

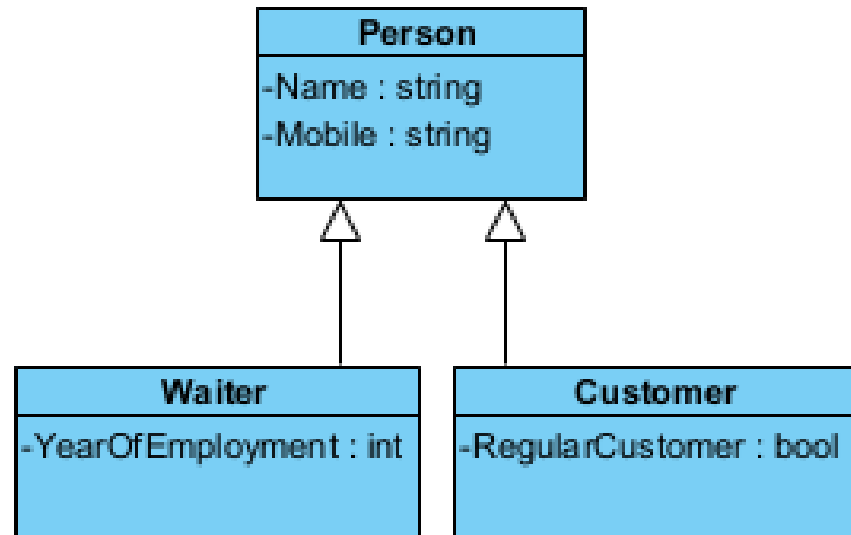
Database sjob

Peter Levinsky, IT Roskilde

21.03.2024

Hvordan håndterer jeg arv i en database

- Hvordan håndterer jeg følgende klasse diagram?



Hvordan håndterer jeg arv i en database

Løsning 1 : En tabel

Person

<u>Mobile</u>	Name	YearOfEmployment	RegularCustomer
---------------	------	------------------	-----------------

- Alle properties fra de tre klasser samlet I en tabel
- YearOfEmployment & RegularCustomer skal være mulighed for NULL
- **Fordel** alle objekter er i samme tabel, (lettere at lave id – gælder ikke i dette eksempel)
- **Ulempe** tager mere plads fx Waiter række har også RegularCustomer
Kan bliver uoverskueligt ved flere klasser og et større arvehierarki

Hvordan håndterer jeg arv i en database

Løsning 2 : To Tabeller

Waiter

<u>Mobile</u>	Name	YearOfEmployment
---------------	------	------------------

Customer

<u>Mobile</u>	Name	RegularCustomer
---------------	------	-----------------

- Der laves en tabel for hver af de specialiserede klasser (waiter+Customer)
- **Fordel** de tager kun den plads der er brug for (ikke kolonner der er NULL)
- **Ulempe** Det er svært at styre en fælles automatisk ID tildeling
Det er svært (læs tid) at få alle objekter fra begge tabeller

Hvis der skal ændres i typer i base-klassen (Person) eller tilføjes nye properties, så skal hver af tabellerne opdateres

Hvordan håndterer jeg arv i en database

Løsning 3 : Tre Tabeller



- Der laves en tabel for hver af klasserne (Person+Waiter+Customer)
- **Fordel** de afspejler klasse modellen og benytter kun lidt mere plads end løsning 2, du kan lave ID generering
- **Ulempe** Det tager tid at få oplysninger om et objekt (waiter eller Customer) fordi der skal laves en join mellem to tabeller

Andet Database sjob

- View
- Referential integrity

Database sjov - View

- Du kan lave et view af tabeller, som så i sig selv kan læses, sorteres grupperes, søges mm.
- Eksempel (Running - Member)

```
CREATE VIEW BlueTeam AS  
SELECT * FROM Member  
WHERE Team = 'blå';
```
- Nu kan jeg så lave en select på view

```
Select * From BlueTeam where name like '%pe%'
```
- Se i øvrigt https://www.w3schools.com/sql/sql_view.asp

Database sjov - Referential integrity

- Når du laver ForeignKey kan du angive hvad der skal ske hvis noget sker med den PrimaryKey, som den peger på
- Eksempel (inspireret fra hotel-opgaven)
Booking har en ForeignKey til Guest

FOREIGN KEY (GuestID) REFERENCES Guest(ID)

On Delete Cascade | No Action | SET Default | SET NULL

On Update Cascade | No Action | SET Default | SET NULL

- Se i øvrigt https://www.w3schools.com/sql/sql_foreignkey.asp