

Immigration Database.

Database Modulopgave 1. Semester 2.



(Source in bottom)

Group 5: Andreas L, Andreas B, Søren, Wajid.

Class: Dat18c.

Data: 12 - 02 - 2019. (DD/MM/YYYY)

Content

Immigration Database.	1
Content	2
Immigration database.	3
Data	3
Domain model (Udkast 1)	4
Domain model (Udkast 2)	4
Domain model (Udkast 3)	5
ER (Befolkning)	5
ER (Fra / Til)	6
ER (Til / Fra)	7
Layers	8
Report	8
Diskussion	9
Konklusion	9
Sources.	9

Immigration database.

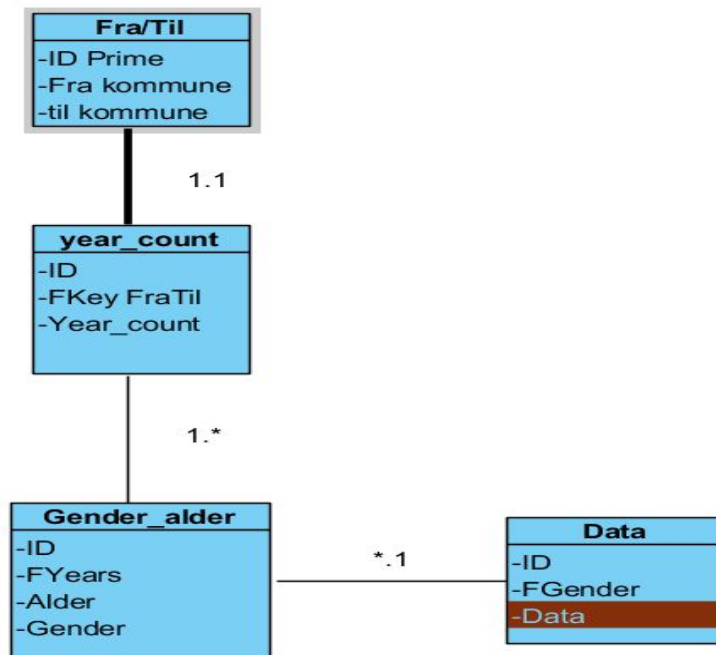
Immigrations database er en opgave i at lave en database for at holde øje med befolknings flytning fra de danske byer København og Frederiksberg ved hjælp af en sql database.

Data

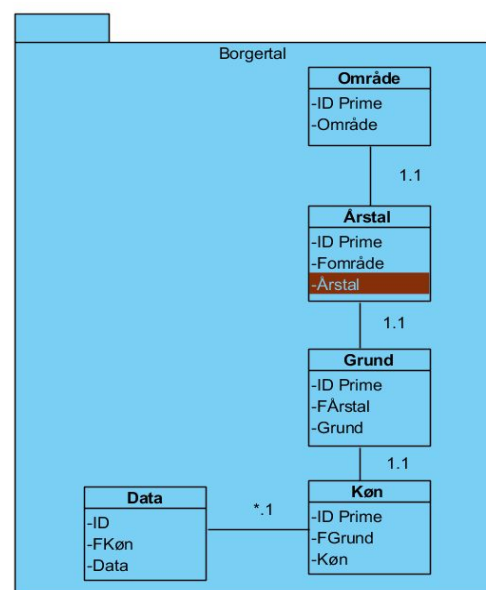
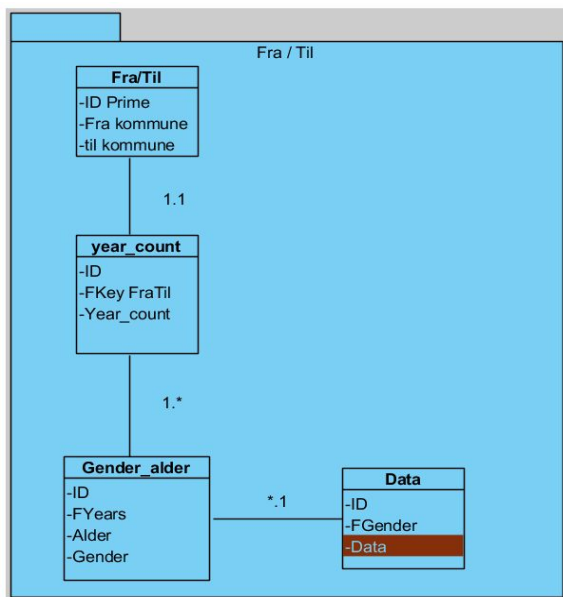
Køn	Bevægelsesart	År	København	Frederiksberg		Kategorier.
Mænd	Befolkningen ultimo forrige år	2006	Tal	Tal		Køn.
Kvinde	Levendefødte	2007	Tal	Tal		Bevægelsesart.
	Døde	2008	Tal	Tal		År.
	Fødselsoverskud	2009	Tal	Tal		Sted.
	Tilflyttede	2010	Tal	Tal		
	Fraflyttede	2011	Tal	Tal		
	Nettoilflyttede	2012	Tal	Tal		
	Indvandrede i alt	2013	Tal	Tal		
	Indvandret i indeværende år	2014	Tal	Tal		
	Indvandret før indeværende år	2015	Tal	Tal		
	Udvandrede i alt	2016	Tal	Tal		
	Udvandret i indeværende år	2017	Tal	Tal		
	Udvandret før indeværende år					
	Nettoindvandrede					
	Korrektioner					
	Befolkningstilvækst					
	Befolkningen ultimo indeværende år					

<https://docs.google.com/spreadsheets/d/1bPTgcbfubtKheWlStEPwB3wDaHLnAKzQyFNxYN4ILhs/edit?usp=sharing>

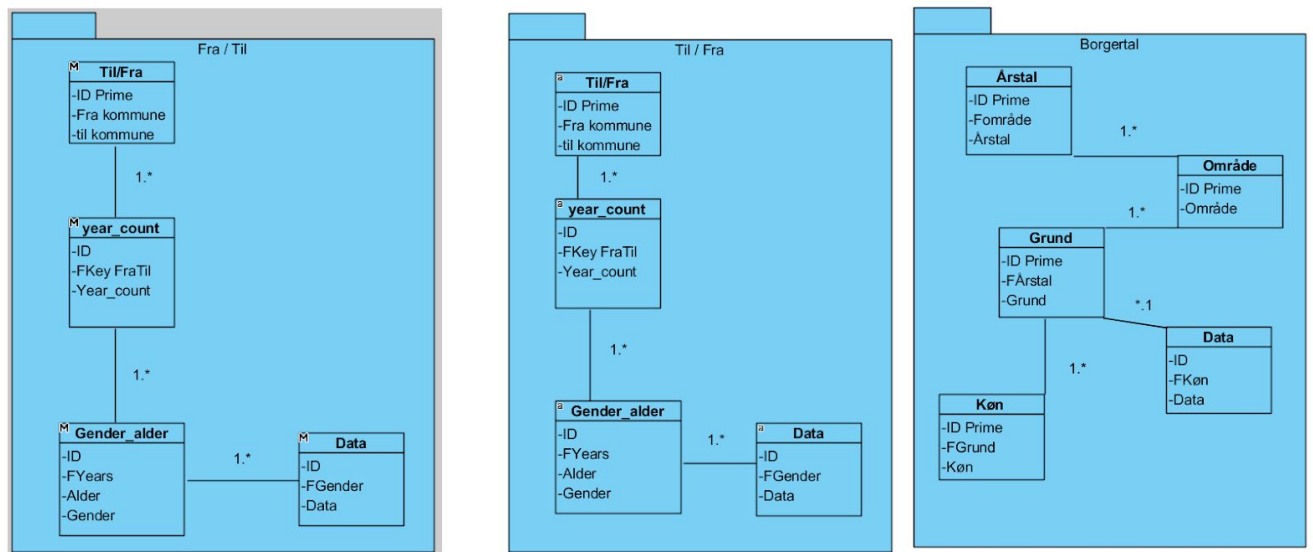
Domain model (Udkast 1)



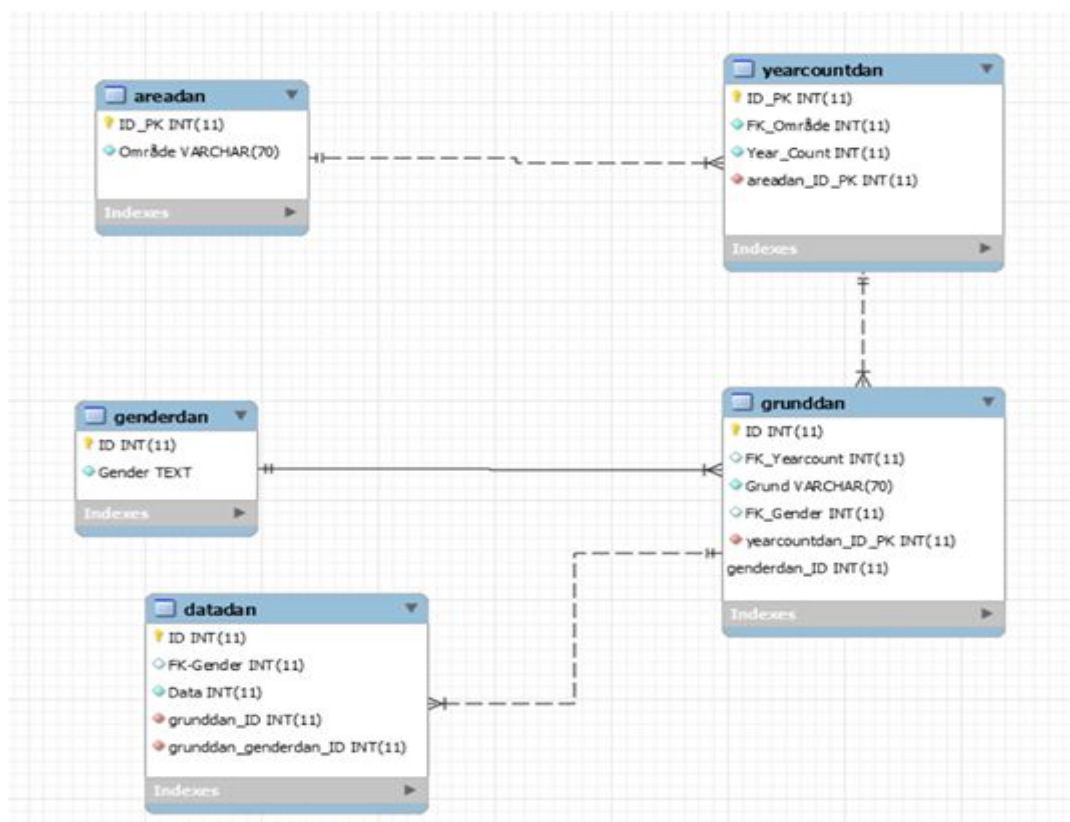
Domain model (Udkast 2)



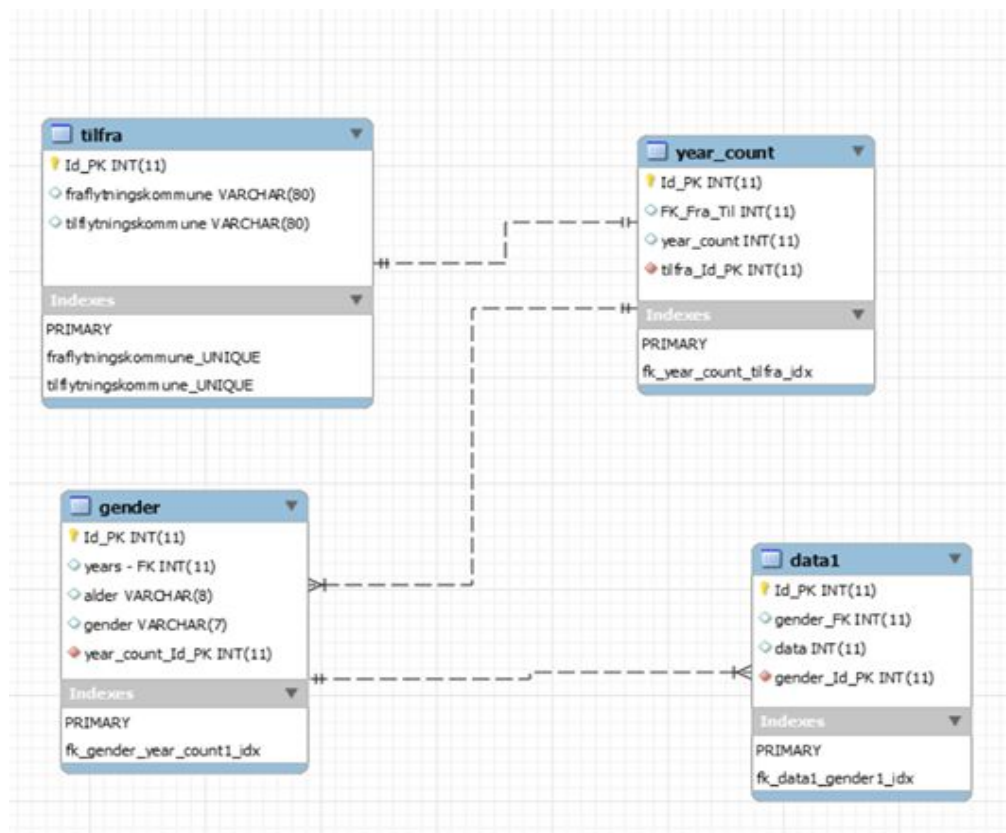
Domain model (Udkast 3)



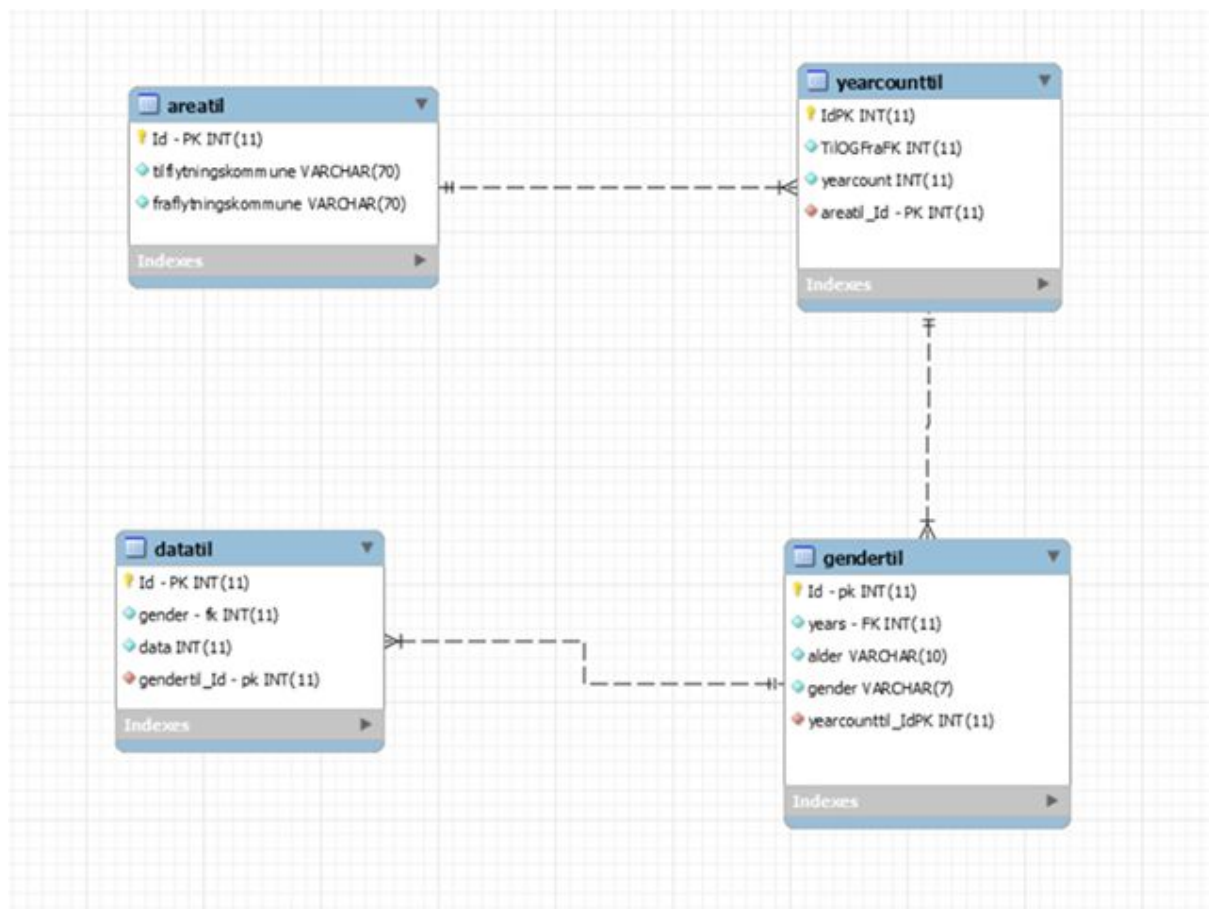
ER (Befolkning)



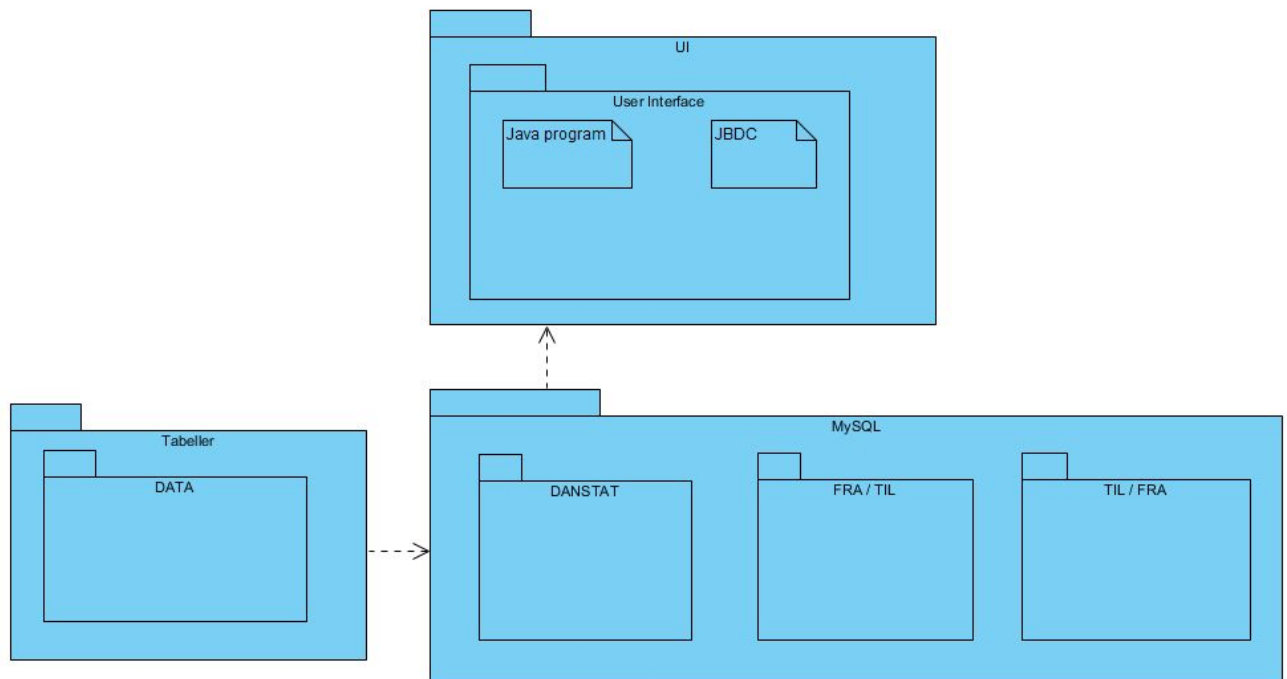
ER (Fra / Til)



ER (Til / Fra)



Layers



Report

Projektet går ud på at vi er blevet givet 3 datasæt med forskellige informationer i hver. Hvad vi så har gjort ved disse datasæt er at vi har flyttet data'en lidt rundt så det lettere kan passe ind i en SQL database. Vi har også holdt normaliserings reglerne ved tanke og dette afspejles i vores database. Derudover har vi også oprettet primærnøgler og fremmednøgler til vores data så vi slipper for at have for meget data et sted i vores database.

Når det kommer til normaliserings reglerne så gør tre ting sig gældende.

First Normal Form – The information is stored in a relational table and each column contains atomic values, and there are not repeating groups of columns.

Second Normal Form – The table is in first normal form and all the columns depend on the table's primary key.

Third Normal Form – the table is in second normal form and all of its columns are not transitively dependent on the primary key.

Hvad vi har gjort for at overholde regel nummer 1, er at vi kun har en form for data pr. "column" / linje.

Hvad vi har gjort for at overholde regel nummer 2, er at vi har oprette en prime key et sted som vi så arbejder vider med some foreign key all andre steder.

Hvad vi har gjort for at overholde regel nummer 3, er at vi har delt de tre tabeller op i 4 tabeller hver, så vi ender med 12 tabeller i alt. Dette har vi gjort fordi den tredje normalform bliver overholdt da alt data i tabellerne kun er afhængig af den primære nøgle.

Diskussion

Selv om vi er kommet godt omkring mange af de formelle krav, så har vi stadig valgt at prioritere vores tid og har derfor udeladt nogle få forskellige ting. Disse ting er ting som frontend delen og < > søgning. Grunden til at disse dele af opgaven er udeladt er simpelthen på grund af en mangel på tid og forståelse.

Konklusion

Vi er kommet godt omkring mange af de formelle krav men mangler stadig at opfylde nogle få. Vi er all tilfreds med vores database men vi er knap så tilfredse med vores arbejdsfordeling. Dette er dog et problem vi kunne havde løst hvis blot vi havde indgået i gruppekontrakter eller hvis vi havde brugt et andet gruppe planlægnings værktøj såsom scrum, gantt eller måske KANBAN; dette er noget vi bliver nødt til at arbejde på, hvis vi på et andet tidspunkt bliver stillet en større opgave.

Sources.

Front picture:

<https://techcrunch.com/2018/06/12/yugabytes-new-database-software-rakes-in-16-million-so-developers-can-move-to-any-cloud/?guccounter=1>

NEEDS:

Add a simple application to the front end and connect to the database using JDBC.

Make a document that explains the diagrams. Especially important are the ERD and how the 1st, 2nd and 3rd normal form is met.

The application must be able to

- Search based on criteria's entered by the user
- Show a list of content from a table based on a criteria
- Show a list as a combination of several tables (INNER JOIN)
- If time allows search by less than, greater than criteria's

<https://www.youtube.com/watch?v=OSPJhCkm850>