## Udvalgte SQL-queries i patentsystemet

# Alle patienter

Med kolonnerne både fra patient og patientnavn

```
SELECT * FROM patient p

JOIN patientNavn pn

ON p.patientId = pn.patientId

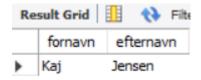
SELECT * FROM patientbase.patient;
```

#### Output:



## Søg efter en patient på cpr-nummer:

```
SELECT fornavn, efternavn FROM patient p, patientnavn pn
WHERE cpr = '0210104003'
AND p.patientId = pn.patientId;
```



### Søg efter alle patienter, som har en bestemt diagnose

Forbindelsen mellem tabellen Patient og Daignose er mange til mange, så jeg har lavet en mellemtabel PatientDiagnoser. Søgningen kan laves med joins eller med where. Her er søgningen med joins lidt hurtigere, når man kigger på Qurey Statics, selv om den er længere.

Man finder først alle match mellem patient og patientnavn, dernæst match med patientdiagnose og til sidst match med diagnose.

```
SELECT d.kort_note_om_diagnose, pn.fornavn, pn.efternavn

FROM patient p, diagnose d, patientdiagnoser pd, patientNavn pn

WHERE p.patientId = pd.patientId

AND p.patientId = pn.patientId

AND pd.diagnoseId = d.diagnoseId

AND d.diagnoseId= 1;

SELECT d.kort_note_om_diagnose, pn.fornavn, pn.efternavn

FROM patient p

JOIN patientNavn pn

ON p.patientId = pn.patientId

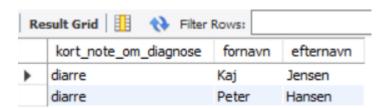
JOIN patientDiagnoser pd

ON p.patientId = pd.patientId

JOIN diagnose d

ON pd.diagnoseId = d.diagnoseId

WHERE d.diagnoseId= 1;
```

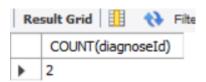


## Søg efter antallet af patienter med en bestemt diagnose

Forbindelsen mellem tabellen Patient og Daignose er mange til mange, så jeg har lavet en mellemtabel PatientDiagnoser. Søgningen kan laves med joins eller med where. Her er søgningen med joins lidt hurtigere, når man kigger på Qurey Statics.

Jeg laver først subquery på diagnoseld og så tæller jeg antallet af rækker. Count(\*) tæller også rækker med null-værdier med, mens COUNT(kolonnenavn) tæller ikke NULL-værdier med.

```
-- Disse søgninger udfører det samme, men den her gang er det joins
-- som er lidt hurtigere
SELECT COUNT(diagnoseId) FROM
    (SELECT d.diagnoseId FROM
    diagnose d, patient p, patientdiagnoser pd
    WHERE p.patientId = pd.patientId
    AND pd.diagnoseId = d.diagnoseId
    AND d.diagnoseId= 1) AS antal;
SELECT COUNT(diagnoseId) FROM
    (SELECT d.diagnoseId
    FROM patient p
    INNER JOIN patientDiagnoser pd
        ON p.patientId = pd.patientId
    INNER JOIN diagnose d
        ON pd.diagnoseId = d.diagnoseId
    WHERE d.diagnoseId= 1) AS antal;
```



## Søg efter én patients alle diagnoser på cpr-nummer

Forbindelsen mellem tabellen Patient og Daignose er mange til mange, så jeg har lavet en mellemtabel PatientDiagnoser, som i det tidligere eksempel.

Søgningen kan laves med joins eller med where. Her er søgningen med where lidt hurtigere, når man kigger på Qurey Statics.

```
-- Her er søgningen med where hurtigere
SELECT pn.fornavn, pn.efternavn, d.kort note om diagnose
FROM patient p
JOIN patientNavn pn
    ON p.patientId = pn.patientId
JOIN patientDiagnoser pd
    ON p.patientId = pd.patientId
JOIN diagnose d
    ON pd.diagnoseId = d.diagnoseId
WHERE p.cpr= '0210104011';
SELECT pn.fornavn, pn.efternavn, d.kort note om diagnose
FROM patient p, diagnose d, patientDiagnoser pd, patientNavn pn
WHERE p.patientId = pd.patientId
AND pd.diagnoseId = d.diagnoseId
AND p.patientId = pn.patientId
AND p.cpr= '0210104011'
```



## Søg efter patientens alder

Alderen er ikke atomisk data, fordi den regnes ud fra fødselsdatoen. Ellers skulle man opdatere alderen hvert år for patienten. Fødselsdatoen findes i tabellen PatientOplysninger.

Curcate() henter dagens dato og metoden datediff regner forskellen mellem fødselsdatoen og dagensdato. Datatypen for fødselsdagen er Date. Jeg laver først subquery til at udregne alderen og så viser jeg den sammen med fornavn og efternavn.

Kolonnen alder eksisterer ikke i nogle af tabellerne, men det kan vi selv lave til resultSet med alias.

