

DAT107 Oblig 1

Gruppe: 587728

Database ligger på bruker 587728.

Joachim Leiros

Lager Ansatt:

```
CREATE TABLE Ansatt (  
  AnsNr    int4 NOT NULL,  
  Fornavn  varchar(20) NOT NULL,  
  Etternavn varchar(20) NOT NULL,  
  Adresse  varchar(30),  
  Fodselsdato date,  
  Kjonn    char(1),  
  Stilling varchar(20),  
  Aarslonn numeric(10, 2),  
  PostNr   int4 NOT NULL,  
  PRIMARY KEY (AnsNr));
```

Setter inn i Ansatt:

```
INSERT into Ansatt (AnsNr, Fornavn, Etternavn, Adresse, Fodselsdato, Kjonn, Stilling,  
Aarslonn, PostNr)  
values ('1234','Joachim','Leiros', 'Gml. postveg 55', '1992-07-18',  
'M','Sjef','500000.00','6968');
```

Lager Poststed:

```
CREATE TABLE Poststed (  
  PostNr   int4,  
  Poststed varchar(30),  
  PRIMARY KEY (PostNr));
```

Setter inn i Poststed:

```
INSERT into Poststed (PostNr, Poststed)  
Values  ('6968', 'Flekke'),  
        ('9900', 'Kirkenes');
```

Lager table Kunde:

```
CREATE TABLE Kunde (  
  KNr    SERIAL NOT NULL,  
  Fornavn varchar(20) NOT NULL,  
  Etternavn varchar(20) NOT NULL,  
  Adresse  varchar(30) NOT NULL,  
  PostNr   int4 NOT NULL,  
  PRIMARY KEY (KNr));
```

Setter inn i Kunde:

```
INSERT into Kunde (Knr, Fornavn, Etternavn, Adresse, PostNr)
values ('1', 'Harry','Potter', 'Bakeveien 2','9900');
```

```
INSERT into Kunde (Knr, Fornavn, Etternavn, Adresse, PostNr)
values ('123', 'William','Nordmann', 'Bakeveien 1','9900');
```

```
INSERT into Kunde(Knr,Fornavn,Etternavn,Adresse,Postnr)
values('2','Hermine','Granger','Galtvort 1', '6968');
```

Lag table ordre:

```
CREATE TABLE Ordre (
  OrdreNr SERIAL NOT NULL,
  OrdreDato date NOT NULL,
  SendtDato date,
  BetaltDato date,
  KNr int4 NOT NULL,
  PRIMARY KEY (OrdreNr));
```

Setter inn i ordre:

```
INSERT into Ordre (OrdreNr, OrdreDato, SendtDato, BetaltDato, KNr)
values (DEFAULT,'2020-06-01','2020-06-02','2020-06-01','1');
```

```
INSERT into Ordre (OrdreNr, OrdreDato, SendtDato, BetaltDato, KNr)
values (DEFAULT,'2020-03-01','2020-02-02','2020-06-01','2');
```

Lager Ordrelinje:

```
CREATE TABLE Ordrelinje (
  OrdreNr SERIAL NOT NULL,
  VNr int4,
  PrisPrEnhet numeric(5, 2),
  Antall int4,
  PRIMARY KEY (OrdreNr,
  VNr));
```

Setter inn i Ordrelinje:

```
INSERT into Ordrelinje (OrdreNr, VNr, PrisPrEnhet, Antall)
values (DEFAULT,'15161','10.50','5');
```

```
INSERT into Ordrelinje (OrdreNr, VNr, PrisPrEnhet, Antall)
values (2,'15161','10.50','10');
```

Lager tabell Vare:

```
CREATE TABLE Vare (  
  VNr      int4 NOT NULL,  
  Betegnelse varchar(20),  
  Pris     int4  
  Antall   int4,  
  Hylle    int4,  
  KatNr    int4 NOT NULL,  
  PRIMARY KEY (VNr));
```

Setter inn i tabell vare:

```
INSERT into Vare (VNr,Betegnelse,Pris,Antall,Hylle,KatNr)  
values (55, 'Godteri', 10.50, 76, 3.2, 9);
```

Lager:

```
CREATE TABLE Prishistorikk (  
  VNr      int4 NOT NULL,  
  Dato     date NOT NULL,  
  Gammelpris numeric(5, 2),  
  PRIMARY KEY (VNr,  
  Dato));
```

Setter inn i prishistorikk:

```
INSERT into Prishistorikk (VNr, Dato, Gammelpris)  
values (55, '1992-07-14', 9.5);
```

Lager:

```
CREATE TABLE Kategori (  
  KatNr SERIAL NOT NULL,  
  Navn varchar(30),  
  PRIMARY KEY (KatNr));
```

Setter inn i:

```
INSERT into Kategori(KatNr, Navn)  
values (55, 'Sjokolade'),  
      (66, 'Gronnsaker');
```

Legger inn referanser til fremmednøkler:

```
ALTER TABLE Ansatt ADD CONSTRAINT FKAnsatt FOREIGN KEY (PostNr) REFERENCES  
Poststed (PostNr) ON DELETE RESTRICT ON UPDATE CASCADE;
```

```
ALTER TABLE Kunde ADD CONSTRAINT FKKunde FOREIGN KEY (PostNr) REFERENCES Poststed  
(PostNr) ON DELETE RESTRICT ON UPDATE CASCADE;
```

```
ALTER TABLE Ordre ADD CONSTRAINT FKOrdre FOREIGN KEY (KNr) REFERENCES Kunde (KNr)  
ON DELETE RESTRICT ON UPDATE CASCADE;
```

```
ALTER TABLE Ordrelinje ADD CONSTRAINT FKOrdrelinje FOREIGN KEY (OrdreNr) REFERENCES  
Ordre (OrdreNr) ON DELETE RESTRICT ON UPDATE CASCADE;
```

```
ALTER TABLE Vare ADD CONSTRAINT FKVare FOREIGN KEY (KatNr) REFERENCES Kategori  
(KatNr) ON DELETE RESTRICT ON UPDATE CASCADE;
```

Ansatt:

AnsNr int4 NOT NULL,

- Bruker int for å representere ett ansattnr, ikke NULL da en ansatt må ha ett ansattnr. Brukes som primærnøkkel da ansattnr er individuelt.

Fornavn varchar(20) NOT NULL,

- Varchar på lengde 20 for navn, kunne vært lengre og. NOT NULL da en person må ha ett navn registrert på seg.

Etternavn varchar(20) NOT NULL,

- Varchar på lengde 20 for etternavn, kunne også vært lengre, f.eks 30 her da noen etternavn er lange. NOT NULL da en person må være registrert med etternavn.

Adresse varchar(30),

- Varchar på 30 da gateadresser kan være lange.

Fodselsdato date,

- Dato format for noe som kun skal være en fødselsdato.

Kjonn char(1),

- En char for å representere (M)ann eller (K)vinne.

Stilling varchar(20),

- Varchar på 20 for en stilling.

Aarslonn numeric (10, 2),

- Numeric 10,2 altså 10 siffer før og 2 siffer etter komma. For å representere lønn på en god måte. Dersom en person skulle ende opp med ett rart desimal (si .345 så vil numeric runde av og gi ett mer leselig tall).

PostNr int4 NOT NULL,

- Int, må være fylt ut.

Alternative primærnøkler kan være adresse eller fødselsdato. Disse er dog mer sannsynlig at flere personer kan dele og er derfor lite passende. Fordeler med ansNr er at vi selv kan styre inkrementering og utdeling av nummer. "Ulempen" er at vi må ha ett ekstra datafelt per ansatt.

Legg til Ansattdato for ansatt:

```
ALTER TABLE Ansatt
```

```
ADD Ansattdato date;
```

Bruker dato da det er mest logisk da vi ønsker en dato, kunne også brukt varchar for å beskrive dato ved tekst dersom det ikke skulle vært nøyaktig når, men ca når en person ble ansatt. Velger å unnlate NOT NULL da dette er noe vi legger på i etterkant og eksisterende ansatte vil stå med tomt felt.

SQL for hver kunde viser salg og produkter:

```
SELECT
```

```
    Kunde.knr,
```

```
    Kunde.Fornavn,
```

```
    Kunde.Etternavn,
```

```
    Ordrelinje.Ordrenr,
```

```
    Vare.betegnelse
```

```
FROM ((Kunde INNER JOIN Ordre ON Kunde.Knr = Ordre.Knr)
```

```
      INNER JOIN Ordrelinje ON Ordre.OrdreNr = Ordrelinje.Ordrenr)
```

```
      INNER JOIN Vare ON Vare.Vnr = Ordrelinje.Vnr
```

```
WHERE Kunde.knr IS NOT NULL;
```

Bruker inner join da vi vil ha fram data som forekommer hvor vi har en union fra de forskjellige tabellene. Mulig det er en liten feil i min database men jeg får ikke opp dataene som kulle kommet fram. Kun kolonnene, som da stemmer. Rakk ikke få fikset før innlevering.

Spørring for å finne sum av salg per poststed, sortert synkende etter postnummer.

SELECT

Poststed.postnr,
Poststed.poststed,
Ordrelinje.prisprenhet,
Ordrelinje.antall,

SUM(Ordrelinje.prisprenhet*Ordrelinje.antall)

FROM ((Poststed INNER JOIN Kunde ON Poststed.postnr = kunde.postnr)

INNER JOIN Ordre ON Kunde.Knr = Ordre.Knr)

INNER JOIN Ordrelinje ON Ordrelinje.Ordrenr = Ordre.Ordrenr

WHERE Ordre.Ordrenr IS NOT NULL

GROUP BY Poststed.postnr,Ordrelinje.prisprenhet,Ordrelinje.antall

ORDER BY Poststed.postnr DESC;

Resultat av spørring: Sortert på synkende postnr. Valgte og bruke innerjoin da vi vil ha data fra flere tabeller som delte flere kolonner, deriblant postnr.

	postnr integer	poststed character varying (30)	prisprenhet numeric (5,2)	antall integer	sum numeric
1	9900	Kirkenes	10.50	5	52.50
2	6968	Flekke	10.50	10	105.00