

## SQL-opgaver 2 – løsning

Opret betingelser for følgende enten ved at droppe de eksisterende tabeller og oprette dem på ny eller vha. Alter table.

For hver enkel type constraint viser jeg forskellige lovlige måder, hvormed disse kan oprettes vha. ALTER TABLE. Dette omfatter anvendelsen af MODIFY og ADD samt med og uden navngivning af betingelserne. *I praksis vil jeg altid anvende ADD til at tilføje nye betingelser på nær NOT NULL betingelser, hvor jeg vil anvende MODIFY. Endvidere vil jeg altid vælge at navngive betingelser måske dog igen på nær NOT NULL betingelser, da disse er en anelse mindre relevante at følge op på end de andre.* I øvrigt er flertallet af ALTER TABLE .. MODIFY kommandoerne (igen på nær ved NOT NULL) først blevet tilladt fra version 10g af Oracle database.

Efter oprettelsen af constraints, så kan man se disse ved at lave forespørgsler på følgende 2 systemtabeller. Den første viser flest data om de enkelte constraints, men ikke hvilke konkrete kolonner, de omfatter. Denne information kan kun findes i den anden systemtabel.

```
SELECT * FROM user_constraints;  
SELECT * FROM user_cons_columns;
```

```
SELECT * FROM user_cons_columns NATURAL JOIN user_constraints;
```

### Not null

Tabel	Kolonne der skal være not null
Postnummer	bynavn
Kunder	kunde
Ordrer	oprettelsesdato
Varer	vare

Det kan ikke lade sig gøre at tilføje en NOT NULL constraint vha. "ALTER TABLE <tabelnavn> ADD CONSTRAINT ..." med mindre NOT NULL skrives som del af en CHECK constraint.

```
ALTER TABLE postnummer  
MODIFY bynavn NOT NULL;
```

```
ALTER TABLE kunder  
MODIFY kunde CONSTRAINT kunder_kunde_nn NOT NULL;
```

```
ALTER TABLE ordrer  
ADD CHECK (oprettelsesdato IS NOT NULL);
```

```
ALTER TABLE varer  
ADD CONSTRAINT varer_vare_nn CHECK (vare IS NOT NULL);
```

## Unikke nøgler

Tabel	Kolonne(r) der skal være unikke
Postnummer	bynavn
Kunder (sammensat unik nøgle)	kunde adresse postnr

Den anden betingelse kan, da den omfatter flere kolonner, kun tilføjes med ADD CONSTRAINT.

```
ALTER TABLE postnummer  
MODIFY bynavn CONSTRAINT postnummer_bynavn_uk UNIQUE;
```

```
ALTER TABLE kunder  
ADD CONSTRAINT kunder_kunde_adresse_postnr_uk UNIQUE (kunde,  
adresse, postnr);
```

## Primærnøgler

Tabel	Kolonne i primærnøgle
postnummer	postnr
kunder	kunde_id
varer	vare_id
ordrer	ordre_id
ordrelinier	ordrelinie_id

```
ALTER TABLE postnummer  
MODIFY postnr NUMBER(4) PRIMARY KEY;
```

```
ALTER TABLE kunder  
MODIFY kunde_id CONSTRAINT kunder_pk PRIMARY KEY;
```

```
ALTER TABLE varer  
ADD PRIMARY KEY (vare_id);
```

```
ALTER TABLE ordrer  
ADD CONSTRAINT ordrer_pk PRIMARY KEY (ordre_id);
```

```
ALTER TABLE ordrelinier  
ADD CONSTRAINT ordrelinier_pk PRIMARY KEY (ordrelinie_id);
```

## Fremmednøgler

Tabel	Kolonne i fremmednøgle	Tabel der refereres til	Kolonne
kunder	postnr	postnummer	postnr
ordrer	kunde_id	kunder	kunde_id
ordrelinier	ordre_id	ordrer	ordre_id
ordrelinier	vare_id	varer	vare_id

```
ALTER TABLE kunder
MODIFY postnr REFERENCES postnummer(postnr);
```

```
ALTER TABLE ordrer
MODIFY kunde_id CONSTRAINT ordrer_kunder_fk REFERENCES kunder
(kunde_id);
```

ON DELETE CASCADE i følgende fremmednøgle gør, at hvis den ordre, som fremmednøglen refererer til, bliver slettet, så bliver alle tilhørende ordrelinier også automatisk slettet.

```
ALTER TABLE ordrelinier
ADD CONSTRAINT ordrelinier_ordrer_fk FOREIGN KEY (ordre_id)
REFERENCES ordrer (ordre_id)
ON DELETE CASCADE;
```

```
ALTER TABLE ordrelinier
ADD FOREIGN KEY (vare_id)
REFERENCES varer (vare_id);
```

## Check Constraint

1. Tallet der indsættes i postnr i tabellen postnummer skal være præcis 4 cifre.
2. Leveringsdato skal være lig eller senere ( $\geq$ ) end oprettelsesdatoen i tabellen ordrer.

```
ALTER TABLE postnummer
MODIFY postnr CONSTRAINT postnummer_postnr_cc CHECK (LENGTH(postnr)
= 4);
```

```
ALTER TABLE ordrer
ADD CONSTRAINT ordrer_levdato_etter_opdato_cc CHECK (leveringsdato
 $\geq$  oprettelsesdato);
```