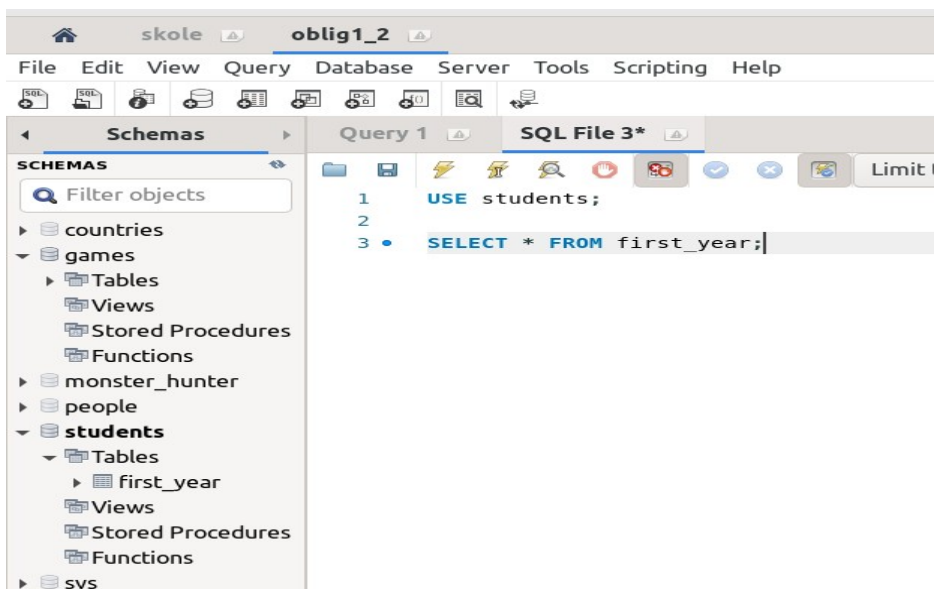


# Oblig1\_3 – Yahye Abdi Ahmed

## Table of Contents

SQL utvalgsspøringer for å hente ut data fra én tabell.....	2
Velg ut rader og kolonner:.....	3
Kolonner spørring:.....	3
Rad spørringer:.....	4
Sortere radene med hensyn på bestemte kolonner:.....	5
Alfabetisk rekkefølge sortering:.....	5
Sortering med nummer.....	5
Begrense nummer med WHERE nøkkelord.....	6
Aggregere data (beregne sum, gjennomsnitt og lignende).....	6
Nummer datatype.....	7
Sum() - sum.....	7
AVG() - average/gjennomsnitt.....	7
MAX() og MIN().....	7
String datatype.....	8
UPPER() og LOWER().....	8
CONCAT().....	8
Funksjoner og operatorer.....	9
AND.....	9
OR.....	9
NOT.....	10

## SQL utvalgsspørringer for å hente ut data fra én tabell



Jeg bruker samme tabell fra oblig1\_2. For å hente ut data fra tabellen, kjører jeg først sql query `USE students;`. Det er for å velge hvilke schema/database jeg ønsker å jobbe videre med. Dette kan også gjøres å bare dobbelklikke students.

`SELECT * FROM first_year;` er query som henter alt data ligger tabellen first\_year. \* er wildcard som velger ut alt. Alternativt er det fult mulig å velge bestemte kolonner fra tabellen.

Sånn ser alt data tabellen first\_year ut:

Result Grid					Filter Rows:
#	id	first_name	last_name	age	
1	1	yahye	abdi	28	
2	2	mostafa	ali	22	
3	3	astrid	johan	25	
4	4	peter	kiir	19	
5	5	kahn	ozi	18	
6	6	anne	marie	22	
7	7	jon	arne	32	
8	8	martin	ger	26	
9	9	seline	sami	21	
10	10	arild	wilhaven	30	
*	NULL	NULL	NULL	NULL	

## Velg ut rader og kolonner:

Isteden for å velge ut alt data med wildcard \*, så er det mulig å velge ut bestemte kolonner og rader.

## Kolonner spørring:

5 #velge ut id kolonnen fra tabellen  
 6 • **SELECT** id **FROM** first\_year;

#velg ut first\_name fra tabellen first\_year  
**SELECT** first\_name **FROM** first\_year;

Result Grid		Filter Rows:
#	id	
1	1	
2	2	
3	3	
4	4	
5	5	
6	6	
7	7	
8	8	
9	9	
10	10	
*	NULL	



Result Grid		Filter Rows:
#	first_name	
1	yahye	
2	mostafa	
3	astrid	
4	peter	
5	kahn	
6	anne	
7	jon	
8	martin	
9	seline	
10	arild	

11 #velg ut last\_name fra tabellen  
 12 • **SELECT** last\_name **FROM** first\_year;



```

14 #velg ut age kolonnen fra tabellen
15 • SELECT age FROM first_year;
16

```

Result Grid   Filter

#	last_name
1	abdi
2	ali
3	johan
4	kiir
5	ozi
6	marie
7	arne
8	ger
9	sami
10	wilhaven

Result Grid   Filter f

#	age
1	28
2	22
3	25
4	19
5	18
6	22
7	32
8	26
9	21
10	30

## Rad spørringer:


Bestemt kolonne kan hentes med for eksempel WHERE nøkkelord. Her henter alt med \* og fra tabellen first\_year med FROM nøkkelord, tilslutt bestemmer WHERE ved å se etter first\_name kolonne med navnet yahye. Nå viser resultatet bare 1 rekord, men om det andre rekord med like first\_name, så vil spørrigen alle som oppfyller lik krav. Vi kan legge til mer begrensning ved å se last\_name og til og med age kolonne om vi ønsker akkurat 1 rekord fra tabellen.

I disse 2 spørringene ser vi en med en rekord siden det finnes bare som har first\_name yahye. På den andre er 2 i tabellen er 22 år gamle.

```

18 #velg ut alt i en rad
19 • SELECT *
20 FROM first_year
21 WHERE first_name = "yahye";

```




Result Grid   Filter Rows:  Edit

#	id	first_name	last_name	age
1	1	yahye	abdi	28
*	NULL	NULL	NULL	NULL

```

22
23 #hent alle som er 22 år gamle
24 • SELECT *
25 FROM first_year
26 WHERE age = 22;

```

Result Grid   Filter Rows:  Edit

#	id	first_name	last_name	age
1	2	mostafa	ali	22
2	6	anne	marie	22
*	NULL	NULL	NULL	NULL

# Sortere radene med hensyn på bestemte kolonner:

## Alfabetisk rekkefølge sortering:

Her er listen sortert alfabetisk, default at den sorterer i stigende rekkefølge. Det er mulig å gjøre motsatt legge nøkkelordet DESC i ORDER BY. Her under er 2 bilder. En stigende og synkende rekkefølge.

```
28 #alfabetisk rekkefølge
29 • SELECT *
30 FROM first_year
31 ORDER BY first_name;
```

#	id	first_name	last_name	age
1	6	anne	marie	22
2	10	arild	wilhaven	30
3	3	astrid	johan	25
4	7	jon	arne	32
5	5	kahn	ozi	18
6	8	martin	ger	26
7	2	mostafa	ali	22
8	4	peter	kiir	19
9	9	seline	sami	21
10	1	yahye	abdi	28
*				

first\_year 12

```
28 #alfabetisk rekkefølge
29 • SELECT *
30 FROM first_year
31 ORDER BY first_name DESC;
```

#	id	first_name	last_name	age
1	1	yahye	abdi	28
2	9	seline	sami	21
3	4	peter	kiir	19
4	2	mostafa	ali	22
5	8	martin	ger	26
6	5	kahn	ozi	18
7	7	jon	arne	32
8	3	astrid	johan	25
9	10	arild	wilhaven	30
10	6	anne	marie	22
*				

## Sortering med nummer

Vi kan også sortere med nummer. Her med alder. Den sorterer stigende(ASC), som også default eller descending(DESC) synkende.

```
33 #sorter med nummer. Her med alder
34 • SELECT *
35 FROM first_year
36 ORDER BY age ASC;
```

#	id	first_name	last_name	age
1	5	kahn	ozi	18
2	4	peter	kiir	19
3	9	seline	sami	21
4	2	mostafa	ali	22
5	6	anne	marie	22
6	3	astrid	johan	25
7	8	martin	ger	26
8	1	yahye	abdi	28
9	10	arild	wilhaven	30
10	7	jon	arne	32
*				

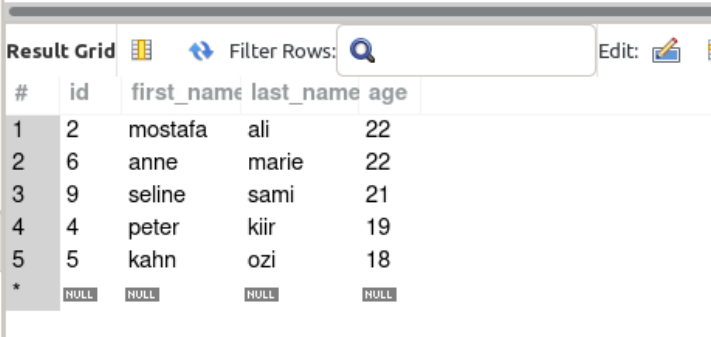
```
37
38 #sorter med nummer og begrensn på alder. stigende
39 • SELECT *
40 FROM first_year
41 WHERE age < 22
42 ORDER BY age DESC;
```

#	id	first_name	last_name	age
1	7	jon	arne	32
2	10	arild	wilhaven	30
3	1	yahye	abdi	28
4	8	martin	ger	26
5	3	astrid	johan	25
6	2	mostafa	ali	22
7	6	anne	marie	22
8	9	seline	sami	21
9	4	peter	kiir	19
10	5	kahn	ozi	18
*				

## Begrense nummer med WHERE nøkkelord

Vi kan legge mer betingelser i sorteringen. Her legger begrens vi antall i resultat til kun personer som er på alder 22 eller mindre med WHERE.

```
43  #sorter med nummer og begrens på alder.
44  • SELECT *
45  FROM first_year
46  WHERE age <= 22
47  ORDER BY age DESC;
48
49
```



Result Grid

#	id	first_name	last_name	age
1	2	mostafa	ali	22
2	6	anne	marie	22
3	9	seline	sami	21
4	4	peter	kiir	19
5	5	kahn	ozi	18
*	NULL	NULL	NULL	NULL

## Aggregere data (beregne sum, gjennomsnitt og lignende)

I MySQL kan vi bruke mange innbygde funksjoner. Det finnes funksjoner å manipulere String datatype, funksjoner for nummer datatype, funksjoner for dato datatype og funksjoner for mer avansert funksjonalitet. Vi skal se på noe funksjoner for å manipulere nummer og String datatype.

For nummer har vi en funksjon til å regne ut sum av alle tall i kolonne. For eksempel i Schema first\_year, kan vi regne ut sum av alle alder med funksjonen SUM().

Vi kan regne ut gjennomsnittsalder med AVG() funksjon, average function.

Størst alder og mist alder kan vi finne med funksjonene MAX() og MIN()

For String har vi funksjoner blant annet UPPER() for å gjøre tekst til store bokstaver, LOWER() for små bokstaver. CONCAT() til legge sammen tekst etter hverandre.

## Nummer datatype

Det som er spesielt med funksjoner er bruken av parentes(). Dette ligner på måten funksjoner fungerer i typisk programmeringsspråk som python, javascript, java etc.

## Sum() - sum

Denne funksjonen legger sammen alle tall i alder kolonnen.

```
49 #sum av alle alder
50 • SELECT SUM(age)
51 FROM first_year;
```

Result Grid  Filter Rows: 

#	SUM(age)
1	243

## AVG() - average/gjennomsnitt

AVG funksjonen finner gjennomsnittalderen på de 10 personene i first\_year database.

Gjennomsnittalderen for denne gruppen er 24.3 år.

```
53 #AVG() gjennomsnitt alder
54 • SELECT AVG(age)
55 FROM first_year;
```

Result Grid  Filter Rows: 

#	AVG(age)
1	24.3000

```
56
57 #MAX() finn største nummer
58 • SELECT MAX(age)
59 FROM first_year;
```

Result Grid  Filter Rows: 

#	MAX(age)
1	32

```
61 #finn minste nummer
62 • SELECT MIN(age)
63 FROM first_year;
```

Result Grid  Filter Rows: 

#	MIN(age)
1	18

og M

## String datatype

Funksjonene for String datatype har parentes vi kan legge kolonnene vi ønsker å manipulere og funksjonene returnerer noe tilbake.

### UPPER() og LOWER()

UPPER() Gjør tekst i kolonnen til storebokstaver og LOWER() til små bokstaver. I javascript, java har funksjonnavnet toUpperCase() og toLowerCase()

```
64
65 #UPPER() storbokstav
66 • SELECT UPPER(first_name)
67 FROM first_year;
```

#	UPPER(first_name)
1	YAHYE
2	MOSTAFA
3	ASTRID
4	PETER
5	KAHN
6	ANNE
7	JON
8	MARTIN
9	SELINE
10	ARILD

```
69 #LOWER() små bokstav
70 • SELECT LOWER(first_name)
71 FROM first_year;
```

#	LOWER(first_name)
1	yahye
2	mostafa
3	astrid
4	peter
5	kahn
6	anne
7	jon
8	martin
9	seline
10	arild

### CONCAT()

Denne funksjonen legger sammen innhold i valgt kolonner. Schema first\_year har to kolonner med String datatype, first\_name og last\_name. Vi kan legge dem sammen i en kolonne som full\_name med alias. I funksjon parentesen kan vi legge tom streng med mellom akkurat som i python, java etc. CONCAT(first\_name, " ", last\_name).

```
73 #CONCAT() legge sammen kolonner
74 • SELECT CONCAT(first_name, " ", last_name) AS full_name
75 FROM first_year;
```

#	full_name
1	yahye abdi
2	mostafa ali
3	astrid johan
4	peter kiir
5	kahn ozi
6	anne marie
7	jon arne
8	martin ger
9	seline sami
10	arild wilhaven



# Funksjoner og operatører

Funksjoner har vi sett litt på allerede i avsnittet over, men operatører har vi ikke brukt ennå. Når vi snakker om operatører i MySQL, så handler det om AND, OR, NOT operatører. Matematisk operator som +, -, \*, / og % er også operatører. >, <, =, <=, >=, <>, større, mindre, lik og mindre, like og større, ulike er også logiske operatører vi kan bruke til å gi betingelser til MySQL spørringene.

## AND

Det er en av operatørene til å lage 2 sammensatt betingelser. Begge MÅ være sanne eller stemme. Eksempelet her ser at vi at først må first\_name starte med bokstaven "a" og den andre betingelsen er at alderen må være lik 25 eller mindre. Resultat er 2 personer av 10 møter disse 2 betingelsene. Astrid og Anne.

```
77 #AND operator. finn person hvor navn start på 'a' og alder er 25 eller mindre
78 • SELECT *
79 FROM first_year
80 WHERE first_name LIKE "a%" AND age <= 25;
```

Result Grid					Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
#	id	first_name	last_name	age				
1	3	astrid	johan	25				
2	6	anne	marie	22				
*								

## OR

OR er også operator for sjekke 2 betingelser, men begge må IKKE være sanne for at spørringen skal resultat. Hvis 1 av betingelsene er sann, så får vi resultat på den. Om begge er sanne, så vi enda mer resultat.

Når vi kjører samme spørring som den i AND, men bare bytter ut AND med OR, får Astrid og Anne som starter first\_navnet sitt 'a' og alle andre personer som på alder 25 eller mindre. I tillegg får personen Arild som er større enn 25 år, fordi navnet fornavnet hans begynner med "a".







```
82 #OR operator. finn alle som har first_name som starter med 'a' eller på alder 25 eller mindre.
83 • SELECT *
84 FROM first_year
85 WHERE first_name LIKE "a%" OR age <= 25;
```

Result Grid					Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
#	id	first_name	last_name	age				
1	2	mostafa	ali	22				
2	3	astrid	johan	25				
3	4	peter	kiir	19				
4	5	kahn	ozi	18				
5	6	anne	marie	22				
6	9	seline	sami	21				
7	10	arild	wilhaven	30				
*								

# NOT

Med NOT operator, kan vi velge ut data vi ikke ønsker. Her blir alle med first\_name som ikke begynner på 'a' og som har er på alder mindre eller lik 25 år hentet.

```
87 #NOT.  
88 • SELECT *  
89 FROM first_year  
90 WHERE first_name NOT LIKE "a%" AND age <= 25;
```

Result Grid   Filter Rows:  Edit:   

#	id	first_name	last_name	age
1	2	mostafa	ali	22
2	4	peter	kiir	19
3	5	kahn	ozi	18
4	9	seline	sami	21
*	NULL	NULL	NULL	NULL