

IN2090 – Databaser og datamodellering

05 – WHERE-klausulen

Leif Harald Karlsen
leifhka@ifi.uio.no



Universitetet i Oslo

Velge over et intervall av verdier

Spørring som finner navnet på alle kunder som har kjøpt mer enn 10 produkter

```
SELECT Name  
FROM Customer  
WHERE NrProducts > 10
```

Resultat

Velge over et intervall av verdier

Spørring som finner navnet på alle kunder som har kjøpt mer enn 10 produkter

```
SELECT Name
FROM Customer
WHERE NrProducts > 10
```

Resultat

CustomerID	Name	Birthdate	NrProducts
0	Anna Consuma	1978-10-09	19
1	Peter Young	2009-03-01	1
2	Carla Smith	1986-06-14	8
3	Sam Penny	1961-01-09	14
4	John Mill	1989-11-16	8
5	Yvonne Potter	1971-04-12	6

Velge over et intervall av verdier

Spørring som finner navnet på alle kunder som har kjøpt mer enn 10 produkter

```
SELECT Name
FROM Customer
WHERE NrProducts > 10
```

Resultat

CustomerID	Name	Birthdate	NrProducts
0	Anna Consuma	1978-10-09	19
1	Peter Young	2009-03-01	1
2	Carla Smith	1986-06-14	8
3	Sam Penny	1961-01-09	14
4	John Mill	1989-11-16	8
5	Yvonne Potter	1971-04-12	6

Velge over et intervall av verdier

Spørring som finner navnet på alle kunder som har kjøpt mer enn 10 produkter

```
SELECT Name
FROM Customer
WHERE NrProducts > 10
```

Resultat

CustomerID	Name	Birthdate	NrProducts
0	Anna Consuma	1978-10-09	19
1	Peter Young	2009-03-01	1
2	Carla Smith	1986-06-14	8
3	Sam Penny	1961-01-09	14
4	John Mill	1989-11-16	8
5	Yvonne Potter	1971-04-12	6

Velge over et intervall av verdier

Spørring som finner navnet på alle kunder som har kjøpt mer enn 10 produkter

```
SELECT Name  
FROM Customer  
WHERE NrProducts > 10
```

Resultat

Name
Anna Consuma
Sam Penny

Kombinere betingelser

Spørring som finner fødselsdatoen og navnet til kunder som kjøpte mellom 4 og 10 produkter

```
SELECT Birthdate, Name
FROM Customer
WHERE NrProducts > 4 AND
      NrProducts < 10
```

Resultat

Kombinere betingelser

Spørring som finner fødselsdatoen og navnet til kunder som kjøpte mellom 4 og 10 produkter

```
SELECT Birthdate, Name
FROM Customer
WHERE NrProducts > 4 AND
      NrProducts < 10
```

Resultat

CustomerID	Name	Birthdate	NrProducts
0	Anna Consuma	1978-10-09	19
1	Peter Young	2009-03-01	1
2	Carla Smith	1986-06-14	8
3	Sam Penny	1961-01-09	14
4	John Mill	1989-11-16	8
5	Yvonne Potter	1971-04-12	6

Kombinere betingelser

Spørring som finner fødselsdatoen og navnet til kunder som kjøpte mellom 4 og 10 produkter

```
SELECT Birthdate, Name
FROM Customer
WHERE NrProducts > 4 AND
      NrProducts < 10
```

Resultat

CustomerID	Name	Birthdate	NrProducts
0	Anna Consuma	1978-10-09	19
1	Peter Young	2009-03-01	1
2	Carla Smith	1986-06-14	8
3	Sam Penny	1961-01-09	14
4	John Mill	1989-11-16	8
5	Yvonne Potter	1971-04-12	6

Kombinere betingelser

Spørring som finner fødselsdatoen og navnet til kunder som kjøpte mellom 4 og 10 produkter

```
SELECT Birthdate, Name
FROM Customer
WHERE NrProducts > 4 AND
      NrProducts < 10
```

Resultat

CustomerID	Name	Birthdate	NrProducts
0	Anna Consuma	1978-10-09	19
1	Peter Young	2009-03-01	1
2	Carla Smith	1986-06-14	8
3	Sam Penny	1961-01-09	14
4	John Mill	1989-11-16	8
5	Yvonne Potter	1971-04-12	6

Kombinere betingelser

Spørring som finner fødselsdatoen og navnet til kunder som kjøpte mellom 4 og 10 produkter

```
SELECT Birthdate, Name
FROM Customer
WHERE NrProducts > 4 AND
      NrProducts < 10
```

Resultat

Birthdate	Name
1986-06-14	Carla Smith
1989-11-16	John Mill
1971-04-12	Yvonne Potter

Kombinere betingelser med OR

Spørring som finner navnet til kunder som har kjøpt færre enn 5 produkter eller fler enn 15 produkter

```
SELECT Name
  FROM Customer
 WHERE NrProducts < 5 OR
       NrProducts > 15
```

Resultat

Kombinere betingelser med OR

Spørring som finner navnet til kunder som har kjøpt færre enn 5 produkter eller fler enn 15 produkter

```
SELECT Name
FROM Customer
WHERE NrProducts < 5 OR
      NrProducts > 15
```

Resultat

CustomerID	Name	Birthdate	NrProducts
0	Anna Consuma	1978-10-09	19
1	Peter Young	2009-03-01	1
2	Carla Smith	1986-06-14	8
3	Sam Penny	1961-01-09	14
4	John Mill	1989-11-16	8
5	Yvonne Potter	1971-04-12	6

Kombinere betingelser med OR

Spørring som finner navnet til kunder som har kjøpt færre enn 5 produkter eller fler enn 15 produkter

```
SELECT Name
FROM Customer
WHERE NrProducts < 5 OR
      NrProducts > 15
```

Resultat

CustomerID	Name	Birthdate	NrProducts
0	Anna Consuma	1978-10-09	19
1	Peter Young	2009-03-01	1
2	Carla Smith	1986-06-14	8
3	Sam Penny	1961-01-09	14
4	John Mill	1989-11-16	8
5	Yvonne Potter	1971-04-12	6

Kombinere betingelser med OR

Spørring som finner navnet til kunder som har kjøpt færre enn 5 produkter eller fler enn 15 produkter

```
SELECT Name
FROM Customer
WHERE NrProducts < 5 OR
      NrProducts > 15
```

Resultat

CustomerID	Name	Birthdate	NrProducts
0	Anna Consuma	1978-10-09	19
1	Peter Young	2009-03-01	1
2	Carla Smith	1986-06-14	8
3	Sam Penny	1961-01-09	14
4	John Mill	1989-11-16	8
5	Yvonne Potter	1971-04-12	6

Kombinere betingelser med OR

Spørring som finner navnet til kunder som har kjøpt færre enn 5 produkter eller fler enn 15 produkter

```
SELECT Name
FROM Customer
WHERE NrProducts < 5 OR
      NrProducts > 15
```

Resultat

Name
Anna Consuma
Peter Young

Bruke både AND og OR

Spørring som finner navn på kunder som har kjøpt mindre enn 5 eller mer enn 15 produkter og er født etter '2000-01-01'

```
SELECT Name FROM Customer
WHERE (NrProducts < 5 OR
       NrProducts > 15) AND
       Birthdate > '2000-01-01'
```

Resultat

Bruke både AND og OR

Spørring som finner navn på kunder som har kjøpt mindre enn 5 eller mer enn 15 produkter og er født etter '2000-01-01'

```
SELECT Name FROM Customer
WHERE (NrProducts < 5 OR
      NrProducts > 15) AND
      Birthdate > '2000-01-01'
```

Resultat

CustomerID	Name	Birthdate	NrProducts
0	Anna Consuma	1978-10-09	19
1	Peter Young	2009-03-01	1
2	Carla Smith	1986-06-14	8
3	Sam Penny	1961-01-09	14
4	John Mill	1989-11-16	8
5	Yvonne Potter	1971-04-12	6

Bruke både AND og OR

Spørring som finner navn på kunder som har kjøpt mindre enn 5 eller mer enn 15 produkter og er født etter '2000-01-01'

```
SELECT Name FROM Customer
WHERE (NrProducts < 5 OR
      NrProducts > 15) AND
      Birthdate > '2000-01-01'
```

Resultat

CustomerID	Name	Birthdate	NrProducts
0	Anna Consuma	1978-10-09	19
1	Peter Young	2009-03-01	1
2	Carla Smith	1986-06-14	8
3	Sam Penny	1961-01-09	14
4	John Mill	1989-11-16	8
5	Yvonne Potter	1971-04-12	6

Bruke både AND og OR

Spørring som finner navn på kunder som har kjøpt mindre enn 5 eller mer enn 15 produkter og er født etter '2000-01-01'

```
SELECT Name FROM Customer
WHERE (NrProducts < 5 OR
      NrProducts > 15) AND
      Birthdate > '2000-01-01'
```

Resultat

CustomerID	Name	Birthdate	NrProducts
0	Anna Consuma	1978-10-09	19
1	Peter Young	2009-03-01	1
2	Carla Smith	1986-06-14	8
3	Sam Penny	1961-01-09	14
4	John Mill	1989-11-16	8
5	Yvonne Potter	1971-04-12	6

Bruke både AND og OR

Spørring som finner navn på kunder som har kjøpt mindre enn 5 eller mer enn 15 produkter og er født etter '2000-01-01'

```
SELECT Name FROM Customer
WHERE (NrProducts < 5 OR
      NrProducts > 15) AND
      Birthdate > '2000-01-01'
```

Resultat

Name
Peter Young

Velge TVer

Spørring som henter navnet, merket og pris på 48 og 50 tommer TVer

```
SELECT Name, Brand, Price
FROM Product
WHERE Name = 'TV 50 inch' OR
       Name = 'TV 48 inch'
```

Resultat

Velge TVer

Spørring som henter navnet, merket og pris på 48 og 50 tommer TVer

```
SELECT Name, Brand, Price
FROM Product
WHERE Name = 'TV 50 inch' OR
       Name = 'TV 48 inch'
```

Resultat

ProductID	Name	Brand	Price	Stock
0	TV 50 inch	Sony	8999	29
1	Laptop 2.5GHz	Lenovo	7499	12
2	Laptop 8GB RAM	HP	6999	80
3	Speaker 500	Bose	4999	42
4	TV 48 inch	Panasonic	11999	31
5	Phone S6	IPhone	5195	65

Velge TVer

Spørring som henter navnet, merket og pris på 48 og 50 tommer TVer

```
SELECT Name, Brand, Price
FROM Product
WHERE Name = 'TV 50 inch' OR
       Name = 'TV 48 inch'
```

Resultat

ProductID	Name	Brand	Price	Stock
0	TV 50 inch	Sony	8999	29
1	Laptop 2.5GHz	Lenovo	7499	12
2	Laptop 8GB RAM	HP	6999	80
3	Speaker 500	Bose	4999	42
4	TV 48 inch	Panasonic	11999	31
5	Phone S6	IPhone	5195	65

Velge TVer

Spørring som henter navnet, merket og pris på 48 og 50 tommer TVer

```
SELECT Name, Brand, Price
FROM Product
WHERE Name = 'TV 50 inch' OR
       Name = 'TV 48 inch'
```

Resultat

ProductID	Name	Brand	Price	Stock
0	TV 50 inch	Sony	8999	29
1	Laptop 2.5GHz	Lenovo	7499	12
2	Laptop 8GB RAM	HP	6999	80
3	Speaker 500	Bose	4999	42
4	TV 48 inch	Panasonic	11999	31
5	Phone S6	IPhone	5195	65

Velge TVer

Spørring som henter navnet, merket og pris på 48 og 50 tommer TVer

```
SELECT Name, Brand, Price
FROM Product
WHERE Name = 'TV 50 inch' OR
       Name = 'TV 48 inch'
```

Resultat

Name	Brand	Price
TV 50 inch	Sony	8999
TV 48 inch	Panasonic	11999

- ◆ Med det vi har lært hittil har vi ingen måte å spørre etter alle TVer

- ◆ Med det vi har lært hittil har vi ingen måte å spørre etter alle TVer
 - ◆ (altså alle produkter som har navn som starter med 'TV')

- ◆ Med det vi har lært hittil har vi ingen måte å spørre etter alle TVer
 - ◆ (altså alle produkter som har navn som starter med 'TV')
- ◆ Vi kan kun bruke likhet, ingen måte å søke i tekst

- ◆ Med det vi har lært hittil har vi ingen måte å spørre etter alle TVer
 - ◆ (altså alle produkter som har navn som starter med 'TV')
- ◆ Vi kan kun bruke likhet, ingen måte å søke i tekst
- ◆ Dette kan gjøres med SQLs `LIKE`

- ◆ Med det vi har lært hittil har vi ingen måte å spørre etter alle TVer
 - ◆ (altså alle produkter som har navn som starter med 'TV')
- ◆ Vi kan kun bruke likhet, ingen måte å søke i tekst
- ◆ Dette kan gjøres med SQLs `LIKE`
- ◆ Kan så bruke '%' som "wildcard" som matcher alt

LIKE

For eksempel:

LIKE

For eksempel:

- ◆ Name `LIKE 'TV%'`
 - ◆ Sant for alle Name-verdier som starter med 'TV'
 - ◆ f.eks. 'TV 50 inch' og 'TVSHOW'
 - ◆ men ikke f.eks. 'hello' eller 'MTV'

LIKE

For eksempel:

- ◆ Name **LIKE** 'TV%'
 - ◆ Sant for alle Name-verdier som starter med 'TV'
 - ◆ f.eks. 'TV 50 inch' og 'TVSHOW'
 - ◆ men ikke f.eks. 'hello' eller 'MTV'
- ◆ Name **LIKE** '%TV'
 - ◆ sant for alle Name-verdier som slutter med 'TV'
 - ◆ f.eks. '50 inch TV' og 'MTV'
 - ◆ men ikke f.eks. 'TV2' eller 'Fun TV program'

LIKE

For eksempel:

- ◆ Name **LIKE** 'TV%'
 - ◆ Sant for alle Name-verdier som starter med 'TV'
 - ◆ f.eks. 'TV 50 inch' og 'TVSHOW'
 - ◆ men ikke f.eks. 'hello' eller 'MTV'
- ◆ Name **LIKE** '%TV'
 - ◆ sant for alle Name-verdier som slutter med 'TV'
 - ◆ f.eks. '50 inch TV' og 'MTV'
 - ◆ men ikke f.eks. 'TV2' eller 'Fun TV program'
- ◆ Name **LIKE** '%TV%'
 - ◆ sant for alle Name-verdier som inneholder 'TV' (hvor som helst)
 - ◆ f.eks. '50 inch TV' og 'Fun TV program'
 - ◆ men ikke f.eks. 'T2V' eller 'hello'

LIKE

For eksempel:

- ◆ Name `LIKE 'TV%'`
 - ◆ Sant for alle Name-verdier som starter med 'TV'
 - ◆ f.eks. 'TV 50 inch' og 'TVSHOW'
 - ◆ men ikke f.eks. 'hello' eller 'MTV'
- ◆ Name `LIKE '%TV'`
 - ◆ sant for alle Name-verdier som slutter med 'TV'
 - ◆ f.eks. '50 inch TV' og 'MTV'
 - ◆ men ikke f.eks. 'TV2' eller 'Fun TV program'
- ◆ Name `LIKE '%TV%'`
 - ◆ sant for alle Name-verdier som inneholder 'TV' (hvor som helst)
 - ◆ f.eks. '50 inch TV' og 'Fun TV program'
 - ◆ men ikke f.eks. 'T2V' eller 'hello'
- ◆ Name `LIKE '%TV%inch'`
 - ◆ sant for alle Name-verdier som inneholder 'TV' og slutter med 'inch'
 - ◆ f.eks. 'TV 50 inch' og 'Fun TV program pinch'
 - ◆ men ikke f.eks. 'TV 50 inches' eller '50 inch TV'

Velge TVer med LIKE

Spørring som finner navn, pris og merke på alle TVer

```
SELECT Name, Brand, Price  
FROM Product  
WHERE Name LIKE 'TV%'
```

Resultat

Velge TVer med LIKE

Spørring som finner navn, pris og merke på alle TVer

```
SELECT Name, Brand, Price
FROM Product
WHERE Name LIKE 'TV%'
```

Resultat

ProductID	Name	Brand	Price	Stock
0	TV 50 inch	Sony	8999	29
1	Laptop 2.5GHz	Lenovo	7499	12
2	Laptop 8GB RAM	HP	6999	80
3	Speaker 500	Bose	4999	42
4	TV 48 inch	Panasonic	11999	31
5	Phone S6	IPhone	5195	65

Velge TVer med LIKE

Spørring som finner navn, pris og merke på alle TVer

```
SELECT Name, Brand, Price
FROM Product
WHERE Name LIKE 'TV%'
```

Resultat

ProductID	Name	Brand	Price	Stock
0	TV 50 inch	Sony	8999	29
1	Laptop 2.5GHz	Lenovo	7499	12
2	Laptop 8GB RAM	HP	6999	80
3	Speaker 500	Bose	4999	42
4	TV 48 inch	Panasonic	11999	31
5	Phone S6	IPhone	5195	65

Velge TVer med LIKE

Spørring som finner navn, pris og merke på alle TVer

```
SELECT Name, Brand, Price
FROM Product
WHERE Name LIKE 'TV%'
```

Resultat

ProductID	Name	Brand	Price	Stock
0	TV 50 inch	Sony	8999	29
1	Laptop 2.5GHz	Lenovo	7499	12
2	Laptop 8GB RAM	HP	6999	80
3	Speaker 500	Bose	4999	42
4	TV 48 inch	Panasonic	11999	31
5	Phone S6	IPhone	5195	65

Velge TVer med LIKE

Spørring som finner navn, pris og merke på alle TVer

```
SELECT Name, Brand, Price
FROM Product
WHERE Name LIKE 'TV%'
```

Resultat

Name	Brand	Price
TV 50 inch	Sony	8999
TV 48 inch	Panasonic	11999

Regulære uttrykk

- ◆ `LIKE` støtter kun % (og _ for wildcard enkelt karakter)

Regulære uttrykk

- ◆ `LIKE` støtter kun % (og `_` for wildcard enkelt karakter)
- ◆ Ønsker man komplisert matching kan man bruke `SIMILAR TO` eller `~`

Regulære uttrykk

- ◆ `LIKE` støtter kun % (og _ for wildcard enkelt karakter)
- ◆ Ønsker man komplisert matching kan man bruke `SIMILAR TO` eller `~`
- ◆ `SIMILAR TO` bruker litt rar miks av `LIKE`-syntaks (%) og vanlige regulære uttrykk

Regulære uttrykk

- ◆ `LIKE` støtter kun % (og _ for wildcard enkelt karakter)
- ◆ Ønsker man komplisert matching kan man bruke `SIMILAR TO` eller `~`
- ◆ `SIMILAR TO` bruker litt rar miks av `LIKE`-syntaks (%) og vanlige regulære uttrykk
- ◆ F.eks. er `Name = 'abc'` et mulig svar for

```
SELECT Name
FROM Products
WHERE Name SIMILAR TO '%(b|d)%'
```

Regulære uttrykk

- ◆ `LIKE` støtter kun % (og `_` for wildcard enkelt karakter)
- ◆ Ønsker man komplisert matching kan man bruke `SIMILAR TO` eller `~`
- ◆ `SIMILAR TO` bruker litt rar miks av `LIKE`-syntaks (%) og vanlige regulære uttrykk
- ◆ F.eks. er `Name = 'abc'` et mulig svar for

```
SELECT Name
FROM Products
WHERE Name SIMILAR TO '%(b|d)%'
```

- ◆ Man kan også bruke `~` for vanlige (POSIX) regulære uttrykk

Regulære uttrykk

- ◆ `LIKE` støtter kun % (og `_` for wildcard enkelt karakter)
- ◆ Ønsker man komplisert matching kan man bruke `SIMILAR TO` eller `~`
- ◆ `SIMILAR TO` bruker litt rar miks av `LIKE`-syntaks (%) og vanlige regulære uttrykk
- ◆ F.eks. er `Name = 'abc'` et mulig svar for

```
SELECT Name
FROM Products
WHERE Name SIMILAR TO '%(b|d)%'
```

- ◆ Man kan også bruke `~` for vanlige (POSIX) regulære uttrykk
- ◆ F.eks.

```
Name ~ '.*(b|d).*'
```

er samme som over

Regulære uttrykk

- ◆ `LIKE` støtter kun % (og `_` for wildcard enkelt karakter)
- ◆ Ønsker man komplisert matching kan man bruke `SIMILAR TO` eller `~`
- ◆ `SIMILAR TO` bruker litt rar miks av `LIKE`-syntaks (%) og vanlige regulære uttrykk
- ◆ F.eks. er `Name = 'abc'` et mulig svar for

```
SELECT Name
FROM Products
WHERE Name SIMILAR TO '%(b|d)%'
```

- ◆ Man kan også bruke `~` for vanlige (POSIX) regulære uttrykk
- ◆ F.eks.

```
Name ~ '.*(b|d).*'
```

er samme som over

- ◆ `LIKE` finnes fordi den er sikrere mhp. ytelse (kan alltid eksekveres raskt)

- ◆ Av og til vil vi bare ha svar som *ikke* tilfredstiller et uttrykk

Negasjon

- ◆ Av og til vil vi bare ha svar som *ikke* tilfredstiller et uttrykk
- ◆ Bruker da NOT-nøkkelordet

- ◆ Av og til vil vi bare ha svar som *ikke* tilfredstiller et uttrykk
- ◆ Bruker da **NOT**-nøkkelordet
- ◆ For eksempel:

```
SELECT Name
  FROM Products
 WHERE NOT Description LIKE '%simple%'
```

er sant for alle rader som ikke har order 'simple' i sin Description

- ◆ Av og til vil vi bare ha svar som *ikke* tilfredstiller et uttrykk
- ◆ Bruker da **NOT**-nøkkelordet
- ◆ For eksempel:

```
SELECT Name
  FROM Products
 WHERE NOT Description LIKE '%simple%'
```

er sant for alle rader som ikke har order 'simple' i sin Description

- ◆ Merk at

- ◆ Av og til vil vi bare ha svar som *ikke* tilfredstiller et uttrykk
- ◆ Bruker da **NOT**-nøkkelordet
- ◆ For eksempel:

```
SELECT Name
FROM Products
WHERE NOT Description LIKE '%simple%'
```

er sant for alle rader som ikke har order 'simple' i sin Description

- ◆ Merk at
 - ◆ **NOT** (E1 **AND** E2) er ekvivalent med (**NOT** E1) **OR** (**NOT** E2)

- ◆ Av og til vil vi bare ha svar som *ikke* tilfredstiller et uttrykk
- ◆ Bruker da `NOT`-nøkkelordet
- ◆ For eksempel:

```
SELECT Name
  FROM Products
 WHERE NOT Description LIKE '%simple%'
```

er sant for alle rader som ikke har order 'simple' i sin Description

- ◆ Merk at
 - ◆ `NOT (E1 AND E2)` er ekvivalent med `(NOT E1) OR (NOT E2)`
 - ◆ `NOT (E1 OR E2)` er ekvivalent med `(NOT E1) AND (NOT E2)`

Null

- ◆ Når vi setter inn data vil vi av og til mangle en verdi (f.eks. fordi den er ukjent eller ikke finnes)

Null

- ◆ Når vi setter inn data vil vi av og til mangle en verdi (f.eks. fordi den er ukjent eller ikke finnes)
- ◆ For eksempel, kan det være vi ikke vet fødselsdatoen til en bestemt student

SID	StdName	StdBirthdate
0	Anna Consuma	1978-10-09
1	Anna Consuma	1978-10-09
2	Peter Young	2009-03-01
3	Carla Smith	1986-06-14
4	Sam Penny	?

Null

- ◆ Når vi setter inn data vil vi av og til mangle en verdi (f.eks. fordi den er ukjent eller ikke finnes)
- ◆ For eksempel, kan det være vi ikke vet fødselsdatoen til en bestemt student
- ◆ Likevel ønsker vi å legge studenten inn i databasen slik at vi kan lagre informasjon om studenten

SID	StdName	StdBirthdate
0	Anna Consuma	1978-10-09
1	Anna Consuma	1978-10-09
2	Peter Young	2009-03-01
3	Carla Smith	1986-06-14
4	Sam Penny	?

Null

- ◆ Når vi setter inn data vil vi av og til mangle en verdi (f.eks. fordi den er ukjent eller ikke finnes)
- ◆ For eksempel, kan det være vi ikke vet fødselsdatoen til en bestemt student
- ◆ Likevel ønsker vi å legge studenten inn i databasen slik at vi kan lagre informasjon om studenten
- ◆ Men hva skal vi sette inn?

SID	StdName	StdBirthdate
0	Anna Consuma	1978-10-09
1	Anna Consuma	1978-10-09
2	Peter Young	2009-03-01
3	Carla Smith	1986-06-14
4	Sam Penny	?

Null

- ◆ Når vi setter inn data vil vi av og til mangle en verdi (f.eks. fordi den er ukjent eller ikke finnes)
- ◆ For eksempel, kan det være vi ikke vet fødselsdatoen til en bestemt student
- ◆ Likevel ønsker vi å legge studenten inn i databasen slik at vi kan lagre informasjon om studenten
- ◆ Men hva skal vi sette inn?
 - ◆ Den tomme teksten? Feil type!

SID	StdName	StdBirthdate
0	Anna Consuma	1978-10-09
1	Anna Consuma	1978-10-09
2	Peter Young	2009-03-01
3	Carla Smith	1986-06-14
4	Sam Penny	' '

Null

- ◆ Når vi setter inn data vil vi av og til mangle en verdi (f.eks. fordi den er ukjent eller ikke finnes)
- ◆ For eksempel, kan det være vi ikke vet fødselsdatoen til en bestemt student
- ◆ Likevel ønsker vi å legge studenten inn i databasen slik at vi kan lagre informasjon om studenten
- ◆ Men hva skal vi sette inn?
 - ◆ Den tomme teksten? Feil type!
 - ◆ År 0? Ikke korrekt!

SID	StdName	StdBirthdate
0	Anna Consuma	1978-10-09
1	Anna Consuma	1978-10-09
2	Peter Young	2009-03-01
3	Carla Smith	1986-06-14
4	Sam Penny	0000-01-01

Null

- ◆ Når vi setter inn data vil vi av og til mangle en verdi (f.eks. fordi den er ukjent eller ikke finnes)
- ◆ For eksempel, kan det være vi ikke vet fødselsdatoen til en bestemt student
- ◆ Likevel ønsker vi å legge studenten inn i databasen slik at vi kan lagre informasjon om studenten
- ◆ Men hva skal vi sette inn?
 - ◆ Den tomme teksten? Feil type!
 - ◆ År 0? Ikke korrekt!
- ◆ For ukjente og manglende verdier har SQL **NULL**

SID	StdName	StdBirthdate
0	Anna Consuma	1978-10-09
1	Anna Consuma	1978-10-09
2	Peter Young	2009-03-01
3	Carla Smith	1986-06-14
4	Sam Penny	0000-01-01

Null

- ◆ Når vi setter inn data vil vi av og til mangle en verdi (f.eks. fordi den er ukjent eller ikke finnes)
- ◆ For eksempel, kan det være vi ikke vet fødselsdatoen til en bestemt student
- ◆ Likevel ønsker vi å legge studenten inn i databasen slik at vi kan lagre informasjon om studenten
- ◆ Men hva skal vi sette inn?
 - ◆ Den tomme teksten? Feil type!
 - ◆ År 0? Ikke korrekt!
- ◆ For ukjente og manglende verdier har SQL **NULL**
- ◆ Så, for å sette inn studenten Sam Penny med ukjent fødselsdato, bruker vi **NULL**

SID	StdName	StdBirthdate
0	Anna Consuma	1978-10-09
1	Anna Consuma	1978-10-09
2	Peter Young	2009-03-01
3	Carla Smith	1986-06-14
4	Sam Penny	NULL

SQL og null

- ◆ Hvordan sjekker vi om en verdi er `NULL`?

SQL og null

- ◆ Hvordan sjekker vi om en verdi er `NULL`?
- ◆ Dersom vi prøver

```
SELECT StdName
FROM Students
WHERE StdBirthdate = NULL
```

får vi ingen svar!

SQL og null

- ◆ Hvordan sjekker vi om en verdi er `NULL`?
- ◆ Dersom vi prøver

```
SELECT StdName
FROM Students
WHERE StdBirthdate = NULL
```

får vi ingen svar!

- ◆ Faktisk så er `NULL = NULL` ikke sant

SQL og null

- ◆ Hvordan sjekker vi om en verdi er `NULL`?
- ◆ Dersom vi prøver

```
SELECT StdName
FROM Students
WHERE StdBirthdate = NULL
```

får vi ingen svar!

- ◆ Faktisk så er `NULL = NULL` ikke sant
- ◆ og heller ikke `NOT (NULL = NULL)`!

SQL og null

- ◆ Hvordan sjekker vi om en verdi er `NULL`?
- ◆ Dersom vi prøver

```
SELECT StdName
FROM Students
WHERE StdBirthdate = NULL
```

får vi ingen svar!

- ◆ Faktisk så er `NULL = NULL` ikke sant
- ◆ og heller ikke `NOT (NULL = NULL)`!
- ◆ Grunnen til dette er at `NULL` representerer en manglende eller ukjent verdi

SQL og null

- ◆ Hvordan sjekker vi om en verdi er `NULL`?
- ◆ Dersom vi prøver

```
SELECT StdName
FROM Students
WHERE StdBirthdate = NULL
```

får vi ingen svar!

- ◆ Faktisk så er `NULL = NULL` ikke sant
- ◆ og heller ikke `NOT (NULL = NULL)`!
- ◆ Grunnen til dette er at `NULL` representerer en manglende eller ukjent verdi
- ◆ Så `NULL` kan potensielt representere en hvilken som helst verdi

SQL og null

- ◆ Hvordan sjekker vi om en verdi er `NULL`?
- ◆ Dersom vi prøver

```
SELECT StdName
      FROM Students
      WHERE StdBirthdate = NULL
```

får vi ingen svar!

- ◆ Faktisk så er `NULL = NULL` ikke sant
- ◆ og heller ikke `NOT (NULL = NULL)`!
- ◆ Grunnen til dette er at `NULL` representerer en manglende eller ukjent verdi
- ◆ Så `NULL` kan potensielt representere en hvilken som helst verdi
- ◆ Så `StdBirthdate = NULL` og `NULL = NULL` er begge ukjente, altså `NULL`

SQL og null

- ◆ Hvordan sjekker vi om en verdi er `NULL`?
- ◆ Dersom vi prøver

```
SELECT StdName
      FROM Students
      WHERE StdBirthdate = NULL
```

får vi ingen svar!

- ◆ Faktisk så er `NULL = NULL` ikke sant
- ◆ og heller ikke `NOT (NULL = NULL)`!
- ◆ Grunnen til dette er at `NULL` representerer en manglende eller ukjent verdi
- ◆ Så `NULL` kan potensielt representere en hvilken som helst verdi
- ◆ Så `StdBirthdate = NULL` og `NULL = NULL` er begge ukjente, altså `NULL`
- ◆ Og `NULL` er ikke `TRUE` (sant) så det tilfredstiller ikke `WHERE`-klausulen

Sjekke for NULLs

- ◆ For å sjekke om en verdi er `NULL` må vi bruke `IS NULL`.

Sjekke for NULLs

- ◆ For å sjekke om en verdi er `NULL` må vi bruke `IS NULL`.
- ◆ For eksempel:

```
SELECT StdName
FROM Students
WHERE StdBirthdate IS NULL
```

så får vi Sam Penny som svar

Sjekke for NULLs

- ◆ For å sjekke om en verdi er `NULL` må vi bruke `IS NULL`.
- ◆ For eksempel:

```
SELECT StdName
FROM Students
WHERE StdBirthdate IS NULL
```

så får vi Sam Penny som svar

- ◆ Vi kan også bruke `IS NOT NULL` for å sjekke at en verdi ikke er `NULL`

NULLs oppførsel

- ◆ Merk at `NULL` oppfører seg som *ukjent*:

NULLs oppførsel

- ◆ Merk at `NULL` oppfører seg som *ukjent*:
 - ◆ `NULL AND TRUE` resulterer i

NULLs oppførsel

- ◆ Merk at `NULL` oppfører seg som *ukjent*:
 - ◆ `NULL AND TRUE` resulterer i `NULL`

NULLs oppførsel

- ◆ Merk at `NULL` oppfører seg som *ukjent*:
 - ◆ `NULL AND TRUE` resulterer i `NULL`
 - ◆ `NULL OR FALSE` resulterer i

NULLs oppførsel

- ◆ Merk at `NULL` oppfører seg som *ukjent*:
 - ◆ `NULL AND TRUE` resulterer i `NULL`
 - ◆ `NULL OR FALSE` resulterer i `NULL`

NULLs oppførsel

- ◆ Merk at `NULL` oppfører seg som *ukjent*:
 - ◆ `NULL AND TRUE` resulterer i `NULL`
 - ◆ `NULL OR FALSE` resulterer i `NULL`
 - ◆ `NULL AND FALSE` resulterer i

NULLs oppførsel

- ◆ Merk at `NULL` oppfører seg som *ukjent*:
 - ◆ `NULL AND TRUE` resulterer i `NULL`
 - ◆ `NULL OR FALSE` resulterer i `NULL`
 - ◆ `NULL AND FALSE` resulterer i `FALSE`

- ◆ Merk at `NULL` oppfører seg som *ukjent*:
 - ◆ `NULL AND TRUE` resulterer i `NULL`
 - ◆ `NULL OR FALSE` resulterer i `NULL`
 - ◆ `NULL AND FALSE` resulterer i `FALSE`
 - ◆ `NULL OR TRUE` resulterer i

NULLs oppførsel

- ◆ Merk at `NULL` oppfører seg som *ukjent*:
 - ◆ `NULL AND TRUE` resulterer i `NULL`
 - ◆ `NULL OR FALSE` resulterer i `NULL`
 - ◆ `NULL AND FALSE` resulterer i `FALSE`
 - ◆ `NULL OR TRUE` resulterer i `TRUE`

NULLs oppførsel

- ◆ Merk at `NULL` oppfører seg som *ukjent*:
 - ◆ `NULL AND TRUE` resulterer i `NULL`
 - ◆ `NULL OR FALSE` resulterer i `NULL`
 - ◆ `NULL AND FALSE` resulterer i `FALSE`
 - ◆ `NULL OR TRUE` resulterer i `TRUE`
 - ◆ `10 + NULL` resulterer i

- ◆ Merk at `NULL` oppfører seg som *ukjent*:
 - ◆ `NULL AND TRUE` resulterer i `NULL`
 - ◆ `NULL OR FALSE` resulterer i `NULL`
 - ◆ `NULL AND FALSE` resulterer i `FALSE`
 - ◆ `NULL OR TRUE` resulterer i `TRUE`
 - ◆ `10 + NULL` resulterer i `NULL`

NULLs oppførsel

- ◆ Merk at `NULL` oppfører seg som *ukjent*:
 - ◆ `NULL AND TRUE` resulterer i `NULL`
 - ◆ `NULL OR FALSE` resulterer i `NULL`
 - ◆ `NULL AND FALSE` resulterer i `FALSE`
 - ◆ `NULL OR TRUE` resulterer i `TRUE`
 - ◆ `10 + NULL` resulterer i `NULL`
 - ◆ (Prøv å lese hver setning over med *ukjent* i stedet for `NULL`)

NULLs oppførsel

- ◆ Merk at `NULL` oppfører seg som *ukjent*:
 - ◆ `NULL AND TRUE` resulterer i `NULL`
 - ◆ `NULL OR FALSE` resulterer i `NULL`
 - ◆ `NULL AND FALSE` resulterer i `FALSE`
 - ◆ `NULL OR TRUE` resulterer i `TRUE`
 - ◆ `10 + NULL` resulterer i `NULL`
 - ◆ (Prøv å lese hver setning over med *ukjent* i stedet for `NULL`)
- ◆ Så resultatet av et uttrykk med `NULL` er `NULL` dersom svaret avhenger av hva `NULL` kan være

Eksempel fra Northwind-databasen

Finn navnet og prisen på alle produkter som selges i flasker eller glass og som koster mer enn 30 dollar. [4 rader]

Eksempel fra Northwind-databasen

Finn navnet og prisen på alle produkter som selges i flasker eller glass og som koster mer enn 30 dollar. [4 rader]

```
SELECT product_name, unit_price
FROM products
WHERE (quantity_per_unit LIKE '%bottles' OR
       quantity_per_unit LIKE '%jars')
AND unit_price > 30;
```


Takk for nå!

Neste video vil se mer om [SELECT](#)-klausulen.