

DAT151 – OBLIG5

Optimalisering og implementasjon av større database.

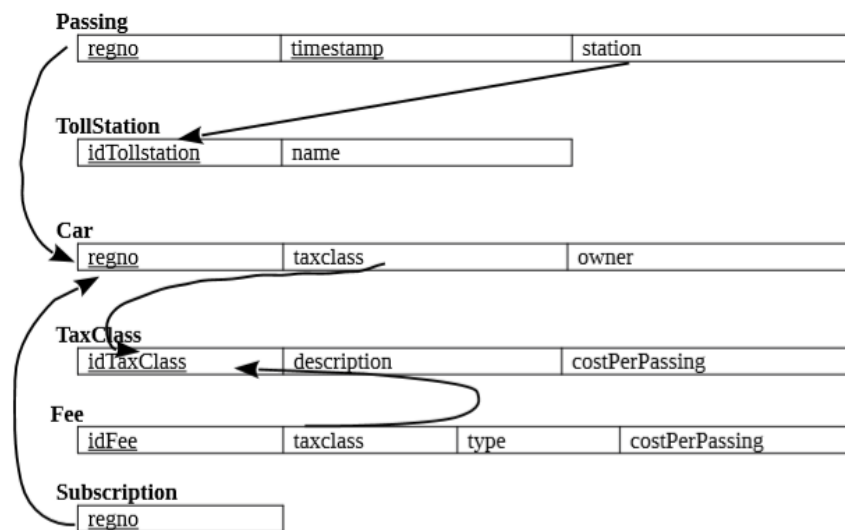


Adrian R.J. Mortensen

Innhold

Oppgave 1: Normal form.....	2
1. Er skjemaet 1NF?.....	2
2. Er det 2NF?	2
3. er den 3nf	2
Oppgave 2: Implementasjon av fysisk skjema og test miljø.....	3
Oppgave 3: Optimalisering av database.....	4
a) Query 1	4
b) Query 2	4
c) Query 3	5
d) Query 4 (same as 2).....	5
e) Query 5 (same as 3).....	6
f) Query 5	6

Oppgave 1: Normal form



1. Er skjemaet 1NF?

Et skjema er 1NF hvis og bare hvis alle underliggende domener har atomiske verdier.

All data er atome om vi antar «timestamp» er det.

2. Er det 2NF?

Et skjema er 2NF hvis og bare hvis det er 1NF og alle ikke nøkkel attributter er knyttet til en kandidat nøkkel eller en annen ikke nøkkel attributt.

Her kan man diskutere at Passing ikke følger denne regelen. Ettersom tollstasjon ikke er avhengig av hverken «regno» eller «timestamp».

3. er den 3nf

Om man mener station er knyttet til timestamp og regno er den det. Ellers er den ikke det siden den ikke fyller 2nf.

Oppgave 2: Implementasjon av fysisk skjema og test miljø.

Jeg bruker skjemaet fra forrige oppgave uten endringer.

SQL filer kan finnes på <https://github.com/H571531/DAT151/tree/master/Oblig5>

På grunn av hardware limitasjoner ble denne oppgaven litt mindre ferdig.

Jeg kjører MariaDB på en raspberry pi 4 (4GB Ram, med ARM prosessor) så enkelte av importene av data ble for heftige.

Dette skaper litt følge feil.

Oppgave 3: Optimalisering av database

a) Query 1

Finner navnet og tiden til noen som har kjørt igjennom en bomstasjon en spesifikk dag.

```
> SELECT SQL_NO_CACHE C.owner, P.timestamp
-> FROM Car C JOIN Passing P USING(regno)
-> WHERE YEAR(P.timestamp)=2018 AND MONTH(P.timestamp)=3
-> AND DAYOFWEEK(P.timestamp)=1;
```

```
+-----+-----+
| owner          | timestamp          |
+-----+-----+
| Urfan Sandøy    | 2018-03-04 04:30:00 |
| Urfan Sandøy    | 2018-03-04 05:42:00 |
| Birte Fossum    | 2018-03-04 02:18:00 |
| Birte Fossum    | 2018-03-04 23:31:00 |
| Stanley Ingvaldsen | 2018-03-04 02:18:00 |
| Stanley Ingvaldsen | 2018-03-04 23:31:00 |
| Amar Wiig       | 2018-03-04 05:53:00 |
| Minda Larssen   | 2018-03-04 14:24:00 |
| Storm Nordstrand | 2018-03-04 03:09:00 |
| Marcus Hafstad  | 2018-03-04 14:24:00 |
| Mathilde Lillevik | 2018-03-04 04:30:00 |
| Mathilde Lillevik | 2018-03-04 05:42:00 |
| Zilan Solbakken  | 2018-03-04 05:53:00 |
| Kurt Aslaksen   | 2018-03-04 19:02:00 |
| Dilara Skar     | 2018-03-04 03:09:00 |
| Annette Sara    | 2018-03-04 19:02:00 |
+-----+-----+
16 rows in set (18.286 sec)
```

b) Query 2

Finner de som har abonnement med total kostnad over 4000

```
MariaDB [oblig5]> SELECT SQL_NO_CACHE C.owner AS carowner,
Sum(F.costPerPassing) AS totalfee
-> FROM Car C JOIN Passing P USING(regno)
-> JOIN TaxClass T ON C.taxclass=T.idTaxClass
-> JOIN Fee F ON F.taxclass=T.idTaxClass
-> JOIN Subscription S USING(regno)
-> WHERE F.type='withsubscription'
-> GROUP BY C.owner HAVING totalfee > 4000;
```

```
+-----+-----+
| carowner        | totalfee |
+-----+-----+
| Ansgar Oftedal  | 4210.00  |
| Rina Kvalheim   | 4605.00  |
+-----+-----+
2 rows in set (1 min 21.136 sec)
```

c) Query 3

Nesten samme spørringen som 2 men med en mindre join. Dermed litt raskere også.

```
MariaDB [oblig5]> SELECT SQL_NO_CACHE C.owner AS carowner,
Sum(F.costPerPassing) AS totalfee
-> FROM Car C JOIN Passing P USING(regno)
-> JOIN TaxClass T ON C.taxclass=T.idTaxClass
-> JOIN Fee F ON F.taxclass=T.idTaxClass
-> WHERE F.type='withsubscription'
-> AND C.regno IN (SELECT regno FROM Subscription)
-> GROUP BY C.owner HAVING totalfee > 4000;
```

carowner	totalfee
Ansgar Oftedal	4210.00
Rina Kvalheim	4605.00

d) Query 4 (same as 2)

Samme som spørring 2 bare med

```
MariaDB [oblig5]> SELECT SQL_NO_CACHE C.owner AS carowner,
Sum(F.costPerPassing) AS totalfee
-> FROM Car C JOIN Passing P USING(regno)
-> JOIN TaxClass T ON C.taxclass=T.idTaxClass
-> JOIN Fee F ON F.taxclass=T.idTaxClass
-> JOIN Subscription S USING(regno)
-> WHERE F.type='regular'
-> GROUP BY C.owner HAVING totalfee > 4000;
```

carowner	totalfee
Ansgar Oftedal	4300.00
Rina Kvalheim	4605.00

2 rows in set (1 min 18.016 sec)

Noe merkelig har skjedd i denne spørringen. (eventuelt med import av data. Får samme output)

e) Query 5 (same as 3)

Her skal man også få samme som i spørring 2/3 bare med 'regular' ikke 'withsubscription'

```
MariaDB [oblig5]> SELECT SQL_NO_CACHE C.owner AS carowner,
Sum(F.costPerPassing) AS totalfee
-> FROM Car C JOIN Passing P USING(regno)
-> JOIN TaxClass T ON C.taxclass=T.idTaxClass
-> JOIN Fee F ON F.taxclass=T.idTaxClass
-> WHERE F.type='regular'
-> AND C.regno IN (SELECT regno FROM Subscription)
-> GROUP BY C.owner HAVING totalfee > 4000;

+-----+-----+
| carowner      | totalfee |
+-----+-----+
| Ansgar Oftedal | 4300.00  |
| Rina Kvalheim  | 4605.00  |
+-----+-----+
2 rows in set (1 min 18.017 sec)
```

f) Query 5

Denne spørringen finner navnet til de som har gjort en passering på en spesifikk dag og tid.

```
ariaDB [oblig5]> SELECT SQL_NO_CACHE C.owner FROM Car C WHERE C.regno
-> IN (SELECT P.regno FROM Passing P JOIN Tollstation T
-> ON P.tollstation = T.idTollstation
-> WHERE T.name LIKE 'Gravdal'
-> AND YEAR(P.timestamp)=2018
-> AND MONTH(P.timestamp)=2
-> AND DAYOFWEEK(P.timestamp)=1
-> AND HOUR(P.timestamp) = 3);

+-----+-----+
| owner      |
+-----+-----+
| Olav Hætta |
| Bjarnhild Reistad |
| Jarle Aarnes |
| Årstein Sele |
| Kay Nettet |
|----Forkortning----|
| Noor Evensen |
| Danny Straume |
| Hossein Kolberg |
| Hassan Haugland |
| Odin Persson |
| Enid Nicolaysen |
| Oddveig Roald |
| Xhavit Høiland |
| Vebjørn Bauge |
| Ramona Søvik |
| Oda Dammen |
| Nicholas Heiberg |
+-----+-----+
177 rows in set (0.526 sec)
```

Denne ville jeg denormalisert til å ha mer info i passing som f.eks Tollstasjon navn og Bileier navn.
Dermed sluppet unna

