Database Design

Øvelse 1: Terminologi

Forventning: At forstå og skelne mellem grundlæggende begreber som data og information, samt at kunne forklare deres forskelle og sammenhænge. Derudover at forstå og kunne forklare, hvad en database og et relationelt database management system (R-DBMS) er, samt deres rolle i datahåndtering.

*Tidsramme: **45** minutter*

Øvelse 1.1: Data og Information

- **Hver studerende skal give minimum 2 eksempler på data og information fra jeres tidligere projekter eller fra virkeligt liv.**
  TheMovies:
  Data: CSV-filer med rå data.
  Information: En organiseret plan, der kombinerer filmdetaljer og detaljer for den specifikke forestilling.

  Data: Websidetrafik
  Information: Analyse på den trafik. Eks. Ændring i besøgstal.

- **Brug Ordet rundt til at reflektere over begreberne data og information.**

- **Forklar forskellen mellem data og information.**
  Data er rå fakta
  Information er struktureret og organiserede data, som præsenteres i en kontekst, der gør den relevant og nyttig.

- **Diskutér jeres refleksioner over begreberne til at kvalitetssikre jeres svar ud fra de I har læst i læringsobjektet om Database design.**

Øvelse 1.2: Database og R-DBMS

Brug **Ordet rundt** til at reflektere over begreberne:

- **Skriv en kort definition af, hvad en database er og beskriv et eksempel på en database, du bruger i din dagligdag (f.eks. en musiktjeneste, et skolesystem osv.).**
  Definition:
  Et sted hvor data bliver opbevaret og kan tilgås
  Eksempel:
  Spotify's back-end
  Github

- **Skriv en kort definition af, hvad et R-DBMS er.**
  Et system der tillader at arbejde med relationelle databaser.
  Et system, hvor data er struktureret i tabeller(relationer). Man kan oprette, opdatere, styre og forespørge deisse realtionelle databaser vha. SQL

- **Diskutér, hvorfor R-DBMS systemer er vigtige og forklar deres fordelene.**
  Fordi det giver bedre data integritet og sikkerhed og tillader at lave operationer/transformationer af data.
  Nemt at ændre. Kan håndtere store mængder af data.

- **Forklar ”Tabel”, ”Relation”, ”Relationsskema (en: Relation schema)”, ”Relationelt databaseskema (en: Relational Database schema)”**
  Tabel: En tabel er en struktur i en relationel database, hvor data gemmes. Tabellen består af rækker (tuples) og kolonner (attributter), hvor hver række repræsenterer en unik post, og hver kolonne repræsenterer en type data.
  Relation: En relation svarer til en tabel, der opfylder et bestemt skema.
  Relationsskema (Relation schema): Et relationsskema definerer strukturen af en relation. Det omfatter en liste over attributter (kolonner).
  Relationelt databaseskema (Relational Database schema): Et relationelt databaseskema er en samling af relationsskemaer, der beskriver strukturen og organisationen af en hel relationel database.

  Forklar ”Primærnøgle (en: Primary Key)”, ”Fremmednøgle (en: Foreign key)”, ”Sammensat nøgle (en: ’compound key’ eller ’composite key’)”, ”Kandidatnøgle (en: Candidate key)”.
  Primærnøgle (Primary Key): En primærnøgle er en kolonne i en tabel, der entydigt identificerer hver række i tabellen. Der kan kun være én primærnøgle i hver tabel, og den skal indeholde unikke værdier. (Typisk et ID).
  Fremmednøgle (Foreign Key): En fremmednøgle er en kolonne eller et sæt af kolonner i en tabel, der refererer til en primærnøgle i en anden tabel. Fremmednøgler bruges til at skabe relationer mellem tabeller.

  Sammensat nøgle (Compound Key/Composite Key): En sammensat nøgle er en primærnøgle, der består af to eller flere kolonner. Disse kolonner sammen sikrer, at hver række er unik.

  Kandidatnøgle (Candidate Key): En kandidatnøgle er en kolonne eller et sæt af kolonner, der kunne være en primærnøgle. Det er en unik identifikator for rækker i tabellen. En tabel kan have flere kandidatnøgler, men kun én af dem vælges som primærnøgle.

- **Forklar ”En til mange (1:M)”, ”En til en (1:1)”, ”Mange til mange (M:M)” relationer**

- **En tabel skal opfylde nogle karakteristika, for at betegnes som en relation. Beskriv kort de karakteristika?**

Regel	Beskrivelse af karakteristika
1	Hver række i tabellen indeholder data (helt eller delvist) om en bestemt entitet
2	Hver kolonne i tabellen indeholder data om en bestemt attribut (egenskab) ved entiteten
3	En celle i en tabel må kun indeholde en enkelt værdi
4	Alle værdier i celler under en bestemt kolonne skal være af samme type
5	Hver kolonne skal have et unikt navn
6	Rækkefølgen af kolonner har ingen betydning
7	Rækkefølgen af rækker har ingen betydning
8	To rækker må ikke have identiske sæt af værdier i dets celler

Øvelse 2: Fra Domænemodel til UML-Databasemodel (Webshop case)

Forventning: At lære, hvordan man kan designe en UML-databasemodel baseret på en domænemodel, ved at gennemgå webshopcasen.

*Tidsramme: **60** minutter*

Aftal med minimum to andre teams, hvornår I videndeler, diskuter og sammenlign jeres individuelle UML-databasemodeller samt hvordan I gør dette.

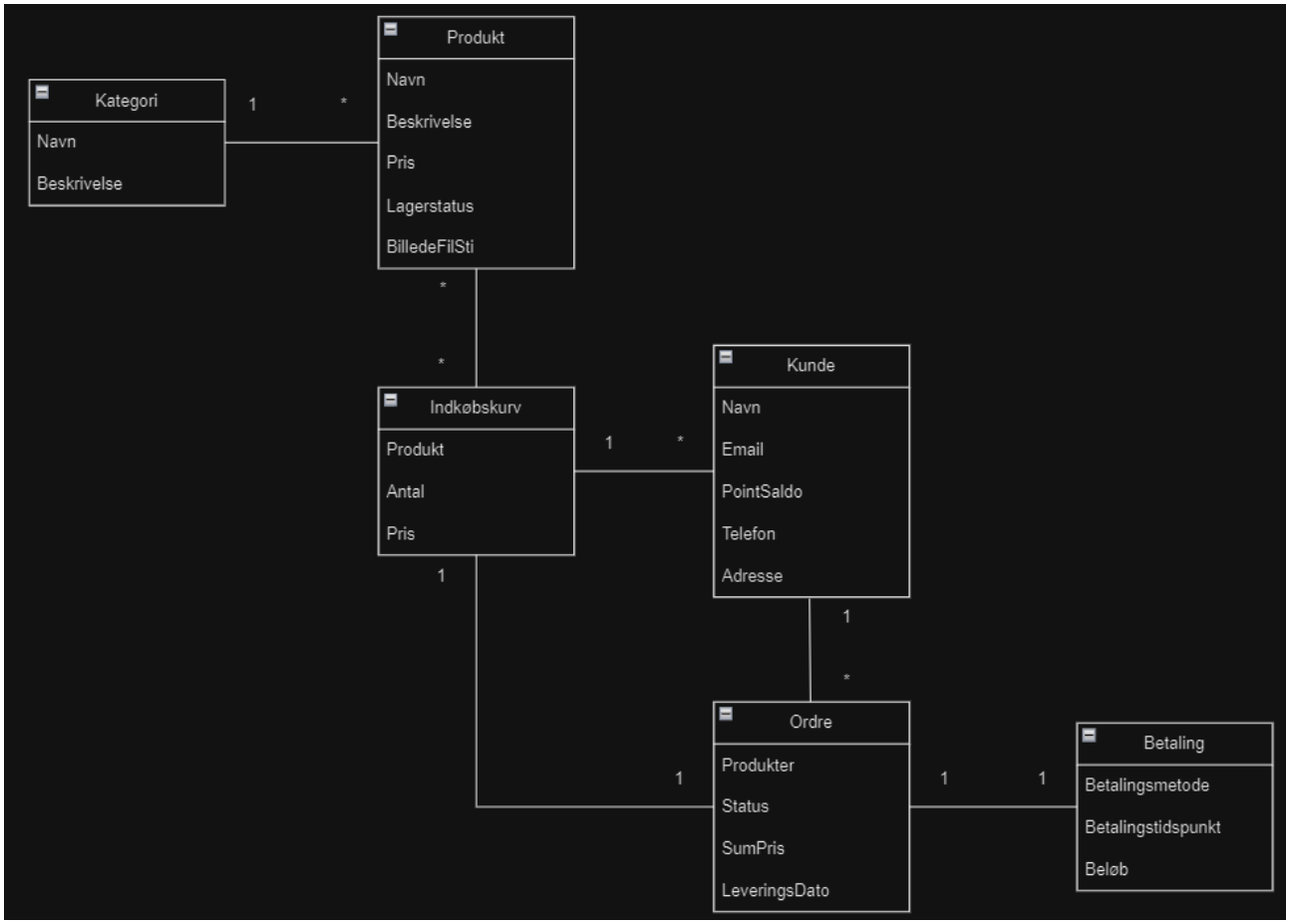
Case:
https://scorm.itslearning.com/data/3289/C20150/ims_import_11/scormcontent/index.html#/lessons/WyC42RT_EpgrtTCVfYv9KwLI46fRPpYf

Draw.io:

https://app.diagrams.net/#Wb!2ZvumUSy5kaTxX9DdntarkwBS9uu9EFMqt0PTZeH00WpAHjXfAitRIPCBFyVKFFi%2F01U5PXUQIMSNVYOZYHXVCI6H3LADNWLDQZ#%7B%22pageId%22%3A%224YYs4n2Sa3wD8fY8W-fy%22%7D

Øvelse 2.1: Etabler domæneforståelse gennem Domænemodellen

Lav en domænemodel til casen, til at analysere casen og etabler domæneforståelse. (Vi ved godt at dette trin er en del af systemudviklingsfag, men formålet er at du vaner sig til at benytte den iterative tilgang også i programmering).



Øvelse 2.2: Fra Domænemodellen til relationelsskemaer

Individuelt opret relationsskemaer for hver af de konceptuelle klasser i domænemodellen. (Hver studerende opretter en eller to relationsskemaer). Sørg for at inkludere primærnøgler (PK) og fremmednøgler (FK) i skemaerne. Husk UML-stereotyper <<table>>, <<PK>>, <<surrogate>>.

CUSTOMER(CustomerID, E-mail, PhoneNumber, Adress, PointSaldo, DateOfCreation)
ORDER(OrderID, TotalPrice, PointsUsed, DateOfCreation, DeliveryStatus)
PRODUCT(ProductID, Name, Description, Price, InStock, ImagePath)
PAYMENT(PaymentID, PaymentMethod, PaymentDate, PaymentAmount)
POINTTRANSACTION(TransactionID, PointsEarned, PointsSpend, TransactionDate)
ORDERLINE(OrderLineID, Amount, SubTotal)
CATEGORY(CatagoryID, Name, Description)

Øvelse 2.3: Fra relationelsskemaer til relationelt databaseskema

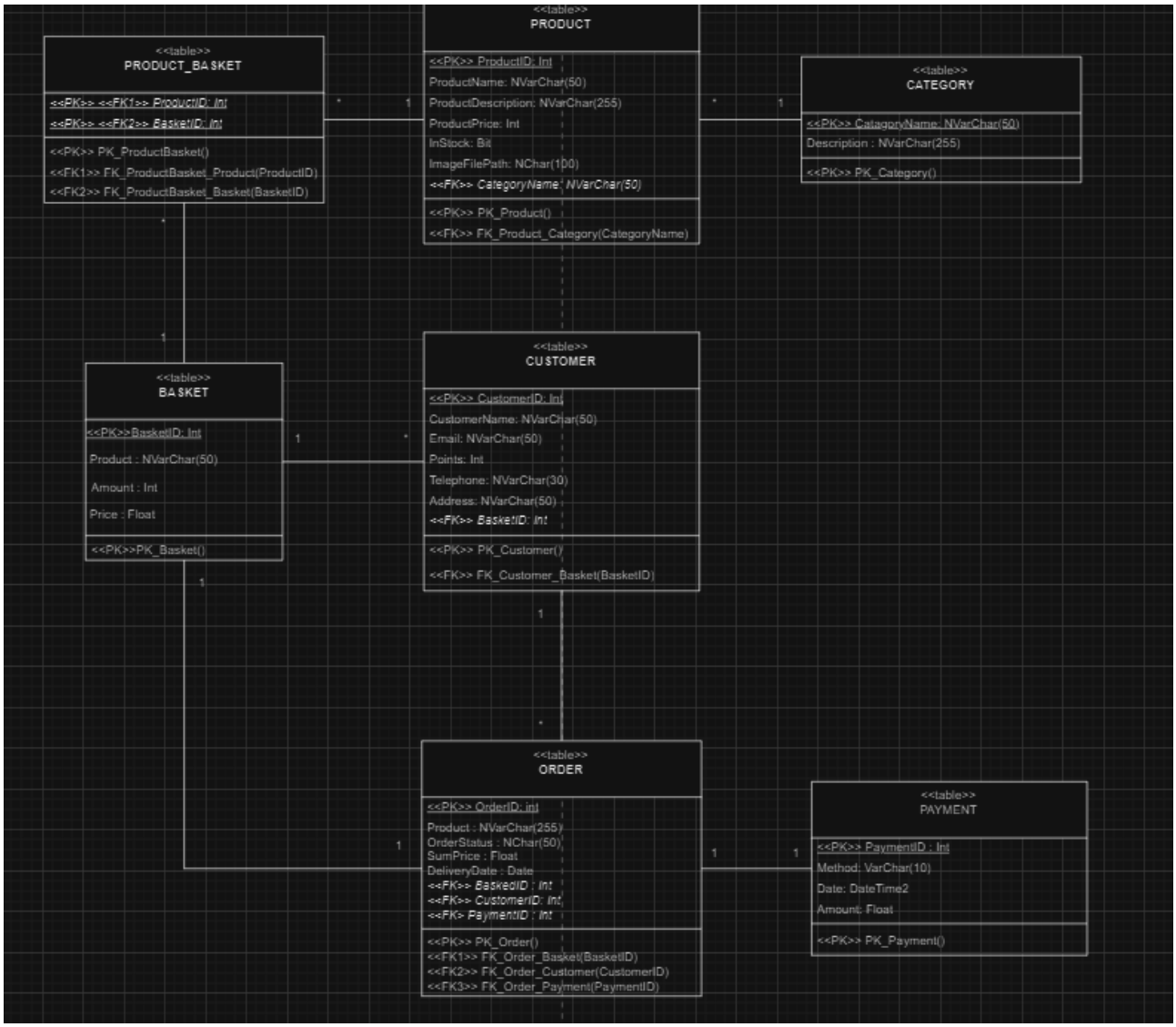
Som team identificer relationerne mellem de relationsskemaer og dan et relationelt databaseskema.

CATEGORY (CategoryName, Description)
PRODUCT(ProductID, ProductName, ProducDescription, ProductPrice, InStock, ImageFilepath, *CategoryName*)
PRODUCT_BASKET(*ProductID*, *BasketID*)
BASKET(BasketID, Product, Amount, Price)
CUSTOMER (CustomerID, CustomerName, Email, Points, Telephone, Address, *BasketID*)

ORDER (OrderID, Product, OrderStatus, SumPrice, DeliveryDate, *BasketID*, *CustomerID*, *PaymentID*)
PAYMENT(PaymentID, Method, PaymentTime, Amount)

Øvelse 2.4: Fra relationelt databaseskema til UML-database model

Nu, hvor du har fået styr på det relationelle databaseskema, så er du og dit team klar til at udarbejde en databasemodel. Opret UML-Databasemodel og brug et værktøj på dit eget valg til at tegne UML-databasemodellen baseret på de definerede relationsskemaer. Husk mappingen mellem datatyperne i .Net (C#) og SQL Server Database Engine.



Øvelse 3: Vidensdeling kl. 11:15

Forventning: At diskutere og sammenligne jeres individuelle UML-databasemodeller.

- Diskuter og sammenlign jeres individuelle UML-databasemodeller med de andre grupper, som I har aftale med i Øvelse 2.

- Lav en samlet version af UML-databasemodellen for webshoppen baseret på jeres diskussioner.
- Vælg en gruppe, der fremviser jeres fælles UML-databasemodellen i vidensdeling kl. 14:30.

*Tidsramme: **15** minutter*

Øvelse 4: Fra domænemodel til Modellaget i C# (Webshop case)

Forventning: At implementere modellaget i en C#-applikation baseret på en domænemodel ved at følge en agil udviklingsproces. Denne opgave indeholder planlægning, oprettelse af projektstruktur, implementering af modellaget, samt diskussion og kvalitetssikring.

*Tidsramme: **60** minutter*

Øvelse 4.1: Planlægning af løsning

Som gruppe udfør følgende:

- Diskuter, hvordan WPF (Windows Presentation Foundation) kan bruges til at skabe en brugergrænseflade for webshoppen.
- Forklar, hvordan MVVM-mønstre kan anvendes i jeres case.
- Dokumenter planen for projektets opbygning og hvilke komponenter der skal implementeres.
- Hvordan I vil organisere arbejdet og dele opgaverne blandt teammedlemmerne.

Øvelse 4.2: Projektstruktur

Som gruppe udfør følgende:

- Opret en ny løsning i Visual Studio kaldet "Webshop".
- Tilføj følgende mapper til løsningen: View, Model og ViewModel.
- Opret et nyt repository på GitHub kaldet "Webshop".
- Tilføj alle teammedlemmer som samarbejdspartnere til repository'et.
- Initialiser Git i jeres løsning og skub den til GitHub.

Øvelse 4.3: Implementering af Modellaget

Hvert teammedlem vælger en eller to klasser fra domænemodellen og implementerer dem i Model laget.

Øvelse 4.4: Diskussion og Kvalitetssikring

- Hvert teammedlem præsenterer deres implementerede klasser for de andre i gruppen.
- Diskuter kodekvalitet, følg kodestandarder, og identificer eventuelle forbedringsområder.
- Gennemgå og test de implementerede klasser for at sikre, at de fungerer som forventet. (Her kan I lave unittests, hvis I har nok tid til det).
- Lav eventuelle nødvendige ændringer baseret på feedback fra teammedlemmerne.

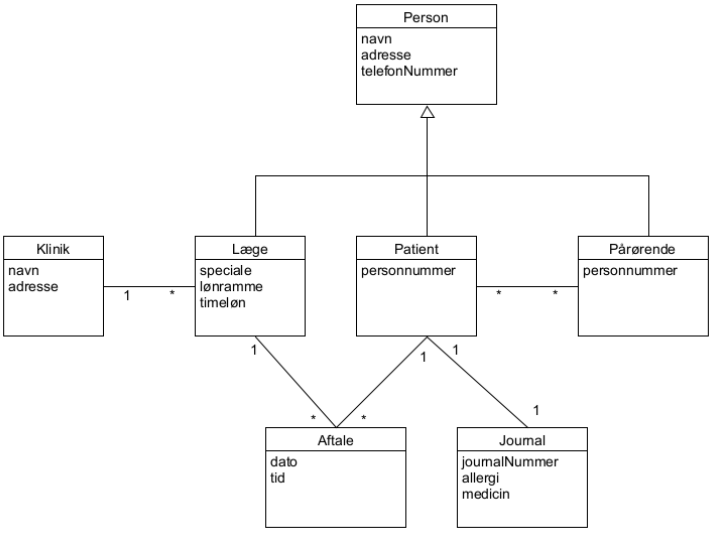
Øvelse 5: Fra domænemodel til UML-Databasemodel (Klinik case)

Forventning: At øve dig på at designe en UML-databasemodel baseret på en domænemodel.

*Tidsramme: **45** minutter*

Nu er det tid til at se på et lidt mere komplekst eksempel. Benyt den CL-struktur, du og dit team finder bedst egnet.

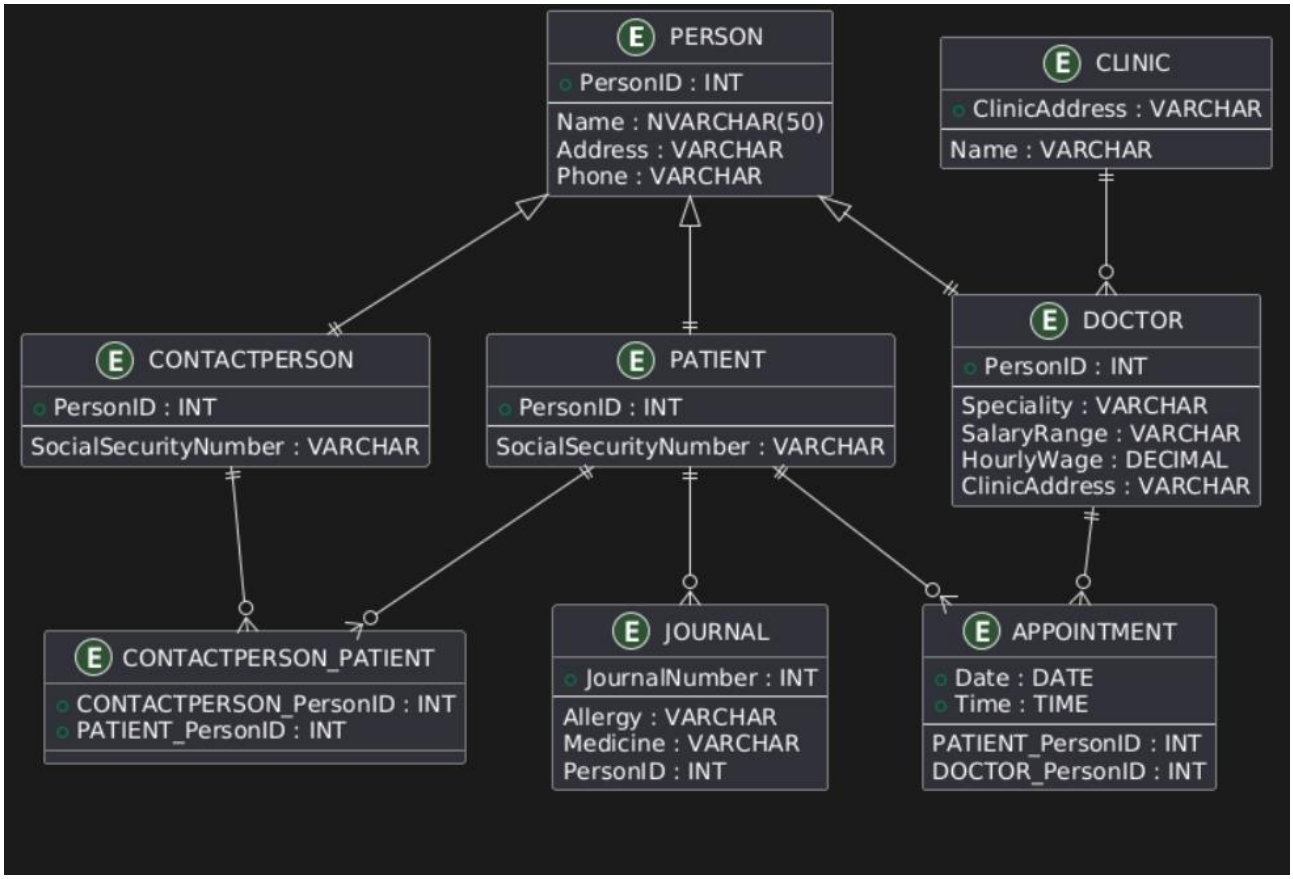
Med udgangspunkt i fremgangsmåden angivet i forberedelsen skal du nu udarbejde et relationelt databaseskema og UML-databasemodel ud fra nedenstående domænemodel:



- Beskriv alle tabeller, der repræsenterer ovenstående model
- For hver tabel, beskriv alle dets attributter og opret relationsskemaer for hver af de konceptuelle klasser i domænemodellen.
- Angiv primærnøgler og fremmednøgler til at repræsentere alle relationer mellem tabeller.
- Udarbejd en databasemodel og brug et værktøj på dit eget valg til at tegne UML-databasemodellen baseret på de definerede relationsskemaer.

Brug tekstnotationen, som angivet i forberedelsen til at repræsentere hver enkelt tabel.

PERSON(PersonID, Name, Address, Phone)
CONTACTPERSON(*PersonID*, SocialSecurityNumber)
CONTACTPERSON_PATIENT(*CONTACTPERSON_PersonID*, *PATIENT_PersonID*)
PATIENT(*PersonID*, SocialSecurityNumber)
DOCTOR(*PersonID*, Speciality, SalaryRange, HourlyWage, *ClinicAdress*)
JOURNAL(JournalNumber, Allergy, Medicine, *PersonID*)
CLINIC (ClinicAddress, Name)
APPOINTMENT(Date, Time, *PATIENT_PersonID*, *DOCTOR_PersonID*)



Øvelse 6: Vidensdeling kl. 14:30

Forbered og fremvis en præsentation af jeres UML-databasemodeller for klassen. I Øvelse 3 har I valgt en gruppe til fremvisningen. *Tidsramme: **30** minutter*