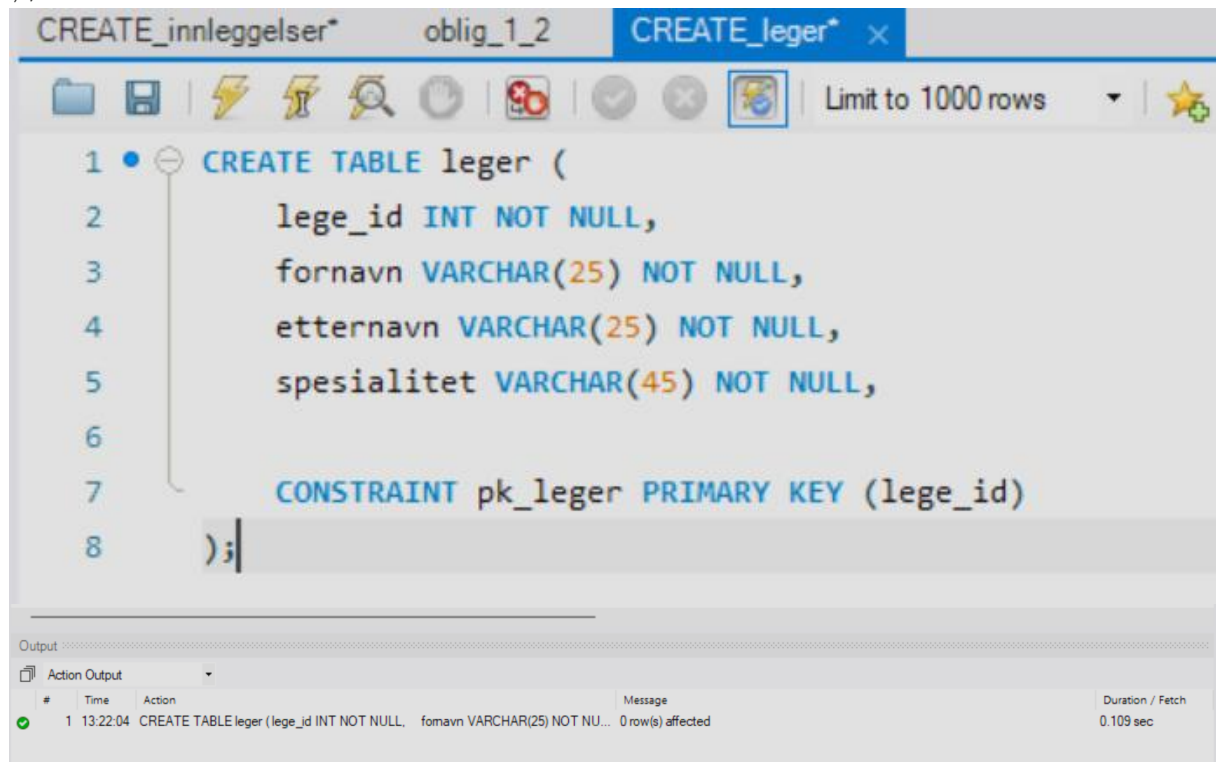


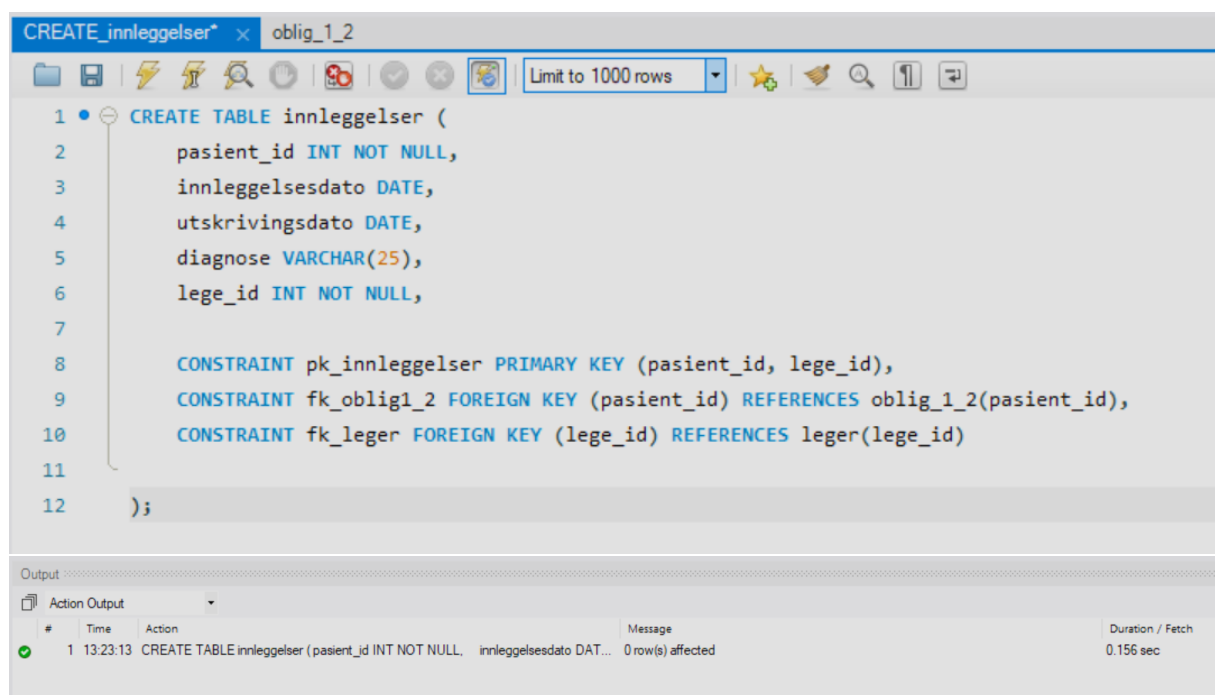
Opprett leger:

```
CREATE TABLE leger (  
    lege_id INT NOT NULL,  
    fornavn VARCHAR(25) NOT NULL,  
    etternavn VARCHAR(25) NOT NULL,  
    spesialitet VARCHAR(45) NOT NULL,  
  
    CONSTRAINT pk_leger PRIMARY KEY (lege_id)  
);
```




Opprett innleggelser:

```
CREATE TABLE innleggelser (  
    pasient_id INT NOT NULL,  
    innleggelsesdato DATE,  
    utskrivingsdato DATE,  
    diagnose VARCHAR(25),  
    lege_id INT NOT NULL,  
  
    CONSTRAINT pk_innleggelser PRIMARY KEY (pasient_id, lege_id),  
    CONSTRAINT fk_oblig1_2 FOREIGN KEY (pasient_id) REFERENCES  
oblig_1_2(pasient_id),  
    CONSTRAINT fk_leger FOREIGN KEY (lege_id) REFERENCES leger(lege_id)  
  
);
```



Fylle tabellene med data:


 Table Data Import

Import Results

File C:\Users\emilb\OneDrive\Dokumenter\HIOF\V_2024\Databasesystemer\oblig2\2_3\leger.csv was imported in 0.192 s

Table oblig.leger was created

27 records imported

 Table Data Import

Import Results

File C:\Users\emilb\OneDrive\Dokumenter\HIOF\V_2024\Databasesystemer\oblig2\2_3\innleggelser.csv was imported in 3.195 s

Table oblig.innleggelser has been used

480 records imported

Innhold i tabellene:

Innleggelser:

The screenshot shows a database application interface. At the top, there's a toolbar with 'Result Grid', 'Filter Rows', 'Edit', 'Export/Import', and 'Wrap Cell Content'. Below this is a table with the following data:

| | pasient_id | innleggesdato | utskrivingsdato | diagnose | lege_id |
|---|------------|---------------|-----------------|--------------------------|---------|
| ▶ | 1 | 2018-11-06 | 2018-11-08 | Ovarian Dermoid-Cyct | 21 |
| | 3 | 2019-01-24 | 2019-01-29 | Cardiac Arrest | 2 |
| | 3 | 2018-10-21 | 2018-10-27 | Congestive Heart Failure | 8 |
| | 6 | 2018-06-13 | 2018-06-15 | Asthma Exacerbation | 3 |
| | 6 | 2018-11-08 | 2018-11-09 | Uterine Fibroid | 22 |
| | 7 | 2018-06-24 | 2018-07-03 | Cancer | 8 |
| | 8 | 2018-09-18 | 2018-09-21 | Amigima | 6 |
| | 9 | 2019-03-02 | 2019-03-09 | Osteoarthritis | 8 |
| | 9 | 2018-12-31 | 2018-12-31 | Ruptured Appendicitis | 19 |
| | 10 | 2018-12-30 | 2019-01-05 | Zenkers Diverticulitis | 14 |
| | 10 | 2019-02-27 | 2019-02-27 | Lower Quadrant Pain | 27 |
| | 11 | 2018-12-14 | 2018-12-16 | Prostatectomy | 15 |
| | 12 | 2019-04-27 | 2019-05-04 | Cerebral Aneurysm Rup... | 21 |
| | 13 | 2019-04-28 | 2019-05-01 | Renal Failure | 1 |

Below the table, there's a tab labeled 'innleggelser 1'. Underneath, there's an 'Output' section with a dropdown menu set to 'Action Output'. Below that, there's a table showing the result of a SQL query:

| # | Time | Action | Message |
|-----|----------|--|---------------------|
| ✓ 1 | 14:50:53 | SELECT * FROM oblig.innleggelser LIMIT 0, 1000 | 480 row(s) returned |

Leger:

The screenshot shows a database application interface. At the top, there's a toolbar with 'Result Grid', 'Filter Rows', 'Edit', 'Export/Import', and 'Wrap Cell Content'. Below this is a table with the following data:

| | lege_id | fornavn | etternavn | spesialitet |
|---|---------|---------|-----------|---------------------------|
| ▶ | 1 | Claude | Walls | Internist |
| | 2 | Joshua | Green | Cardiologist |
| | 3 | Miriam | Tregre | General Surgeon |
| | 4 | James | Russo | Obstetrician/Gynecologist |
| | 5 | Scott | Hill | Gastroenterologist |
| | 6 | Tasha | Phillips | Psychiatrist |
| | 7 | Hazel | Patterson | Oncologist |
| | 8 | Mickey | Duval | Pediatrician |
| | 9 | Jon | Nelson | Neurologist |
| | 10 | Monica | Singleton | Orthopaedic Surgeon |

Below the table, there's a tab labeled 'leger 2'. Underneath, there's an 'Output' section with a dropdown menu set to 'Action Output'. Below that, there's a table showing the result of a SQL query:

| # | Time | Action | Message |
|-----|----------|---|--------------------|
| ✓ 1 | 14:51:58 | SELECT * FROM oblig.leger LIMIT 0, 1000 | 27 row(s) returned |

Spørringer mot tabellen:

Oppgave 1:

```
SELECT COUNT(innleggesdato) AS Antall_Innleggelser FROM innleggelser;
```

I denne spørringen spør vi om antall innleggingsdatoer fra innleggelser-tabellen. Her har det egentlig ikke noe å si hvilken rad man velger, ettersom ingen var «null». Antallet fikk også et alias, for bedre oversikt.

The screenshot shows a SQL query execution interface. At the top, the query is entered in a text area:

```
1 -- Antall Innleggelser --  
2 • SELECT COUNT(innleggesdato) AS Antall_Innleggelser FROM innleggelser;
```

Below the query area, there is a "Result Grid" section. It contains a table with one column, "Antall_Innleggelser", and one row with the value "480".

| Antall_Innleggelser |
|---------------------|
| 480 |

At the bottom, there is an execution log table with columns: #, Time, Action, and Message.

| # | Time | Action | Message |
|-----|----------|---|-------------------|
| ✓ 1 | 13:42:11 | SELECT COUNT(innleggesdato) AS Antall_Innleggelser FROM innleggelser LIM... | 1 row(s) returned |

Oppgave 2:

```
SELECT * FROM innleggelser WHERE innleggesdato =  
utskrivingsdato;
```

Her spør vi om alt av informasjon om en pasient hvor innleggesdatoen er det samme som utskrivingsdatoen.

```
4 -- Alle innleggelser hvor en pasient ble innlagt og utskrevet på samme dag --  
5 • SELECT * FROM innleggelser WHERE innleggesdato = utskrivingsdato;
```

| | pasient_id | innleggesdato | utskrivingsdato | diagnose | lege_id |
|---|------------|---------------|-----------------|---------------------------|---------|
| ▶ | 9 | 2018-12-31 | 2018-12-31 | Ruptured Appendicitis | 19 |
| | 10 | 2019-02-27 | 2019-02-27 | Lower Quadrant Pain | 27 |
| | 17 | 2019-03-04 | 2019-03-04 | Diabetes Mellitus | 9 |
| | 28 | 2019-03-30 | 2019-03-30 | 2019-03-04 Of The Stomach | 26 |
| | 31 | 2018-09-26 | 2018-09-26 | Cardiovascular Disease | 19 |
| | 53 | 2018-10-24 | 2018-10-24 | Urinary Tract Infection | 8 |
| | 70 | 2018-07-17 | 2018-07-17 | Migraine | 20 |
| | 78 | 2018-06-17 | 2018-06-17 | Hypertension | 17 |
| | 91 | 2018-08-30 | 2018-08-30 | Congestive Heart Failure | 3 |
| | 92 | 2019-01-03 | 2019-01-03 | Osteo Arthritis Knee | 5 |
| | 110 | 2019-02-18 | 2019-02-18 | Biliary Colic | 14 |

2 13:44:27 SELECT * FROM innleggelser WHERE innleggesdato = utskrivingsdato LIMIT 0,... 44 row(s) returned

Oppgave 3:

```
SELECT pasient_id, COUNT(innleggesdato) as  
Antall_innleggelser FROM innleggelser WHERE pasient_id = 179;
```

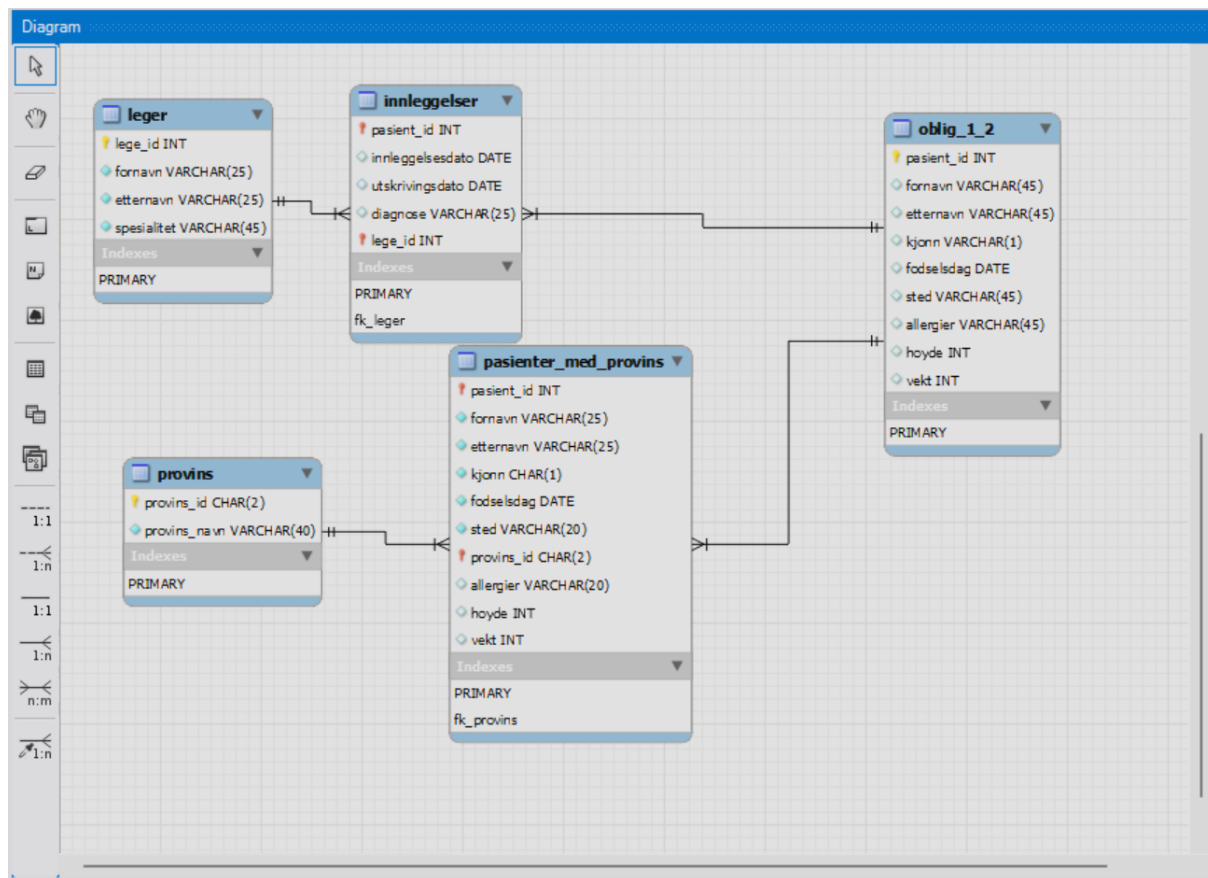
I denne spørringen vil vi ha ut pasient_id og antall innleggesdatoer (med alias), men kun hvor pasient id = 179.

```
7 -- Pasient id og antall innleggelser til pasient 179 --  
8 • SELECT pasient_id, COUNT(innleggesdato) as Antall_innleggelser FROM innleggelser WHERE pasient_id = 179;
```

| Result Grid | | |
|-------------|------------|---------------------|
| | pasient_id | Antall_innleggelser |
| ▶ | 179 | 2 |

| # | Time | Action | Message |
|-----|----------|---|-------------------|
| ✓ 1 | 13:48:43 | SELECT pasient_id, COUNT(innleggesdato) as Antall_innleggelser FROM innleg... | 1 row(s) returned |

Skjema/tabeller



Forklaring av database:

Hvordan er tabellene koblet sammen?

Tabellene er koblet sammen via foreign-keys. Tar vi utgangspunktet i leger-tabellen først, så legger denne grunnlaget for innleggelser-tabellen. I «leger» setter vi lege_id som primærnøkkel. Denne primærnøkkel er en foreign-key i «innleggelser» som skaper en referanse mellom disse to tabellene. «innleggelser» har også en referanse til «pasienter» (for meg: oblig1_2) hvor pasient_id også er en foreign-key som refererer til en annen tabell. Ved å ha disse fremmednøkklene etableres det en forbindelse mellom innleggelser og pasienter, og innleggelser og leger.

I hvilken rekkefølge fyller du tabellene med informasjon?

Ettersom tabellene er koblet sammen, må man fylle de på en spesiell måte. Man må derfor starte med å fylle tabellene som ikke har foreign-keys, som i dette tilfellet var «leger». Leger-tabellen hadde kun en primærnøkkel og andre verdier som ikke var direkte oppkoblet mot noen andre tabeller (ennå). «leger» ble derfor fylt først.

Deretter kunne man fylle «innleggelser». Grunnen til dette, var at begge referansene (pasienter og leger) var i orden. Det samme gjaldt også de tidligere oblig-oppgavene, med pasienter, pasienter_med_provins og provins. Ettersom «pasienter_med_provins» skulle ha to fremmednøkler, måtte man vente til disse var i orden, før man lagde denne tabellen.