

IN2090 – Databaser og datamodellering

09 – Eksempler: Aggregering og sortering

Leif Harald Karlsen
leifhka@ifi.uio.no



Universitetet i Oslo

Sortering: Oppgave 1

Finn alle produktnavn, kategorinavn og pris på produkter, sortert etter kategorinavn og deretter produktnavn.

```
SELECT c.category_name, p.product_name, p.unit_price
FROM categories AS c INNER JOIN products AS p
ON (c.category_id = p.category_id)
ORDER BY c.category_name, p.product_name;
```

Sortering: Oppgave 2

Finn navnet og ratingen på de 10 høyest ratede filmene i Filmdatabasen med mer enn 1000 stemmer

```
SELECT f.title, r.rank
  FROM film AS f
       INNER JOIN filmrating AS r USING (filmid)
 WHERE r.votes > 1000
ORDER BY r.rank DESC
LIMIT 10;
```

Aggregering i grupper: Oppgave 3

Finn ut hvor mange kunder som kommer fra hvert land

```
SELECT country, count(*) AS nr_customers
  FROM customers
 GROUP BY country;
```

Aggregering i grupper: Oppgave 4

Finn navn og total regning for hver kunde (antar at ingen bestillinger ennå er betalt for)

```
SELECT c.company_name ,
       sum(d.unit_price * d.quantity * (1 - d.discount)) AS customertotal
FROM customers AS c
     INNER JOIN orders AS o USING (customer_id)
     INNER JOIN order_details AS d USING (order_id)
GROUP BY c.customer_id, c.company_name;
```

Aggregering i grupper: Oppgave 5

Finn navn og total regning for hver kunde som har kjøpt fler enn 1000 varer, sortert alfabetisk etter firmanavn

```
SELECT c.company_name ,
       sum(d.unit_price * d.quantity * (1 - d.discount)) AS customertotal
FROM customers AS c
     INNER JOIN orders AS o USING (customer_id)
     INNER JOIN order_details AS d USING (order_id)
GROUP BY c.customer_id, c.company_name
HAVING sum(d.quantity) > 1000
ORDER BY c.company_name;
```

Avansert SQL: Oppgave 6

Finn de 10 skuespillerne som har spilt i flest filmer, men som har spilt i filmer med gjennomsnittsrating høyere enn 9, sortert etter antall filmer de har spilt i

```
WITH
  played_in AS (
    SELECT DISTINCT fp.personid, fp.filmid, r.rank
    FROM filmparticipation AS fp
      INNER JOIN film AS f USING (filmid)
      INNER JOIN filmrating AS r USING (filmid)
    WHERE fp.parttype = 'cast'
  )
SELECT p.personid, p.firstname, p.lastname, count(*) AS nr_played_in
FROM person AS p
  INNER JOIN played_in AS pi USING (personid)
GROUP BY p.personid, p.firstname, p.lastname
HAVING avg(pi.rank) > 9
ORDER BY nr_played_in DESC
LIMIT 10;
```