IN2090 – Databaser og datamodellering

05 – `SELECT`-klausulen

Leif Harald Karlsen
leifhka@ifi.uio.no

Universitetet i Oslo

Uttrykk i `SELECT`

- Hittil har vi bare hentet ut data direkte fra tabeller
- Ofte ønsker man å transformere dataene før vi returnerer svaret
- Dette kan gjøres med bruke uttrykk for å manipulere verdiene i `SELECT`-klausulen
- For eksempel, for å få alle priser i NOK fremfor USD (antar at 1 USD = 8 NOK) kan vi gjøre:

```
SELECT product_name, unit_price * 8
  FROM products
```

- Eller, for å få alle kunders kontaktpersoners navn med tittel først:

```
SELECT contact_title || ' ' || contact_name
  FROM customers
```

- `||` konkatenerer strenger (f.eks. `'hel'|| 'lo'= 'hello'`)

Gi navn til kolonner

- Når vi har et uttrykk i en `SELECT`-klausul får den resulterende kolonnen ingen navn
- Vi kan gi kolonner resultat-tabellen navn ved å bruke `AS`-nøkkelordet
- F.eks.:

```
SELECT product_name, unit_price * 8 AS unit_price_nok
  FROM products
```

```
SELECT contact_title || ' ' || contact_name AS contact_person
  FROM customers
```

Dupliserte svar

- Svarene fra en spørring kan inneholde duplikater
- F.eks. dersom vi kjører

```
SELECT contact_title
  FROM customers
 WHERE contact_title LIKE '%Manager%'
```

over `northwind`-databasen får vi 33 svar:

contacttitle
Marketing Manager
Accounting Manager
Marketing Manager
Sales Manager
Accounting Manager
Marketing Manager
Marketing Manager
⋮

Fjerning av duplikater

- Duplikater er av og til uønsket
- Vi kan fjerne duplikater med `DISTINCT`-nøkkelordet i `SELECT`-klausulen
- F.eks.:

```
SELECT DISTINCT contact_title
  FROM customers
 WHERE contact_title LIKE '%Manager%'
```

gir kun 3 svar:

contacttitle
Sales Manager
Marketing Manager
Accounting Manager

Aggregering

- En aggregeringsfunksjon er en funksjon som returnerer en enkel verdi fra en samling verdier
- I SQL har vi mange aggregeringsfunksjoner, slik som `sum`, `avg`, `count`, osv.
- Disse funksjonene kan enten bli anvendt på alle verdier i en kolonne (f.eks. summere alle priser)
- eller anvendes på grupper av rader (kommer tilbake til dette om noen uker)

Aggregering: Sum

- For å summere en hel kolonne, kan vi putte `sum(<column>)` i `SELECT`-klausulen
- For eksempel, for å finne det totale antallet varer på lager kan vi summere `units_in_stock`-kolonnen i `products`-tabellen slik:

```
SELECT sum(units_in_stock) AS total_nr_products
  FROM products
```

- Tilsvarende har vi:
  - `avg` – gjennomsnitt
  - `max` – maksimum
  - `min` – minimum
  - `count` – antall rader

Kombinere aggregering og andre kolonner

- En aggregeringsfunksjon returnerer én enkel verdi
- Altså gir det ikke mening å direkte kombinere denne med andre kolonner i samme `SELECT`-klausul
- F.eks. følgende gir ikke mening:

```
SELECT product_name,                                    -- ERROR!
       sum(units_in_stock) AS total_nr_products
  FROM products
```

- Merk at man derimot kan kombinere flere aggregater i samme `SELECT`-klausul, f.eks.:

```
SELECT max(unit_price) AS highest,
       min(unit_price) AS lowest,
       max(unit_price) - min(unit_price) AS difference,
  FROM products
```

Aggregering: `Count`

- For eksempel, for å finne antall produkter som koster mer enn 20 dollar kan vi kjøre:

```
SELECT count(*) AS nr_expensive_products
  FROM products
 WHERE unit_price > 20
```

- `count(*)` teller antall rader i resultatet
- `count(product_id)` teller antall ikke-`NULL` verdier i `product_id`-kolonnen
- Merk at det kan være duplikater i svaret, og disse blir telt med
- Skal senere se hvordan man kan telle kun unike svar

Takk for nå!

Neste video vil se mer om `FROM`-klausulen og joins.