SQL-opgaver 4 – løsning

Opret følgende sekvenser til anvendelsen i tabellernes primærnøgler

Navn på sekvens	Værdi der skal øges med	Værdi der skal startes fra
kunde_seq	1	1001
vare_seq	1	101
ordre_seq	1	1
ordrelinie_seq	1	1

```
CREATE SEQUENCE kunde_seq
INCREMENT BY 1
START WITH 1000;

CREATE SEQUENCE vare_seq
START WITH 101;

CREATE SEQUENCE ordre_seq;

CREATE SEQUENCE ordrelinie_seq;
```

Indsæt følgende værdier i tabellen postnumre

Postnr	Bynavn
9200	Aalborg SV
9000	Aalborg
8200	Århus N

```
INSERT INTO postnumre
(postnr, bynavn)
VALUES
(9200, 'Aalborg SV');

INSERT INTO postnumre
VALUES
(9000, 'Aalborg');

INSERT INTO postnumre
VALUES
(8200, 'Århus N');
```

Indsæt følgende værdier i tabellen kunder

Kunde_id	Kunde	Adresse	Postnr	Telefonnr	Email
Anvend sekvens	IDEmøbler	Nålemagervej 6	9000	98138744	aalborg@ide.dk
Anvend sekvens	IKEA	Graham Bells vej 9-11	8200		

```
INSERT INTO kunder
VALUES
(kunde_seq.nextval, 'IDEmøbler', 'Nålemagervej 6', 9000, 98138744,
'aalborg@ide.dk');

INSERT INTO kunder
(kunde_id, kunde, adresse, postnr)
VALUES
(kunde_seq.nextval, 'IKEA', 'Graham Bells vej 9-11', 8200);
```

Indsæt følgende værdier i tabellen varer

Vare_id	**Vare**	**Pris**
Anvend sekvens	Sofabord	799,50
Anvend sekvens	TV-bord	349,00
Anvend sekvens	2-personers sofa	2199,00

```
INSERT INTO varer
VALUES
(vare_seq.nextval, 'Sofabord', 799.50);

INSERT INTO varer
VALUES
(vare_seq.nextval, 'TV-bord', 349);

INSERT INTO varer
VALUES
(vare_seq.nextval, '2-personers sofa', 2199);
```

Du er lige blevet ansat i virksomhedens salgsafdeling. Opret derfor dig selv i tabellen medarbejdere med medarbejder_id 1030 og en løn på 25.000. Det korrekte afdeling_id kan findes i tabellen afdelinger.

```
SELECT * FROM afdelinger;

INSERT INTO medarbejdere
(medarbejder_id, fornavn, efternavn, loen, email, afdeling_id)
VALUES
(1030, 'Bent Møller', 'Madsen', 25000, 'bmm@business.aau.dk', 20);
```

IKEA har dags dato bestilt 150 stk sofaborde, 50 TV-borde og 80 2-personers sofaer (i alt 3 ordrelinier på en ordre), hvilket skal indtastes. Du har betjent dem, produkterne skal leveres om 14 dage, mens betalingsdatoen endnu ikke er aftalt. Ordre og ordrelinier skal anvende deres respektive sekvenser til at danne id'er. Status skal sættes til 'oprettet'.

Da vi ikke kan være sikre på hvor langt sekvensen til kunde_id er kommet (uden manuelt at skulle tjekke indholdet i kundetabellen) kan vi sikre at vi får det rigtige kunde_id ved at lave en sub-select, der henter kunde_id'et for IKEA.

```
INSERT INTO ordrer
(ordre_id, oprettelsesdato, leveringsdato, kunde_id, status,
ansvarlig_id)
VALUES
(ordre_seq.nextval, SYSDATE, SYSDATE + 14, (SELECT kunde_id FROM
kunder WHERE kunde = 'IKEA'), 'oprettet', 1030);


INSERT INTO ordrelinier
(ordrelinie_id, ordre_id, vare_id, antal)
VALUES
(ordrelinie_seq.nextval, 1, 101, 150);
```

I stedet for at lave et opslag i tabellen ordrer for at finde det rigtige ordre_id til ordrelinierne, kan vi anvende en sub-select til at finde det største (sidst indtastede) ordre_id fra tabellen, hvilket vil virke korrekt så længe, at flere ordre(-hoveder) ikke oprettes inden ordrelinierne.

```
INSERT INTO ordrelinier
VALUES
(ordrelinie_seq.nextval, (select max(ordre_id) from ordrer), 102,
50);
```

I forbindelse med den tredje ordrelinie vises her tilsvarende ved indtastningen af ordrehovedet, hvordan vi via en sub-select kan finde det korrekte vare_id fra tabellen varer.

```
INSERT INTO ordrelinier
VALUES
(ordrelinie_seq.nextval, 1, (SELECT vare_id FROM varer WHERE vare =
'2-personers sofa'), 80);
```

Vi forestiller os nu at der er gået 14 dage og ovenstående ordre er blevet leveret og derfor skal status opdateres til 'leveret'. Endvidere er betalingsdato nu fastsat til den sidste dag i denne måned. Opdater disse to felter på ordren.

For at være sikre på hvordan datoer skal indtastes bestemmer vi først datoformatet i denne session (så længe den pågældende SQL-prompt er åben).

```
ALTER SESSION
SET NLS_DATE_FORMAT = 'DD-MM-YYYY';

UPDATE ordrer
SET
status = 'leveret',
betalingsdato = '30-09-2006'
WHERE
ordre_id = 1;

COMMIT;
```

Følgende forespørgsel skal nu gerne vise et resultat på 3 rækker:

```
SELECT * FROM kunder NATURAL JOIN ordrer NATURAL JOIN ordrelinier
NATURAL JOIN varer;
```