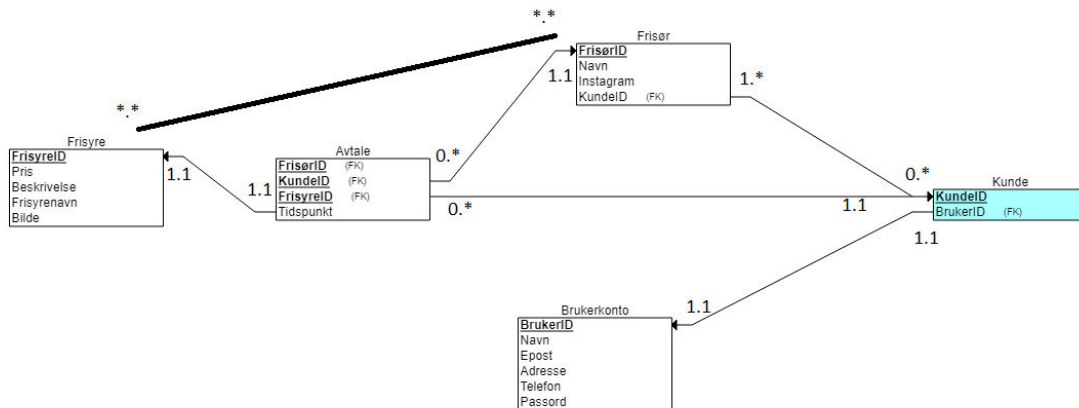


Eksamen DB1102 H2022

Oppgave 1. Modelling



Oppgave 2. SQL

a)

```
SELECT *  
FROM spiller  
ORDER BY kode;
```

Result Grid			
Filter Rows:			
	SpillerId	Dato	Kode
▶	1	1974-01-21	ABC
	2	1999-05-01	BCA
	5	2004-03-25	JES
	3	2000-04-05	XYZ
	4	2001-03-03	Z_Z
*	NULL	NULL	NULL

b)

```
SELECT *  
FROM spill  
WHERE lansert > '2020-01-01';
```

Result Grid

Filter Rows:

Edit:

Export/Import:

	SpillId	Tittel	Beskrivelse	Lansert	Kategori
▶	1	Blockbusters	En reise i filmens verden	2020-03-05	Adventure
	2	Tetrimania	Velkjente utfordringer i ukjent terreng	2020-04-06	Annet
	3	Valheimia	Tilbake til vikingetiden	2021-07-07	RPG
*	NULL	NULL	NULL	NULL	NULL

c)

```
SELECT Kategori, count(kategori) AS AntallSpill  
FROM spill  
GROUP BY kategori;
```

Result Grid	Filter Ro
Kategori	AntallSpill
Adventure	2
Annet	1
RPG	2
Puzzle	2
Action	2
Simulation	1

d)

```
SELECT spiller.spillerId, spiller.Dato, spiller.Kode,  
COUNT(resultat.spillerId) AS antResultater  
FROM Spiller  
INNER JOIN resultat ON spiller.spillerId = Resultat.SpillerId  
GROUP BY spiller.spillerId  
ORDER BY antResultater DESC  
LIMIT 1;
```

Result Grid				
	spillerId	Dato	Kode	antResultater
▶	5	2004-03-25	JES	22

e)

```
SELECT spill.tittel, spill.kategori, AVG(resultat.score) AS  
gjSnittligScore  
FROM spill  
INNER JOIN resultat ON spill.spillID = resultat.spillID  
GROUP BY spill.tittel
```

Result Grid			
	tittel	kategori	gjSnittligScore
▶	Blockbusters	Adventure	59.3333
	Tetrimania	Annet	51.8182
	Valheimia	RPG	95.0000
	Pakk meg	Puzzle	46.6667
	Metropolis	Adventure	119.3333
	Atalanta	Puzzle	36.7778
	Space Deflators	RPG	68.1667
	Galagu	Action	65.5000
	Dug Dig	Simulation	35.3333
	Masteroids	Action	103.0000

f)

```
SELECT spill.tittel, spill.kategori, spiller.kode, resultat.score AS  
maxScore, resultat.tidspunkt  
FROM spill  
INNER JOIN resultat ON spill.spillID = resultat.spillID  
INNER JOIN spiller ON resultat.spillerID = spiller.spillerID  
WHERE resultat.score = (  
SELECT MAX(SCORE)  
FROM resultat as result  
WHERE result.spillID = resultat.spillId)  
ORDER BY spill.tittel ASC;
```

Result Grid					
Filter Rows:					
Export: Wrap Cell Co					
	tittel	kategori	kode	maxScore	tidspunkt
►	Atalanta	Puzzle	XYZ	64	2021-03-12 12:13:22
	Blockbusters	Adventure	BCA	112	2021-01-05 10:15:16
	Dug Dig	Simulation	Z_Z	121	2022-05-11 14:33:19
	Galagu	Action	BCA	108	2020-02-06 11:19:16
	Masteroids	Action	JES	119	2022-08-23 11:32:55
	Metropolis	Adventure	XYZ	182	2021-02-12 12:30:18
	Pakk meg	Puzzle	Z_Z	111	2022-02-11 14:33:19
	Space Deflators	RPG	ABC	121	2019-10-24 13:23:16
	Tetrimania	Annet	JES	216	2022-02-21 09:11:12
	Valheimia	RPG	BCA	198	2021-01-06 11:19:16

g)

```
CREATE OR REPLACE VIEW resultatTavle_dugDig AS
SELECT spiller.kode, resultat.tidspunkt, resultat.score
FROM resultat
INNER JOIN spiller ON resultat.spillerID = spiller.spillerID
INNER JOIN spill ON resultat.spillID = spill.spillID
WHERE spill.tittel = 'Dug Dig'
ORDER BY resultat.score DESC;

SELECT *
FROM resultatTavle_dugDig;
```

Result Grid				Filter Rows:			
	kode	tidspunkt	score				
▶	Z_Z	2022-05-11 14:33:19	121				
	JES	2022-08-23 10:12:23	67				
	JES	2022-07-22 13:12:13	52				
	JES	2022-07-23 08:12:12	36				
	JES	2022-07-22 13:22:22	19				
	JES	2022-07-23 09:22:33	12				
	BCA	2020-03-06 15:17:46	10				
	BCA	2020-04-06 15:27:32	1				
	JES	2022-07-22 13:44:44	0				

h)

INSERT INTO spill VALUES

(11, 'Ping Pong', 'Det er som en diskusjon med ball, det går att og fram', '2022-9-8', 'Sports');

Result Grid

Filter Rows:

Edit:

Export/Import:



	SpillId	Tittel	Beskrivelse	Lansert	Kategori
▶	1	Blockbusters	En reise i filmens verden	2020-03-05	Adventure
	2	Tetrimania	Velkjente utfordringer i ukjent terreng	2020-04-06	Annet
	3	Valheimia	Tilbake til vikingetiden	2021-07-07	RPG
	4	Pakk meg	Pakk riktig og raskt!	2019-09-01	Puzzle
	5	Metropolis	Byen blir til...	2019-12-02	Adventure
	6	Atalanta	Den ville vesten og ville vester	2018-02-09	Puzzle
	7	Space Deflators	En oppdagelse i verdensrommet	2018-04-06	RPG
	8	Galagu	Skyt i vei!	2019-01-02	Action
	9	Dug Dig	Her skal det graves!	2019-02-03	Simulation
	10	Masteroids	Pang, pang i verdensrommet	2019-09-09	Action
	11	Ping Pong	Det er som en diskusjon med ball, det...	2022-09-08	Sports
✱	NULL	NULL	NULL	NULL	NULL

INSERT INTO utvikler VALUES

(9, 'billy.betong@boeljeband.no', 'Billy Betong');

Result Grid	Filter Rows:	Edit:
UtvId	KontaktInfo	Navn
1	jens@online.no	Jens Jensen
2	ida@online.no	Ida Idesen
3	kvakker@gmail.no	Karius Olsen
4	oda@hotmail.com	Oda Hansen
5	jens.petrus@prk.no	Jens Petrus
6	helene.olsen@online.no	Helene Olsen
7	pelle.parafin@boeljeband.no	Pelle Parafin
8	frida.frosk@boeljeband.no	Frida Frosk
9	billy.betong@boeljeband.no	Billy Betong
NULL	NULL	NULL

```
INSERT INTO spillutvikling VALUES
(5, 11, "Programmerer"),
(8, 11, "Designer"),
(9, 11, "Manusforfatter");
```

Result Grid   Filter Rows: <input type="text"/>			
	UtvId	SpillId	Rolle
▶	1	1	Programmerer
	1	2	Designer
	1	10	Tester
	2	3	Programmerer
	2	4	Programmerer
	2	6	Designer
	2	7	Programmerer
	3	5	Tester
	3	8	Programmerer
	3	9	Designer
	4	3	Programmerer
	4	10	Programmerer
	5	1	Programmerer
	5	2	Manusforfatter
	5	4	Tester
	5	8	Designer
	5	11	Programmerer
	6	3	Programmerer
	6	5	Designer
	6	6	Programmerer
	6	7	Manusforfatter
	6	9	Manusforfatter
	7	4	Manusforfatter
	7	8	Tester
	7	10	Annet
	8	5	Annet
	8	9	Programmerer
	8	11	Designer
	9	11	Manusforfatter
*	NULL	NULL	NULL

i)

```
SELECT utvikler.navn, count(resultat.score) AS ant_resultater  
FROM utvikler
```

```
INNER JOIN spillutvikling ON utvikler.utvID = spillutvikling.utvID
```

```
INNER JOIN resultat ON spillutvikling.spillID = resultat.spillID
```

```
GROUP BY utvikler.navn
```

```
HAVING count(resultat.score) > 30
```

```
ORDER BY ant_resultater DESC;
```

Result Grid			Filter Rows:
	navn	ant_resultater	
▶	Helene Olsen	36	
	Ida Idesen	33	
	Jens Petrus	31	

j)

```
ALTER TABLE spill
```

```
MODIFY Kategori ENUM('Adventure','Action', 'RPG',  
'Simulation', 'Sports', 'Puzzle', 'Annet', 'Shooter', 'Platformer',  
'Fighter');
```

```
UPDATE spill
```

```
SET kategori ='Shooter'
```

```
WHERE kategori = 'Action';
```

Result Grid			Filter Rows:
	tittel	kategori	
▶	Blockbusters	Adventure	
	Tetrimania	Annet	
	Valheimia	RPG	
	Pakk meg	Puzzle	
	Metropolis	Adventure	
	Atalanta	Puzzle	
	Space Deflators	RPG	
	Galagu	Shooter	
	Dug Dig	Simulation	
	Masteroids	Shooter	
	Ping Pong	Sports	

Oppgave 3 – Normalisering

a) Redundans i tabellen er at navn på kunde og tittel gjentas, selv om kunde nummer og film ID allerede står.

b) Den er ikke 2NF fordi den har sammensatt PK med filmid og kundennummer. Dette betyr at info i tabellen ikke er avhengig av PK alene. For å oppnå 2NF må hver del av en sammensatt PK ikke være redunant og UNIK.

c) For å normalisere tabellen til 2NF kan vi dele den i to. En for kundene og en for filmene. Så har vi en tabell med oversikt over hvem som har hatt hvilken film

Kundeinfo

Kundennummer PK	kundenavn
8	Morten Hanssen
9	Lene Jenssen
11	Hans Hanssen
12	Andre Jenssen

Filminfo

Filmid PK	filmnavn
15	Pretty Woman
24	Terminator 2
37	Tatt av vinden

Utleie info

Kundenummer FK	Filmid FK
8	37
9	15
11	24
12	15
12	24
12	37

Her har vi en til mange relasjon mellom kunder og film.

d) Denne er ikke 3NF fordi avdelingssted er basert på (avdelingsnummer og ansattnummer) som ikke er en PK.

e) For å gjøre den til 3NF, må vi først gjøre den til 2NF. Da gjør vi slik:

Avdelingstabell

Avdelingsnr PK	avdelingssted
2	Fredrikstad
3	Hovseter
4	Drammen

Ansatttabell

Ansattnummer PK	fornavn	etternavn	Avdelingsnr FK
3	Per	Persson	3
5	Ole	Olsen	4
8	Liv	Hanssen	3
9	Beate	Jensen	2

Denne er i 3NF også fordi ingen av informasjonen i ansattabellen er basert på blandede PK'er.