

Database project specification



Danmarks Tekniske Universitet

Gruppe 14



Reza Haddang

(s176481)



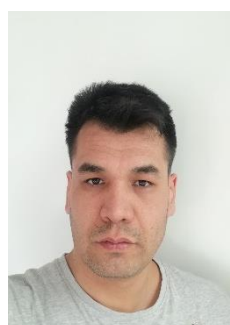
Amer Biro

(s176356)



Sulaiman Kasas

(s195462)



Mohammad Javad Zamani

(s176485)



Bashar Khaled Bdewi

(s183356)

Rapport i kurset Indledende databaser og database programmering

Danmarks Tekniske Universitet

Afleveringsfrist: 21. April 2020

Indledning

Denne rapport omhandler et database opsætning, for et fiktivt firma, i henhold til faget 02327 (Indledende databaser og database programmering).

Opgavebeskrivelsen lyder på oprettelse af database, skema og tabeller, for det fiktive handelsfirma TrainModels, hvor handelsfirmaet ønsker implementering af deres produkter, til et nyt IT-system som skal administrere køb og salg af TrainModels' skalamodeller.

Gruppe 14 har således, ud fra bedste evner, at gennemføre de givne kravspecifikationer, givet fra opgavebeskrivelsen.

I de kommende sider vil der forefindes indholdsfortegnelse, og ligeledes beskrivelse vores fremgangsmetode for det udførte arbejde.

Indhold

1.	DB tabel skemaer og brugerprofiler	5
1.1	DB tabel skemaer	5
1.2	Brugerprofiler.....	6
2.	Entity Relationship Diagram (UML notation).....	7
3.	Extended Entity Relationship Diagram (Crow's Foot notation)	8
4.	Database instans	9
5.	Normalformer	12
6.	SQL tabelmodifikationer	14
7.	SQL forespørgsler.....	16
8.	SQL programmering	18
8.1	Trigger	18
8.2	Funktionen	19
8.3	Proceduren.....	19
8.4	Eventen	20
9.	SQL brugerprofiler.....	21
10.	Java database adgang	23
11.	Deltagernes indsats i gruppearbejdet.....	25

1. DB tabel skemaer og brugerprofiler

1.1 DB tabel skemaer

- **Medarbejdere** (MID, Efternavn, Fornavn, Adresse, Postnummer, Bynavn, Land, Mobil, Email, Månedsløn, Timeløn, Lønkonto, Jobtitel, Afdeling, ChefID, Startdato, Slutdato)
- **Timerapporter**(MID, Arbejdsdato, Arbejdstime, Status, StatudID, Bemærkning)
- **Kunder**(KID, Kreditgrænse, Efternavn, Fornavn, Adresse, Postnummer, Bynavn, Land, Mobil, Email)
- **Produkter**(PID, Produkttype, Produktbeskrivelse, HTMLBeskrivelse, Billede, Leverandørnavnbeskrivelse, Købspris, Salgspris, Lagerantal, Transportdage_L_V, Transportdage_V_K)
- **Produkttyper**(Produkttype, Beskrivelse)
- **SalgsOrdre**(SOID, KID, Bestillingsdato, Afsendelsesdato, Faktureringsdato, Betalingsdato)
- **Salgsordrelinie**(SOID, PID, SalgsordrelinieBeskrivelse, Kvantitet, Salgsstykkepris)
- **Leverandører**(LID, Leverandørnavnbeskrivelse, Efternavn, Fornavn, Adresse, Postnummer, Bynavn, Land, Mobil, Email)
- **Købsordrer**(KOID, LID, Bestillingsdato, Modtagelsesdato, Betalingsdato)
- **Købsordrelinie**(KOID, PID, KøbsordrelinieBeskrivelse, Kvantitet, Købsstykkepris)

DB tabel skema er en essentielt del af projektet, da det ligger til grund for hvordan hele databasen skal se ud. Vi havde uenigheder om hvordan det skulle bygges, da medarbejderne ville have for mange informationer i én tabel.

Vi har derfor valgt at opdele medarbejdernes tabel i 3 tabeller.

Medarbejderes tabel og kontakt oplysningers tabel, samt en lønoplysningers tabel.

Vi søgte konsultation hos forelæseren og fik at vide, at vi skulle have kontaktoplysninger samt medarbejderes oplysninger i én tabel, men lønoplysninger kan være i en anden tabel, hvilket vi har gjort nytte af med vores Timerapporter tabel.

Medarbejdere tabel har MID (medarbejder ID) og Startdato som PRIMARY KEYS. Idéen med at startdato som PRIMARY KEY er, at den tabel bliver temporal, så den husker fortiden, som er en del af opgaven.

Timerapporter tabel har MID som både PRIMARY KEY og FOREIGN KEY, som refererer til medarbejdere tabel. Den har også Arbejdsdato som PRIMARY KEY.

Kunder tabel har også kontaktoplysninger i samme tabel. Den har også KID som PRIMARY KEY.

1.2 Brugerprofiler

Bruger Privileger	Administrator afdeling	Salgsafdeling	Indkøbsafdeling	Direktør	Salgschef	Indkøbschef
Produkter	Skrive			Skrive		
Produkttyper	Skrive			Skrive		
Købsordrer	Skrive		Skrive	Skrive		Skrive
Købsordrelinje	Skrive		Skrive	Skrive		Skrive
Salgsordrer	Skrive	Skrive		Skrive	Skrive	
Salgsordrelinje	Skrive	Skrive		Skrive	Skrive	
Kunder	Skrive	Læse		Skrive	Skrive	
Leverendør	Skrive		Læse	Skrive		Skrive
Timerapporter	Skrive, kun for egne	Skrive, kun for egne	Skrive, kun for egne	Skrive	Skrive, OG godkend/afvise medarbejder-timer	Skrive, OG godkend/afvise medarbejder-timer
Medarbejdere	Skrive, undtagen lønkonto, månedsløn, timeløn	Læse, undtagen lønkonto, månedsløn, timeløn	Læse, undtagen lønkonto, månedsløn, timeløn	Skrive	Læse, undtagen lønkonto, månedsløn, timeløn	Læse, undtagen lønkonto, månedsløn, timeløn

Figur 1: tabel oversigt for bruger privilegier - Excel

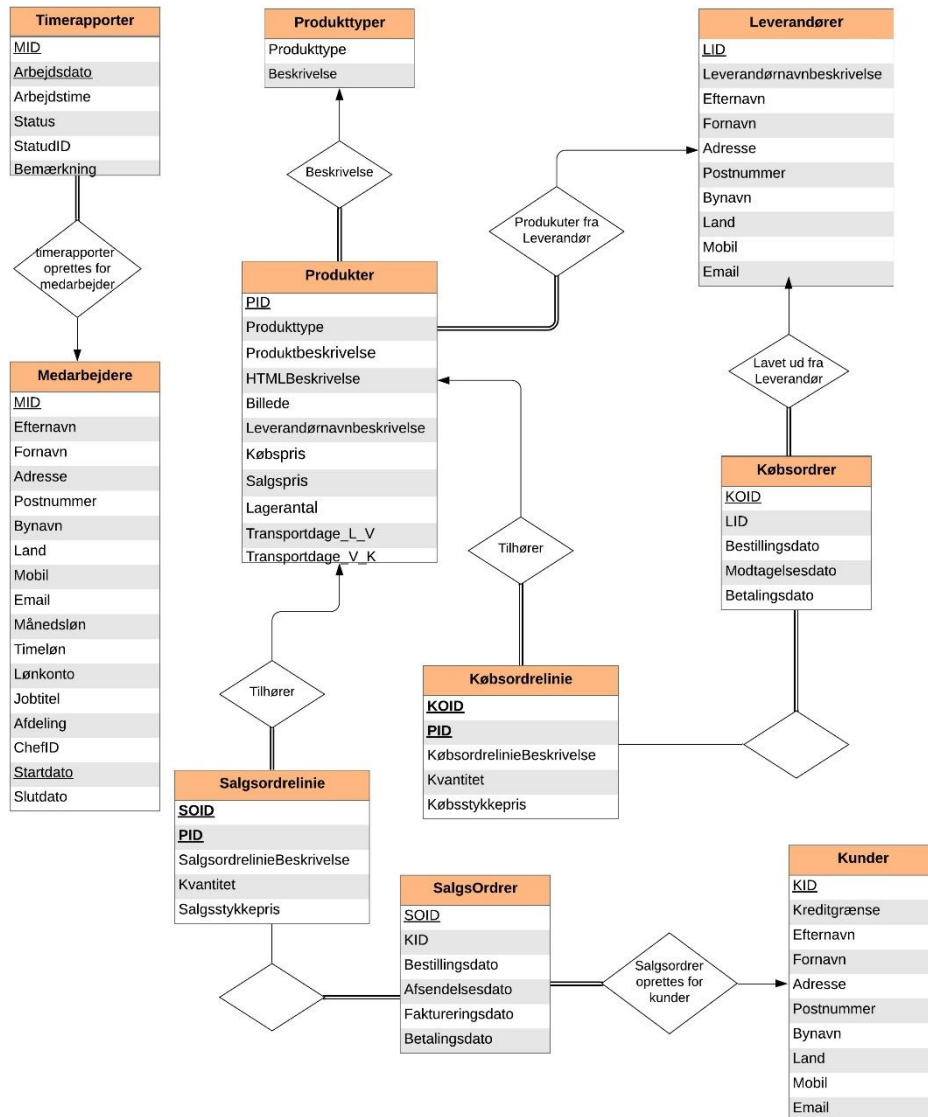
- "Skrive" er tilladelse til både at skrive og læse data i tabellen
- "Læse" er kun tilladelse til at læse data i tabellen

Overstående tabel er blevet konstrueret ud fra de givne oplysninger fra opgavebeskrivelsen. Tabellen viser de enkelte bruger-rettigheder for DB, i form af læse og skrive tilladelser.

F.eks. **direktøren** har adgang til alle funktioner i databasen. Han kan skrive og læse data i alle tabeller.

Salgschef kan læse sine medarbejders lønoplysninger. Kan også godkende eller afvise arbejdstimer for sine medarbejdere. Ligeledes kan de også skrives i kunder, salgsordrer og salgsordrelinje tabellerne.

2. Entity Relationship Diagram (UML notation)



Figur 2: Entity Relationship diagram - Lucidcharts

Ovenstående tabel viser forhold mellem forskellige tabeller. F.eks. en til mange/ mange til en. F.eks. timerapporter tabel SKAL have én medarbejder, men en medarbejder skal nødvendigvis ikke have en timerapport. Det nævnte forhold vises med, at der går 2 linjer fra timerapporter tabel mod medarbejdere tabel.

Forhold mellem timerapporter og medarbejdere tabel er en til mange. En medarbejder kan have en/ flere timerapporter, men en timerapport kan have en medarbejder og kun en.

3. Extended Entity Relationship Diagram (Crow's Foot notation)

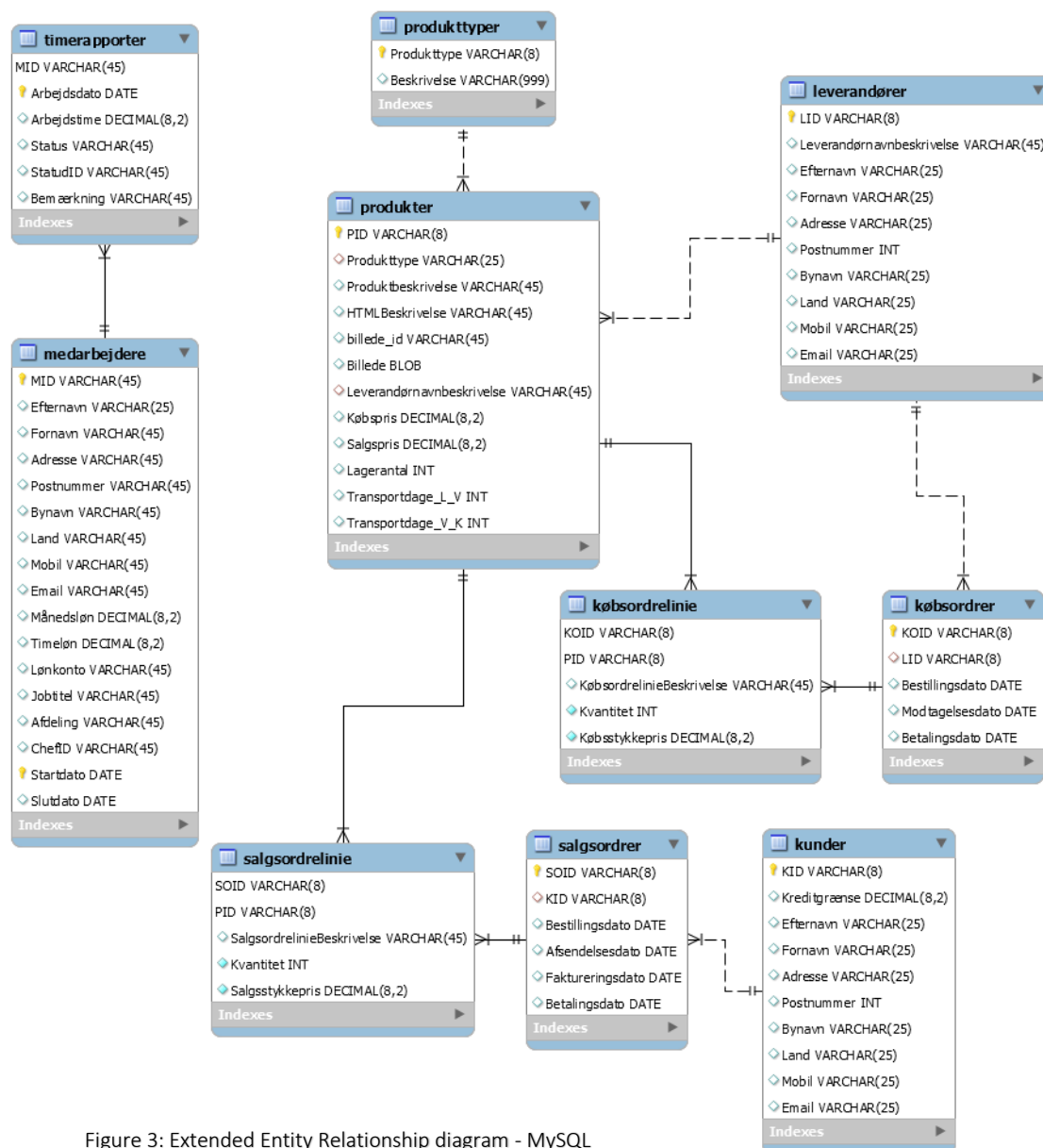


Figure 3: Extended Entity Relationship diagram - MySQL

Extended Entity Relationship diagram viser forhold mellem attributterne i forskellige tabeller, og forhold mellem tabellerne.

F.eks. **Timerapporter** tabel har MID som både PRIMARY KEY og FOREIGN KEY. MID PRIMARY KEY refererer til medarbejdere tabel. Den har også Arbejdsdato som PRIMARY KEY.

EER diagrammet er lavet i MSQl og bagefter har vi lavet Forward Engineering, for at få genere databasen (Se i Database instans V)

Produkttyper tabel viser at en produkttype kan have en eller mange produkter, men en produkt kan have én produkttype.

4. Database instans

Select * from Medarbejdere;

	MID	Efternavn	Fornavn	Adresse	Postnummer	Bynavn	Land	Mobil	Email	Månedsløn	Timeløn	Lønkonto	Jobtitel	Afdeling	ChefID	Startdato	Slutdato
▶	M00001	Bentsen	Anders	Elmedalsvej 1	4200	Slagelse	Danmark	+45 54 78 95 25	absn@gmail.com	100000.00	666.00	1234 5678910112	direktør	admin	selv	2017-01-01	9999-12-31
	M00002	Petersen	Malte	Skovduesien 31	2400	København	Danmark	+45 40 66 54 19	mpns@gmail.com	43500.00	290.00	1234 5438254324	produktchef	admin	M20001	2017-01-01	2022-12-31
	M00003	Dahl	Christine	Tinghøjvej 46	2860	Søborg	Danmark	+45 71 47 23 30	cdl@gmail.com	37500.00	250.00	1234 2635253387	salgschef	salg	M20001	2017-01-01	2020-12-31
	M00004	Frederiksen	Eva Annette	Vesterbo Vænge 34 1	3500	Værløse	Danmark	+45 42 32 17 72	efn@gmail.com	35000.00	230.00	1234 5247827447	indkøbschef	indkøb	M20001	2018-01-01	2024-12-31
	M10001	Kristensen	Hans	Måløv parkvej 2	2750	Ballerup	Danmark	+45 78 95 22 18	hln@gmail.com	26250.00	175.00	1234 5625527866	sælger	salg	M20005	2017-08-01	2024-12-31
	M10002	Hansen	Ole	Bybjergvej 15	2970	Hørsholm	Danmark	+45 91 95 47 17	ohn@gmail.com	28500.00	190.00	1234 6785784514	sælger	salg	M20005	2017-10-01	2019-12-31
	M10003	Alberg	Viggo	Søndre Ringvej 55	2605	Brøndby	Danmark	+45 42 72 12 21	vag@gmail.com	28500.00	190.00	4434 7531594026	indkøber	indkøb	M20012	2019-10-01	2025-12-31
	M20001	Jacobsen	Lise	Hulgårdsvvej 131	2400	København	Danmark	+45 34 25 98 11	ljne@gmail.com	0.00	195.00	1234 5638450134	administrassistent	admin	M20001	2018-01-01	2025-12-31
	M20002	Thomsen	Frederik	Sølsortvej 131	2000	Frederiksberg	Danmark	+45 70 22 48 78	frns@gmail.com	0.00	195.00	1234 5678253321	administrassistent	admin	M20001	2017-01-01	2030-12-31
	M20003	Leth	Iben	smakkegårdsvej 197	2820	Gentofte	Danmark	+45 93 89 99 10	lbi@gmail.com	0.00	180.00	1234 7854253595	salgsassistent	salg	M20005	2019-01-01	2026-12-31
	M20004	Mortensen	Hans Peter	Birkebakken 24	2750	Ballerup	Danmark	+45 70 40 70 52	hpm@gmail.com	0.00	185.00	1234 2133454850	salgsassistent	salg	M20005	2017-10-01	2022-12-31
	M20005	Jensen	Leif	Dronninggårds Alle 109	2840	Holte	Danmark	+45 70 45 32 71	ljn@gmail.com	0.00	180.00	1234 5771815338	salgsassistent	salg	M20005	2018-10-01	2020-12-31
	M20006	Petersen	Morten	Skodsborg Strandvej...	2942	Skodsborg	Danmark	+45 42 50 50 98	mpn@gmail.com	0.00	150.00	1234 9876543210	salgsassistent	salg	M20005	2018-10-01	2020-12-31
	M20007	Flyvholm	Ea	H C Andersens Vej 23	3000	Helsingør	Danmark	+45 32 78 49 01	efm@gmail.com	0.00	155.00	4434 1593574620	indkøbsassistent	indkøb	M20012	2019-10-01	2021-12-31
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Select * from Timerapporter;

	MID	Arbejdsdato	Arbejdstime	Status	StatudID	Bemærkning
▶	M10001	2019-12-23	7.50	Godkendt	M00003	Ingen
	M10002	2019-12-23	6.00	Afvist	M00003	Du tog tidligt hjem!
	M10003	2019-12-24	12.00	Godkendt	M00004	Ingen
	M20001	2019-12-23	4.00	Godkendt	M00001	Godt arbejde
	M20002	2019-12-23	4.50	Afvist	M00002	Du glemte at skrive under
	M20003	2019-12-23	2.00	Godkendt	M00003	Ingen
	M20004	2019-12-23	4.00	Afvist	M00003	Ingen
	M20005	2019-12-23	3.00	Godkendt	M00003	Ingen
	M20006	2019-12-23	2.00	Godkendt	M00003	Ingen
	M20007	2019-12-24	4.00	Godkendt	M00004	Ingen
•	NULL	NULL	NULL	NULL	NULL	NULL

Select * from Kunder;

	KID	Kreditgrænse	Efternavn	Fornavn	Adresse	Postnummer	Bynavn	Land	Mobil	Email
▶	K00001	150000.00	Reza	Reza	MagstrÅde-Snaregade	777	Copenhagen	Denmark	Unknown	re@dk.com
	K00002	225000.00	Bshar	Bshar	Champs ElysÅes	758	Paris	France	Unknown	ba@fr.com
	K00003	75000.00	Jawad	Jawad	Via Montnapoleone	92631	Milan	Italy	Unknown	ja@it.com
	K00004	325000.00	Amer	Amer	Fifth Avenue, Manhattan	7744	New York	USA	Unknown	am@us.com
	K00005	700000.00	Donald	Trump	Sheikh Zayed Road	0	Dubai	UAE	Unknown	dt@ua.com
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Select * from Produkter;

	PID	Produkttype	Produktbeskrivelse	HTMLBeskrivelse	billede_id	Billede	Leverandørnavnbeskrivelse	Købspris	Salgspris	Lagerantal	Transportdage_L_V	Transportdage_V_K
▶	P00001	High speed trains	High speed	High speed trains	P00001	NULL	Frankfort	7400.00	9200.00	21	18	28
	P00002	Inter-city trains	Inter-city	Inter-city trains	P00002	NULL	Dublin	3300.00	6700.00	14	7	9
	P00003	Commuter/regional trains	Commuter/regional	Commuter/regional trains	P00003	NULL	Berlin	3700.00	8400.00	21	2	6
	P00004	Rapid transit	Rapid	Rapid transit	P00004	NULL	Madrid	8200.00	9200.00	29	5	9
	P00005	Light rail	Light	Light rail	P00005	NULL	Oslo	8900.00	11200.00	5	16	21
	P00006	Modern streetcar	Modern	Modern streetcar	P00006	NULL	Brussel	10100.00	13000.00	3	8	14
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Select * from Produkttyper;

Produkttype	Beskrivelse
Commuter/regional trains	Commuter trains generally mean trains connecting suburban areas with the central city and primarily serves riders to and from work. Commuter trains typically run on weekdays, during rush hours, and only in the peak directions. A prime example would be Al...
High speed trains	High speed trains are generally defined as trains that can operate 125mph or faster. High speed trains generally connect large metropolitan areas (with very few stops in between) and are meant to be competitive with airlines in terms of overall travel ti...
Inter-city trains	Inter-city trains generally mean trains traveling long distances connecting metropolitan areas. Although the distances covered by some of these trains are comparable to airlines, inter-city trains generally operate at highway speed. Long distance inter-c...
Light rail	Light rail, which might be also known as trolley and streetcars, mean trains that function as local transit in an urban-core and can operate on the street-level. Compared to rapid transit, light rail costs less, is more pedestrian friendly, but has less ...
Modern streetcar	In some cities such as Portland and Seattle, they have a urban streetcar system that is somewhat compatible but operated separately from their light rail system. Those streetcars typically have smaller dimensions and operate at slower speed than their li...
Rapid transit	Rapid transit, which is also known as metro, subway, and heavy rail, mean trains that generally serve the urban-core, have large passenger capacity, and operate totally separate from road traffic. In order to run separately from road traffic in the city-...
NULL	NULL

Select * from SalgsOrder;

	SOID	KID	Bestillingsdato	Afsendelsesdato	Faktureringsdato	Betalingsdato
►	SO00001	K00001	2020-04-17	2020-04-19	2020-04-18	2020-04-17
	SO00002	K00002	2020-04-12	2020-04-15	2020-04-13	2020-04-12
	SO00003	K00003	2020-03-15	2020-03-19	2020-03-14	2020-03-15
	SO00004	K00004	2019-01-21	2019-01-23	2019-01-22	2019-01-21
	SO00005	K00005	2007-07-07	2007-07-11	2007-07-08	2007-07-07

Select * from Salgsordrelinie;

	SOID	PID	SalgsordrelinieBeskrivelse	Kvantitet	Salgsstykkepris
►	SO00001	P00001	High speed trains	12	9200.00
	SO00002	P00002	Inter-city trains	15	6700.00
	SO00003	P00003	Commuter/regional trains	17	8400.00
	SO00004	P00004	Rapid transit	29	9200.00
	SO00005	P00005	Light rail	47	11200.00
	SO00006	P00003	Commuter/regional trains	3	8400.00
	SO00006	P00004	Rapid transit	13	9200.00

Select * from Leverandører;

	LID	Leverandørnavnbeskrivelse	Efternavn	Fornavn	Adresse	Postnummer	Bynavn	Land	Mobil	Email
►	L00001	High speed trains	Mason	Smith	Florida	4500	Florida	US.	NULL	NULL
	L00002	Inter-city trains	Murphy	Johnson	New York	51385	New York	US.	NULL	NULL
	L00003	Commuter/regional trains	Brown	Williams	Las Vegas	5124	Las Vegas	US.	NULL	NULL
	L00004	Rapid transit	Morton	Brown	Paris	1268	Paris	France	NULL	NULL
	L00005	Light rail	Li	Jones	Dublin	89412	Dublin	Ireland	NULL	NULL
	L00006	Modern streetcar	Wilson	Oliver	Berlin	51903	Berlin	Germany	NULL	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Select * from Købsordrer;

	KOID	LID	Bestillingsdato	Modtagelsesdato	Betalingsdato
▶	K00001	L00001	2019-12-21	2019-12-23	2019-12-21
	K00002	L00002	2017-04-21	2017-04-27	2017-04-21
	K00003	L00003	2017-11-13	2017-11-17	2017-11-13
	K00004	L00004	2017-09-07	2017-09-11	2017-09-07
	K00005	L00005	2006-08-04	2006-08-08	2006-08-04
✱	NULL	NULL	NULL	NULL	NULL

Select * from Købsordrelinie;

	KOID	PID	KøbsordrelinieBeskrivelse	Kvantitet	Købsstykkepris
▶	K00001	P00001	High speed trains	7	7400.00
	K00002	P00002	Inter-city trains	12	3300.00
	K00003	P00003	Commuter/regional trains	13	3700.00
	K00004	P00004	Rapid transit	24	8200.00
	K00005	P00005	Light rail	40	8900.00
✱	NULL	NULL	NULL	NULL	NULL

Vi har udfyldt medarbejdere tabel udefra de givet oplysninger i opgavebeskrivelsen. F.eks. Cheferne er tildelt et MID, der starter med M0. Fuldtidsmedarbejdere er tildelt et MID, der starter med M1, og deltidsmedarbejdere er tildelt et MID, der starter med M2. Vi havde tidligere oplevet nogle problemer, når vi skulle udfylde tabellerne. Problemer kunne bl.a. være at en VARCHAR(value) er for begrænset, at den ikke kan indeholde en længere tekst beskrivelse. Andre problemer havde været, at også en INT(value) var også for begrænset, eller at man ikke kunne oploade et billede. Alle de nævnte problemer er blevet fixet undervejs, enten ved at redigere value i f.eks. VARCHAR og INT, eller ved at tilføje SQL kode, som kan ses i nedenstående skærmudklip.

```
1  -- MySQL Workbench Forward Engineering
2  • SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
3  • SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
4  • SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
5  • SET SQL_MODE=@OLD_SQL_MODE;
6  • SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
7  • SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
8  • SET @@global.sql_mode= '';
9  • SET GLOBAL FOREIGN_KEY_CHECKS=0;
10 • SET FOREIGN_KEY_CHECKS=0;
11 • SET GLOBAL log_bin_trust_function_creators = 1;
--
```

5. Normalformer

De fleste tabeller står allerede i 3NF. Men man kan godt gøre noget forbedringer.

F.eks. Medarbejdere tabel er i 1NF, fordi

- 1- Der er ikke flere værdier i én cell (atomic)
- 2- Der er ikke værdier af forskellige typer i én attribut. Dvs. alle værdier er fra samme domæne
- 3- Ikke nogen af attributterne har samme navn
- 4- Hver række har et unikt PRIMARY KEY
- 5- Rækkefølgen af attributterne er ligegyldigt.

Medarbejdere tabel er i 2NF, fordi alle ikke PRIMARY KEYS attributter er fuldt afhængige af hele PRIMARY KEY (MID, Startdato)

Medarbejdere tabel er ikke i 3NF, selvom alle attributterne er fuldt afhængige af hele PRIMARY KEY (MID, Startdato), men nogle af dem er transitivt afhængige af PRIMARY KEY bl.a.

(Postnummer, Bynavn, Land)

Bynavn er fuldt afhængig af hele PRIMARY KEY, men er transitivt afhængig af Postnummer

(Afdeling, ChefID)

Afdeling er fuldt afhængig af hele PRIMARY KEY, men er transitivt afhængig af ChefID

Det skaber gentagelser af attributternes værdier, således skaber det også krænkelse af 3NF.

Man kan godt dele tabellen i 3 mindre tabeller.

Normalisering ved projektion:

- 1- Medarbejdere3NF \equiv Π _MID, Efternavn, Fornavn, Adresse, Postnummer, Mobil, Email, Månedsløn, Timeløn, Lønkonto, ChefID, Startdato, Slutdato (Medarbejdere)
- 2- Adresse \equiv Π _Postnummer, Bynavn, Land (Medarbejdere)
- 3- Job \equiv Π _Afdeling, ChefID (Medarbejdere)

Ved Natural Join kan man komme tilbage til den fulde tabel.

Medarbejdere \equiv Medarbejdere3NF \bowtie Adresse \bowtie Job

- 1- Medarbejdere(MID, Efternavn, Fornavn, Adresse, Postnummer, Mobil, Email, Månedsløn, Timeløn, Lønkonto, Jobtitel, ChefID, Startdato, Slutdato)
Foreign Key(Postnummer) references Adresse(Postnummer)
Foreign Key(ChefID) references Job(ChefID)
- 2- Adresse(Postnummer, Bynavn, Land)
- 3- Job(Afdeling, ChefID)

select distinct MID, Efternavn, Fornavn, Adresse, Postnummer, Mobil, Email, Månedsløn, Timeløn, Lønkonto, ChefID, Startdato, Slutdato from medarbejdere;

	MID	Efternavn	Fornavn	Adresse	Postnummer	Mobil	Email	Månedsløn	Timeløn	Lønkonto	ChefID	Startdato	Slutdato
►	M00001	Bentsen	Anders	Elmedalsvej 1	4200	+45 54 78 95 25	absn@gmail.com	100000.00	666.00	1234 5678910112	M00001	2017-01-01	9999-12-31
	M00002	Petersen	Malte	Skovduetien 31	2400	+45 40 66 54 19	mpns@gmail.com	43500.00	290.00	1234 5438254324	M00001	2017-01-01	2022-12-31
	M00003	Dahl	Christine	Tinghøjvej 46	2860	+45 71 47 23 30	cdl@gmail.com	37500.00	250.00	1234 2635253387	M00001	2017-01-01	2020-12-31
	M00004	Frederiksen	Eva Annette	Vesterbo Vænge 34 1	3500	+45 42 32 17 72	efn@gmail.com	35000.00	230.00	1234 5247827447	M00001	2018-01-01	2024-12-31
	M10001	Kristensen	Hans	Måløv parkvej 2	2750	+45 78 95 22 18	hkn@gmail.com	26250.00	175.00	1234 5625527866	M00003	2017-08-01	2024-12-31
	M10002	Hansen	Ole	Bybjergvej 15	2970	+45 91 95 47 17	ohn@gmail.com	28500.00	190.00	1234 6785784514	M00003	2017-10-01	2019-12-31
	M10003	Alberg	Viggo	Søndre Ringvej 55	2605	+45 42 72 12 21	vag@gmail.com	28500.00	190.00	4434 7531594026	M00004	2019-10-01	2025-12-31
	M20001	Jacobsen	Lise	Hulgårdsvej 131	2400	+45 34 25 98 11	ljne@gmail.com	0.00	195.00	1234 5638450134	M00001	2018-01-01	2025-12-31
	M20002	Thomsen	Frederik	Solsortvej 131	2000	+45 70 22 48 78	frns@gmail.com	0.00	195.00	1234 5678253321	M00001	2017-01-01	2030-12-31
	M20003	Leth	Iben	smakkegårdsvej 197	2820	+45 93 89 99 10	ibl@gmail.com	0.00	180.00	1234 7854253595	M00003	2019-01-01	2026-12-31
	M20004	Mortensen	Hans Peter	Birkebakken 24	2750	+45 70 40 70 52	hpm@gmail.com	0.00	185.00	1234 2133454850	M00003	2017-10-01	2022-12-31
	M20005	Jensen	Leif	Dronninggårds Alle 109	2840	+45 70 45 32 71	ljn@gmail.com	0.00	180.00	1234 5771815338	M00003	2018-10-01	2020-12-31
	M20006	Petersen	Morten	Skodsborg Strandvej...	2942	+45 42 50 50 98	mpn@gmail.com	0.00	150.00	1234 9876543210	M00003	2018-10-01	2020-12-31
	M20007	Flyvholm	Ea	H C Andersens Vej 23	3000	+45 32 78 49 01	efm@gmail.com	0.00	155.00	4434 1593574620	M00004	2019-10-01	2021-12-31
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

select distinct Postnummer, Bynavn, Land from medarbejdere;

	Postnummer	Bynavn	Land
►	4200	Slagelse	Danmark
	2400	København	Danmark
	2860	Søborg	Danmark
	3500	Værløse	Danmark
	2750	Ballerup	Danmark
	2970	Hørsholm	Danmark
	2605	Brøndby	Danmark
	2000	Frederiksberg	Danmark
	2820	Gentofte	Danmark
	2840	Holte	Danmark
	2942	Skodsborg	Danmark
	3000	Helsingør	Danmark

select distinct Afdeling, ChefID from medarbejdere;

Afdeling	ChefID
admin	M00001
salg	M00003
indkøb	M00004

Man mister ikke data ved normalisering.

6. SQL tabelmodifikationer

I de tre nedenstående tabeller kan det ses hvordan SQL kommandoerne kan benyttes til at sælge produkter til kunden Reza Reza (både fornavn og efternavn er angivet som Reza).

I 28.april.2020 vil kunde (K00001) Reza Reza købe:

Kvantitet	ProduktID	Produkttype	Salgsstykkepris	SalgsOrdreID
7.stk	P00001	High speed trains	9200	SO00006
13.stk	P00004	Rapid transit	9200	SO00006
3.stk	P00005	Light rail	11200	SO00006

Indsættelse af salgsordre:

```
insert salgsordrer values ('SO00006','K00001','2020-04-28','2020-04-30','2020-05-03','2020-04-30');
```

Indsættelse af salgsordrelinier:

```
INSERT Salgsordrelinie VALUES  
('SO00006', 'P00001', 'High speed trains', '7', '9200'),  
('SO00006', 'P00004', 'Rapid transit', '13', '9200'),  
('SO00006', 'P00005', 'Light rail', '3', '11200');
```

Reza Reza vil købe produkt nr P00003 i stedet for P00005

```
UPDATE Salgsordrelinie SET PID='P00003',  
Salgsstykkepris='8400',  
SalgsordrelinieBeskrivelse='Commuter/regional trains' WHERE  
SOID='SO00006' AND PID='P00005';
```

I tilfælde af at kunden Reza Reza vil aflyse produkt nr P00001m kan følgende gøres:

```
DELETE FROM Salgsordrelinie WHERE SOID='SO00006' and  
PID='P00001';
```

```
SELECT SOID, KID, Bestillingsdato FROM salgsordrer WHERE  
SOID='SO00006';
```

	SOID	KID	Bestillingsdato
1	SO00006	K00001	2020-04-28

```
SELECT * FROM Salgsordrelinie WHERE SOID='SO00006';
```

	SOID	PID	SalgsordrelinieBeskrivelse	Kvantitet	Salgsstykkepris
1	SO00006	P00003	Commuter/regional trains	3	8400.00
2	SO00006	P00004	Rapid transit	13	9200.00

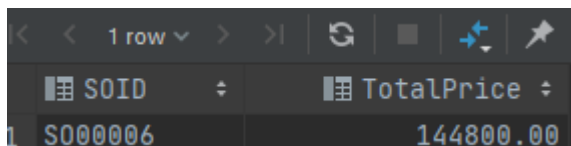
7. SQL forespørgsler

Hvis vi skal beregne totalprisen for kunden Reza Reza ordren SO00006, vil følgende fremgangsmåde kunne benyttes, som forevist i de kommende skærmudklip.

```
SELECT SOID, PID, SalgsordrelinieBeskrivelse, Kvantitet,
Salgsstykkepris, Kvantitet*Salgsstykkepris AS TotalPrice FROM
salgsordrelinie where SOID='SO00006'
```

	SOID	PID	SalgsordrelinieBeskrivelse	Kvantitet	Salgsstykkepris	TotalPrice
1	SO00006	P00003	Commuter/regional trains	3	8400.00	25200.00
2	SO00006	P00004	Rapid transit	13	9200.00	119600.00

```
SELECT SOID, SUM(Kvantitet*Salgsstykkepris) AS TotalPrice
FROM salgsordrelinie WHERE SOID='SO00006' GROUP BY SOID;
```



SOID	TotalPrice
SO00006	144800.00

For at vise produktsalg i 2020:

Vi opretter en tabel Produktsalg2020

```
CREATE TABLE Produktsalg2020 SELECT PID,
SUM(Kvantitet*Salgsstykkepris) AS TotalPrice FROM
salgsordrelinie WHERE SOID IN (SELECT SOID FROM salgsordrer
WHERE Bestillingsdato BETWEEN 20200101 AND 20201231)
GROUP BY PID WITH ROLLUP;

select * from Produktsalg2020;
```


	PID	TotalPrice
1	P00001	42000.00
2	P00002	67500.00
3	P00003	118700.00
4	P00004	119600.00
5	<null>	347800.00

```
SELECT PID, Produkttype, TotalPrice FROM Produktsalg2020
NATURAL LEFT OUTER JOIN produkter;
```

	PID	Produkttype	TotalPrice
1	P00001	High speed trains	42000.00
2	P00002	Inter-city trains	67500.00
3	P00003	Commuter/regional trains	118700.00
4	P00004	Rapid transit	119600.00
5	<null>	<null>	347800.00

8. SQL programming

8.1 Trigger

```
CREATE TABLE købTid like købsordrelinie;  
ALTER TABLE købTid ADD LogTime TIMESTAMP(6);  
  
CREATE TRIGGER købsordrelinie_After_Insert  
AFTER INSERT ON købsordrelinie FOR EACH ROW  
BEGIN  
INSERT købTid VALUES (New.koid, New.PID,  
New.KøbsordrelinieBeskrivelse, New.Kvantitet, new.Købsstykkepris, NOW(6));  
END;
```

```
INSERT INTO `trainmodels`.`Købsordrelinie`  
(`KOID`, `PID`, `KøbsordrelinieBeskrivelse`, `Kvantitet`, `Købsstykkepris`)  
VALUES  
(`K00006`, `P00006`, `Modern streetcar`, `22`, `10100`);
```

```
SELECT * FROM Købsordrelinie;
```

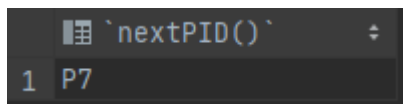
	KOID	PID	KøbsordrelinieBeskrivelse	Kvantitet	Købsstykkepris
1	K00001	P00001	High speed trains	7	4500.00
2	K00002	P00002	Inter-city trains	12	5500.00
3	K00003	P00003	Commuter/regional trains	13	6500.00
4	K00004	P00004	Rapid transit	24	7500.00
5	K00005	P00005	Light rail	40	8500.00
6	K00006	P00006	Modern streetcar	22	10100.00

```
SELECT * FROM købTid;
```

	KOID	PID	KøbsordrelinieBeskrivelse	Kvantitet	Købsstykkepris	LogTime
1	K00006	P00006	Modern streetcar	22	10100.00	2020-04-20 22:04:05.028401000

8.2 Funktionen

```
CREATE FUNCTION nextPID () #for at vise den næste ledige
ProdukterID
  RETURNS VARCHAR(6)
BEGIN
  DECLARE vMaxID VARCHAR (6);
  SELECT MAX(PID) INTO vMaxID FROM produkter;
  RETURN CONCAT ('P',(RIGHT(vMaxID,5)+1));
END;
SELECT nextPID();
```



8.3 Proceduren

```
CREATE TABLE GamleKøb LIKE Købsordrelinie;

CREATE PROCEDURE KøbBackup ()
BEGIN
  DELETE FROM GamleKøb;
  INSERT INTO GamleKøb SELECT * FROM Købsordrelinie;
  DELETE FROM købTid;
END;

CALL KøbBackup;

SELECT * FROM Købsordrelinie;
SELECT * FROM GamleKøb;
SELECT * FROM købTid;
```

```
DROP PROCEDURE KøbBackup; #for at slette den
```

8.4 Eventen

```
CREATE EVENT KøbEvent
ON SCHEDULE EVERY 1 WEEK
STARTS '2020-04-21 00:00:01'
DO CALL KøbEvent;

SET GLOBAL event_scheduler = 1;

SHOW VARIABLES LIKE 'event_scheduler';
```

	Variable_name	Value
1	event_scheduler	ON

```
SET GLOBAL event_scheduler = 0; #for at slette EVENT KøbEvent;
```

	Variable_name	Value
1	event_scheduler	OFF

9. SQL brugerprofiler

Vi har oprettet tre brugere til Database:

Leder	Anders Bentsen	Kan gøre hvad han vil
-------	----------------	-----------------------

```
CREATE USER 'Leder'@'localhost' IDENTIFIED BY '1234';
GRANT ALL ON TrainModels.* TO 'Leder'@'localhost';
SHOW GRANTS FOR 'Leder'@'localhost';
```

```
Grants for Leder@localhost
1 GRANT USAGE ON *.* TO `Leder`@`localhost`
2 GRANT ALL PRIVILEGES ON `trainmodels`.* TO `Leder`@`localhost`
```

Salgschefen	Christine Dahl	Kan gøre hvad han vil med det kun følgende tabler: 1. Kunder 2. Produkter 3. SalgsOrdre 4. Salgsordrelinie
-------------	----------------	--

```
CREATE USER 'Salgschefen'@'localhost' IDENTIFIED BY '1234';
GRANT SELECT, INSERT, UPDATE, DELETE ON trainmodels.salgsordrer TO 'Salgschefen'@'localhost';
GRANT SELECT, INSERT, UPDATE, DELETE ON trainmodels.salgsordrelinie TO 'Salgschefen'@'localhost';
GRANT SELECT, INSERT, UPDATE, DELETE ON trainmodels.kunder TO 'Salgschefen'@'localhost';
GRANT SELECT, INSERT, UPDATE, DELETE ON trainmodels.produkter TO 'Salgschefen'@'localhost';
SHOW GRANTS FOR 'Salgschefen'@'localhost';
```

```
Grants for Salgschefen@localhost
1 GRANT USAGE ON *.* TO `Salgschefen`@`localhost`
2 GRANT SELECT, INSERT, UPDATE, DELETE ON `trainmodels`.`kunder` TO `Salgschefen`@`localhost`
3 GRANT SELECT, INSERT, UPDATE, DELETE ON `trainmodels`.`produkter` TO `Salgschefen`@`localhost`
4 GRANT SELECT, INSERT, UPDATE, DELETE ON `trainmodels`.`salgsordrelinie` TO `Salgschefen`@`localhost`
5 GRANT SELECT, INSERT, UPDATE, DELETE ON `trainmodels`.`salgsordren` TO `Salgschefen`@`localhost`
```

Indkøbschefen	Eva Annette	<p>Kan gøre hvad han vil med det kun følgende tabler:</p> <ol style="list-style-type: none"> 1. Leverandører 2. Produkter 3. Købsordrer 4. Købsordrelinie
---------------	-------------	---

```
CREATE USER 'Indkøbschefen'@'localhost' IDENTIFIED BY '1234';
GRANT SELECT, INSERT, UPDATE, DELETE ON trainmodels.købsordrer TO
'Indkøbschefen'@'localhost';
GRANT SELECT, INSERT, UPDATE, DELETE ON trainmodels.købsordrelinie TO
'Indkøbschefen'@'localhost';
GRANT SELECT, INSERT, UPDATE, DELETE ON trainmodels.leverandører TO
'Indkøbschefen'@'localhost';
GRANT SELECT, INSERT, UPDATE, DELETE ON trainmodels.produkter TO
'Indkøbschefen'@'localhost';
SHOW GRANTS FOR 'Indkøbschefen'@'localhost';
```

```
1 GRANT USAGE ON *.* TO 'Indkøbschefen'@'localhost'
2 GRANT SELECT, INSERT, UPDATE, DELETE ON 'trainmodels'. 'købsordrelinie' TO 'Indkøbschefen'@'localhost'
3 GRANT SELECT, INSERT, UPDATE, DELETE ON 'trainmodels'. 'købsordrer' TO 'Indkøbschefen'@'localhost'
4 GRANT SELECT, INSERT, UPDATE, DELETE ON 'trainmodels'. 'leverandører' TO 'Indkøbschefen'@'localhost'
5 GRANT SELECT, INSERT, UPDATE, DELETE ON 'trainmodels'. 'produkter' TO 'Indkøbschefen'@'localhost'
```

10. Java database adgang

Data manipulation language by Java

```
package Database;

import java.sql.*;
// Java package for accessing and processing data in a relational database
import java.util.Scanner;
// Scanner class found in java.util package gets user console input

// Declaring the Database.QueryTrainmodels class, main method and rightPad method
public class QueryTrainmodels {
    public static void main(String[] args) {
        //Edit the variables below to match a specific setup
        String host = "localhost";
        String port = "3306";
        String username = "root";
        String password = "1234";

        //Do not edit these variables
        String driver = "com.mysql.cj.jdbc.Driver";
        String url = "jdbc:mysql://"
            + host + ":" + port + "/trainmodels" + "?characterEncoding=latin1";
        // ?characterEncoding=latin1 makes codepages in database and Java compatible

        // "try" defines a block to be tested for errors while it is being executed
        try {
            Class.forName(driver);
            Scanner scanner = new Scanner(System.in);
            String columnWidth;
            String sqlQuery;

            System.out.print("Type columnwidth: ");
            // Minimum length for all attribute names and values. For instance type: 10
            columnWidth = scanner.nextLine();
            int j = Integer.parseInt(columnWidth);

            System.out.println("Type sql query: ");
            // Provide user sql query. For instance type: SELECT * FROM medarbejdere;
            sqlQuery = scanner.nextLine();

            Connection connection = DriverManager.getConnection(url, username,
password);
```

```

Statement statement = connection.createStatement();
ResultSet resultSet = statement.executeQuery(sqlQuery);
ResultSetMetaData resultSetMetaData = resultSet.getMetaData();
int columnCount = resultSetMetaData.getColumnCount();

// Print all attribute names
for (int i = 1; i <= columnCount; i++) {
    System.out.print(rightPad(resultSetMetaData.getColumnName(i), j));
}

System.out.println();

// Print all table rows
while (resultSet.next()) {
    // Print all attribute values in the row
    for (int i = 1; i <= columnCount; i++) {
        System.out.print(rightPad(resultSet.getString(i), j));
    }
    System.out.println();
}

connection.close();
// "catch" defines a block to be executed, if an error occurs in the try block
} catch (Exception e) {
    e.printStackTrace();
}
}

// rightPad method pads short attributenames and values to same columnwidth
public static String rightPad(String str, int num) {
    return String.format("%1${-}" + num + "s", str);
}
}

```

Run: QueryTrainmodels

"C:\Program Files (x86)\Java\jdk1.8.0_231\bin\java.exe" ...

Type columnwidth: 17

Type sql query:

```
select * from medarbejdere;
```

MID	Efternavn	Fornavn	Adresse	Postnummer	Bynavn	Land	Mobil
M20001	Bentsen	Anders	Elmedalsvej 1	4200	Slagelse	Danmark	+45 54 78 95 25
M20002	Jacobsen	Lise	Hulgårdsvvej 131	2400	København	Danmark	+45 34 25 98 11
M20003	Thomsen	Frederik	Solsortvej 131	2000	Frederiksberg	Danmark	+45 70 22 48 78
M20004	Petersen	Malte	Skovduetien 31	2400	København	Danmark	+45 40 66 54 19
M20005	Dahl	Christine	Tinghøjvej 46	2860	Søborg	Danmark	+45 71 47 23 30
M20006	Kristensen	Hans	Måløv parkvej 2	2750	Ballerup	Danmark	+45 78 95 22 18

Vores DAL (Data Access Layer) viser således vores forbindelse mellem vores server og javakode. Host, Port, User og Password indskrives i koden, for at oprette databasetilknytning, ligeledes skrives de rette parameter for ind i koden, for fremvisning af de ønskede informationer.

Det vises i de overstående skærmudklip, hvordan koden er implementeret og kommenteret.

11. Deltagernes indsats i gruppearbejdet

Deltagere i gruppe 14	1. DB tabel skemaer og brugerprofiler	2. Entity Relationship Diagram (UML notation)	3. Extended Entity Relationship Diagram	4. Database instans	5. Normalformer	6. SQL tabelmodifikationer	7. SQL forespørgsler	8. SQL programmering	9. SQL brugerprofiler	10. Java database adgang
Sulaiman kasas	15%	30%	20%	14%	10%	8%	10%	20%	5%	70%
Reza Haddang	15%	15%	20%	12%	10%	70%	10%	20%	5%	10%
Amer Biro	30%	30%	20%	50%	60%	5%	5%	20%	10%	5%
Mohammad Zamani	25%	15%	20%	12%	10%	7%	70%	20%	10%	10%
Bashar Khaled Bdewi	15%	10%	20%	12%	10%	10%	5%	20%	70%	10%