



Technical University of Denmark

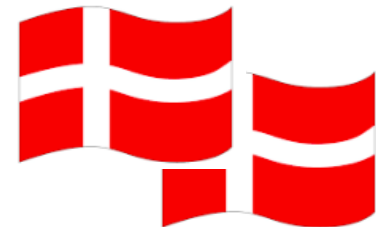
DTU Compute

Department of Applied Mathematics and Computer Science

Database project solution

En løsning til det obligatorisk database projekt

©Flemming Schmidt



1. Forside og indholdsfortegnelse

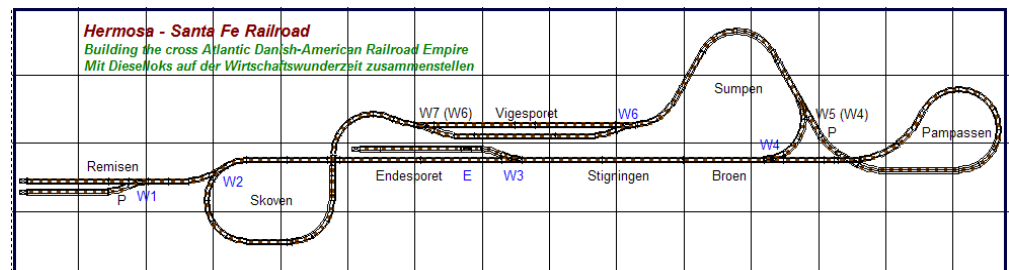
■ Forside oplysninger

- Gruppenummer: 00
- Kursusnummer: 02327
- Kursusnavn: Indledende databaser og database programmering
- Projekt navn: TrainModels
- Afleveringsdato: 21. april 2020
- Deltager navn: Flemming Schmidt
- DTU nummer: 3685046307
- Tilhørende fil: TrainModels.sql
- Deltagerbillede:



■ Indholdsfortegnelse

1. Forside og indholdsfortegnelse
2. DB tabel skemaer og brugerprofiler
3. Entity Relationship diagram (UML notation)
4. Extended Entity Relationship diagram (Crow's Foot notation)
5. Database instans
6. Normalformer
7. SQL tabelmodifikationer
8. SQL forespørgsler
9. SQL programmering
10. SQL brugerprofiler
11. Java database adgang
12. Deltagernes indsats i gruppearbejdet



2. Database tabel skemaer

Database tabel skemaer (10)

medarbejdere (mID, efternavn, fornavn, adresse, postnr, bynavn, land, mobil, email, månedsløn, timeløn, lønkonto, jobtitel, afdeling, chefID, startdato, slutdato)

timerapporter (mID, arbejdsdato, arbejdstimer, status, statusID, bemærkning)

kunder (kID, efternavn, fornavn, adresse, postnr, bynavn, land, mobil, email, kreditgrænse)

leverandører (IID, leverandør, efternavn, fornavn, adresse, postnr, bynavn, land, mobil, email)

produkttyper (produkttype, beskrivelse, html, billede)

produkter (pID, produkt, produkttype, beskrivelse, html, billede, IID, købspris, salgspris, lagerantal, fralagerdage, fraleverandørdage)

salgsordrer (soID, kID, bestillingsdato, afsendelsesdato, faktureringsdato, betalingsdato)

salgsordrelinier (soID, pID, linienr, linieantal, liniestkpris)

købsordrer (koID, IID, bestillingsdato, modtagelsesdato, betalingsdato)

købsordrelinier (koID, pID, linienr, linieantal, liniestkpris)

Views (2) & materialized views (2)

ansatte (afdeling, efternavn, fornavn, mID, mobil, email, jobtitel, chefefternavn, cheffornavn)

timesedler (fornavn, efternavn, arbejdsdato, arbejdstimer, status, statusfornavn, statusefternavn, bemærkning)

lønudbetalinger (løndato, mID, efternavn, fornavn, månedsløn, timer, timeløn, variabel løn, lønudbetaling, lønkonto)

produktsalgYTD (kørselsdato, pID, produkt, subtotal)



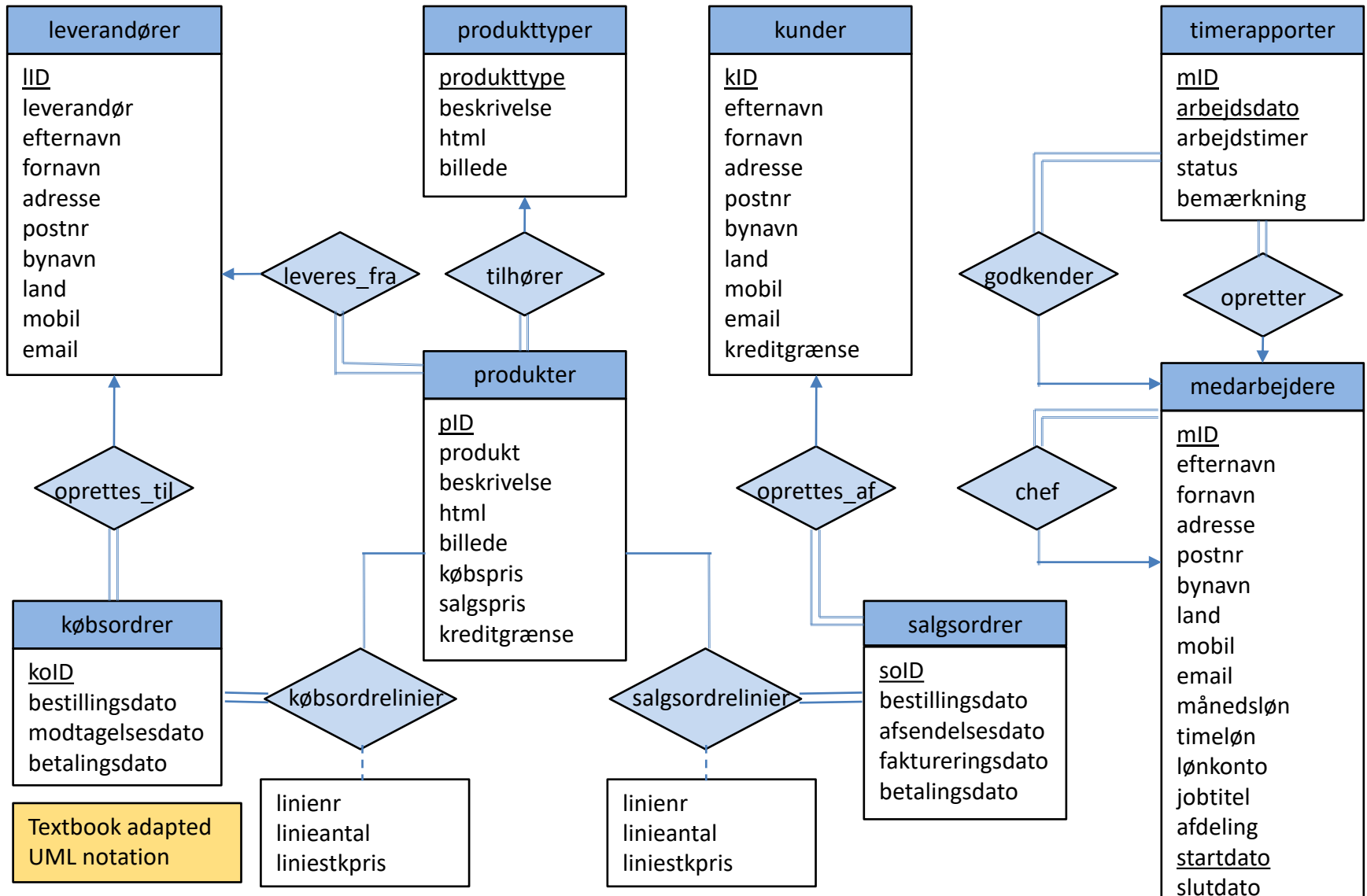
2. Database brugerprofiler

- Brugerprofiler; Medarbejdernes tilgang til tabellerne

Tabeller	Direktør	Admin-afdelingen	Salgschef	Salgsafdelingen	Indkøbschef	Indkøbsafdelingen
medarbejdere	Skrive	Skrive på nær: månedsløn timeløn lønkonto	Læse på nær: månedsløn timeløn lønkonto	Læse på nær: månedsløn timeløn lønkonto	Læse på nær: månedsløn timeløn lønkonto	Læse på nær: månedsløn timeløn lønkonto
timerapporter	Skrive	Skrive egne	Skrive egne og godkende/afvise medarbejderes	Skrive egne	Skrive egne og godkende/afvise medarbejderes	Skrive egne
produkttyper	Skrive	Skrive				
produkter	Skrive	Skrive				
kunder	Skrive	Skrive	Skrive	Læse		
salgsordrer	Skrive	Skrive	Skrive	Skrive		
salgsordrelinier	Skrive	Skrive	Skrive	Skrive		
leverandører	Skrive	Skrive			Skrive	Læse
købsordrer	Skrive	Skrive			Skrive	Skrive
købsordrelinier	Skrive	Skrive			Skrive	Skrive

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

3. Entity Relationship diagram

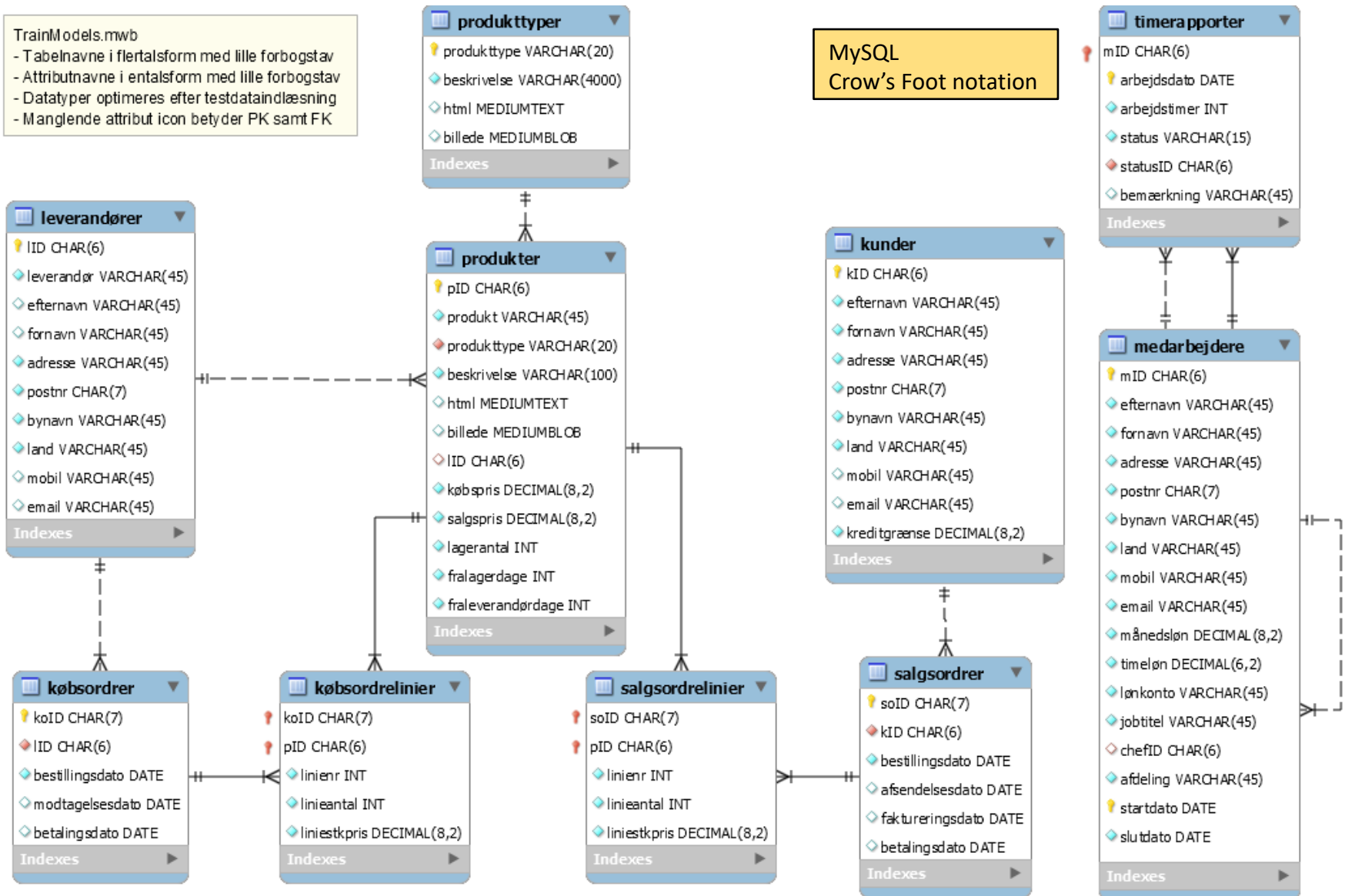


4. Extended Entity Relationship diagram

TrainModels.mwb

- Tabelnavne i flertalsform med lille for bogstav
- Attributnavne i entalsform med lille for bogstav
- Datatyper optimeres efter testdataindlæsning
- Manglende attribut icon betyder PK samt FK

MySQL
Crow's Foot notation



5. Database instans (1 af 3)

select * from medarbejdere

mID	efternavn	fornavn	adresse	postnr	bynavn	land	mobil	email	månedsløn	timeløn	lønkonto	jobtitel	chefID	afdeling	startdato	slutdato
M20001	Bentsen	Anders	Tranehaven 76, 2.tv	DK-2605	Brøndby	Danmark	+45 70 62 35 21	ab@gmail.com	75000.00	500.00	4434 4440123456	direktør	M20001	admin	2017-01-01	9999-12-31
M20002	Dahl	Christine	Æblevej 6	DK-2400	Kbh NV	Danmark	+45 21 24 37 75	cd@gmail.com	36000.00	235.00	4434 4440234561	salgschef	M20001	salg	2017-01-01	2019-01-01
M20002	Dahl	Christine	Æblevej 6	DK-2400	Kbh NV	Danmark	+45 21 24 37 75	cd@gmail.com	40000.00	275.00	4434 4440234561	salgschef	M20001	salg	2019-01-01	9999-12-31
M20003	Kristensen	Hans	Jægersborg Alle 185	DK-2500	Gentofte	Danmark	+45 35 27 19 02	hk@gmail.com	36000.00	235.00	4434 4440345612	sælger	M20002	salg	2017-08-01	9999-12-31
M20004	Hansen	Ole	C. N. Petersensvej Vej 25	DK-2000	Frederiksberg	Danmark	+45 21 28 73 52	oh@gmail.com	34000.00	220.00	4434 4440456123	sælger	M20002	salg	2018-10-01	9999-12-31
M20005	Jensen	Leif	Birkebakken 24	DK-2750	Ballerup	Danmark	+45 70 24 63 91	lej@gmail.com	0.00	185.00	4434 4440561234	salgsassistent	M20002	salg	2017-10-01	9999-12-31
M20006	Frederiksen	Eva Annette	Kystvej 23	DK-3050	Humblebæk	Danmark	+45 26 43 63 29	ef@gmail.com	35000.00	230.00	4434 4440612345	indkøbschef	M20001	indkøb	2018-08-01	9999-12-31
M20007	Mortensen	Hans Peter	Tirsbækvej 17	DK-2720	Vanløse	Danmark	+45 25 73 16 21	hm@gmail.com	0.00	180.00	4434 4440456789	salgsassistent	M20002	salg	2018-01-01	9999-12-31
M20008	Jacobsen	Lise	Moseholmene 14	DK-2610	Rødovre	Danmark	+45 21 39 37 28	lj@gmail.com	0.00	190.00	4434 4440567894	adminassistent	M20001	admin	2018-04-01	9999-12-31
M20009	Thomsen	Frederik	Tranehaven 84	DK-2605	Brøndby	Danmark	+45 25 91 35 03	ft@gmail.com	0.00	180.00	4434 4440678945	adminassistent	M20001	admin	2019-08-01	9999-12-31
M20010	Petersen	Morten	Nørre Søgade 19 3. Sal	DK-1370	Kbh K	Danmark	+45 70 42 19 75	mop@gmail.com	0.00	180.00	4434 4440789456	salgsassistent	M20002	salg	2018-04-01	9999-12-31
M20011	Alberg	Viggo	Nørre Søgade 19 3. sal	DK-1370	Kbh K	Danmark	+45 21 58 43 29	va@gmail.com	28000.00	190.00	4434 4440894567	indkøber	M20006	indkøb	2019-01-01	9999-12-31
M20012	Flyvholm	Ea	Nørregårdsvej 185	DK-2610	Rødovre	Danmark	+45 70 14 72 39	eaf@gmail.com	0.00	180.00	4434 4440945678	indkøbsassistent	M20006	indkøb	2019-03-01	9999-12-31
M20013	Petersen	Malte	Hørbakken 31	DK-2765	Smørum	Danmark	+45 36 84 27 15	mp@gmail.com	45000.00	300.00	4434 4440654321	produktchef	M20001	admin	2017-01-01	9999-12-31
M20014	Leth	Iben	Nystedvej 29, 1. th	DK-2500	Valby	Danmark	+45 21 92 35 03	il@gmail.com	0.00	185.00	4434 4440543216	salgsassistent	M20002	salg	2019-08-01	9999-12-31
M20015	Berg	Lisa	Tranehaven 65, 1. tv	DK-2605	Brøndby	Danmark	+45 36 14 76 98	lb@gmail.com	32000.00	205.00	4434 4440432165	indkøbschef	M20001	indkøb	2017-01-01	2018-08-01

select * from timerapporter

mID	arbejdsdato	arbejdstimer	status	statusID	bemærkning
M20005	2020-01-06	4	godkendt	M20002	Ingen
M20005	2020-01-07	2	godkendt	M20002	Ingen
M20008	2020-01-06	4	godkendt	M20001	Mails og oprydning
M20008	2020-01-07	4	oprettet	M20008	Produktgennemgang
M20009	2020-01-06	2	godkendt	M20001	Ingen
M20009	2020-01-07	2	oprettet	M20009	Ingen
M20011	2020-01-06	3	afvist	M20006	Arbejdede du ikke 6 timer?
M20011	2020-01-07	6	godkendt	M20006	Ingen



select * from kunder

kID	efternavn	fornavn	adresse	postnr	bynavn	land	mobil	email	kreditgrænse
K30001	Larsen	Bent	Hermosavej 27	DK-2605	Brøndby	Danmark	+45 28 54 36 47	bl51@gmail.com	100000.00
K30002	Svendsen	Kurt	Bygaden 28	DK-2605	Brøndby	Danmark	+45 86 25 64 38	ks23@gmail.com	100000.00
K30003	Bjerg	Arne	Nystedvej 24	DK-2500	Valby	Danmark	+45 35 27 54 38	ab17@gmail.com	150000.00
K30004	Schnack	Freddy	Kalvehaven 6	DK-4690	Haslev	Danmark	+45 21 45 37 32	fs12@gmail.com	100000.00

5. Database instans (2 af 3)

select * from leverandører

IID	leverandør	efternavn	fornavn	adresse	postnr	bynavn	land	mobil	email
L40001	Gebr. Märklin & Cie. GmbH	Schmidt	Herman	Stuttgarter Strasse 55-57	D-73033	Göppingen	Germany	+49 (0) 71 61 608-0	marklin@gmail.com
L40002	Modelleisenbahn GmbH	Ahler	Franz	Plainbachstr. 4	5101	Bergheim	Austria	+43 (0) 820-200668	fleischmann@gmail.com
L40003	PostNord Danmark	Hannibal	Michael	Hedegaardsvej 48	2300	Kbh S	Danmark	+45 38 56 53 91	mh25@gmail.com
L40004	GLS Danmark A/S	Christiansen	Lublav	Kokmose 7	6000	Kolding	Danmark	+45 79 84 36 55	lc23@gmail.com

select * from produkttyper

produkttype	beskrivelse	html	billede
damptog	Et damplokomotiv er et lokomotiv der drives frem af en dampmaskine placeret på lokomotivet. Dampmaskinen påvirker drivakslerne ved hjælp af trækstænger (gangtøjet) som driver hjulene rundt. Det første damplokomotiv blev bygget af Richard Trevithick, og kørte første gang i England den 21. februar 1804, det tog dog nogle år inden damplokomotiver var effektive og økonomiske nok til at kunne afløse heste og okser der, på dette tidspunkt, anvendtes til at trække vognene. Et gennembrud kom der først i 1829 med the Rainhill Trials som var en konkurrence der skulle finde det bedste lokomotiv til jernbanen mellem Liverpool og Manchester i England. Denne konkurrence vandt det senere berømte lokomotiv Rocket bygget af George og Robert Stephenson. Den første jernbanestrækning i kongeriget Danmark (dvs. fraregnet jernbanerne i hertugdømmerne Slesvig-Holsten) blev indviet lørdag den 26. juni 1847. Til denne havde Sjællandske Jernbane Selskab (SJS) indkøbt fem damplokomotiver hos Sharp Brothers & co. i Manchester. Det første, der ankom fra England i august 1846, døbttes Odin, og anvendtes ved banens bygning. (De andre døbttes "København", "Sjælland", "Danmark", og "Korsør").		
dieseltog	Et diesellokomotiv er et lokomotiv, der er forsynet med en dieselmotor til fremdrift og derfor anvender dieselolie som brændstof. Diesellokomotiver inddeles i underkategorier efter transmissionstype, nemlig dieselmekaniske, dieselhydrauliske og diesel-elektriske lokomotiver. I et dieselmekanisk lokomotiv driver motoren akslerne gennem en gearkasse, evt. en automatgearkasse. I et dieselhydraulisk lokomotiv driver motoren akslerne gennem en hydraulisk momentomformer. I Danmark kendes f.eks. MH rangerlokomotivet fra Henschel/Tyskland Frichs/Danmark. Danske diesel-hydrauliske lokomotiver inkluderer: Litra MH og Litra MK. Diesel-elektriske lokomotiver fungerer i princippet som et elektrisk lokomotiv ved at akslerne drives direkte eller indirekte af kraftige elmotorer (banemotorerne). Disse motorer forsynes med elektricitet fra en elektrisk generator, som drives af en dieselmotor, og dette princip kendes f.eks. fra de klassiske MY lokomotiver fra NOHAB/Sverige. Stort set alle diesel-strækningslokomotiver (ikke rangerlokomotiver) i Danmark er diesel-elektriske. Danske diesel-elektriske lokomotiver inkluderer: Litra Mx, Litra MY, Litra MZ, Litra ME og Litra MT.		
eltog	Et ellokomotiv er et lokomotiv, der drives af elektricitet fra typisk køreledninger eller strømskinner – alternativt batterier. Eksempler på danske ellokomotiver er Litra EA og Litra EG. Litra EA er typebetegnelsen for 22 ellokomotiver erhvervet af DSB, men hvoraf halvdelen i dag er i drift i Østeuropa. EA blev anskaffet i forbindelse med elektrificeringen af de danske hovedbaner og er leveret i to serier i hhv. 1984-1986 (3001-3010) og 1992-1993 (3011-3022). EA sættes i drift på Kystbanen i 1986 og på Vestbanen i 1987-1989.[2] Fra Storebæltsforbindelsens åbning i 1997 fremførte EA også godstog, indtil litra EG leveredes fra 2000. Efterhånden som nyere eltogset kom til, reduceredes driften med EA til et par lokomotiver, der fremførte internationale nattog mellem København og Padborg, indtil disse ophørte i 2014. DSB's 6 tilbageblevne EA'ere er efterfølgende blevet ombygget, så de siden 2015 har kunnet fremføre dobbeltdekkervogne mellem Østerport og Ringsted (Slagelse). Litra EA's banemotorer er asynkrone el-motorer (vekselstrømsmotorer). Den tidligere norm for såvel elektriske som diesel-elektriske lokomotiver havde været jævnstrøms-motorer. Skiftet skyldtes, at man nu endelig havde udviklet en pålidelig styring til variabel ydelse fra vekselstrøms-motorer, og at jævnstrøms-motorer efterhånden ikke kunne levere samme ydelse som vekselstrøms-motorer.		
pakkestransport	Pakker udbringes af en række leverandører til varierende priser		
togskinner	Modeljernbaneanlæg anlægges typisk enten på faste plader eller som en rammekonstruktion, eller det sammensættes af moduler. Moduler er aktuelle, hvor et anlæg skal kunne bygges op, tages ned og flyttes forholdsvis nemt og hurtigt. Desuden har de den fordel, at de kan kombineres efter behov. Til gengæld følger de ofte standardmål, hvilket lægger begrænsninger på udfoldelsesmuligheder, ligesom der skal tages hensyn til, at de skal kunne transporteres. Faste anlæg er derimod beregnet til at blive, hvor de er, og alle dele er som navnet siger fast forbundne. Det sparer arbejdet med flytninger men mindsker muligheden for fleksibilitet. Afhængig af forholdene kan faste anlæg laves midt i et lokale eller langs væggene, evt. med "tunger" ud i lokalet.		
togvogne	En jernbanevogn eller togvogn er et skinnekøretøj uden egen fremdrift, som anvendes til transport på jernbane. Jernbanevogne trækkes af et eller flere lokomotiver. Man skelner mellem godsvogne, der anvendes til transport af gods, og personvogne, der primært anvendes til persontransport i den kollektive trafik. Tidligere fandtes der et utal af specielle jernbanevogne. Blandt andet havde postvæsenet et antal Postbureau- og postpakvogne.		



5. Database instans (3 af 3)

select * from produkter

pID	produkt	produkttype	beskrivelse	html	billede	IID	købspris	salgspris	lagerantal	fralagerdage	fraleverandordage
P50001	DSB MY1100	dieseltog	Flot Märklin DSB lokomotiv fra 1998. Lokomotivet har et flot sort og rødt farvedesign.			L40001	1250.00	2499.00	10	2	5
P50002	SBB klasse C 5/6 "Elefant"	damp tog	Lokomotivet har kontrolleret højeffektiv fremdrift, 5 trækaksler og trækdek.			L40001	1450.00	3599.00	12	2	5
P50003	BR 491 DB	eltog	Class 491 toget bliver også kaldet "glas toget" grundet de store glas partier.			L40001	1425.00	2649.00	0	2	5
P50004	Ae 6/6 "Kantonslok", SBB	eltog	Succeshistorien for Ae 6/6, hvoraf 120 blev leveret til SBB, begyndte allerede i 1955.			L40002	800.00	1555.00	5	2	7
P50005	Märklin 47803 Maersk containervogn	togvogne	Vognene er type Y 25 og 2 40-foot Maersk containere			L40001	420.00	799.00	23	2	7
P50006	Märklin 46913 Eaos 106 freight car	togvogne	Flot Märklin togvogn i rødbrun farve og last med skrotmetal			L40001	475.00	849.99	19	2	7
P50007	Märklin 48936 Type g 10 Boxcar	togvogne	Märklin boxcar af den privatejede Coca Cola vogn. Kørt i Holland			L40001	135.00	269.95	27	2	7
P50008	Märklin 48935 Kølevogn Coca Cola	togvogne	Coca Colass reklamevogn, som kørte på De Danske Statsbaner			L40001	135.00	269.95	18	2	7
P50009	Märklin 24188 lige C skinne	togskinner	H0 Lige spor 188,3mm			L40001	11.00	21.00	248	2	7
P50010	Märklin 24172 lige C skinne	togskinner	H0 Lige spor 171,7mm			L40001	11.00	21.00	173	2	7
P50011	Märklin 24094 lige C skinne	togskinner	H0 Lige spor 94,2mm			L40001	13.00	24.50	55	2	7
P50012	Märklin 24130 sving/buet C skinne	togskinner	H0 Kurvespor R1=360mm 30°			L40001	12.00	21.00	212	2	7
P50013	Märklin 24611 C skinne skiftespor	togskinner	H0 Sporskifte venstre			L40001	76.00	149.95	73	2	7
P50014	Märklin 24612 C skinne skiftespor	togskinner	H0 Sporskifte højre			L40001	76.00	149.00	84	2	7
P59001	PostNord	pakketransport	Udbringning af pakker til døren			L40003	75.00	75.00	0	0	0
P59002	GLS	pakketransport	Udbringning af pakker til døren			L40004	65.00	65.00	0	0	0

select * from salgsordrer

soID	kID	bestillingsdato	afsendelsesdato	faktureringsdato	betalingsdato
SO60002	K30003	2020-01-06	2020-01-06	2020-01-06	2020-01-06
SO60003	K30002	2020-01-06			
SO60004	K30003	2020-01-08			

select * from salgsordrelinier

soID	pID	linienr	linieantal	liniestkpris
SO60002	P50001	1	1	2499.00
SO60002	P50003	2	1	2649.00
SO60002	P59001	3	1	75.00
SO60003	P50001	1	1	2499.00
SO60003	P59002	2	1	65.00
SO60004	P50002	1	1	3599.00
SO60004	P59001	2	1	75.00

select * from købsordrer

koID	IID	bestillingsdato	modtagelsesdato	betalingsdato
KO70005	L40001	2020-01-06		
KO70006	L40002	2020-01-06		

select * from købsordrelinier

koID	pID	linienr	linieantal	liniestkpris
KO70005	P50001	1	10	1250.00
KO70005	P50003	2	10	1425.00
KO70005	P59001	3	1	75.00
KO70006	P50004	1	15	800.00
KO70006	P59002	2	1	75.00

6. Normalformer

■ Tabellerne i TrainModels er på 3NF?

- 1. Normal Form betingelser
 - Alle tabel værdier er atomare
 - Alle værdier i samme attribut er fra samme domæne.
 - Hver attribut i en tabel har et unikt navn
 - Hver række har en unik primærnøgle
 - Rækkefølgen af rækker og kolonner har ingen signifikans.
- 2NF er 1NF plus betingelsen
 - En ikke primær attribut afhænger af alle attributterne i den primære nøgle.
- 3NF er 2NF plus betingelsen
 - En ikke primær attribut afhænger ikke transitivt, via andre attributter, af den primære nøgle.

■ Kun binære entitets relationer

- Ingen tertiære relationer mellem entiteter

■ Selvstændig tabel for postnr/bynavn

- Kunne godt være tilføjet, og så være i relation med tabellerne kunder, leverandører og medarbejdere, disse tabeller er kun på 2NF.

■ Tabellerne salgs-/købsordrelinier

- Supernøgle: {sold, pID, linienr}
- Kandidatnøgler: {sold, pID} og {sold, linienr}
- Valgt primærnøgle: {sold, pID}
Linienr burde vel helt udelades og håndteres i brugergrænsefladen?

■ Tabellen medarbejdere

- Er i en relation til sig selv:
`FOREIGN KEY (chefID) REFERENCES
medarbejdere (mID)
ON DELETE SET NULL
ON UPDATE CASCADE)`

6. Normalformer

Eksempel: Projektions Normalisering

Tabellen "salg" er på 1NF men ikke på 2NF, idet kunde oplysninger lagres gentaget på hver salgsordre!

```
SELECT * FROM salg;
```

soID	kID	dato	navn	adresse
SO60002	K30003	2020-01-06	Bjerg	Nystedvej 24
SO60003	K30002	2020-01-06	Svensen	Bygaden 28
SO60004	K30003	2020-01-08	Bjerg	Nystedvej 24

Salg erstattes derfor af to tabeller på 2NF

```
SELECT DISTINCT soID, kID, dato FROM salg;
```

soID	kID	dato
SO60002	K30003	2020-01-06
SO60003	K30002	2020-01-06
SO60004	K30003	2020-01-08

```
SELECT DISTINCT kID, navn, adresse FROM salg;
```

kID	navn	adresse
K30003	Bjerg	Nystedvej 24
K30002	Svensen	Bygaden 28

Normalisering sker ved to projektioner. Salg kan altid genskabes med en natural join af de to tabeller.

Eksempel: Armstrong Normalisering

salg(soID, kID, dato, navn, adresse)

Funktionelle afhængigheder: {soID → kID},
{soID → dato}, {kID → navn, adresse}

1. soID → soID Regel 4 Selfdetermination
2. soID → kID Funktionel afhængighed
3. soID → dato Funktionel afhængighed
4. kID → kID Regel 4 Selfdetermination
5. kID → navn Regel 5 Decomposition
6. kID → adresse Regel 5 Decomposition
7. soID → soID, kID, dato Regel 6 Union
8. kID → kID, navn, adresse Regel 6 Union
9. Kandidat/Primærnøgler: soID og kID
10. Tabeller:
 salgsortrer(soID, kID, dato)
 kunder(kID, navn, adresse)

Domænet for timerapporter(status)

Et bedre domæne end VARCHAR(15) ville være:

```
Status ENUM (  
  'oprettet', 'godkendt', 'afvist', 'udbetalt'),
```

Så MySQL kan lave test i stedet for Java programmet

7. SQL tabelmodifikationer

■ Registrering af et salg

Den 12. marts ønsker Arne Bjerg (K30003) at købe:

10 stk. Märklin lige C skinne (P50009) á 21.00 Kr.

12 stk. Märklin buet C skinne (P50012) á 21.00 Kr.

1 stk. Märklin venstre skiftespor (P50013) á 149.95

1 stk. GLS pakketransport (P59002) á 65.00 Kr.

Næste ledige sold er SO60005

Indsættelse af salgsordre og salgsordrelinier:

```
INSERT salgsordrer VALUES  
( 'SO60005', 'K30003', '2020-03-12', NULL, NULL,  
NULL );
```

```
INSERT salgsordrelinier VALUES  
( 'SO60005', 'P50009', '1', '10', '21.00' ),  
( 'SO60005', 'P50012', '2', '12', '21.00' ),  
( 'SO60005', 'P50013', '3', '1', '149.95' ),  
( 'SO60005', 'P59002', '4', '1', '65.00' );
```

Arne Bjerg oplyser nu, at skiftesporet skulle være et højre (i.e. P50013 skal være P50014 á 149.00 Kr.)

```
UPDATE salgsordrelinier SET pID='P50014',  
liniestkpris='149.00' WHERE soID='SO60005' AND  
pID='P50013';
```

Arne Bjerg oplyser nu, at han selv vil afhente pakken og derfor ikke ønsker GLS pakketransport.

```
DELETE FROM salgsordrelinier  
WHERE soID='so60005' and pID='P59002';
```

Den endelige salgsordre med salgsordrelinier:

```
SELECT soID, kID, bestillingsdato FROM  
salgsordrer WHERE soID='SO60005';
```

soID	kID	bestillingsdato
SO60005	K30003	2020-03-12

```
SELECT * FROM salgsordrelinier WHERE  
soID='SO60005';
```

soID	pID	liniennr	linieantal	liniestkpris
SO60005	P50009	1	10	21.00
SO60005	P50012	2	12	21.00
SO60005	P50014	3	1	149.00

8. SQL forespørgsler

■ Beregning af en ordres salgsværdi

Beregning af Arne Bjerg ordren (SO60005):

```
SELECT soID, pID, linienr, linieantal,
liniestkpris, linieantal*liniestkpris AS
subtotal FROM salgsordrelinier WHERE
soID='SO60005';
```

soID	pID	linienr	linieantal	liniestkpris	subtotal
SO60005	P50009	1	10	21.00	210.00
SO60005	P50012	2	12	21.00	252.00
SO60005	P50014	3	1	149.00	149.00

```
SELECT soID, SUM(linieantal*liniestkpris)
AS subtotal FROM salgsordrelinier WHERE
soID='SO60005' GROUP BY soID;
```

soID	subtotal
SO60005	611.00

■ Beregning af produksalg i 2020

Produktomsætning og samlet omsætning:

```
CREATE TABLE S SELECT pID,
SUM(linieantal*liniestkpris) AS subtotal
FROM salgsordrelinier WHERE soID IN
(SELECT soID FROM salgsordrer WHERE
bestillingsdato BETWEEN 20200101 AND
20201231) GROUP BY pID WITH ROLLUP;
```

```
SELECT * from S;
```

pID	subtotal
P50001	4998.00
P50002	3599.00
P50003	2649.00
P50009	210.00
P50012	252.00
P50014	149.00
P59001	150.00
P59002	65.00
NULL	12072.00

```
SELECT pID, produkt, subtotal FROM
S NATURAL LEFT OUTER JOIN produkter;
```

pID	produkt	subtotal
P50001	DSB MY1100	4998.00
P50002	SBB klasse C 5/6 "Elefant"	3599.00
P50003	BR 491 DB	2649.00
P50009	Marklin 24188 lige C skinne	210.00
P50012	Marklin 24130 sving/buet C skinne	252.00
P50014	Marklin 24612 C skinne skiftespor	149.00
P59001	PostNord	150.00
P59002	GLS	65.00
NULL	NULL	12072.00

9. SQL programmering

■ Indhold

- View ansatte; En let adgang for alle til information om medarbejdere
- View timesedler; En let adgang til timerapporter
- Trigger timerapporter_Before_Insert; Test af nye timerapporter før oprettelse
- Funktion nextsold; Finder næste ledige sold
- Procedure lønudbetaling; For perioden startdato til slutdato
- Procedure lønudbetaling som transaktion; For perioden startdato til slutdato
- Event TrainModelsHero; Månedlig udtrækning af én medarbejderbonus
- Event produktsalg_i_år; Beregning af det samlede produktsalg hver måned



View ansatte

■ View ansatte; En let adgang for alle til information om medarbejdere

```
CREATE VIEW ansatte AS
SELECT T.afdeling, T.efternavn, T.fornavn, T.mID, T.mobil, T.email, T.jobtitel,
       S.efternavn AS chefefternavn, S.fornavn AS cheffornavn
FROM medarbejdere AS T JOIN medarbejdere AS S
WHERE T.chefID=S.mID AND curdate() BETWEEN T.startdato AND T.slutdato AND
       curdate() BETWEEN S.startdato AND S.slutdato
ORDER BY T.afdeling, T.efternavn;
```

afdeling	efternavn	fornavn	mID	mobil	email	jobtitel	chefefternavn	cheffornavn
admin	Bentsen	Anders	M20001	+45 70 62 35 21	ab@gmail.com	direktør	Bentsen	Anders
admin	Jacobsen	Lise	M20008	+45 21 39 37 28	lj@gmail.com	adminassistent	Bentsen	Anders
admin	Petersen	Malte	M20013	+45 36 84 27 15	mp@gmail.com	produktchef	Bentsen	Anders
admin	Thomsen	Frederik	M20009	+45 25 91 35 03	ft@gmail.com	adminassistent	Bentsen	Anders
indkøb	Alberg	Viggo	M20011	+45 21 58 43 29	va@gmail.com	indkøber	Frederiksen	Eva Annette
indkøb	Flyvholm	Ea	M20012	+45 70 14 72 39	eaf@gmail.com	indkøbsassistent	Frederiksen	Eva Annette
indkøb	Frederiksen	Eva Annette	M20006	+45 26 43 63 29	ef@gmail.com	indkøbschef	Bentsen	Anders
salg	Dahl	Christine	M20002	+45 21 24 37 75	cd@gmail.com	salgschef	Bentsen	Anders
salg	Hansen	Ole	M20004	+45 21 28 73 52	oh@gmail.com	sælger	Dahl	Christine
salg	Jensen	Leif	M20005	+45 70 24 63 91	lej@gmail.com	salgsassistent	Dahl	Christine
salg	Kristensen	Hans	M20003	+45 35 27 19 02	hk@gmail.com	sælger	Dahl	Christine
salg	Leth	Iben	M20014	+45 21 92 35 03	il@gmail.com	salgsassistent	Dahl	Christine
salg	Mortensen	Hans Peter	M20007	+45 25 73 16 21	hm@gmail.com	salgsassistent	Dahl	Christine
salg	Petersen	Morten	M20010	+45 70 42 19 75	mop@gmail.com	salgsassistent	Dahl	Christine

View timesedler

■ View timesedler; En let adgang til timerapporter

```
CREATE VIEW timesedler AS
SELECT T.fornavn, T.efternavn, S.arbejdsdato, S.arbejdstimer, S.status,
       U.fornavn AS statusfornavn, U.efternavn AS statusefternavn, bemærkning
FROM timerapporter AS S, medarbejdere AS T, medarbejdere AS U
WHERE S.mID=T.mID AND S.statusID=U.mID AND curdate() BETWEEN T.startdato AND T.slutdato
      AND curdate() BETWEEN U.startdato AND U.slutdato
ORDER BY by arbejdsdato, T.fornavn, T.efternavn;
```

fornavn	efternavn	arbejdsdato	arbejdstimer	status	statusfornavn	statusefternavn	bemærkning
Frederik	Thomsen	2020-01-06	2	godkendt	Anders	Bentsen	Ingen
Leif	Jensen	2020-01-06	4	godkendt	Christine	Dahl	Ingen
Lise	Jacobsen	2020-01-06	4	godkendt	Anders	Bentsen	Mails og oprydning
Viggo	Alberg	2020-01-06	3	afvist	Eva Annette	Frederiksen	Arbejdede du ikke 6 timer?
Frederik	Thomsen	2020-01-07	2	oprettet	Frederik	Thomsen	Ingen
Leif	Jensen	2020-01-07	2	godkendt	Christine	Dahl	Ingen
Lise	Jacobsen	2020-01-07	4	oprettet	Lise	Jacobsen	Produktgennemgang
Viggo	Alberg	2020-01-07	6	godkendt	Eva Annette	Frederiksen	Ingen

Trigger timerapporter_Before_Insert

- Trigger timerapporter_Before_Insert; Tester nye timerapporter før oprettelse

```
CREATE TRIGGER timerapporter_Before_Insert
BEFORE INSERT ON timerapporter FOR EACH ROW
BEGIN
  IF NOT (NEW.mID IN (SELECT mID FROM ansatte))
    THEN SIGNAL SQLSTATE 'HY000' SET MYSQL_ERRNO = 1525,
    MESSAGE_TEXT = 'mID er ukendt!';
  END IF;

  IF NOT (NEW.mID = NEW.statusID)
    THEN SIGNAL SQLSTATE 'HY000' SET MYSQL_ERRNO = 1525,
    MESSAGE_TEXT = 'Arbejdstimer kan kun rapporteres for een selv!';
  END IF;

  IF NOT (NEW.status = 'oprettet')
    THEN SIGNAL SQLSTATE 'HY000' SET MYSQL_ERRNO = 1525,
    MESSAGE_TEXT = '"oprettet" skal være status for nye rækker!';
  END IF;

  IF NOT (NEW.arbejdstimer BETWEEN 0 and 24)
    THEN SIGNAL SQLSTATE 'HY000' SET MYSQL_ERRNO = 1525,
    MESSAGE_TEXT = 'Højst 24 arbejdstimer i døgnet!';
  END IF;

  IF NOT (NEW.arbejdsdato BETWEEN '20200101' AND CURDATE())
    THEN SIGNAL SQLSTATE 'HY000' SET MYSQL_ERRNO = 1525,
    MESSAGE_TEXT = 'Kun afholdte arbejdsdage i 2020 kan registreres!';
  END IF;

  SET NEW.bemærkning = TRIM(LEADING ' ' FROM NEW.bemærkning);
END;

INSERT timerapporter VALUES ('M20014','2020-08-08','8','oprettet','M20014','Produktoversigt');
```

Error Code: 1525. Kun afholdte arbejdsdage i 2020 kan registreres!

Funktionen nextsold, som finder næste ledige sold

- Funktion nextsold; Finder næste ledige sold

```
CREATE FUNCTION nextsold () RETURNS CHAR(7)
BEGIN
DECLARE vMaxID CHAR(7);
SELECT MAX(soldID) INTO vMaxID FROM salgsordrer;
RETURN CONCAT ('SO', (RIGHT(vMaxID,5)+1));
END;
```

```
SELECT nextsold();
```

nextsold()
SO60006

- Funktion nextsokoid, for næste ledige sold eller koid

```
CREATE FUNCTION nextsokoid (vTypeID CHAR(2)) RETURNS CHAR(7)
BEGIN
DECLARE vMaxID CHAR(7);
CASE
WHEN vTypeID='so' THEN SELECT MAX(soldID) INTO vMaxID FROM salgsordrer;
RETURN CONCAT ('SO', (RIGHT(vMaxID,5)+1));
WHEN vTypeID='ko' THEN SELECT MAX(koid) INTO vMaxID FROM købsordrer;
RETURN CONCAT ('KO', (RIGHT(vMaxID,5)+1));
ELSE RETURN 'ERRORID';
END CASE;
END;
```

```
SELECT nextsokoid('ko');
```

nextsokoid('ko')
KO70007



Proceduren lønudbetaling

■ Procedure lønudbetaling; For perioden startdato til slutdato

```
CREATE PROCEDURE lønudbetaling (IN vstartdato DATE, IN vslutdato DATE) BEGIN
CREATE TABLE S SELECT mID, efternavn, fornavn, månedsløn, timeløn, lønkonto
FROM medarbejdere WHERE startdato <= vstartdato AND slutdato >= vslutdato;
CREATE TABLE T SELECT mID, SUM(arbejdstimer) AS timer
FROM timerapporater WHERE status = 'godkendt' AND arbejdsdato <= vslutdato GROUP BY mID;
INSERT lønudbetalinger SELECT vslutdato AS løndato, mID, efternavn, fornavn, månedsløn,
COALESCE(timer,0) AS timer, timeløn, COALESCE(timer,0)*timeløn AS variabelsløn,
månedsløn + COALESCE(timer,0)*timeløn AS lønudbetaling, lønkonto
FROM S NATURAL LEFT OUTER JOIN T;
UPDATE timerapporater SET status='udbetalt' WHERE status='godkendt' AND arbejdsdato<=vslutdato;
DROP TABLE S;DROP TABLE T; END;
```

Tabellen lønudbetalinger forudsættes allerede oprettet og klar til at indeholde de månedlige lønudbetalinger.

```
CALL lønudbetaling('20200101','20200131');
```

```
SELECT * FROM lønudbetalinger; SELECT mID, arbejdstimer, status FROM timerapporater;
```

løndato	mID	efternavn	fornavn	månedsløn	timer	timeløn	variabelsløn	lønudbetaling	lønkonto	mID	arbejdstimer	status
2020-01-31	M20001	Bentsen	Anders	75000.00	0	500.00	0.00	75000.00	4434 4440123456	M20005	4	udbetalt
2020-01-31	M20002	Dahl	Christine	40000.00	0	275.00	0.00	40000.00	4434 4440234561	M20005	2	udbetalt
2020-01-31	M20003	Kristensen	Hans	36000.00	0	235.00	0.00	36000.00	4434 4440345612	M20008	4	udbetalt
2020-01-31	M20004	Hansen	Ole	34000.00	0	220.00	0.00	34000.00	4434 4440456123	M20008	4	oprettet
2020-01-31	M20005	Jensen	Leif	0.00	6	185.00	1110.00	1110.00	4434 4440561234	M20009	2	udbetalt
2020-01-31	M20006	Frederiksen	Eva Annette	35000.00	0	230.00	0.00	35000.00	4434 4440612345	M20009	2	oprettet
2020-01-31	M20007	Mortensen	Hans Peter	0.00	0	180.00	0.00	0.00	4434 4440456789	M20011	3	afvist
2020-01-31	M20008	Jacobsen	Lise	0.00	4	190.00	760.00	760.00	4434 4440567894	M20011	6	udbetalt
2020-01-31	M20009	Thomsen	Frederik	0.00	2	180.00	360.00	360.00	4434 4440678945			
2020-01-31	M20010	Petersen	Morten	0.00	0	180.00	0.00	0.00	4434 4440789456			
2020-01-31	M20011	Alberg	Viggo	28000.00	6	190.00	1140.00	29140.00	4434 4440894567			
2020-01-31	M20012	Flyvholm	Ea	0.00	0	180.00	0.00	0.00	4434 4440945678			
2020-01-31	M20013	Petersen	Malte	45000.00	0	300.00	0.00	45000.00	4434 4440654321			
2020-01-31	M20014	Leth	Iben	0.00	0	185.00	0.00	0.00	4434 4440543216			

Proceduren lønudbetaling med en Transaktion

- Procedure lønudbetaling med en transaktion; For perioden startdato til slutdato

```
CREATE PROCEDURE lønudbetaling (IN vstartdato DATE, IN vslutdato DATE)
BEGIN
    DECLARE vSQLSTATE CHAR(5) DEFAULT '00000';
    DECLARE CONTINUE HANDLER FOR SQLEXCEPTION
    BEGIN GET DIAGNOSTICS CONDITION 1 vSQLSTATE = RETURNED_SQLSTATE;
END;

START TRANSACTION;

CREATE TABLE S SELECT mID, efternavn, fornavn, månedsløn, timeløn, lønkonto
    FROM medarbejdere WHERE startdato <= vstartdato AND slutdato >= vslutdato;
CREATE TABLE T SELECT mID, SUM(arbejdstimer) AS timer
    FROM timerapporter WHERE status='godkendt' AND arbejdsdato<=vslutdato GROUP BY mID;
INSERT lønudbetalinger SELECT vslutdato AS løndato, mID, efternavn, fornavn,
    månedsløn, COALESCE(timer,0) AS timer, timeløn, COALESCE(timer,0)*timeløn AS
    variabelsløn, månedsløn + COALESCE(timer,0)*timeløn AS lønudbetaling, lønkonto
    FROM S NATURAL LEFT OUTER JOIN T;
UPDATE timerapporter SET status = 'udbetalt' WHERE status = 'godkendt' AND
    arbejdsdato <= vslutdato;
DROP TABLE S; DROP TABLE T;
IF vSQLSTATE = '00000' THEN COMMIT; ELSE ROLLBACK; END IF;
END;
```

- Lønudbetaling = fastløn + timer*timeløn; Hvis timer er Null, så benyt COALESCE funktionen

(28000.00+4*150.00)	(28000.00+Null*150.00)	(28000.00+COALESCE(Null,0)*150.00)	(28000.00+COALESCE(4,0)*150.00)
28600.00	NULL	28000.00	28600.00

Eventen TrainModelsHero

■ Eventen TrainModelsHero; Månedlig udtrækning af én medarbejderbonus

- Hver måned udtrækkes en medarbejder, der får en bonus svarende til 1-5 arbejdstimer

```
CREATE EVENT TrainModelsHero ON SCHEDULE EVERY 1 MONTH STARTS '2020-02-01 00:00:01'
DO
BEGIN
DECLARE vmID CHAR(6);
DECLARE varbejdstimer INT;
SET vmID = (SELECT mID FROM medarbejdere WHERE curdate() BETWEEN startdato AND slutdato
            ORDER BY RAND() LIMIT 1);
SET varbejdstimer = 1+FLOOR(5*RAND());
INSERT timerapporter VALUES
    (vmID, curdate(), varbejdstimer, 'oprettet', vmID, 'TrainModelsHero');
END;

SET GLOBAL event_scheduler = 1;
SELECT * FROM timerapporter;
```

mID	arbejdsdato	arbejdstimer	status	statusID	bemærkning
M20004	2020-02-01	2	oprettet	M20004	TrainModelsHero
M20005	2020-01-06	4	udbetalt	M20002	Ingen
M20005	2020-01-07	2	udbetalt	M20002	Ingen
M20008	2020-01-06	4	udbetalt	M20001	Mails og oprydning
M20008	2020-01-07	4	oprettet	M20008	Produktgennemgang
M20009	2020-01-06	2	udbetalt	M20001	Ingen
M20009	2020-01-07	2	oprettet	M20009	Ingen
M20009	2020-03-01	5	oprettet	M20009	TrainModelsHero
M20011	2020-01-06	3	afvist	M20006	Arbejdede du ikke 6 timer?
M20011	2020-01-07	6	udbetalt	M20006	Ingen

Eventen produktsalg_i_år

- Event produktsalg_i_år; Beregning af det samlede produktsalg hver måned
 - Opret tabel skema: produktsalgYTD (kørselsdato, pID, produkt, subtotal)
 - Hver måned ønskes produktsalgsrækkerne tilføjet tabellen

```
CREATE EVENT produktsalg_i_år
ON SCHEDULE EVERY 1 MONTH
STARTS '2020-02-01 00:00:01'
DO
BEGIN
CREATE TABLE S SELECT pID, SUM(linieantal*liniestkpris) AS subtotal
FROM salgsordrelinier WHERE soID IN
(SELECT soID FROM salgsordrer WHERE YEAR(bestillingsdato) = YEAR(CURDATE()))
GROUP BY pID WITH ROLLUP;
INSERT produktsalgYTD SELECT curdate() as kørselsdato, pID, produkt, subtotal
FROM S NATURAL LEFT OUTER JOIN produkter;
DROP TABLE S;
END;

SET GLOBAL event_scheduler = 1;
SELECT * FROM produktsalgYTD;
```

kørselsdato	pID	produkt	subtotal
2020-02-01	P50001	DSB MY1100	4998.00
2020-02-01	P50002	SBB klasse C 5/6 "Elefant"	3599.00
2020-02-01	P50003	BR 491 DB	2649.00
2020-02-01	P50009	Marklin 24188 lige C skinne	210.00
2020-02-01	P50012	Marklin 24130 sving/buet C skinne	252.00
2020-02-01	P50014	Marklin 24612 C skinne skiftespor	149.00
2020-02-01	P59001	PostNord	150.00
2020-02-01	P59002	GLS	65.00
2020-02-01	HULL	HULL	12072.00

10. SQL brugerprofiler

- Database Administrator oprettelse af brugere og tildeling af bruger privilegier

Bruger CEO oprettes med password "1234" samt alle privilegier:

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

Grants for CEO@localhost

```
GRANT USAGE ON *.* TO `CEO`@`localhost`  
GRANT ALL PRIVILEGES ON `trainmodels`.* TO `CEO`@`localhost`
```

Bruger Salgschef oprettes med password "1234" med privilegier:

```
CREATE USER 'Salgschef'@'localhost' IDENTIFIED BY '1234';  
GRANT SELECT ON trainmodels.ansatte TO 'Salgschef'@'localhost';
```

Grants for Salgschef@localhost

```
GRANT USAGE ON *.* TO `Salgschef`@`localhost`  
GRANT SELECT ON `trainmodels`.`ansatte` TO `Salgschef`@`localhost`  
GRANT SELECT, INSERT, UPDATE, DELETE ON `trainmodels`.`kunder` TO `Salgschef`@`localhost`  
GRANT SELECT, INSERT, UPDATE, DELETE ON `trainmodels`.`salgsordrelinier` TO `Salgschef`@`localhost`  
GRANT SELECT, INSERT, UPDATE, DELETE ON `trainmodels`.`salgsordrer` TO `Salgschef`@`localhost`  
GRANT SELECT, INSERT, UPDATE ON `trainmodels`.`timerapporter` TO `Salgschef`@`localhost`
```

Bruger Salgsmedarbejder oprettes med password "1234" med privilegier:

```
CREATE USER 'Salgsmedarbejder'@'localhost' IDENTIFIED BY '1234';  
GRANT SELECT ON trainmodels.kunder TO 'Salgsmedarbejder'@'localhost';
```

Ellers med de samme privilegier som salgschef til de øvrige tabeller

11. Java database adgang

■ Indhold

- Insertsalesorder.java; Indsætter en ny salgsordre
- Salesordervalue.java; Beregner en salgsordres samlede værdi
- Fra University findes tilpasset ManipulateTrainModels.java og QueryTrainModels.java

```
C:\DBSJava>javac Insertsalesorder.java
```

```
C:\DBSJava>java Insertsalesorder
Skriv 'soID', 'kID', 'bestillingsdato':
'SO60005', 'K30002', '2020-03-16'
```

```
C:\DBSJava>java Insertsalesorder
Skriv 'soID', 'kID', 'bestillingsdato':
'SO60006', 'K30003', '2020-03-16'
```

```
mysql> SELECT soID, kID, bestillingsdato
FROM salgsordrer;
```

soID	kID	bestillingsdato
SO60002	K30003	2020-01-06
SO60003	K30002	2020-01-06
SO60004	K30003	2020-01-08
SO60005	K30002	2020-03-16
SO60006	K30003	2020-03-16

```
C:\DBSJava>javac Salesordervalue.java
```

```
C:\DBSJava>java Salesordervalue
Skriv soID:
SO60002
5223.00
```

```
C:\DBSJava>java Salesordervalue
Skriv soID:
SO60003
2564.00
```

```
C:\DBSJava>java Salesordervalue
Skriv soID:
```

```
null
```

Insertsalesorder.java

```
1  import java.sql.*;
2  import java.util.Scanner;
3
4  public class Insertsalesorder {
5      public static void main(String[] args) {
6          //Edit username and password to match database login
7          String url = "jdbc:mysql://localhost:3306/trainmodels?characterEncoding=utf8";
8          String username = "root";
9          String password = "7984";
10
11         try {
12             //Change driver below to "com.mysql.jdbc.Driver" if prompted by IntelliJ
13             Class.forName("com.mysql.jdbc.Driver");
14             Scanner scanner = new Scanner(System.in, "CP850");
15             String sqlInsert;
16             System.out.println("Skriv 'soID', 'kID', 'bestillingsdato': ");
17             sqlInsert = scanner.nextLine();
18             sqlInsert = "INSERT salgsordrer VALUES (" + sqlInsert
19                 + ", NULL, NULL, NULL)";
20
21             Connection connection = DriverManager.getConnection(url, username, password);
22             Statement statement = connection.createStatement();
23             statement.executeUpdate(sqlInsert);
24             connection.close(); }
25         catch (Exception e) { e.printStackTrace(); }
26     }
27 }
```

Salesordervalue.java

```
1  import java.sql.*;
2  import java.util.Scanner;
3
4  public class Salesordervalue {
5      public static void main(String[] args) {
6          //Edit username and password to match database login
7          String url = "jdbc:mysql://localhost:3306/trainmodels?characterEncoding=utf8";
8          String username = "root";
9          String password = "7984";
10
11         try {
12             //Change driver below to "com.mysql.jdbc.Driver" if prompted by IntelliJ
13             Class.forName("com.mysql.jdbc.Driver");
14             Scanner scanner = new Scanner(System.in, "CP850");
15             String soID;
16             String sqlQuery;
17             System.out.println("Skriv soID: ");
18             soID = scanner.nextLine();
19             sqlQuery = "SELECT SUM(linieantal*liniestkpris) FROM salgsordrelinier "
20                 + "WHERE soID='" + soID + "';";
21
22             Connection connection = DriverManager.getConnection(url, username, password);
23             Statement statement = connection.createStatement();
24             ResultSet resultSet = statement.executeQuery(sqlQuery);
25             if (resultSet.next()) { System.out.println(resultSet.getString(1)); }
26             connection.close(); }
27         catch (Exception e) { e.printStackTrace(); }
28     }
29 }
```

12. Deltagernes indsats i gruppearbejdet

■ Deltagernes indsats i gruppearbejdet

Deltagere i Gruppe 00	2. DB tabel skemaer og brugerprofiler	3. Entity Relationship diagram	4. Extended Entity Relationship diagram	5. Database instans	6. Normalformer	7. SQL tabelmodifikationer	8. SQL forespørgsler	9. SQL programmering	10. SQL brugerprofiler	11. Java database adgang
Flemming Schmidt	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
Deltager 2	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Deltager 3	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Deltager 4	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Deltager 5	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Deltager 6	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%



11. Database project solution



12. Deltagernes indsats i gruppearbejdet