

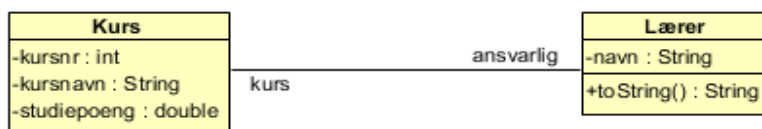
Hibernate ORM

Hibernate ORM (Hibernate Object Relational Mapping), vanligvis bare kalt Hibernate, er en gratis programvare for å mappe fra Java klassediagrammer til relasjonsdatabase. Den konverterer også Java datatyper til SQL datatyper. Hibernate kan legges til som en plug-in i Eclipse, og er allerede integrert i Visual Paradigm. En kort innføring finner dere her:

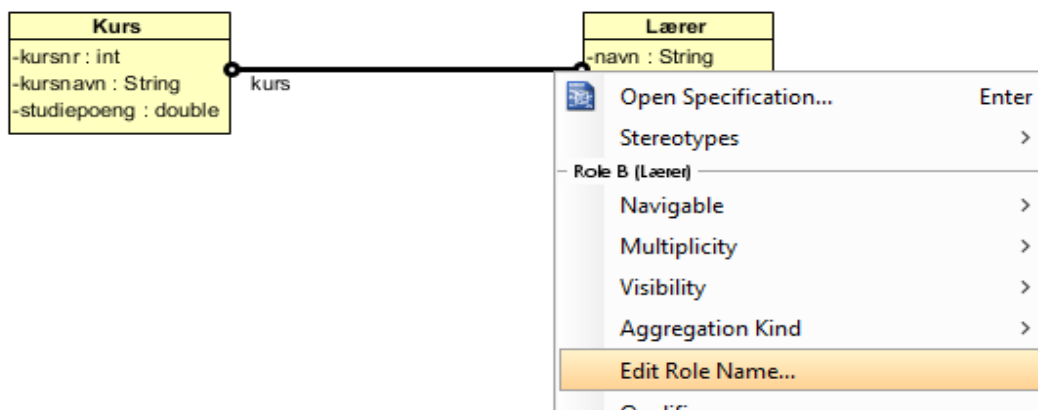
https://www.visual-paradigm.com/support/documents/vpuserguide/3563/3564/85444_mappingobjec.html

Klargjøring av klassediagrammet

Det er nødvendig å sette rollenavn på assosiasjonen mellom to klasser, slik det er vist her:

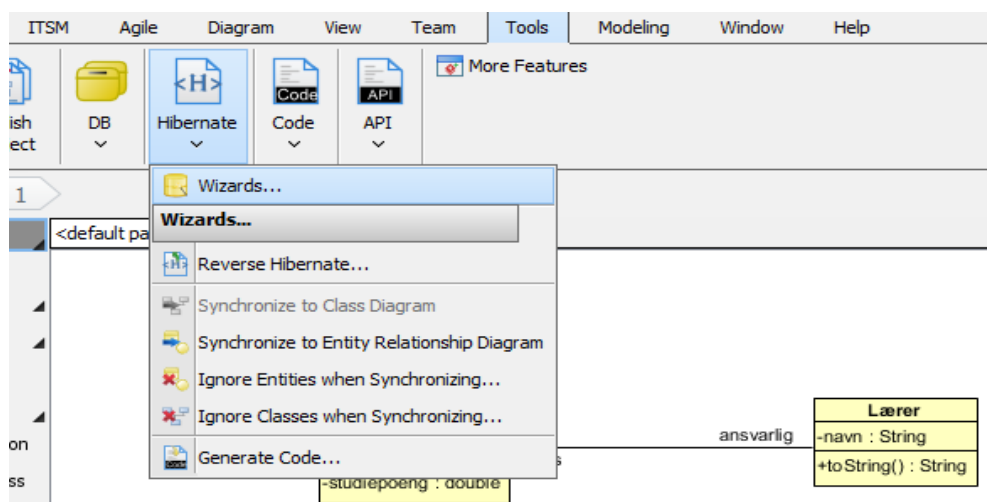


Vi gjør dette ved å høyreklikke på en ende av assosiasjonen, og så velge Edit Role Name:

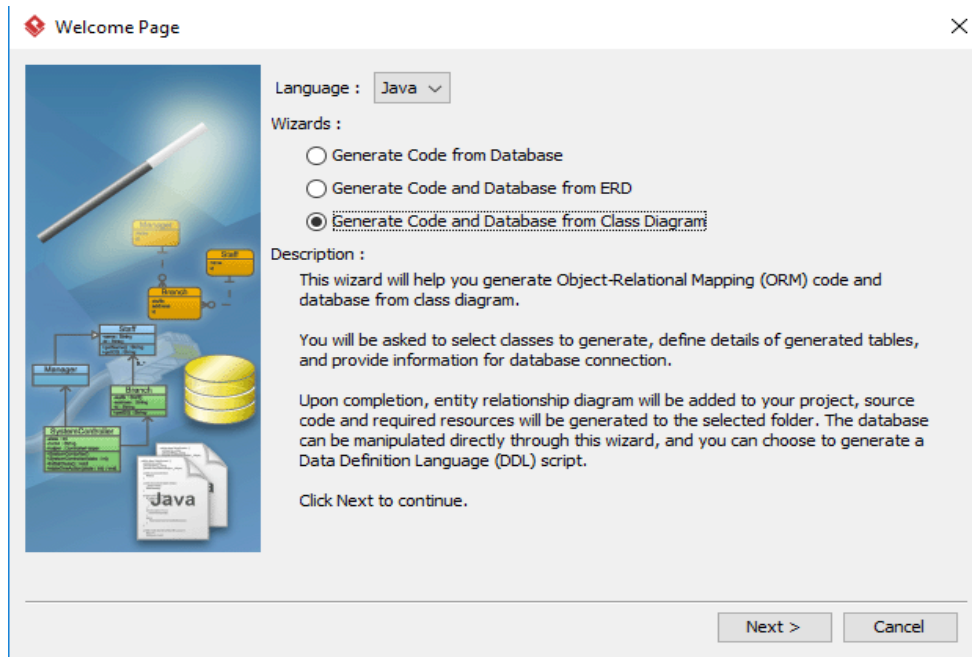


Mapping til ER-diagram

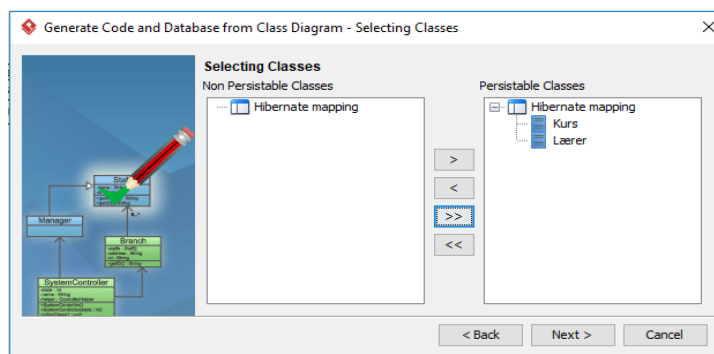
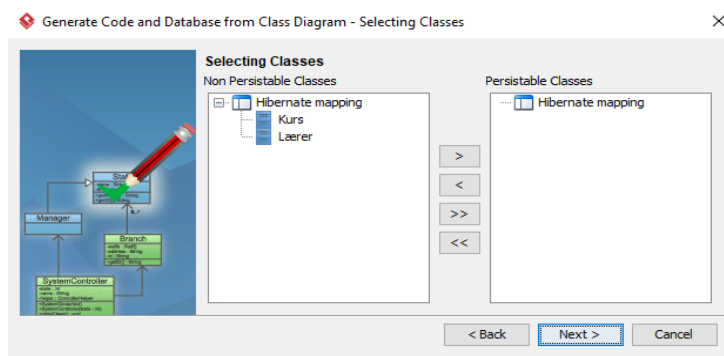
Gå til arkfanen Tools, velg Hibernate og så Wizards:



Velg Generate Code and Database from Class Diagram:



Velg klassene som skal mappes:



Neste trinn er å velge primærnøkkel for ER-diagrammet. I klassen Kurs har vi et unikt attributt i kursnr. Da lar vi Hibernate bruke dette til primærnøkkel. I klassen Lærer har vi ikke noe unikt attributt. Da lar vi Hibernate lage en primærnøkkel:

Generate Code and Database from Class Diagram - Select Primary Key

Target Parent : Same as source model element

Select Primary Key
Select primary key:

Entity	Primary Key
Lærer	Auto Generate
Kurs	Auto Generate
	kursnr
	kursnavn
	studiepoeng
	Do Not Generate
	Auto Generate

Reset all to Auto Generate Reset all to Do Not Generate

< Back Next > Cancel

Neste trinn er å spesifisere databasen:

Generate Code and Database from Class Diagram - Database Configuration

Database Configuration

Generate Database : Create Database

☐ Export to Database ☒ Generate DDL

Quote SQL Identifier: Default(Auto)

Connection : JDBC

JDBC

Connection Pool Options ☒ Use connection pool

Database Setting

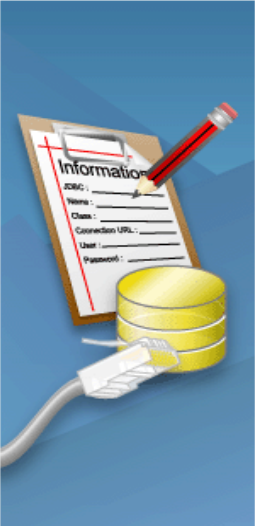
Driver file : ...

Connection URL : Production ?

< Back Next > Cancel

Vi må angi driveren som skal brukes, server, port, database og bruker. Her er JDBC-driveren for MySql valgt.

Generate Code and Database from Class Diagram - Database Configuration



Database Configuration

Generate Database : Create Database

☒ Export to Database ☒ Generate DDL

Quote SQL Identifier: Default(Auto)

Table Charset:

Connection : JDBC

JDBC

Connection Pool Options ☒ Use connection pool

Database Setting

Driver : MySQL5 (Connector/J Driver) ✓

Driver file : J 5.1\mysql-connector-java-5.1.40-bin.jar ... ✓

Connection URL : Production ?

☒ Hostname : localhost : 3306
Database name : studiebase

☐ jdbc:mysql://localhost:3306/studiebase

User : root Password :

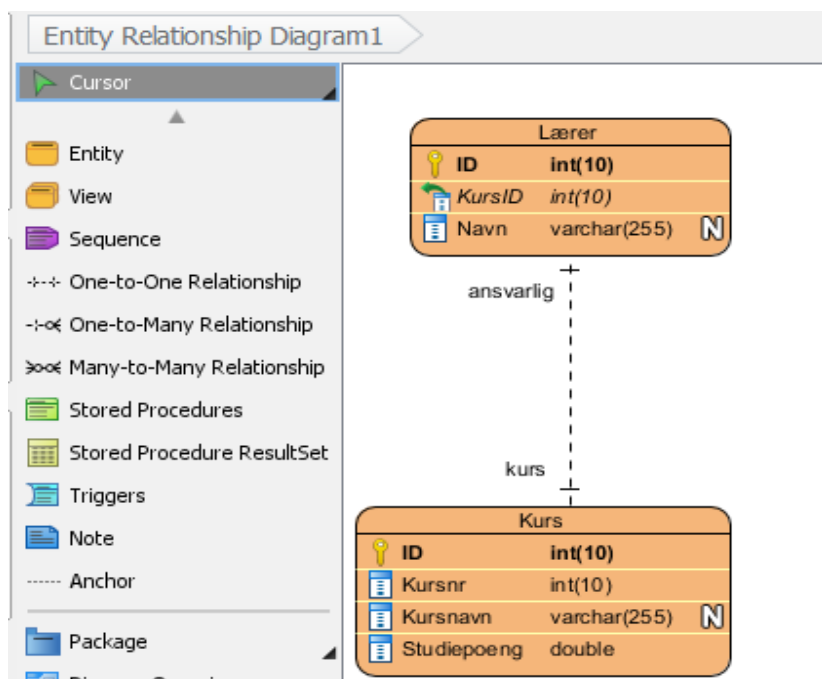
Engine: ☒ Default ☐ InnoDB (with constraint) ☐ MyISAM

☒ Set as default Test Connection

< Back Next > Cancel

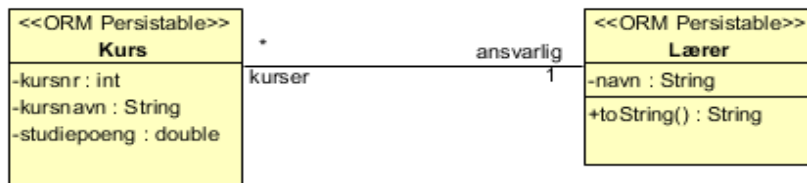
Jeg har på forhånd laget et skjema studiebase i MySQL.

Visual Paradigm lager nå en ER-modell av klassesdiagrammet:



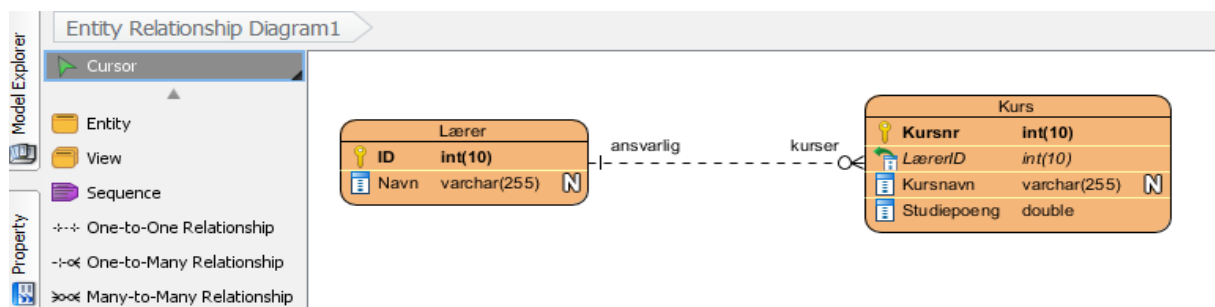
Her ble det imidlertid ikke riktig. Jeg hadde «glemt»   sette multiplisitet p  assosiasjonen i klassesdiagrammet. S  vi g r tilbake til dette, og oppdager at VP ogs  har lagt til stereotyper

på klassene. Her er det modifiserte klassediagrammet, med stereotypier og multiplisiteter på assosiasjonen:

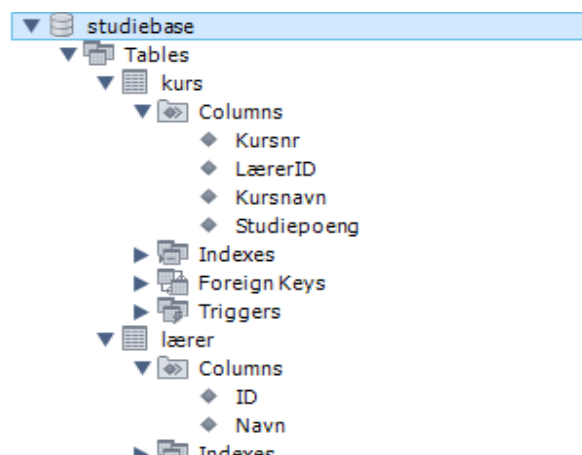


Persistable betyr at dette er klasser med data som skal lagres.

Nå ble det bedre:



Ikke nok med det, men Hibernate har også generert tabellene i MySQL:



I tillegg har den laget en DLL-fil med de nødvendige SQL-statements:

```

Document Selector
Hibernatemapping.ddl
1 ALTER TABLE Kurs DROP FOREIGN KEY FKkurs448899;
2 DROP TABLE IF EXISTS Kurs;
3 DROP TABLE IF EXISTS Lærer;
4 CREATE TABLE Kurs (Kursnr int(10) NOT NULL AUTO_INCREMENT, LærerID int(10) NOT NULL, Kursnavn varchar(255), Studiepoeng double NOT NULL, PRIMARY KEY (Kursnr));
5 CREATE TABLE Lærer (ID int(10) NOT NULL AUTO_INCREMENT, Navn varchar(255), PRIMARY KEY (ID));
6 ALTER TABLE Kurs ADD CONSTRAINT FKkurs448899 FOREIGN KEY (LærerID) REFERENCES Lærer (ID);
7
  
```

I tillegg har den generert en rekke Java-filer:

Documents > Hibernatemapping > src >

Name	Date modified	Type	Size
ormmapping	25.02.2019 13.09	File folder	
CreateHibernatemappingData	25.02.2019 13.27	JAVA File	2 KB
CreateHibernatemappingDatabaseSchema	25.02.2019 13.27	JAVA File	1 KB
DeleteHibernatemappingData	25.02.2019 13.27	JAVA File	1 KB
DropHibernatemappingDatabaseSchema	25.02.2019 13.27	JAVA File	1 KB
HibernatemappingPersistentManager	25.02.2019 13.27	JAVA File	4 KB
Kurs	25.02.2019 13.27	JAVA File	14 KB
KursCriteria	25.02.2019 13.27	JAVA File	2 KB
KursDetachedCriteria	25.02.2019 13.27	JAVA File	3 KB
KursSetCollection	25.02.2019 13.27	JAVA File	3 KB
ListHibernatemappingData	25.02.2019 13.27	JAVA File	3 KB
Lærer	25.02.2019 13.27	JAVA File	14 KB
LærerCriteria	25.02.2019 13.27	JAVA File	2 KB
LærerDetachedCriteria	25.02.2019 13.27	JAVA File	2 KB
ORMConstants	25.02.2019 13.27	JAVA File	1 KB
RetrieveAndUpdateHibernatemappingData	25.02.2019 13.27	JAVA File	2 KB

Hibernate vil også konvertere fra Java datatyper til sql datatyper. Her er en liste over de fleste av datatypene:

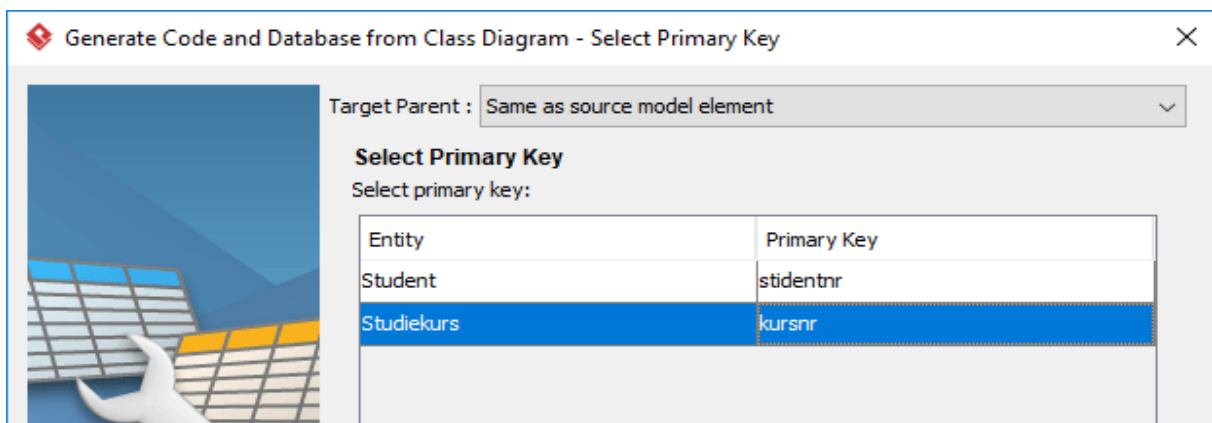
Object Model	Data Model
Boolean	Bit (1)
Byte	Tinyint (3)
Byte[]	Binary (1000)
Blob	Blob
Char	Char (1)
Character	Char (1)
String	varchar (255)
Int	Integer (10)
Integer	Integer (10)
Double	Double (10)
Decimal	Integer
BigDecimal	Decimal (19)
Float	Float (10)
Long	Bigint (19)
Short	Smallint (5)
Date	Date

Mapping av mange-til-mange assosiasjoner

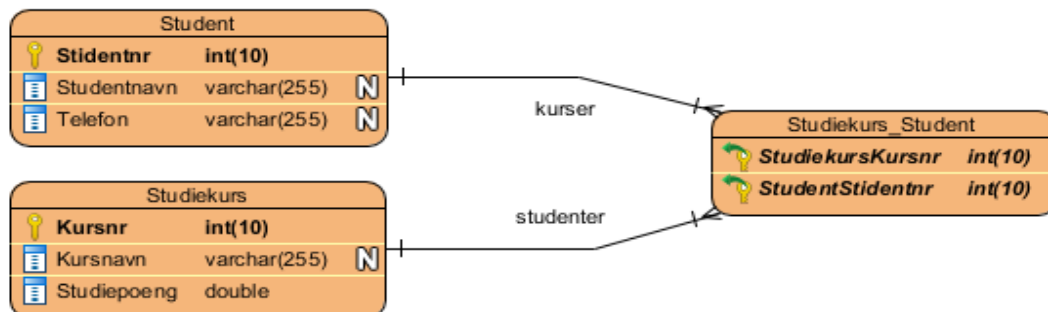
La oss se hva Hibernate gjør med en mange-til-mange assosiasjon. Vi starter med dette klassediagrammet:



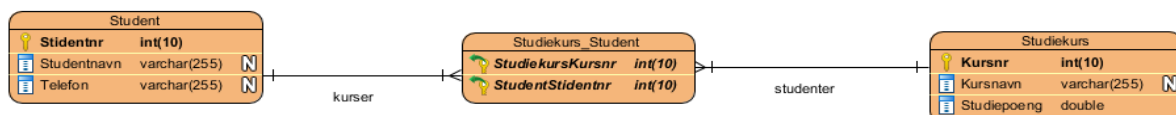
Begge kursene har et unikt attributt som kan brukes som primærnøkkel. Vi angir dette i veiviseren:



Hibernate genererte denne datamodellen:



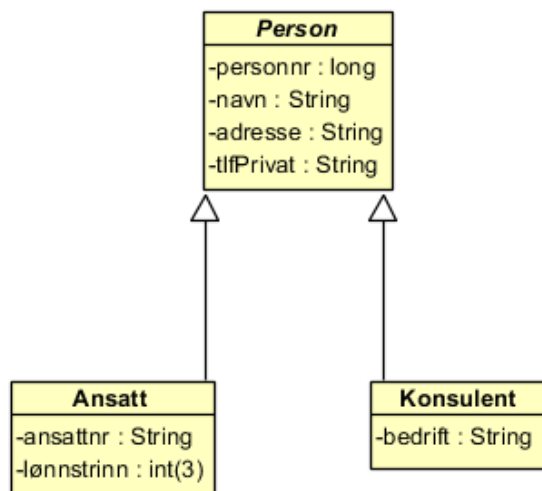
Denne kan vi omredigere slik at den kommer på en mer standardisert form:



Mapping av arvehierarkier

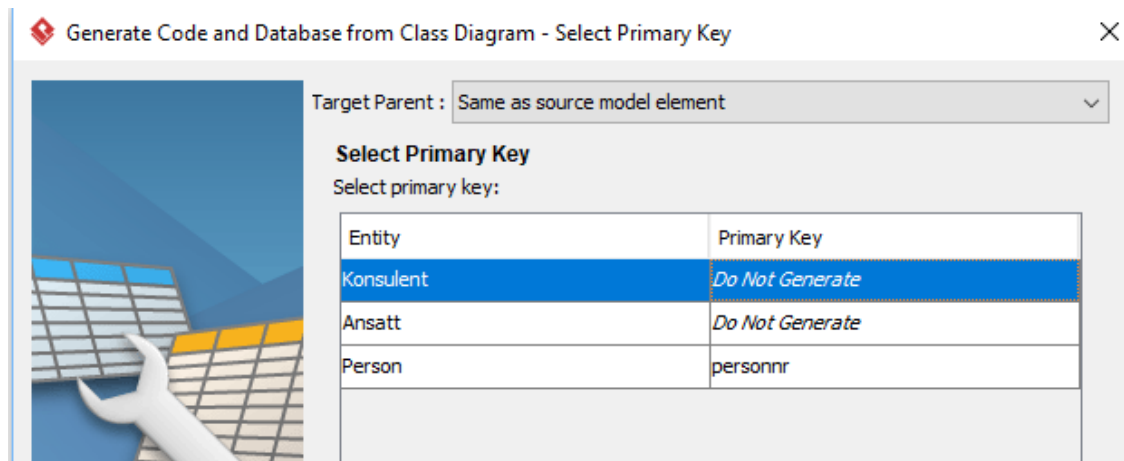
Visual Paradigm dokumentasjon: https://www.visual-paradigm.com/support/documents/vpuserguide/3563/3564/85388_differentinh.html

Vi begynner med dette arvehierarkiet:

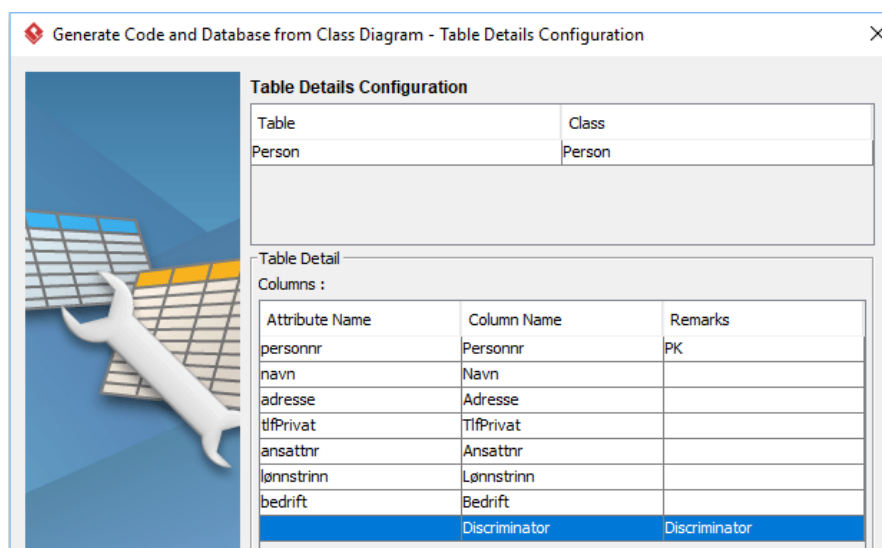


Klassen Person er abstrakt.









I veiviseren angir vi at bare superklassen Person skal definere en primærnøkkel:



Default er horisontal mapping:

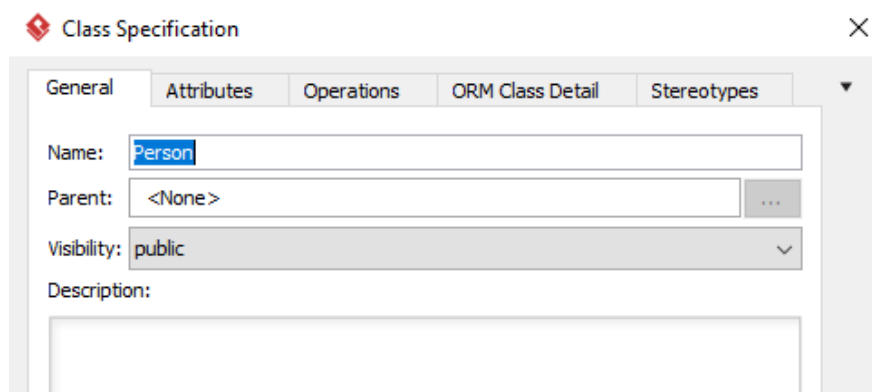


Resultatet ser slik ut som ER-diagram:

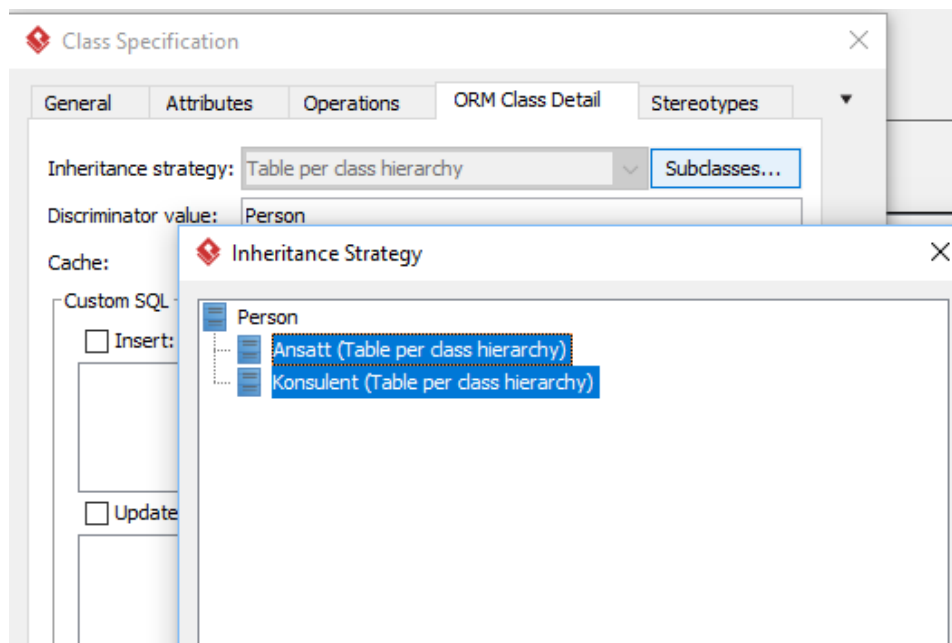
Person		
	Personnr	bigint(19)
	Navn	varchar(255) N
	Adresse	varchar(255) N
	TlfPrivat	varchar(255) N
	Ansattnr	varchar(255) N
	Lønnstrinn	int(11) N
	Bedrift	varchar(255) N
	Discriminator	varchar(255)

Hibernate har lagt til et felt Discriminator, som angir typen (ansatt eller konsulent)

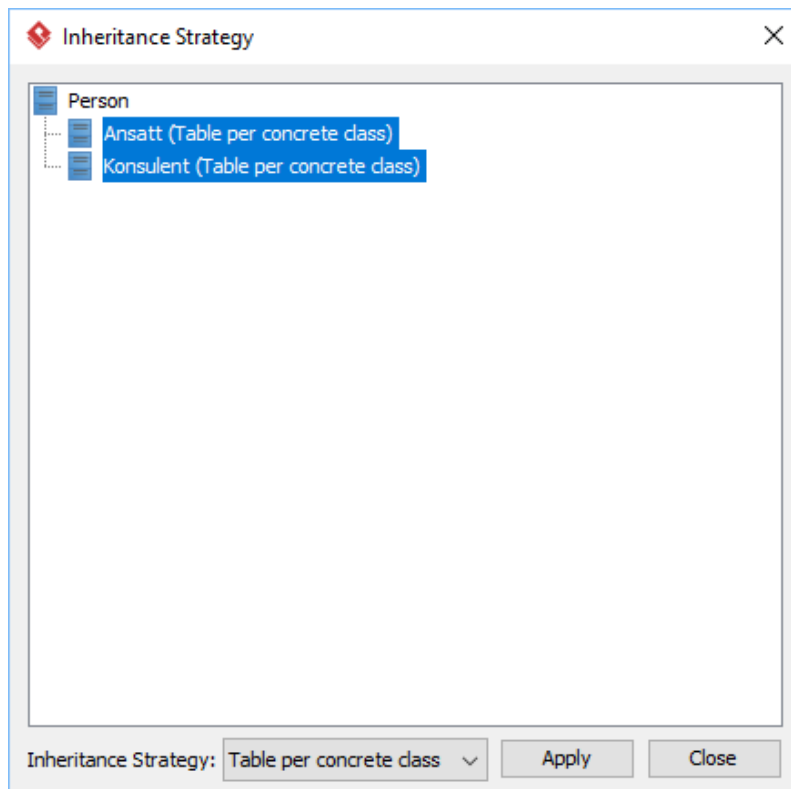
For å få vertikal mapping, høyreklikker vi på superklassen i klassesdiagrammet og velger Open Specification:



Vi velger fanen ORM Class Detail, og kan spesifisere subklasser (subklassene kommer opp automatisk, men må markeres):



I det samme vinduet kan vi nederst spesifisere mappingstrategi. Her er valgt Table per concrete class, som vil tilsvare vertikal mapping:



Valget blir imidlertid ikke effektivert. Det krever en «høyere», kommersiell lisens. Men vi kan iallfall se hvordan det gjøres.