

Funktiot



JYVÄSKYLÄN
AMMATTIKORKEAKOULU

Osaaminen kilpailukyvyksi

Koostefunktiot

- Viisi standardifunktiota:
 - AVG -- keskiarvo
 - MIN -- pienin arvo
 - MAX -- suurin arvo
 - SUM -- summa
 - COUNT -- lukumäärä
- Esim. montako henkilöä on henkilo-taulussa:
 - `SELECT Count(*) AS Lkm
FROM henkilo`

Koostefunktioiden käyttö

- Koostefunktioita käytetään SELECT-lauseen sarakeluetelossa ja HAVING-lauseessa

```
SELECT MAX(hinta), MIN(hinta)      -- suurin ja pienin hinta
FROM tuote
```

- WHERE-lauseessa koostefunktioita ei voi käyttää (vaan ne on laitettava HAVING-lauseeseen), esim. Haetaan myyjät, joiden myymien tuotteiden hinta yhteensä on yli 5000:

```
SELECT myyja, SUM(hinta)           -- tuotteiden hintojen summa
FROM tuote
GROUP BY myyja                    -- ryhmittely myyjän mukaan
HAVING SUM(hinta) > 5000
```

- DISTINCT-rajaus toimii muissa tuotteissa paitsi Accessissa:

```
SELECT COUNT(DISTINCT city)        -- monestako kaupungista
FROM authors                       -- kirjailijoita
```

Ryhmittely - Group By

- GROUP BY -ehdolla tiedot ryhmitellään jonkin sarakkeen tai sarakkeiden mukaan ryhmiin
 - GROUP BY -ehdon ryhmiin kohdistetaan yleensä jokin summafunktio
 - kaikki SELECT-listan lauseet, joissa ei ole funktiota, on lueteltava GROUP BY –lauseessa!
- Esim. kunkin myyjän tuotteiden hintojen keskiarvo:
`SELECT myyja, AVG(hinta) FROM tuote
GROUP BY myyja;`
- Kunkin myyjän myymien tuotteiden määrä
`SELECT myyja, COUNT(myyja) FROM tuote
GROUP BY myyja;`

Ryhmittely - Having

- Sekä HAVING- että WHERE-lauseella voidaan ryhmittelyn tulosta rajata:

```
SELECT myyja, COUNT(*) -- myös NULL kelpaa
FROM tuote
WHERE tuoteHinta > 100
GROUP BY myyja          -- hae myyjät joilla
HAVING COUNT(*) > 2;    -- vähintään 3 tuotetta
```

- **Tulosjoukkoa voidaan siis rajata eri tasoilla:**
 - rajataan ensiksi ryhmiteltävä joukko WHERE-lausekkeella
 - tästä syntynyt tulos ryhmitellään GROUP BY-määrittelyllä
 - lopuksi ryhmiteltyä joukkoa voidaan vielä rajata HAVING-lauseella

Merkkijonofunktioita

- Merkkitiedon käsittely
 - standardissa merkkijono saadaan funktiolla SUBSTRING(mjono FROM mista FOR pituus)
 - monissa tuotteissa on kuitenkin SUBSTR(mjono,mista,mihin)-funktio (Accessissa Mid-funktio), esim.:
`SELECT sukunimi, ' ', SUBSTR(etunimi,1,1), '.'`
`FROM henkilo`
 - Tuloksena lista: Aaltonen, M. ...
- Muita merkkijonofunktioita:
 - CHAR_LENGTH -- merkkien määrä (Oraclessa LENGTH)
 - UPPER(nimi) -- merkit isoiksi (Accessissa UCase)
 - LOWER(nimi) -- merkit pieniksi (Accessissa LCase)
 - LEFT(nimi, n) -- merkkijono vasemmalta (n merkkiä)
 - RIGHT(nimi, n) -- merkkijono oikealta (n merkkiä)

Vielä merkkijonojen käsittelystä

- Merkkijonojen yhdistäminen (konkatenointi)
 - standardia noudattavat Oracle ja DB2, jossa merkkijonot yhdistetään kahdella pystyviivalla, esim.:
`SELECT sukunimi || ', ' || etunimi AS Nimi`
`FROM henkilo`
 - MS:n tuotteissa käytetään + -merkkiä ja MySQL:ssä CONCAT-funktiota
- Tietotyypin muuntaminen:
 - eksplisiittisesti numeerisen ja merkkimuotoisen välillä, esim. ANSI-standardin CAST-funktiolla tai Oraclessa TO_CHAR (n, [format [,lang]]); muutoin implisiittisesti

Muita funktioita

- Numerofunktioita ovat mm. desimaalien pyöristysfunktiot ROUND ja FLOOR; ROUND-funktiolla saadaan luku pyöristettyä kokonaisluvuksi:
 - `SELECT ROUND(palkka*1.1,0) AS palkankorotus`
`FROM henkilo`
- NULL-arvon muuttamiseen voidaan käyttää esim. COALESCE-funktiota, jolla NULL muutetaan esim. nollaksi: `COALESCE(sarake, 0)`
- Päivämääräfunktiot, ks. Hovi s. 56 – 65
 - Tuotekohtaiset maa-asetukset huomioitava
 - ANSI-standardissa CURRENT_DATE: tämä päivä

SQL:2003-standardin uusia funktioita

- Nk. skalaarifunktiot: LN (), EXP (), POWER (), SQRT (), FLOOR (), CEIL[ING] (), WIDTH_BUCKET()
- Koostefunktiot: STDDEV_POP (), STDDEV_SAMP (), VAR_POP (), VAR_SAMP () ym.
- Nk. taulukkofunktiot: RANK () OVER ..., DENSE_RANK () OVER ..., PERCENT_RANK () OVER ..., CUME_DIST () OVER ..., ROW_NUMBER () OVER ...
- Muitakin on ...
- Tuotekohtaisista eroista on hyvä yhteenveto:
<http://troels.arvin.dk/db/rdbms/#functions>

CASE-lause

- Tarvitaan kun sarakkeen otsikko riippuu sarakkeen arvosta
- Kaksi syntaksia (sarakenimen kanssa tai ilman)
- Esimerkiksi haetaan tuotteita ja luokituksia:

`SELECT tuote_nimi,`

`CASE`

`WHEN hinta < 100 THEN 'edullinen'`

`ELSE 'laatutavara'`

`END AS luokitus`

`FROM tuote`

