# Oblig 1

Anders P. Åsbø

(Dated: 11. september 2022)

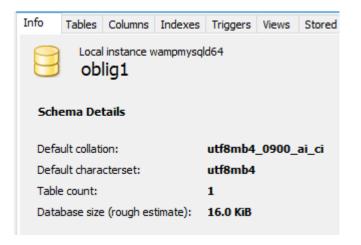
### **OPPGAVE 1**

a)

For å lage databasen "Oblig1bruker jeg sql setningen

```
oblig1> CREATE SCHEMA Oblig1
[2022-09-11 16:51:14] 1 row affected in 9 ms
```

Som resulterer i

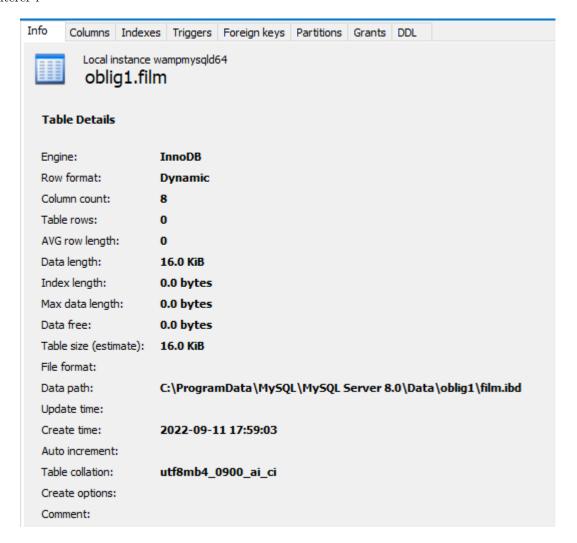


b)

For å lage tabellen Film", så bruker jeg følgende sql:

```
oblig1> CREATE TABLE Oblig1.Film (
FNr INTEGER UNSIGNED NOT NULL,
Tittel VARCHAR(40) NOT NULL,
Ar SMALLINT UNSIGNED,
Land VARCHAR(20),
Sjanger VARCHAR(20),
Alder TINYINT UNSIGNED,
Tid SMALLINT UNSIGNED,
Pris DECIMAL(5, 2),
PRIMARY KEY (FNr)

12 [2022-09-11 17:25:37] completed in 19 ms
```



c)

Jeg legger til dataene fra tabellen i boken med følgende sql:

```
oblig1> INSERT INTO Oblig1.Film
                               VALUES (1, 'Casablanca', 1942, 'USA', 'Drama', 15, 102, 149.00), (2, 'Fort Apachea', 1948, 'USA', 'Western', 15, 127, NULL),
  3
                                                (2, 'Fort Apachea', 1948, 'Usa', 'Western', 18, 12', NULL),
(3, 'Apocalypse Now', 1979, 'USA', 'Action', 18, 155, 123.00),
(4, 'Streets of Fire', 1984, 'USA', 'Action', 15, 93, NULL),
(5, 'High Noon', 1952, 'USA', 'Western', 15, 85, 123.00),
(6, 'Cinema Paradiso', 1988, 'Italia', 'Komedie', 11, 123, NULL),
  5
  6
7
                                                (6, Cinema Paradiso', 1988, 'Italia', 'Komedie', 11, 123, NULL),
(7, 'Asterix hos britene', 1988, 'Frankrike', 'Tegnefilm', 7, 78, 149.00),
(8, 'Veiviseren', 1987, 'Norge', 'Action', 15, 96, 87.00),
(9, 'Salmer fra kjøkkenet', 2002, 'Norge', 'Komedie', 7, 80, 149.00),
(10, 'Anastasia', 1997, 'USA', 'Tegnefilm', 7, 94, 123.00),
(11, 'La Grande bouffe', 1973, 'Frankrike', 'Drama', 15, 129, 87.00),
(12, 'Blues Brothers 2000', 1998, 'USA', 'Komedie', 11, 124, 135.00),
(13, 'Paradisc', 1988, 'Starkritannia', 'Musikk', 11, 144, NULL)
  8
  9
10
11
12
13
                                                 (13, 'Beatles: Help', 1965, 'Storbritannia', 'Musikk', 11, 144, NULL)
14
15
            [2022-09-11 18:13:34] 13 {\tt rows} affected in 6 ms
16
            oblig1> SELECT * FROM Oblig1.Film
17
            [2022-09-11 18:19:18] 13 \frac{1}{1} rows retrieved starting from 1 in 39 ms (execution: 5 ms, fetching: 34 ms)
```

	₽ FNr ÷	.⊞ Tittel ÷	∎∄År ÷	III Land ÷	∎ Sjanger ÷	∎ Alder ÷	∎≣ Tid ÷	I≣ Pris ÷
1	1	Casablanca	1942	USA	Drama	15	102	149.00
2	2	Fort Apachea	1948	USA	Western	15	127	
3	3	Apocalypse Now	1979	USA	Action	18	155	123.00
4		Streets of Fire	1984	USA	Action	15	93	
5		High Noon	1952	USA	Western	15	85	123.00
6		Cinema Paradiso	1988	Italia	Komedie	11	123	
7	7	Asterix hos britene	1988	Frankrike	Tegnefilm	7	78	149.00
8	8	Veiviseren	1987	Norge	Action	15	96	87.00
9	9	Salmer fra kjøkkenet	2002	Norge	Komedie	7	80	149.00
10	10	Anastasia	1997	USA	Tegnefilm	7	94	123.00
11	11	La Grande bouffe	1973	Frankrike	Drama	15	129	87.00
12	12	Blues Brothers 2000	1998	USA	Komedie	11	124	135.00
13	13	Beatles: Help	1965	Storbritannia	Musikk	11	144	

## d)

For å finne tittel, sjanger og pris for filmer produsert i eller etter 1988, og sortere resultatet synkende etter pris, så bruker jeg følgende spørring:

```
oblig1> SELECT Tittel, Sjanger, Pris
FROM Oblig1.Film
WHERE Ar >= 1988
ORDER BY Pris DESC
[2022-09-11 18:13:38] 5 rows retrieved starting from 1 in 44 ms (execution: 5 ms, fetching: 39 ms)
```

som resulterer i

	.⊞ Tittel	<b>■</b> Sjanger	∎ Pris ÷
1	Asterix hos britene	Tegnefilm	149.00
2	Salmer fra kjøkkenet	Komedie	149.00
3	Blues Brothers 2000	Komedie	135.00
4	Anastasia	Tegnefilm	123.00
5	Cinema Paradiso	Komedie	<null></null>

## e)

For å hente alle kolonner for filmer som mangler pris, sortert først etter aldersgrense og så alfabetisk etter sjanger, så bruker jeg følgende spørring:

```
oblig1> SELECT * FROM Oblig1.Film

WHERE Pris IS NULL

ORDER BY Alder, Sjanger ASC

[2022-09-11 18:26:48] 4 rows retrieved starting from 1 in 52 ms (execution: 18 ms, fetching: 34 ms)
```

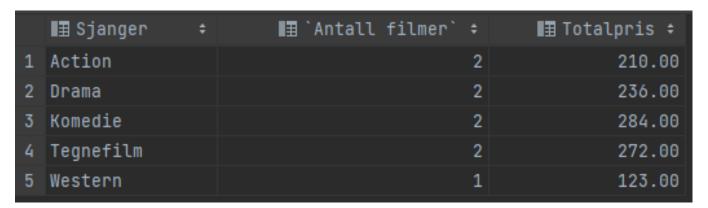


f)

Jeg finner antall filmer i hver sjanger som er til salgs, og totalpris per sjanger, ved hjelp av følgende spørring:

```
oblig1> SELECT Sjanger, COUNT(Sjanger) AS 'Antall filmer', SUM(Pris) AS 'Totalpris'
FROM Oblig1.Film
WHERE Pris IS NOT NULL
GROUP BY Sjanger
ORDER BY Sjanger ASC
[2022-09-11 18:51:56] 5 rows retrieved starting from 1 in 30 ms (execution: 5 ms, fetching: 25 ms)
```

som resulterer i



g)

Jeg legger til data for filmen Ghost in the Shell (1995)"ved følgende spørring:

```
oblig1> INSERT INTO Oblig1.Film

VALUE (14, 'Ghost in the Shell', 1995, 'Japan', 'Tegnefilm', 13, 83, 95.00)

[2022-09-11 19:02:07] 1 row affected in 19 ms
oblig1> SELECT * FROM Oblig1.Film

WHERE Tittel = 'Ghost in the Shell'

[2022-09-11 19:04:31] 1 row retrieved starting from 1 in 29 ms (execution: 4 ms, fetching: 25 ms)
```

som resulterer i

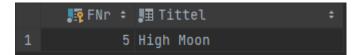
```
      Image: First properties
      Image: I
```

h)

Bruker følgende sql for å korrigere tittelen film nummer 5 fra 'High Noon' til 'High Moon':

```
oblig1> UPDATE Oblig1.Film
SET Tittel = 'High Moon'
WHERE Tittel = 'High Noon'
[2022-09-11 19:15:28] 1 row affected in 7 ms
oblig1> SELECT FNr, Tittel
FROM Oblig1.Film
WHERE Tittel = 'High Moon'
[2022-09-11 19:16:32] 1 row retrieved starting from 1 in 30 ms (execution: 5 ms, fetching: 25 ms)
```

som resulterer i



i)

Øker prisen med 10% på action-filmer med følgende kode:

```
oblig1> UPDATE Oblig1.Film
SET Pris = Pris * 1.1
WHERE Sjanger = 'Action'
[2022-09-11 19:23:18] 3 rows affected in 6 ms
oblig1> SELECT Sjanger, Tittel, Pris
FROM Oblig1.Film
WHERE Sjanger = 'Action'
ORDER BY Tittel ASC
[2022-09-11 19:27:54] 3 rows retrieved starting from 1 in 31 ms (execution: 4 ms, fetching: 27 ms)
```

som resulterer i



J

Sletter filmen 'Anastasia':

```
oblig1> DELETE
FROM Oblig1.Film
WHERE Tittel = 'Anastasia'
[2022-09-11 19:34:24] 1 row affected in 21 ms
oblig1> SELECT Tittel
FROM Oblig1.Film
ORDER BY Tittel ASC
[2022-09-11 19:35:25] 13 rows retrieved starting from 1 in 35 ms (execution: 17 ms, fetching: 18 ms)
```



### **OPPGAVE 2**

a)

Lager tabellen 'Kunde':

```
oblig1> CREATE TABLE Oblig1.Kunde (

KNr INTEGER UNSIGNED NOT NULL,
Fornavn VARCHAR(50),

Etternavn VARCHAR(50),
Adresse VARCHAR(100),
PostNr SMALLINT UNSIGNED,
PRIMARY KEY (KNr)

[2022-09-11 19:45:51] completed in 37 ms
oblig1> SELECT *
FROM Oblig1.kunde
[2022-09-11 19:47:38] O rows retrieved in 29 ms (execution: 4 ms, fetching: 25 ms)
```

som resulterer i

b)

Legger til data for tre kunder:

```
oblig1> INSERT INTO Oblig1.Kunde

VALUES (10001, 'Kari', 'Mo', 'Moldegata 21 H0402', 0445),

(10002, 'Geir', 'Gallestein', 'Grønnegata 68 H0304', 2317),

(10003, 'Reidun', 'Roterud', 'Hylleråsvegen 11', 2440)

[2022-09-11 19:56:01] 3 rows affected in 6 ms

oblig1> SELECT *

FROM Oblig1.Kunde

[2022-09-11 19:56:03] 3 rows retrieved starting from 1 in 35 ms (execution: 5 ms, fetching: 30 ms)
```

	🌇 KNr 🕏	II Fornavn ÷	I⊞ Etternavn ÷	III Adresse ÷	■ PostNr ÷
1	10001	Kari	Мо	Moldegata 21 H0402	445
2	10002	Geir	Gallestein	Grønnegata 68 H0304	2317
3	10003	Reidun	Roterud	Hylleråsvegen 11	2440

### c)

Lager en faktura-tabell med kolonner for fakturanummer, kundenummer, beløp, opprettelsesdato, forfallsdato, filmnummer og betalt. Fakturanummeret er unikt til hver faktura, og brukes derfor som primærnøkkel. Kundenummeret er en fremmednøkkel knyttet til hovednøkkelen i kunde-tabellen, slik at man vet hvilken kunde som har utført bestillingen. Filmnummeret er også en fremmednøkkel knyttet til hovednøkkelen i film-tabellen, slik at man vet hvilken film som er blitt leid. Til slutt har vi en bool i betalt som er True, hvis fakturaen er betalt og false hvis den ikke er betalt.

Ingen av nøklene kan være NULL, siden fakturaen må ha et fakturanummer, må kunne knyttes til en leietager, og må kunne knyttes til produktet som ble utleid, ellers er det ikke en gyldig faktura. Jeg lar beløp være NULL ved default, da må prisen føres inn manuelt. Jeg valgte å ha prisen som DECIMAL(7, 2), siden da kan beløp pål opptil 99.999,- brukes. Datoene skal være på formatet 'YYYY-MM-DD', f.eks. '2022.04.11', og de kan ikke være NULL, for da kan ikke transaksjonen tidfestes, og er ugyldig. Hvis en faktura ikke opprettes med utfylte datoer, så blir opprettelsesdatoen lik datoen når fakturaen ble lagt til i tabellen, og forfallsdato blir 14 dager fra opprettelsesdato.

```
oblig1> CREATE TABLE Oblig1.Faktura
2
3
                 FakturaNr
                                   INTEGER UNSIGNED NOT NULL,
                 KundeNr
                                   INTEGER UNSIGNED NOT NULL,
                 FilmNr
                                   INTEGER UNSIGNED NOT NULL,
5
6
7
                                   DECIMAL(7, 2)
                                                              DEFAULT NULL,
                 Beløp
                                                    NOT NULL DEFAULT (CURRENT_DATE),
                 Opprettelsesdato DATE
                 Forfallsdato
                                  DATE
                                                    NOT NULL DEFAULT (Opprettelsesdato + INTERVAL 14 DAY),
9
                 Betalt BOOL DEFAULT (FALSE),
10
                 PRIMARY KEY (FakturaNr),
11
                 FOREIGN KEY (KundeNr) REFERENCES Oblig1.Kunde (KNr),
                 FOREIGN KEY (FilmNr) REFERENCES Oblig1.Film (FNr)
     [2022-09-11 22:04:56] completed in 24 ms
```

d)

Jeg legger til to fakturaer for Kari Mo:

```
oblig1> INSERT INTO Oblig1.Faktura (FakturaNr, KundeNr, FilmNr, Beløp, Opprettelsesdato, Betalt)
 2
                 VALUES
 4
                      10001,
 5
                      (SELECT KNr FROM Oblig1.Kunde WHERE Fornavn = 'Kari' AND Etternavn = 'Mo'),
 6
                      14,
 7
                      (SELECT Pris FROM Oblig1.Film WHERE FNr = 14),
                      ,2021-12-30,
 8
9
                      TRUE
10
11
12
                      10002,
13
                      (SELECT KNr FROM Oblig1.Kunde WHERE Fornavn = 'Geir' AND Etternavn = 'Gallestein'),
14
15
                      (SELECT Pris FROM Oblig1.Film WHERE FNr = 11),
16
                      ,2022-04-01,
17
                      TRUE
19
20
                      13592,
21
                      (SELECT KNr FROM Oblig1.Kunde WHERE Fornavn = 'Kari' AND Etternavn = 'Mo'),
22
23
                      (SELECT Pris FROM Oblig1.Film WHERE FNr = 5),
\frac{24}{25}
                      CURRENT_DATE,
     [2022-09-11 22:13:52] 3 rows affected in 7 ms
28
    oblig1> SELECT *
29
             FROM Oblig1.Faktura
     [2022-09-11 22:13:59] 3 rows retrieved starting from 1 in 26 ms (execution: 5 ms, fetching: 21 ms)
```

som resulterer i

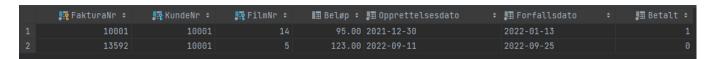
	驔 FakturaNr 🕏	<b>,</b> KundeNr ≎	<b>J≅</b> FilmNr ≎	⊞ Beløp ÷	∰ Opprettelsesdato	<b>‡</b>	,∰ Forfallsdato	÷	∰ Betalt ÷
1	10001	10001	14	95.00	2021-12-30		2022-01-13		1
2	10002	10002	11	87.00	2022-04-01		2022-04-15		1
3	13592	10001		123.00	2022-09-11		2022-09-25		0

e)

Finner Kari Mo sine fakturaer ved navn:

```
1 | oblig1> SELECT * FROM Oblig1.Faktura | WHERE KundeNr = (SELECT KNr FROM Oblig1.Kunde WHERE Fornavn = 'Kari' AND Etternavn = 'Mo') | [2022-09-11 22:18:13] 2 rows retrieved starting from 1 in 38 ms (execution: 4 ms, fetching: 34 ms)
```

som resulterer i



f)

Lagt til forretningsregel for beløpet, slik at beløpet må være i intervallet [0, 10.000]:

```
oblig1> ALTER TABLE Oblig1.Faktura

ADD CONSTRAINT Beløpsintervall

CHECK ( Beløp >= 0.00 AND Beløp <= 10000.00)

[2022-09-11 22:28:29] 3 rows affected in 55 ms
```

g)

Prøver å legge inn faktura med ulovlig beløp:

```
oblig1> INSERT INTO Oblig1.Faktura (FakturaNr, KundeNr, FilmNr, Beløp)

VALUE (1010, 1003, 1, 10000.69)

[2022-09-11 22:33:47] [HY000] [3819] Check constraint 'Beløpsintervall' is violated.

[2022-09-11 22:33:47] [HY000] [3819] Check constraint 'Beløpsintervall' is violated.

oblig1> SELECT *

FROM Oblig1.Faktura

[2022-09-11 22:34:03] 3 rows retrieved starting from 1 in 53 ms (execution: 6 ms, fetching: 47 ms)
```

Her fikk jeg en feilmelding om at forretningsregelen er brutt, og ved å printe tabellen så ser jeg at den ulovlige fakturaen ikke ble lagt til:

	驔 FakturaNr 🕏	<b>⊪</b> KundeNr ÷	<b>J≅</b> FilmNr ≎	⊪⊞ Beløp ÷	.⊞ Opprettelsesdato	÷ 🎛 Forfallsdato ÷	ৣ≣ Betalt ÷
1	10001	10001	14	95.00	2021-12-30	2022-01-13	1
2	10002	10002	11	87.00	2022-04-01	2022-04-15	1
3	13592	10001		123.00	2022-09-11	2022-09-25	0