

Oblig 3_3 – SQL – Yahye Abdi

Table of Contents

Oppgave 1:.....	2
Oppgave 2:.....	3
Oppgave 3:.....	4
Oppgave 4:.....	5
Oppgave 5:.....	5
Oppgave 6:.....	7
Oppgave 7:.....	9

Oppgave 1:

Hvor mange personer har blitt vaksinert, altså mottatt minst en vaksinedose, i hver kommune?

SQL-spørring:

```
SELECT COUNT(DISTINCT pr.Personnummer) AS antall_personer, k.Navn AS kommune
FROM person AS pr INNER JOIN person_has_vaksine AS pv
ON pv.person_Personnummer = pr.Personnummer
INNER JOIN vaksine AS v
ON pv.vaksine_id = v.id
INNER JOIN post AS ps
ON ps.postNr = pr.post_PostNr
INNER JOIN kommune AS k
ON k.kommuneID = ps.kommune_kommuneID
GROUP BY Kommune
ORDER BY COUNT(DISTINCT pr.Personnummer) DESC
;
```

For å finne antall personer vaksiner minst 1 gang i hver kommune, må koble sammen **person** entitet med flere tabeller.

Koblingen skjer ved at fremmed nøkler sammenlignes i mange-til-mange forholdet som mellom disse to tabellene. For eksempel mellom tabellen **person_has_vaksine** inneholder fremmed nøkkelen til **person** entitet.

I SELECT seksjon henter vi to kolonner/attributter. Den først bruker aggregering funksjonen COUNT() som teller opp antall person er i tabellen, men med DISTINCT nøkkelord tar vi hver person kun 1 gang for å duplikat. Så vi kan finne riktig antall personer som er blitt vaksinert i hver kommune. Vi bruker personnummer siden den er unik for hver person i Norge. Andre kolonnen henter navnet på kolonnen.

Etter det, har vi 4 INNER JOIN. Først blir person tabellen koblet mellom tabellen **person_has_vaksine** ved å sammenligne fremmednøkkel og primær nøkkel. Etter det kobler vi **vaksine** tabellen til mellom tabellen person_has_vaksine på samme måte. Dette gjør at person og vaksine tabellene er koblet sammen fult.

Videre kobler vi sammen **person** og **kommune** tabellene. Det er ikke direkte forhold mellom disse to. **Person** har en-til-mange forhold til **post** og samme forhold til **kommune**. Så vi kobler først **post** til **person** og kobler **kommune** til post.

Til slutt grupperer vi resultatet med navn på kommune og sorterer synkende på antall personer vaksinert. Så kommunen med fleste personer vaksinert blir sarp med og Råde og Hvaler med minst personer vaksinert.

Result Grid			Filter Rows:
#	antall_personer	kommune	
1	3	Sarpsborg	
2	2	Fredrikstad	
3	2	Halden	
4	1	Hvaler	
5	1	Råde	

Oppgave 2:

Hvem, fullt navn, har blitt vaksinert?

SQL-spørring på personer som vaksinert:

```
# 2: Hvem, fullt navn, har blitt vaksinert?
# Here we also find how many times each person was vaxed
SELECT DISTINCT pr.Fornavn, pr.Etternavn, COUNT(pr.Persnummer) AS
antall_vaksine
FROM person AS pr
INNER JOIN person_has_vaksine AS pv
ON pv.person_Persnummer = pr.Persnummer
GROUP BY pr.Persnummer
ORDER BY antall_vaksine DESC
;
```

I denne spørringen har vi gjort litt mer enn oppgaven spurte om. Vi henter også antall ganger hver person ble vaksinert og sorterer resultatet synkende fra person som har blitt vaksinert flest ganger til den som er minst vaksinert for covid.

I SELECT clause henter vi 3 kolonner. De 2 første er fornavn og etternavn til person. Den tredje kolonnen er antall ganger hver ble vaksinert. Antall ganger vaksinert gjør med COUNT() aggregering funksjonen på samme måte som i oppgave 1. Dette betyr naturligvis at vi må gruppere resultatet på en eller annet måte. Her grupperer vi med persnummer til personene.

For å finne antall person vaksinert, kobler vi på samme måte **person** tabellen til **person_has_vaksine** mellom tabellen. Da for vi alle instansene hvor en person er vaksinert. For å unngå duplikat setter DISTINCT nøkkelord i SELECT seksjon-

I resultatet får vi 9 personer som er vaksinert i databasen. Den siste person som mangler er blitt vaksinert. Bam barabum og lala lire tok fleste vaksiner, begge ble vaksinert 3 ganger.

Result Grid				Filter Rows:
#	Fornavn	Etternavn	antall_vaksine	
1	bam	barabum	3	
2	lala	lire	3	
3	mara	kilini	2	
4	mr	anderson	2	
5	billie	bin	1	
6	kabada	kii	1	
7	mikel	juqson	1	
8	nisar	read	1	
9	tic	tac	1	

Oppgave 3:

Hvilken vaksine har flest personer mottatt, og hvor mange doser av denne er satt?

SQL-spørring:

3: Hvilken vaksine har flest personer mottatt, og hvor mange doser av denne er satt?

```
SELECT v.Navn AS vaksine, COUNT(pr.Personnummer) AS antall_doser_tatt,  
COUNT(DISTINCT pr.Personnummer) AS antall_personer ,v.Antall_doser AS  
antall_dose  
FROM person AS pr INNER JOIN person_has_vaksine AS pv  
ON pv.person_Personnummer = pr.Personnummer  
INNER JOIN vaksine AS v  
ON pv.vaksine_id = v.id  
GROUP BY pv.vaksine_id  
ORDER BY COUNT(pr.Personnummer) DESC  
LIMIT 1  
;
```

For å finne hvor mange person har blitt vaksinert og hva de heter, trenger vi å koble **person** entitet som inneholder navnene til personene til **person_has_vaksine** has inneholder fremmed nøkkelen til personer som har tatt en vaksine. Vi kobler med inner join **vaksine** entitet. Dette er mange-til-mange forhold. Etter

I SELECT seksjon henter 4 kolonner. Navn på vaksine, antall doser som personer tok, antall personer som tok vaksine type og til slutt antall doser man trenger å ta av vaksinen for å være godt beskyttet. Kolonne number 2 teller opp antall ganger person tok en vaksine ved å telle alle radene i mellom tabellen. Alle person som er knyttet til mellom tabellen person_has_vaksine er de som er vaksinert.

Til slutt grupperer vi med vaksine id som unik for hver vaksine. Vi sorterer synkende sånn at vaksine er brukt mest kommer øverst. LIMIT 1 sikrer at vi får ut kun den vaksinen med høyest antall doser tatt.

I resultat ser at Pfizer er vaksinen som fleste i databasen tok. Den ble totalt tatt 7 doser av. Det var 4 personer som tok 7 dosene mellom. Pfizer trenger man 4 doser for bli godt beskyttet.

Result Grid  Filter Rows:  Export:  Wrap Cell Content:				
#	vaksine	antall_doser_tatt	antall_personer	antall_dose
1	pfizer	7	4	4

Oppgave 4:

Hvilke kommuner har vi en oversikt over?

SQL-spørring som viser alle kommunene i databasen:

```
SELECT *  
FROM kommune;
```

Vi har oversikt over 12 kommuner i denne databasen.

#	kommuneID	Navn
1	3001	Halden
2	3002	Moss
3	3003	Sarpsborg
4	3004	Fredrikstad
5	3011	Hvaler
6	3012	Aremark
7	3013	Marker
8	3014	Indre Østfold
9	3015	Skiptvedt
10	3016	Rakkestad
11	3017	Råde
12	3018	Våler
*	NULL	NULL

Oppgave 5:

Hvem har fått vaksine av den typen det er gitt flest doser av?

SQL-spørring som personene som tok vaksinen det er tatt mest av

5: View which returns only the vaxx which is used most. Here it is pfizer which has 7 doses used

```
CREATE VIEW mest_tatt_vaksine AS  
SELECT v.Navn, COUNT(pv.vaksine_id) AS antall_doser  
FROM person AS pr INNER JOIN person_has_vaksine AS pv  
ON pv.person_Personnummer = pr.Personnummer  
INNER JOIN vaksine AS v  
ON pv.vaksine_id = v.id  
GROUP BY v.Navn  
ORDER BY antall_doser DESC  
LIMIT 1  
;
```

#query to show result of view

```
SELECT *  
FROM mest_tatt_vaksine;
```

5: Hvem har fått vaksine av den typen det er gitt flest doser av?

```
SELECT DISTINCT pr.Fornavn, pr.Etternavn, v.Navn AS vaksine  
FROM person AS pr INNER JOIN person_has_vaksine AS pv
```

```

ON pv.person_Personnummer = pr.Personnummer
INNER JOIN vaksine AS v
ON pv.vaksine_id = v.id
WHERE v.Navn = (SELECT Navn FROM mest_tatt_vaksine)
;

```




Vi bruker view i denne oppgaven. Fra oppgave 3, vet vi allerede hvilke vaksine som har blitt brukt mest. Så vi setter spørringen i oppgave 3 i en view. I view er spørringen litt modifisert. Den henter kun 2 attributter, navn på vaksine mest brukt og antall doser ble tatt av den. Den grupperer på vaksine navn. View blir subquery/delspørring.

Hovedspørringen gjør likt som vi har gjort før. **Person** entitet blir koblet sammen til **vaksine** entitet med inner join ved å bruke mellom tabellen **person_has_vaksine**.

I SELECT seksjon henter vi 3 attributter, fornavn til person, etternavn til person og navn på vaksine.

Spørringen henter alle personer som er vaksinert, men i WHERE seksjonen personer der vaksine de har tatt er lik den vaksinen som view tabellen henter. Det er Pfizer vaksine ble tatt mest. På denne måten kan vi også de personene som tok vaksinen som ble brukt mest.

I resultatet ser vi de personene som 4 dosene av Pfizer vaksine.

Result Grid   Filter Rows: 			
#	Fornavn	Etternavn	vaksine
1	bam	barabum	pfizer
2	billie	bin	pfizer
3	lala	lire	pfizer
4	mara	kilini	pfizer

Oppgave 6:

Hvilke personer har mottatt to eller flere doser, og når ble disse dosene satt?

SQL-spørring som henter personer som tok 2 eller mer doser av vaksine:

```
# 6: view which gets persons who are vaccinated atleast 2 or more times
CREATE VIEW vaxxed_2_or_more_times AS
SELECT pv.person_Personnummer
FROM person_has_vaksine AS pv
GROUP BY pv.person_Personnummer
HAVING COUNT(*) >= 2;
;
```

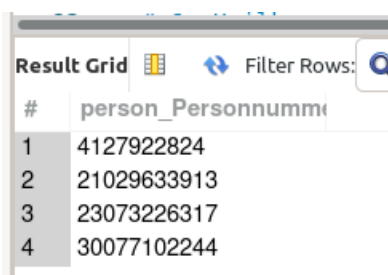
```
#personnummer of people who are vaxxed 2 or more times
SELECT *
FROM vaxxed_2_or_more_times;
```

```
# 6: Hvilke personer har mottatt to eller flere doser, og når ble disse dosene
satt?
# here we do normal inner join of person to person_has_vaksine and vaksine table
# then in WHERE clause we get record where personnummer is equal to personnummer
from the ones in the view
# the personnummers returned from view are ones who have been vaccinated 2 or
more times
SELECT pr.Fornavn, pr.Etternavn, v.Navn AS vaksine_navn, pv.vaksine_tid AS
vaksinert_dato
FROM person AS pr
INNER JOIN person_has_vaksine AS pv
ON pv.person_Personnummer = pr.Personnummer INNER JOIN vaksine AS v
ON pv.vaksine_id = v.id
WHERE pr.Personnummer IN (SELECT person_Personnummer FROM
vaxxed_2_or_more_times)
ORDER BY pr.Fornavn
;
```

Vi bruker også view i denne oppgaven, siden vi trenger delspørring. I view delspørringen henter vi kun 1 kolonne i SELECT seksjonen som er fremmed nøkkelen til person i person_has_vaksine tabell.

Vi kobler **person** entitet med **person_has_vaksine** med inner join og teller vi opp hele spørringen med COUNT() aggregering funksjon som henter personene som dukker opp 2 eller mer ganger spørringen. Vi grupperer på fremmed nøkkelen av **person** i **person_has_vaksine** tabell.

Resultat i view viser personnummer til de personene som tok 2 eller mer vaksine doser.



#	person_Personnummer
1	4127922824
2	21029633913
3	23073226317
4	30077102244

I hoved spørringen henter kobler vi sammen **person** og **vaksine** entitet via mellom tabellen **person_has_vaksine** med inner join.

I SELECT seksjon henter vi 4 attributter. Fornavn til person, etternavn til person, navn på vaksine og dato på når person tok vaksinen.

Vi kobler sammen igjen person og vaksine via mellom tabellen **person_has_vaksine** med inner join. Dette vil gi alle personene som har tatt vaksine.

I WHERE seksjonen begrenser vi antall personer som blir hentet til kun de personene som har tatt 2 doser eller mer. Dette vi gjør med å si hent de personer med personnummer som er lik personnummer fra view delspørringen.

I resultat ser vi de 4 personene som tok 2 eller mer doser i databasen. Resultatet er sortert alfabetisk. Bam Barabum har tatt mest doser med 4 og resten tok 2 doser hver.

Result Grid				
Filter Rows:				
#	Fornavn	Etternavn	vaksine_navn	vaksinert_dato
1	bam	barabum	pfizer	2022-10-10
2	bam	barabum	pfizer	2021-01-12
3	bam	barabum	pfizer	2020-08-22
4	lala	lire	pfizer	2021-11-05
5	lala	lire	Moderna	2022-06-14
6	lala	lire	Moderna	2021-01-23
7	mara	kilini	pfizer	2021-01-19
8	mara	kilini	pfizer	2020-05-11
9	mr	anderson	Moderna	2021-12-28
10	mr	anderson	Moderna	2021-03-14

Oppgave 7:

Hvor mange personer i hver risikogruppe er fullvaksinert, dvs. har mottatt minst to doser?

SQL-spørring til å finne person full vaksinert i hver gruppe:

```
# 7: view which gets persons who are vaccinated atleast 2 or more times
#This is copy from view in oppgave 6
CREATE VIEW vaxxed_2_or_more_times AS
SELECT pv.person_Personnummer
FROM person_has_vaksine AS pv
GROUP BY pv.person_Personnummer
HAVING COUNT(*) >= 2;
;
#Main query
SELECT COUNT(pr.Personnummer) AS antall_personer, Beskrivelse AS risiko_gruppe
FROM person AS pr INNER JOIN person_has_risikogruppe AS phr
ON phr.person_Personnummer = pr.Personnummer
INNER JOIN risikogruppe AS r
ON phr.risikogruppe_RisikogruppeID = r.RisikogruppeID
WHERE pr.Personnummer IN (SELECT person_Personnummer FROM
vaxxed_2_or_more_times)
GROUP BY r.RisikogruppeID
;
```

I denne oppgaven bruker vi samme view vi hadde i oppgave 6 siden den gir oss de personene har tatt 2 eller mer doser vaksine. Disse personene blir full vaksinert mennesker.

I Hoved spørring kobler med person entitet til risikogruppe entitet via mellom tabellen person_has_risikogruppe.

I WHERE seksjon henter vi ut resultat der personnummer til person er som personnummer i view vaxxed_2_or_more_times.

Som i oppgave 6, har vi 4 personer har tatt 2 eller mer doser. De tilhører disse risikogruppene. Ingen av er i samme risikogruppe. Jeg la til en person som er sykehjem risikogruppe bare for å se om spørringen fungerte riktig siden alle andre kategori har kun 1 person som full vaksinert.

Legg ny person i person tabellen:

```
INSERT INTO
person(Fornavn, Etternavn, personnummer, Telefonnummer, Adresse, post_postNr)
VALUES
("final", "person", 21099212791, 4799003340, "lol street", 1602)
;
```

Registrer vaksine og når personen tok vaksine:

```
INSERT INTO person_has_vaksine(person_Personnummer, vaksine_id, vaksine_tid)
VALUES
(21099212791, 3, "2021-05-27"),
(21099212791, 3, "2021-05-28"),
(21099212791, 3, "2021-05-29"),
```

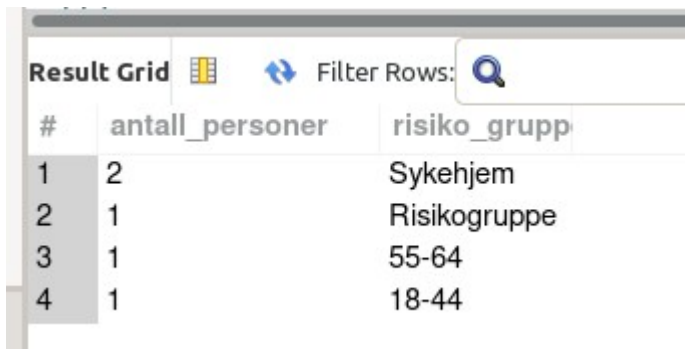
```
(21099212791, 3, "2021-05-30")  
;
```

Legg person i en risikogruppe. Her blir person lagt i sykehjem risikogruppe:

```
INSERT INTO  
person_has_risikogruppe(person_Personnummer, risikogruppe_RisikogruppeID)  
VALUES  
(21099212791, 1)  
;
```

Kjør hovedspørring på nytt som viser at sykehjem risikogruppe har 2 full vaksinert personer:

```
#Main query  
SELECT COUNT(pr.Personnummer) AS antall_personer, Beskrivelse AS risiko_gruppe  
FROM person AS pr INNER JOIN person_has_risikogruppe AS phr  
ON phr.person_Personnummer = pr.Personnummer  
INNER JOIN risikogruppe AS r  
ON phr.risikogruppe_RisikogruppeID = r.RisikogruppeID  
WHERE pr.Personnummer IN (SELECT person_Personnummer FROM  
vaxxed_2_or_more_times)  
GROUP BY r.RisikogruppeID  
;
```



The screenshot shows a database interface with a 'Result Grid' tab. The grid contains 4 rows of data. The first row is highlighted. The columns are labeled '#', 'antall_personer', and 'risiko_grupp'. The data is as follows:

#	antall_personer	risiko_grupp
1	2	Sykehjem
2	1	Risikogruppe
3	1	55-64
4	1	18-44