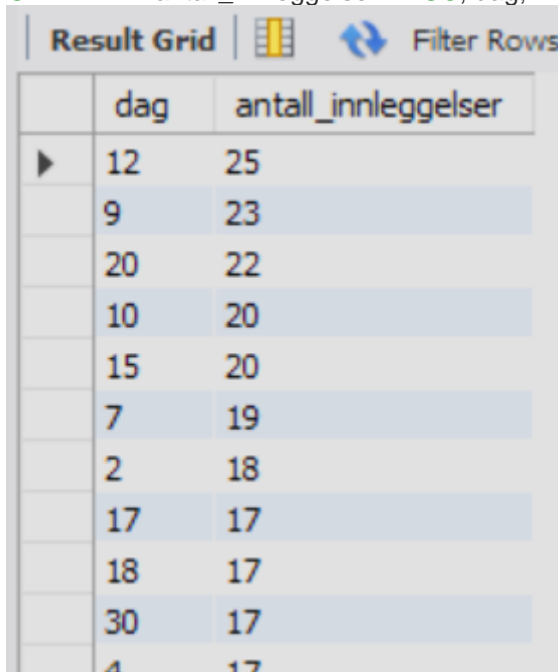


Oppgave 1:

Spørringen under viser dagene i måneden med tall (1-31) og antall innleggelser per dag. Deretter sorteres resultatet fra flest til færrest innleggelser.

```
SELECT DATE_FORMAT(innleggesdato, "%e") as dag, count(*) as antall_innleggelser
FROM innleggelser
GROUP BY dag
ORDER BY antall_innleggelser DESC, dag;
```



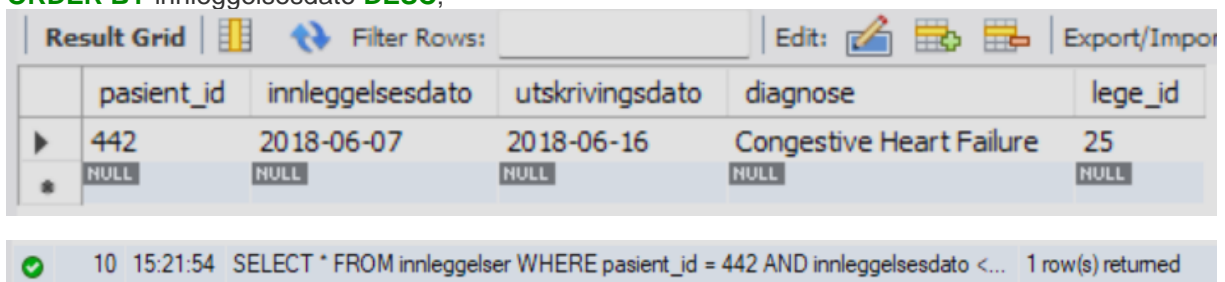
	dag	antall_innleggelser
▶	12	25
	9	23
	20	22
	10	20
	15	20
	7	19
	2	18
	17	17
	18	17
	30	17
	4	17

✓ 5 15:12:25 SELECT DATE_FORMAT(innleggesdato, "%e") as dag, count(*) as antall_innl... 31 row(s) returned

Oppgave 2:

Spørringen under henter alle kolonner for pasienten med ID 442, men da kun den siste innleggelsen. Se kommentaren i spørringen, men ettersom det kun var én innleggelse for 442, skrev jeg <= for å få med denne nå. Hadde det vært flere enn én skulle det vært '<'.

```
SELECT *  
FROM innleggelser  
WHERE pasient_id = 442  
AND innleggelsesdato <= ( /*kun '<' hvis det hadde vært flere innleggelser*/  
    SELECT MAX(innleggelsesdato)  
    FROM innleggelser  
    WHERE pasient_id = 442  
)  
ORDER BY innleggelsesdato DESC;
```



The screenshot shows a database query result grid. The grid has columns: pasient_id, innleggelsesdato, utskrivingsdato, diagnose, and lege_id. The first row shows patient 442 with an admission date of 2018-06-07, discharge date of 2018-06-16, diagnosis of Congestive Heart Failure, and doctor ID 25. Below this row is a row with NULL values. The status bar at the bottom indicates that 1 row(s) returned.

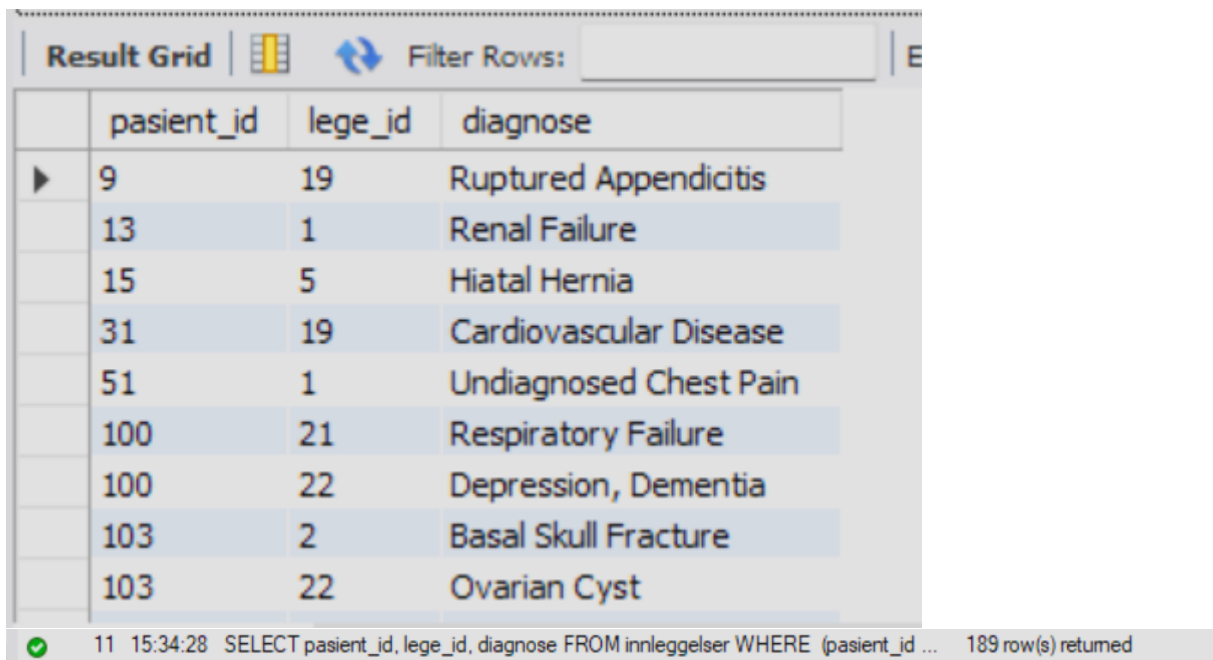
	pasient_id	innleggelsesdato	utskrivingsdato	diagnose	lege_id
▶	442	2018-06-07	2018-06-16	Congestive Heart Failure	25
*	NULL	NULL	NULL	NULL	NULL

10 15:21:54 SELECT * FROM innleggelser WHERE pasient_id = 442 AND innleggelsesdato <... 1 row(s) returned

Oppgave 3:

Spørringen under henter ut pasient id, lege id og diagnose hvis en av følgende kriterer er oppfylt. Første kriterie er at pasient id er et oddetall og lege id må være enten 1, 5, eller 19. Andre kriterie er at Lege id inneholder tallet 2, og pasient id må bestå av 3 tall, altså ha en lengde på 3.

```
SELECT pasient_id, lege_id, diagnose
FROM innleggelser
WHERE
  (pasient_id %2 != 0 and lege_id in (1,5,19))
OR
  (lege_id LIKE "%2%" and length(pasient_id) = 3);
```



The screenshot shows a database query result grid with the following columns: pasient_id, lege_id, and diagnose. The results are as follows:

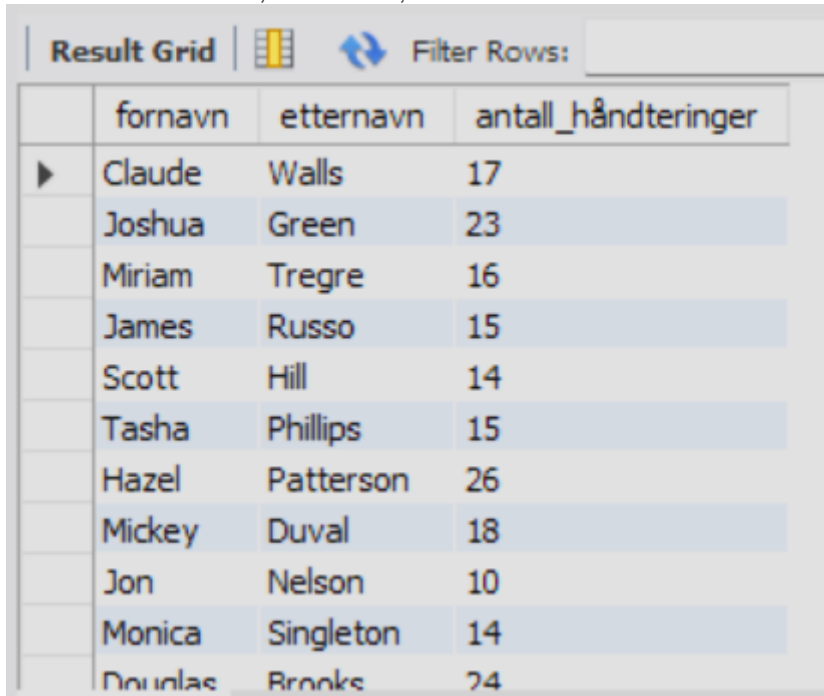
	pasient_id	lege_id	diagnose
▶	9	19	Ruptured Appendicitis
	13	1	Renal Failure
	15	5	Hiatal Hernia
	31	19	Cardiovascular Disease
	51	1	Undiagnosed Chest Pain
	100	21	Respiratory Failure
	100	22	Depression, Dementia
	103	2	Basal Skull Fracture
	103	22	Ovarian Cyst

At the bottom of the screenshot, a status bar shows a green checkmark, the time 11:15:34:28, the SQL query, and the message "189 row(s) returned".

Oppgave 4:

Spørringen under viser fornavn, etternavn og antall innleggelser en lege har håndtert. Resultatet blir deretter sortert på fornavn og etternavn.

```
SELECT l.fornavn, l.etternavn, COUNT(i.innleggesdato) as antall_håndteringer
FROM leger l
JOIN innleggelser i ON l.lege_id = i.lege_id
GROUP BY l.fornavn, l.etternavn;
```



The screenshot shows a 'Result Grid' window with a table of query results. The table has three columns: 'fornavn', 'etternavn', and 'antall_håndteringer'. There are 12 rows of data, each representing a doctor and their count of handled appointments. The rows are: Claude Walls (17), Joshua Green (23), Miriam Tregre (16), James Russo (15), Scott Hill (14), Tasha Phillips (15), Hazel Patterson (26), Mickey Duval (18), Jon Nelson (10), Monica Singleton (14), and Douglas Brooks (24). The table is sorted by first name and then last name.

	fornavn	etternavn	antall_håndteringer
▶	Claude	Walls	17
	Joshua	Green	23
	Miriam	Tregre	16
	James	Russo	15
	Scott	Hill	14
	Tasha	Phillips	15
	Hazel	Patterson	26
	Mickey	Duval	18
	Jon	Nelson	10
	Monica	Singleton	14
	Douglas	Brooks	24



7 16:42:40 SELECT l.fornavn, l.etternavn, COUNT(i.innleggesdato) as antall_håndteringer ... 27 row(s) returned

Oppgave 5:

Spørringen under viser hver lege sin id, fullt navn og første og siste pasienthåndtering (innleggingsdato)

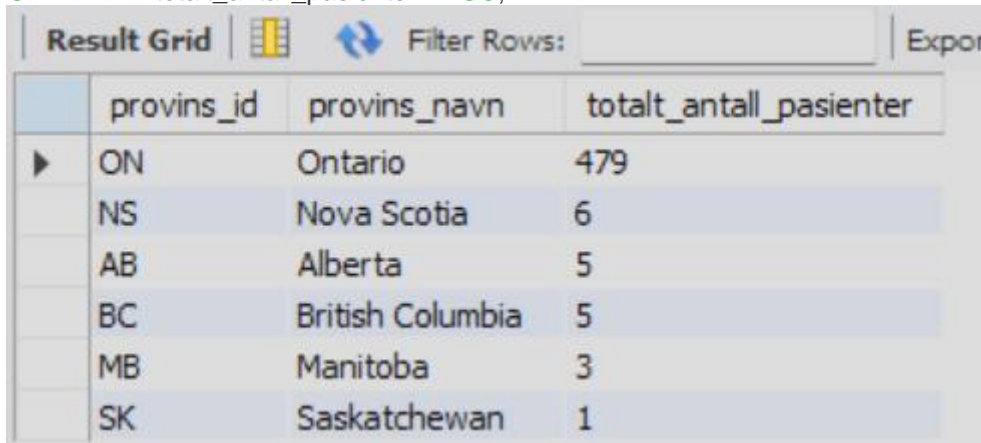
```
SELECT l.lege_id, CONCAT(l.fornavn, ' ', l.etternavn) AS fullt_navn, MIN(i.innleggingsdato) AS  
første_innleggingsdato, MAX(i.innleggingsdato) AS siste_innleggingsdato  
FROM leger l  
JOIN innleggelser i ON l.lege_id = i.lege_id  
GROUP BY l.lege_id, fullt_navn;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Con
	lege_id	fullt_navn	første_innleggingsdato	siste_innleggingsdato
▶	1	Claude Walls	2018-06-20	2019-05-07
	2	Joshua Green	2018-07-03	2019-06-02
	3	Miriam Tregre	2018-06-09	2019-04-07
	4	James Russo	2018-06-15	2019-05-06
	5	Scott Hill	2018-07-11	2019-05-06
	6	Tasha Phillips	2018-08-03	2019-06-02
	7	Hazel Patterson	2018-06-20	2019-05-03
	8	Mickey Duval	2018-06-10	2019-06-05
	9	Jon Nelson	2018-06-30	2019-04-10
	10	Monica Singleton	2018-06-09	2019-04-25
	11	Douglas Brooks	2018-06-14	2019-04-07
✓	11	17:09:31	SELECT l.lege_id, CONCAT(l.fornavn, ' ', l.etternavn) AS fullt_navn, MIN(i.innlegg... 27 row(s) returned	

Oppgave 6:

Spørringen under viser antall pasienter for hver provins. Den viser provins id, provins navn og antall pasienter for hver provins. Resultatet blir sortert fra høyest til lavest antall per provins.

```
SELECT pmp.provins_id, p.provins_navn, COUNT(*) as totalt_antall_pasienter
FROM pasienter_med_provins as pmp
JOIN provins p ON pmp.provins_id = p.provins_id
GROUP BY pmp.provins_id, p.provins_navn
ORDER BY totalt_antall_pasienter DESC;
```



The screenshot shows a database interface with a 'Result Grid' tab. The grid displays the results of the SQL query, sorted by the number of patients in descending order. The columns are 'provins_id', 'provins_navn', and 'totalt_antall_pasienter'. The data rows are as follows:

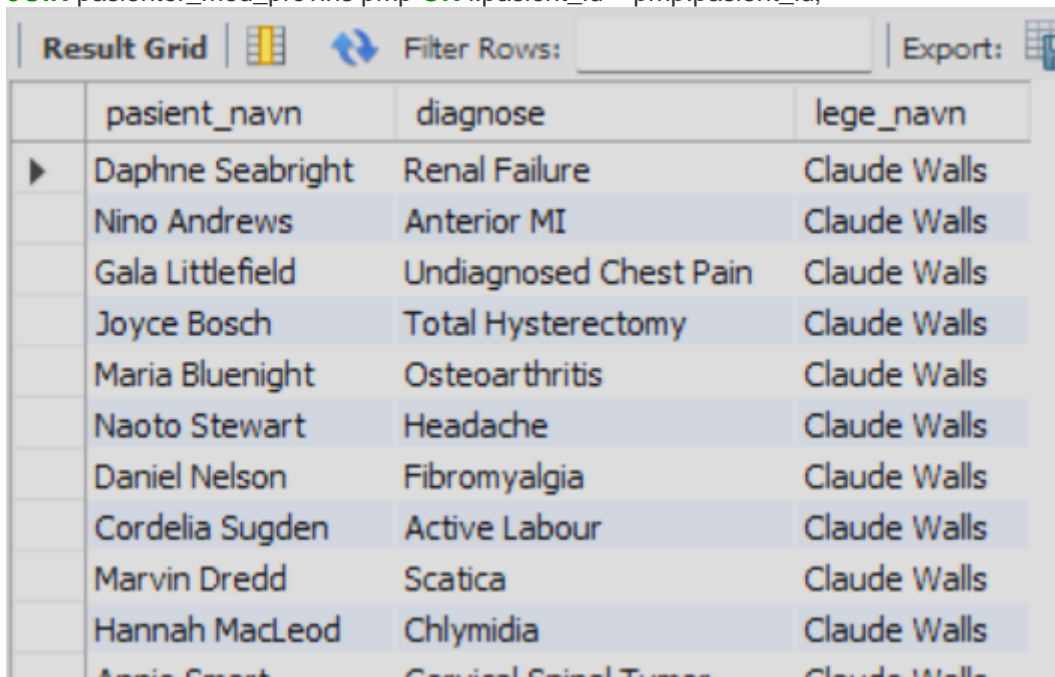
	provins_id	provins_navn	totalt_antall_pasienter
▶	ON	Ontario	479
	NS	Nova Scotia	6
	AB	Alberta	5
	BC	British Columbia	5
	MB	Manitoba	3
	SK	Saskatchewan	1

7 10:28:33 SELECT pmp.provins_id, p.provins_navn, COUNT(*) as totalt_antall_pasienter FR... 6 row(s) returned

Oppgave 7:

Spørringen under viser pasientens navn, diagnose og legens navn, for hver registrerte innleggelse.

```
SELECT CONCAT(pmp.fornavn, ' ', pmp.etternavn) AS pasient_navn, i.diagnose, CONCAT(l.fornavn, ' ', l.etternavn) AS lege_navn
FROM innleggelser i
JOIN leger l ON i.lege_id = l.lege_id
JOIN pasienter_med_provins pmp ON i.pasient_id = pmp.pasient_id;
```



The screenshot shows a database query result grid with the following columns: pasient_navn, diagnose, and lege_navn. The results are as follows:

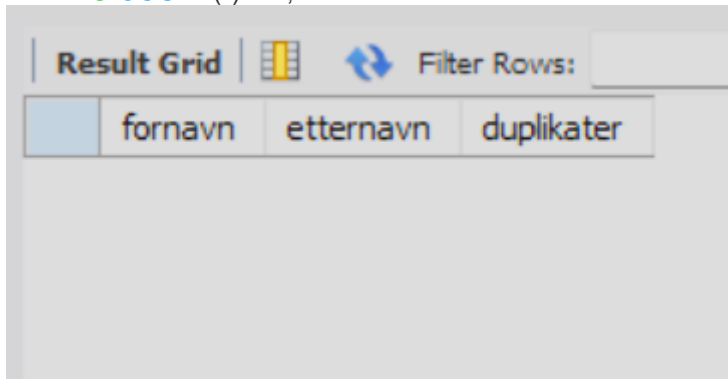
	pasient_navn	diagnose	lege_navn
▶	Daphne Seabright	Renal Failure	Claude Walls
	Nino Andrews	Anterior MI	Claude Walls
	Gala Littlefield	Undiagnosed Chest Pain	Claude Walls
	Joyce Bosch	Total Hysterectomy	Claude Walls
	Maria Bluenight	Osteoarthritis	Claude Walls
	Naoto Stewart	Headache	Claude Walls
	Daniel Nelson	Fibromyalgia	Claude Walls
	Cordelia Sugden	Active Labour	Claude Walls
	Marvin Dredd	Scatica	Claude Walls
	Hannah MacLeod	Chlymidia	Claude Walls
	Aggie Smart	Cervical Spinal Tumor	Claude Walls

9 10:44:17 SELECT CONCAT(pmp.fornavn, ' ', pmp.etternavn) AS pasient_navn, i.diagnose, ... 483 row(s) returned

Oppgave 8:

Spørringen under viser fornavn og etternavn og antall duplikater hvis det finnes noen med samme fornavn og etternavn. Hvis dette er tilfelle, vil «duplikater» øke.

```
SELECT fornavn, etternavn, COUNT(*) as duplikater  
FROM pasienter_med_provins  
GROUP BY fornavn, etternavn  
HAVING COUNT(*) > 1;
```



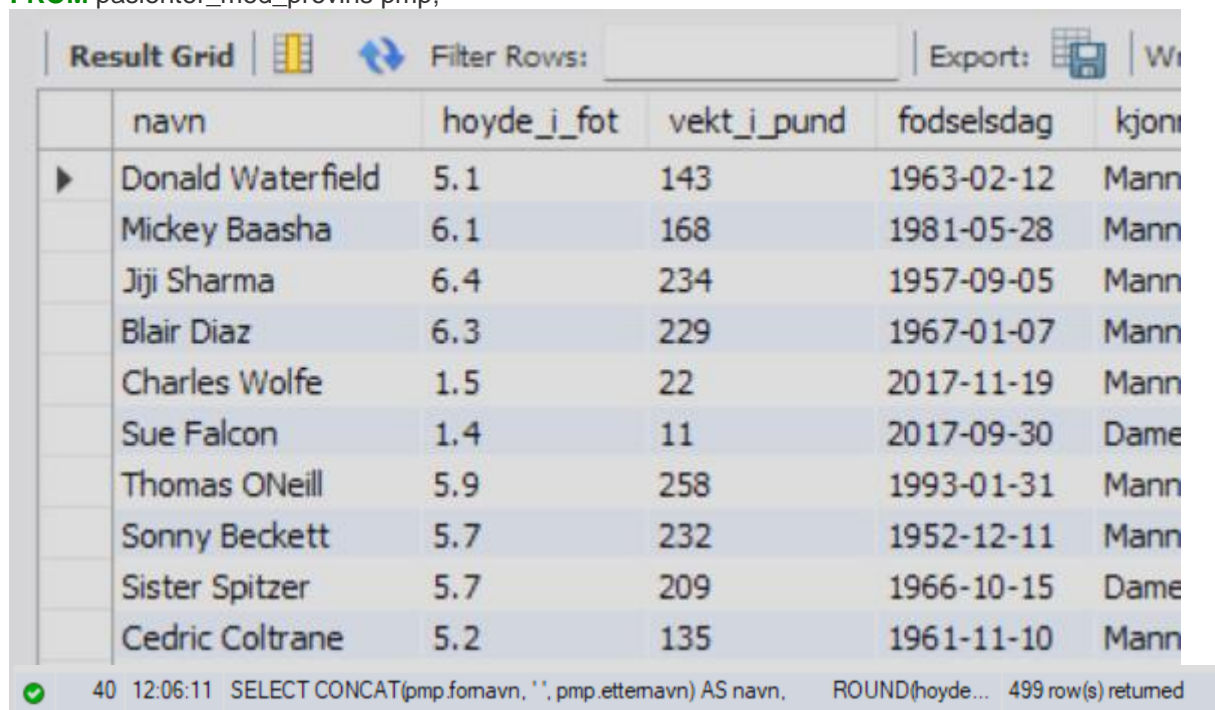
The screenshot shows a database interface with a 'Result Grid' tab. The grid has three columns: 'fornavn', 'etternavn', and 'duplikater'. The grid is currently empty, showing only the column headers.

✓ 48 12:20:34 SELECT fornavn, etternavn, COUNT(*) as duplikater FROM pasienter_med_provi... 0 row(s) returned

Oppgave 9:

Spørringen under viser pasientens navn, høyde i fot, vekt i pund, fødselsdag og kjønn som ord og ikke enkelt bokstav.

```
SELECT CONCAT(pmp.fornavn, ' ', pmp.etternavn) AS navn,  
       ROUND(hoyde/30.48, 1) AS hoyde_i_fot,  
       ROUND(vekt * 2.205) AS vekt_i_pund,  
       fødselsdag,  
       IF(kjonn = 'M', 'Mann', 'Dame') AS kjonn  
FROM pasienter_med_provins pmp;
```



The screenshot shows a database query result grid with the following columns: navn, hoyde_i_fot, vekt_i_pund, fødselsdag, and kjonn. The data is displayed in a table with 10 rows. The status bar at the bottom indicates that 499 rows were returned.

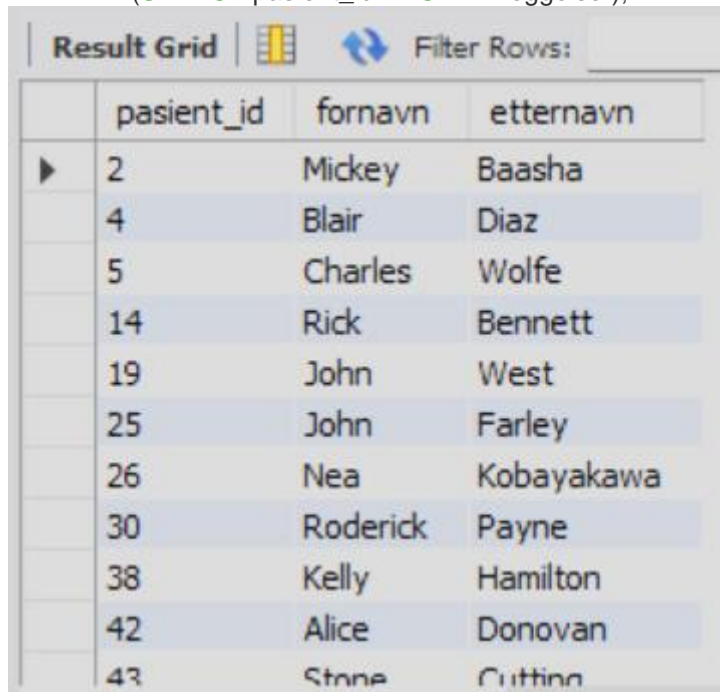
	navn	hoyde_i_fot	vekt_i_pund	fødselsdag	kjonn
▶	Donald Waterfield	5.1	143	1963-02-12	Mann
	Mickey Baasha	6.1	168	1981-05-28	Mann
	Jiji Sharma	6.4	234	1957-09-05	Mann
	Blair Diaz	6.3	229	1967-01-07	Mann
	Charles Wolfe	1.5	22	2017-11-19	Mann
	Sue Falcon	1.4	11	2017-09-30	Dame
	Thomas O'Neill	5.9	258	1993-01-31	Mann
	Sonny Beckett	5.7	232	1952-12-11	Mann
	Sister Spitzer	5.7	209	1966-10-15	Dame
	Cedric Coltrane	5.2	135	1961-11-10	Mann

40 12:06:11 SELECT CONCAT(pmp.fornavn, ' ', pmp.etternavn) AS navn, ROUND(hoyde... 499 row(s) returned

Oppgave 10:

Spørringen under viser pasient id, fornavn og etternavn til personer som aldri har vært innlagt.

```
SELECT pasient_id, fornavn, etternavn  
FROM pasienter_med_provins  
WHERE pasient_id NOT IN  
      (SELECT pasient_id FROM innleggelser);
```



The screenshot shows a database query result grid. At the top, there is a toolbar with 'Result Grid', a table icon, a refresh icon, and a 'Filter Rows:' dropdown. Below the toolbar is a table with four columns: 'pasient_id', 'fornavn', and 'etternavn'. The table contains 11 rows of data. The first row is highlighted with a blue arrow icon in the first column. The last row is partially cut off.

	pasient_id	fornavn	etternavn
▶	2	Mickey	Baasha
	4	Blair	Diaz
	5	Charles	Wolfe
	14	Rick	Bennett
	19	John	West
	25	John	Farley
	26	Nea	Kobayakawa
	30	Roderick	Payne
	38	Kelly	Hamilton
	42	Alice	Donovan
	43	Stone	Cutting

✓ 47 12:14:14 SELECT pasient_id, fornavn, etternavn FROM pasienter_med_provins WHERE p... 155 row(s) returned