgave Beskrivelse: Scandic hotel – kæde – fortsat

De tabeller du har defineret og undersøgt i Trin 3 skal du oprette som tabeller i databasen.

se (Fronter) evt: database-note kapitel 3 eller

https://www.w3schools.com/sql/sql_create_table.asp

En Løsning

Tabellerne er navngivet med demo for ikke at falde sammen med tabeller de Studerende allerede selv har lavet.

```
CREATE TABLE DemoHotel(  
    Hotel_No int NOT NULL PRIMARY KEY,  
    Name VARCHAR(30) NOT NULL,  
    Address VARCHAR(50) NOT NULL  
);  
  
CREATE TABLE DemoRoom(  
    Room_No int NOT NULL,  
    Hotel_No int NOT NULL,  
    Types CHAR(1) DEFAULT 'S',  
    Price FLOAT,  
    CONSTRAINT checkType  
    CHECK (Types IN ('D','F','S') OR Types IS NULL),  
    CONSTRAINT checkPrice  
    CHECK (price BETWEEN 0 AND 9999),  
  
    FOREIGN KEY (Hotel_No) REFERENCES DemoHotel (Hotel_No)  
    ON UPDATE CASCADE ON DELETE NO ACTION,  
  
    Primary KEY (Room_No, Hotel_No)  
);  
  
CREATE TABLE DemoGuest (  
    Guest_No int NOT NULL PRIMARY KEY,  
    Name VARCHAR(30) NOT NULL,  
    Address VARCHAR(50) NOT NULL  
);  
  
CREATE TABLE DemoBooking(  
    Booking_id int IDENTITY(1,1) NOT NULL PRIMARY KEY,  
    Hotel_No int NOT NULL,  
    Guest_No int NOT NULL,  
    Date_From DATE NOT NULL,  
    Date_To DATE NOT NULL,  
    Room_No int NOT NULL,  
  
    FOREIGN KEY (Guest_No) REFERENCES DemoGuest (Guest_No),  
    FOREIGN KEY (Room_No, Hotel_No) REFERENCES DemoRoom (Room_No, Hotel_No)  
);  
  
ALTER TABLE DemoBooking  
    ADD CONSTRAINT incorrect_dates  
    CHECK ((Date_To > Date_From) AND (Date_From <= '2018-04-04'));
```

Trin 5: Udtræk data

Del A – se trin 4

Del B - select fra een tabel

1. List alle oplysninger om alle hoteller.
select * from DemoHotel
2. List alle oplysninger om alle hoteller i Roskilde.
select * from DemoHotel
where Address like '%Roskilde%'

- List navne og adresser på alle gæster fra Roskilde.
`select Name, Address from DemoGuest
where Address like '%Roskilde%'`
- List navne og adresser på alle gæster fra Roskilde sorteret alfabetisk efter navn.
`select Name, Address from DemoGuest
where Address like '%Roskilde%'
order by Name ASC`
- List alle dobbeltværelser med en pris under 200 pr. nat.
`select * from DemoRoom
where Types='D' and Price < 200`
- List alle dobbeltværelser eller familierum med en pris under 400 pr. nat.
`select * from DemoRoom
where Types='D' or Types='F' and Price < 400`
- List alle dobbeltværelser eller familierum med en pris under 400 pr. nat
sorteret i stigende orden efter pris.
`select * from DemoRoom
where Types='D' or Types='F' and Price < 400
order by Price`
- List alle gæster, som har et navn, der starter med 'G'.
`select * from DemoGuest
where Name like 'G%'`

Del C - Aggregate-funktioner

- Hvor mange hoteller er der?
`select count(*) from DemoHotel`
- Hvor mange hoteller er der i Roskilde?
`select count(*) from DemoHotel
where Address like '%Roskilde%'`
- Hvad er gennemsnitsprisen på et værelse?
`select avg(Price) from DemoRoom`
- Hvad er gennemsnitsprisen på et enkeltværelse?
`select avg(Price) from DemoRoom
where Types='S'`
- Hvad er gennemsnitsprisen på et dobbeltværelse?
`select avg(Price) from DemoRoom
where Types='D'`
- Hvad er gennemsnitsprisen på et værelse på Hotel Scandic?
`select avg(Price) from DemoRoom
where Hotel_No = (select Hotel_No from DemoHotel where Name like 'Scandic')`

7. Hvad er den totale indtægt pr. nat for alle dobbeltværelser?

```
select sum(price) from DemoRoom
where Types='D'
```
8. Hvor mange forskellige gæster har foretaget bookinger i marts måned?

```
select count (DISTINCT Guest_No) from DemoBooking
where '2011-03-01' <= Date_From and Date_From <= '2011-03-31'
```
9. Hvor mange bookinger er der i dag på Scandic hotel?

```
select count (*) from DemoBooking
where GETDATE() = Date_From
```
10. Hvor mange bookinger er der i morgen på Scandic hotel?

```
select count (*) from DemoBooking
where (GETDATE()+1) = Date_From
and Hotel_No = (select Hotel_No from DemoHotel where Name like 'Scandic')
```

Del D - Flere tabeller

1. List pris og type på alle værelser på 'Prindsen'.

```
select * from DemoRoom r, DemoHotel h
where h.Name like '%Prindsen%'
and h.Hotel_No = r.Hotel_No
```
2. List alle gæster, der p.t. bor på 'Prindsen'.

```
select g.* from DemoBooking b, DemoHotel h, DemoGuest g
where h.Name like '%Prindsen%'
and h.Hotel_No = b.Hotel_No
and b.Guest_No = g.Guest_No
and GETDATE() = Date_From
```

Gruppering

3. List antal værelser for hvert hotel.

```
select count(*), h.Name from DemoRoom r, DemoHotel h
where r.Hotel_No = h.Hotel_No
group by h.Name
```
4. List antal værelser for hvert hotel i Roskilde.

```
select count(*), h.Name from DemoRoom r, DemoHotel h
where r.Hotel_No = h.Hotel_No
and h.Address Like '%Roskilde%'
group by h.Name
```
5. Hvad er det gennemsnitlige antal bookinger for hvert hotel i denne måned?

```
select avg(r.Price) as 'gns-pris', h.Name from DemoRoom r, DemoHotel h, DemoBooking b
where r.Hotel_No = h.Hotel_No and b.Hotel_No = h.Hotel_No
and month(getdate()) = month(b.Date_From)
group by h.Name
```

Del E - Insert og Update af Tabeller

1. Indsæt nye gæster.
insert into DemoGuest (Guest_No,Name,Address)
Values (123,'Peter', 'Alleen 11B, 4180 Sorø')
2. Indsæt et nyt hotel.
insert into DemoHotel (Hotel_No, Name, Address)
Values (9, 'PetersRest', 'somewhere in Roskilde')
3. Indsæt værelser i det nye hotel.
insert into DemoRoom (Room_No, Hotel_No, Types, Price) Values (111, 9, 'D', 345)
4. Indsæt bookinger.
insert into DemoBooking (Hotel_No, Guest_No, Room_No, Date_From, Date_To)
Values (9, 123, 111, GETDATE(), GETDATE()+2)
5. Forøg prisen på alle værelser med 5%.
Update DemoRoom
set Price = Price * 1.05

Trin 7 Simpel DB-Forbindelse Til Hotel Databasen

Opgave Beskrivelse Lav en almindeligt Konsol-applikation.

Lav en klasse DBClient med en metode:

```
public void Start()
```

I main lav et object af DBClient og kald Start()-metoden.

En Løsning

Se Github: <https://github.com/ro17da2a1-1a/ClassDemoDBConnection>

Trin 8 Opbygning Af En REST-Service

Opgave Beskrivelse NU er det tid til at lave en egentlig REST-Service.

Til det skal du have model-klasser, en controller samt selve REST-servicen.

En Løsning

Model klasser Se Github: <https://github.com/ro17da2a1-1a/HotelLibrary>

RestService Se GitHub: <https://github.com/ro17da2a1-1a/RestServiceHotel>

Servicen kan kaldes på <http://resthotel-pele-easj.azurewebsites.net/>

Fx. (GET) <http://resthotel-pele-easj.azurewebsites.net/api/Guest/4>

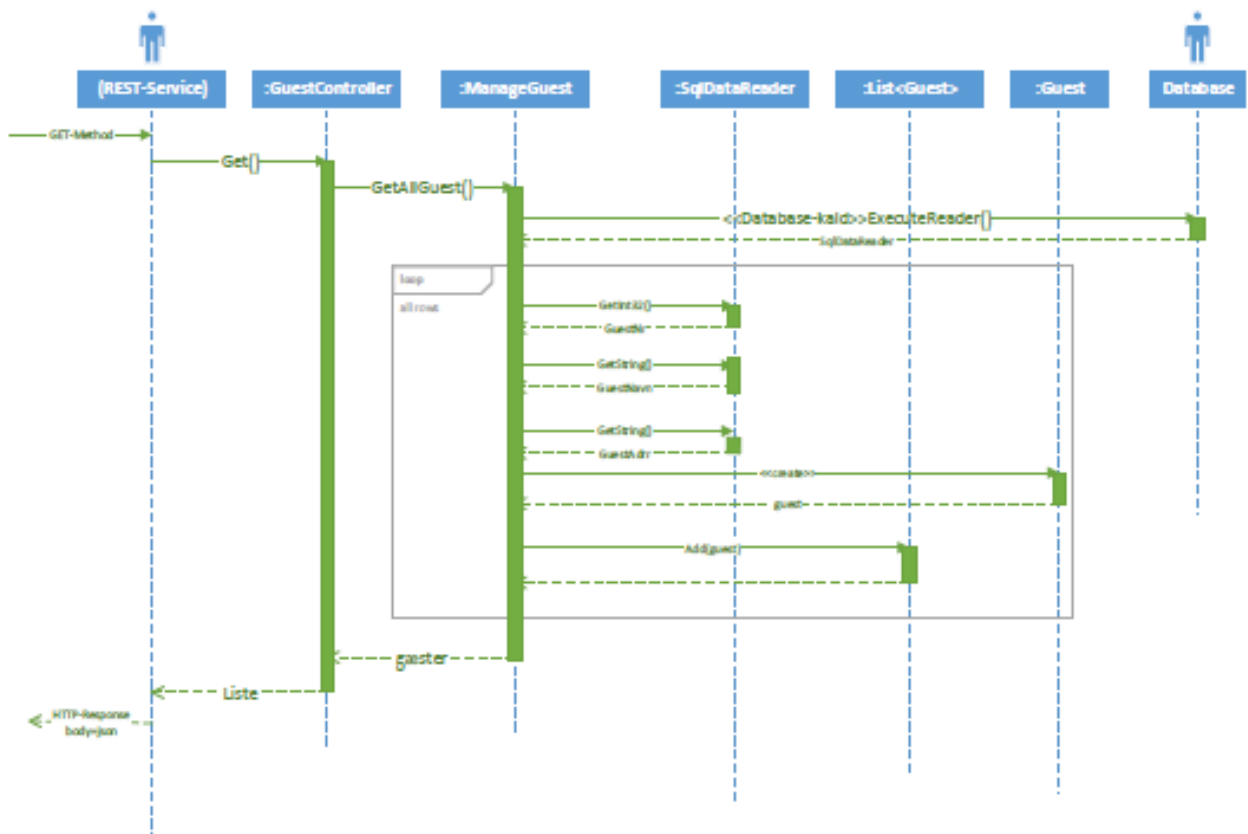
Trin 9 Udarbejd 'design' Sekvens Diagram og Design Klasse Diagram for REST-servicen

Opgave Beskrivelse Lav et DSD og et DCD for REST servicen

For at forstå designet af REST-service systemet (der bliver et support system ('supporting actor')) skal du udarbejde et 'design' Sekvens Diagram (SD) samt et Design Klasse Diagram (DCD).

En løsning

Et eksempel på sekvens diagrammet – for hent alle gæster.



Kommentarer:

Egentlig skulle List<Guest> oprettes (<<create>>) inden kald af databasen.

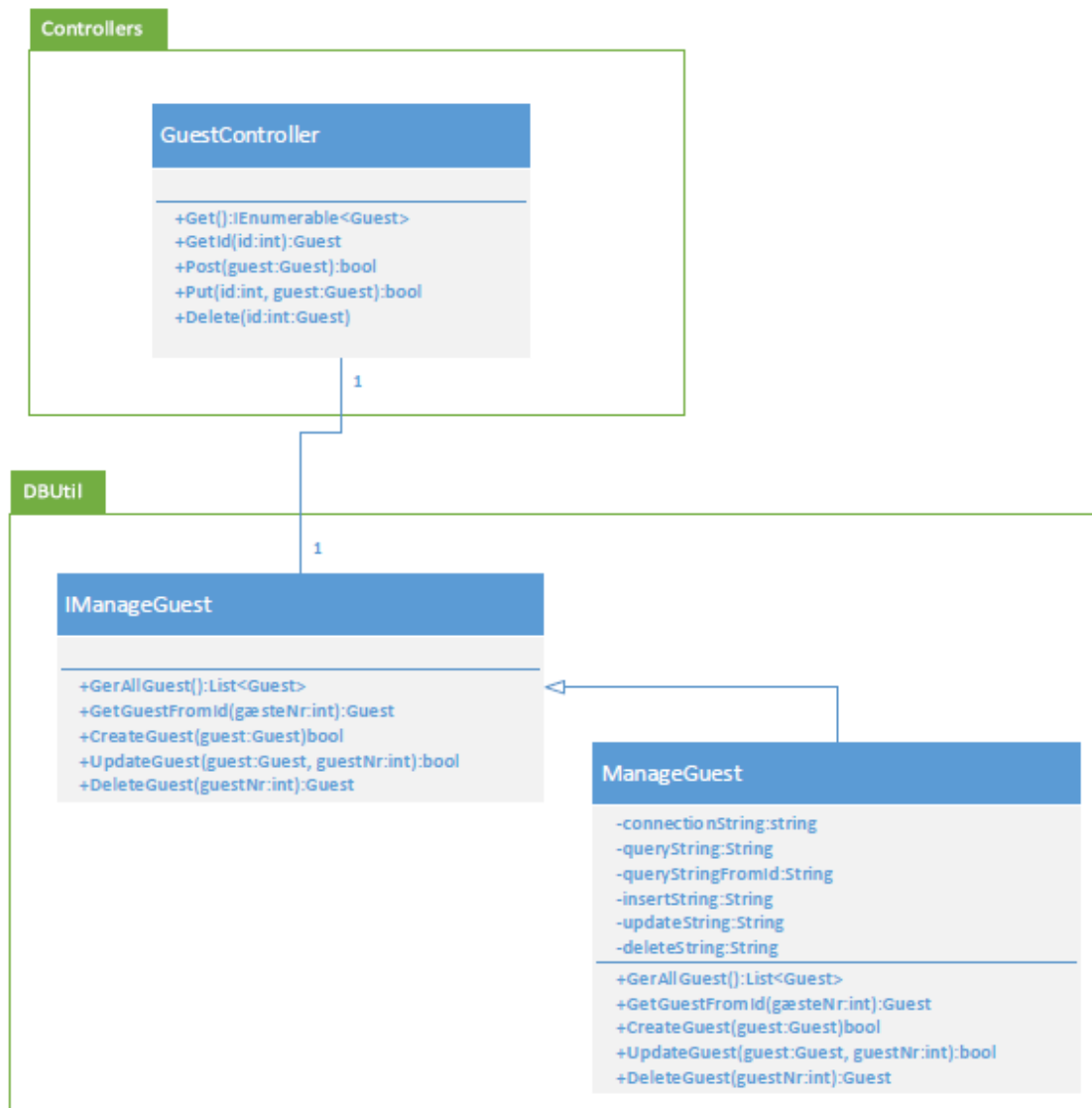
Egentlig skulle Database-kald være

1) opret en SqlConnection (<<create>>)

2) Opret en SqlCommand (<<create>>)

Inden selve ExecuteReader kaldet udføres.

Et eksempel på klasse diagrammet- for CRUD på gæster



Kommentarer

Model klasserne `Hotel`, `Room`, `Guest`, `Booking` findes i et særskilt library-projekt, minder om Domain modellen.

Trin 10 Et program til Hotel REST-service

Opgave Beskrivelse - Lav et program der konsumere Hotel REST-service

Du skal lave et program der kan konsumere din REST-service. Dvs. du skal kunne hente alle gæster, en specifik gæst, indsætte gæster, slette gæster og opdatere gæster.

En Løsning

Se Github: <https://github.com/ro17da2a1-1a/DemoClassConsumeREST>