

Oppgave(1)

TIMELISTE

```

=> SELECT * FROM timeliste;

```

timelistenr	status	levert	utbetalt	beskrivelse
1	utbetalt	2016-07-04	2016-07-13	HMS-kurs
2	utbetalt	2016-07-08	2016-07-13	Innføring
3	utbetalt	2016-07-19	2016-07-27	Test av database
4	levert	2016-07-20		Innlegging av virksomhetsdokumenter
5	utbetalt	2016-07-20	2016-07-27	Oppsporing av manglende underlagsinformasjon
6	aktiv			Identifisering av manglende funksjonalitet
7	utbetalt	2016-08-01	2016-08-10	Opprettelse av testdatabase
8	aktiv	2016-08-10		Videreutvikling av testdatabase

(8 rows)

TIMELISTELINJE

```

=> SELECT * FROM timelistelinje;

```

timelistenr	linjenr	startdato	starttid	sluttid	pause	beskrivelse
1	1	2016-07-01	09:00:00	12:00:00		HMS del 1
1	2	2016-07-04	09:00:00	12:00:00		HMS del 2
2	1	2016-07-01	13:00:00	15:00:00	15	Innføring
3	1	2016-07-01	15:00:00	16:00:00		Test 1
3	2	2016-07-04	13:15:00	17:00:00	40	Test 2
3	3	2016-07-04	22:00:00	01:00:00	30	Test 3
3	4	2016-07-05	14:00:00	18:00:00		Test 4
3	5	2016-07-06	10:00:00	16:50:00	55	Test 5
3	6	2016-07-07	10:00:00	12:00:00		Test 6
3	7	2016-07-07	15:00:00	18:00:00	20	Test 7
3	8	2016-07-08	13:00:00	13:50:00		Test 8
3	9	2016-07-09	22:00:00	03:00:00	25	Retesting
4	1	2016-07-05	13:00:00	14:00:00		innlegging
4	2	2016-07-08	11:00:00	12:00:00		innlegging
4	3	2016-07-11	14:20:00	16:55:00	45	innlegging
4	4	2016-07-15	15:00:00	17:00:00		innlegging
4	5	2016-07-20	10:00:00	11:45:00		innlegging
4	6	2016-07-20	12:00:00	13:45:00		Enhetstesting
5	1	2016-07-13	09:15:00	12:00:00		Leting i arkivet
5	2	2016-07-18	14:30:00	16:00:00		Leting i arkivet
5	3	2016-07-19	15:45:00	17:20:00	20	Søk i databasene
5	4	2016-07-21	13:00:00	14:00:00		Leting i arkivet
6	1	2016-08-01	13:15:00	14:00:00		Diskusjoner
6	2	2016-08-02	11:00:00	12:10:00		Diskusjoner
6	3	2016-08-05	14:00:00	17:00:00	45	Skriving av notat
7	1	2016-07-13	14:05:00	16:10:00		Innlegging av data
7	2	2016-07-14	09:20:00	13:00:00	45	Vasking av data
7	3	2016-07-15	10:00:00	12:00:00		Testing
7	4	2016-07-18	18:00:00	00:15:00	50	Testing
7	5	2016-07-19	18:00:00	20:15:00		Innlegging av data
7	6	2016-07-21	17:15:00	22:00:00	35	Testing
7	7	2016-07-21	23:15:00	01:10:00		Feilsøking
7	8	2016-07-26	09:00:00	11:35:00		Testing
7	9	2016-08-01	10:30:00	12:40:00		Stresstesting

(34 rows)

VARIGHET

```
=> SELECT * FROM varighet;
```

timelistenr	linjenr	varighet
1	1	180
1	2	180
2	1	105
3	1	60
3	2	185
3	3	150
3	4	240
3	5	355
3	6	120
3	7	160
3	8	50
3	9	275
4	1	60
4	2	60
4	3	110
4	4	120
4	5	105
4	6	105
5	1	165
5	2	90
5	3	75
5	4	60
6	1	45
6	2	70
6	3	135
7	1	125
7	2	175
7	3	120
7	4	325
7	5	135
7	6	250
7	7	115
7	8	155
7	9	130

(34 rows)

Oppgave (2)

- a) Timelistelinjer som er lagt inn for timeliste nummer 3:

```
SELECT * FROM timelistelinje WHERE timelistenr = 3;
```

```
> SELECT * FROM timelistelinje WHERE timelistenr = 3;
```

timelistenr	linjenr	startdato	starttid	sluttid	pause	beskrivelse
3	1	2016-07-01	15:00:00	16:00:00		Test 1
3	2	2016-07-04	13:15:00	17:00:00	40	Test 2
3	3	2016-07-04	22:00:00	01:00:00	30	Test 3
3	4	2016-07-05	14:00:00	18:00:00		Test 4
3	5	2016-07-06	10:00:00	16:50:00	55	Test 5
3	6	2016-07-07	10:00:00	12:00:00		Test 6
3	7	2016-07-07	15:00:00	18:00:00	20	Test 7
3	8	2016-07-08	13:00:00	13:50:00		Test 8
3	9	2016-07-09	22:00:00	03:00:00	25	Retesting

(9 rows)

- b) Hvor mange timelister det er:

```
SELECT count(*) FROM timeliste;
```

```
=> SELECT count(*) FROM timeliste;
```

count
8

(1 row)

- c) Hvor mange timelister som det ikke er utbetalt penger for:

```
SELECT count (*) FROM timeliste WHERE NOT status LIKE '%utbetalt%';
```

```
=> SELECT count (*) FROM timeliste WHERE NOT status LIKE '%utbetalt%';
```

count
3

(1 row)

- d) Antall timelistelinjer, og antall timelistelinjer med en pauseverdi:

```
SELECT count(*) AS antall, count(pause) AS antallmedpause FROM timelistelinje;
```

```
=> SELECT count(*) AS antall, count(pause) AS antallmedpause FROM timelistelinje;
```

antall	antallmedpause
34	12

(1 row)

- e) Alle timelistelinjer som ikke har pauseverdier (der pause er satt til null):

```
SELECT * FROM timelistelinje WHERE pause IS NULL;
```

```
=> SELECT * FROM timelistelinje WHERE pause IS NULL;
```

timelistenr	linjenr	startdato	starttid	sluttid	pause	beskrivelse
1	1	2016-07-01	09:00:00	12:00:00		HMS del 1
1	2	2016-07-04	09:00:00	12:00:00		HMS del 2
3	1	2016-07-01	15:00:00	16:00:00		Test 1
3	4	2016-07-05	14:00:00	18:00:00		Test 4
3	6	2016-07-07	10:00:00	12:00:00		Test 6
3	8	2016-07-08	13:00:00	13:50:00		Test 8
4	1	2016-07-05	13:00:00	14:00:00		innlegging
4	2	2016-07-08	11:00:00	12:00:00		innlegging
4	4	2016-07-15	15:00:00	17:00:00		innlegging
4	5	2016-07-20	10:00:00	11:45:00		innlegging
4	6	2016-07-20	12:00:00	13:45:00		Enhetstesting
5	1	2016-07-13	09:15:00	12:00:00		Leting i arkivet
5	2	2016-07-18	14:30:00	16:00:00		Leting i arkivet
5	4	2016-07-21	13:00:00	14:00:00		Leting i arkivet
6	1	2016-08-01	13:15:00	14:00:00		Diskusjoner
6	2	2016-08-02	11:00:00	12:10:00		Diskusjoner
7	1	2016-07-13	14:05:00	16:10:00		Innlegging av data
7	3	2016-07-15	10:00:00	12:00:00		Testing
7	5	2016-07-19	18:00:00	20:15:00		Innlegging av data
7	7	2016-07-21	23:15:00	01:10:00		Feilsøking
7	8	2016-07-26	09:00:00	11:35:00		Testing
7	9	2016-08-01	10:30:00	12:40:00		Stresstesting

(22 rows)

Oppgave (3)

- a) Antall timer som det ikke er utbetalt penger for:

```
SELECT sum(varig.varighet) / 60 AS antall_ubetalt_time FROM (SELECT v.varighet FROM
timeliste AS t INNER JOIN varighet AS v ON (t.timelistenr = v.timelistenr) WHERE NOT t.status
LIKE '%utbetalt%')AS varig;
```

```
> SELECT sum(varig.varighet) / 60 AS antall_ubetalt_time FROM (SELECT v.varighet FROM timeliste AS t INNER JOIN
varighet AS v ON (t.timelistenr = v.timelistenr) WHERE NOT t.status LIKE '%utbetalt%')AS varig;
antall_ubetalt_time
-----
13
(1 row)
```

- b) Hvilke timelister (nr og beskrivelse) har en timelistelinje med en beskrivelse som inneholder 'test' eller 'Test'. Ikke vis duplikater:

```
SELECT DISTINCT tl.timelistenr , tl.beskrivelse FROM timeliste AS tl INNER JOIN timelistelinje
AS tlj ON(tl.timelistenr = tlj.timelistenr) WHERE (tlj.beskrivelse LIKE '%Test%') OR
(tlj.beskrivelse LIKE '%test%');
```

```

=> SELECT DISTINCT tl.timelistenr , tl.beskrivelse FROM timeliste AS tl INNER JOIN timelistelinje AS tlj ON(tl.timelistenr = tlj.timelistenr) WHERE (tlj.beskrivelse LIKE '%Test%') OR (tlj.beskrivelse LIKE '%test%');
timelistenr |      beskrivelse
-----+-----
7 | Opprettelse av testdatabase
4 | Innlegging av virksomhetsdokumenter
3 | Test av database
(3 rows)

```

- c) Hvor mye penger som har blitt utbetalt, dersom man blir utbetalt 200 kr per time:

```

SELECT (sum(v.varighet) / 60) * 200 AS Utbetalt_penger_per_kr FROM timeliste AS tl INNER JOIN varighet AS v ON (tl.timelistenr = v.timelistenr) WHERE tl.status LIKE '%utbetalt%';

```

```

=> SELECT (sum(v.varighet) / 60) * 200 AS Utbetalt_penger_per_kr FROM timeliste AS tl INNER JOIN varighet AS v ON (tl.timelistenr = v.timelistenr) WHERE tl.status LIKE '%utbetalt%';
utbetalt_penger_per_kr
-----
13200
(1 row)

```

Oppgave (4)

- (a) Det som vi får fra NATURAL JOIN spørringen er kun en rad, det vi si at NATURAL JOIN fjerner alle kolonner som har likt navn(timelistenr, beskrivelse) og så fjerner verdiene i rader som ikke er like, derfor får vi bare timelistenr 2 med beskrivelse innføring som er helt likt i både timeliste og timelistelinje, mens det som INNER JOIN gjør, er egentlig kartetiskprodukt hvor relasjonene har noe i felles (fremmednøkkel), dvs at timeliste joiner til timelistelinje hvor timeliste sitt timelistenr er likt til timelistelinje sitt timelistenr, derfor har vi 34 rader med INNER JOIN.
- (b) Resultatet ved å eksekvere spørringene i både NATURAL JOIN og INNER JOIN er 34 rader. Siden det som er felles kolonne mellom relasjonene timeliste og viewet varighet er kun timelistenr som er helt likt i timeliste sitt timelistenr med varighet sitt timelistenr.