

CS-A1150 Tietokannat

26.3.2019

Oppimistavoitteet: tämän luennon jälkeen

- ▶ Osaat tehdä SQL:llä kyselyitä, jotka käyttävät hyväkseen toisen kyselyn tuloksia (alikyselyt).
- ▶ Osaat tehdä SQL:ssä liitoksia suoraan siihen tarkoitetuilla operaatioilla.
- ▶ Osaat SQL:ssä laskea esimerkiksi summia tai keskiarvoja halutun ehdon täyttävien monikoiden jonkin attribuutin arvoista (koostefunktiot ja ryhmittely).

Esimerkkitietokanta

- ▶ Tämän luennon esimerkit käsittelevät aikaisempien luentojen esimerkkitietokantaa, joka koostuu seuraavista relaatioista

Customers(custNo, name, born, bonus, address, email)

Products(number, prodName, description, price, manufID)

Manufacturers(ID, manufName, phone)

Orders(orderNo, deliver, status, custNo)

BelongsTo(orderNo, productNo, count)

Alikyselyt

- ▶ SQL-kyselyn sisään voi kirjoittaa **WHERE**- tai **FROM**-osan sisään toisen kyselyn. Tällaista kyselyä sanotaan *alikäyselyksi* (subquery).
- ▶ Alikäselyn tuloksena on joko yksittäinen arvo (skalaarisen arvon tuottava alikäselly) tai relaatio.

Skalaarisen arvon tuottava alikysely

- ▶ Esimerkki: halutaan löytää tuotteen T-77445 (tuotenumero) valmistajan nimi ja puhelinnumero.
- ▶ Aikaisemmin opituilla tavoilla

```
SELECT manufName, phone  
FROM Manufacturers, Products  
WHERE ID = manufID AND number = 'T-77445';
```

- ▶ Annettu kysely käy hyvin ongelman ratkaisemiseen, mutta tiedot voidaan hakea myös toisella tavalla: haetaan ensin tuotteen T-77445 valmistajan tunnus *Products*-relaatiosta. Sen jälkeen etsitään *Manufacturers*-relaatiosta annettua tunnusta vastaavan valmistajan tiedot. (Tämä ei ole välttämättä kyselyn todellinen suoritusjärjestys.)

Skalaarisen arvon tuottava alikysely, jatkoa

- ▶ Ensimmäinen kysely kirjoitetaan nyt toisen kyselyn **WHERE**-osan sisään:

```
SELECT manufName, phone
FROM Manufacturers
WHERE ID =
    (SELECT manufID
     FROM Products
     WHERE number = 'T-77445'
    );
```

Relaation tuottavat alikyselyt

- ▶ Edellistä tapaa voidaan käyttää sellaisenaan kuitenkin vain silloin, kun alikyselyn tuloksena on yksittäinen arvo, ei relaatio.
- ▶ Alikyselyn tuloksena voi olla myös relaatio. SQL:ssä on tarjolla erilaisia operaattoreita, joiden avulla voi tutkia pääkyselyn ja alikyselyn tulosrelaation suhteita.
- ▶ Voidaan esimerkiksi tutkia, kuuluuko alikyselyn tulosrelaatioon yhtään monikkoa tai vastaako pääkyselyssä tarkasteltava monikko yhtään alikyselyn tulosrelaatioon kuuluvaa monikkoa.

Relaation tuottavat alikyselyt: ehdoissa käytettäviä operaattoreita

- ▶ **EXISTS** R
 - ▶ Tosi, jos R ei ole tyhjä.
- ▶ **s IN** R
 - ▶ Tosi jos ja vain jos s on sama kuin jokin R:ään kuuluva monikko.
- ▶ **s >** **ALL** R
 - ▶ Tosi jos ja vain jos s on suurempi kuin jokainen R:ään kuuluva arvo. R:n täytyy olla unaarinen (yksisarakeinen) relaatio. Operaation > tilalla voi olla myös jokin muista vertailuoperaattoreista <, <=, >=, = tai <>.
- ▶ **s >** **ANY** R
 - ▶ Tosi jos ja vain jos s on suurempi kuin jokin R:ään kuuluva arvo. R:n täytyy olla unaarinen (yksisarakeinen) relaatio. Operaation > tilalla voi olla myös jokin muista vertailuoperaattoreista <, <=, >=, = tai <>.
- ▶ Ehdoista saadaan aikaan negaatio lisäämällä operaattorin eteen sana **NOT**.
- ▶ Huom: SQLitessä ei ole operaattoreita **ALL** (tässä merkityksessä) ja **ANY**.

Esimerkkejä

- ▶ Haetaan niiden tilausten numerot, joihin kuuluu tuote, jonka kuvaus on cellphone:

```
SELECT DISTINCT orderNo
FROM BelongsTo
WHERE productNo IN
  (SELECT number
   FROM Products
   WHERE description = 'cellphone'
  );
```

Välitehtävä 1

- ▶ Kirjoita seuraava kysely SQL:llä käyttämällä alikyselyä:
Hae niiden valmistajien nimet, joilla on verkkokaupassa yli 100 euron arvoisia tuotteita.

Välitehtävän ratkaisu

- ▶ Kirjoita seuraava kysely SQL:llä käyttämällä alikyselyä:
Hae niiden valmistajien nimet, joilla on verkkokaupassa yli 100 euron arvoisia tuotteita.

```
SELECT DISTINCT manufName
FROM Manufacturers
WHERE ID IN
  (SELECT manufID
   FROM Products
   WHERE price > 100
  );
```

Esimerkkejä, jatkuu

- ▶ Haetaan niiden asiakkaiden nimet ja asiakasnumerot, jotka ovat tilanneet tuotteen T-77445 (tuotenumero).

```
SELECT name, custNo
FROM Customers
WHERE custNo IN
    (SELECT custNo
     FROM Orders AS O
     WHERE O.orderNo IN
         (SELECT B.orderNo
          FROM BelongsTo AS B
          WHERE productNo = 'T-77445'
         )
    )
);
```

Ulomman kyselyn monikkoon viittaaminen alikyselyssä

- ▶ Alikyselyssä voidaan myös viitata ulommassa kyselyssä käytettyyn monikkoon.
- ▶ Esimerkki: haetaan kustakin tuoteryhmästä (sama kuvaus) halvin tuote. Jos samanhintaisia halvimpia tuotteita on useita, kysely tuottaa ne kaikki. Tämä kysely ei toimi SQLite:ssä.

```
SELECT DISTINCT number, prodName, description, price
FROM Products AS Cheap
WHERE price <= ALL
  (SELECT price
   FROM Products
   WHERE description = Cheap.description
  );
```

- ▶ Alikysely joudutaan suorittamaan erikseen jokaista ulomman kyselyn monikkoa kohti.

Ulomman kyselyn monikkoon viittaaminen alikyselyssä, jatkoa

- ▶ Vaihtoehtoinen kysely, joka toimii SQLitessä

```
SELECT DISTINCT number, prodName, description, price
FROM Products AS Cheap
WHERE NOT EXISTS
  (SELECT *
   FROM Products
   WHERE description = Cheap.description AND
        price < Cheap.price
  );
```

Ulomman kyselyn monikkoon viittaaminen alikyselyssä

- ▶ Kun kysely sisältää alikyselyitä, joudutaan miettimään, minkä relaation attribuutteja missäkin kyselyssä tarkoitetaan.
- ▶ Sääntö: Kyselyssä oleva attribuutin nimi viittaa saman kyselyn **FROM**-osassa olevaan relaatioon, jos jollain **FROM**-osassa mainitulla relaatiolla on tällainen attribuutti. Jos ei ole, haetaan nimeä seuraavaksi uloman kyselyn **FROM**-osan relaatiosta jne. Rivimuuttujaa käyttämällä voidaan kertoa, että tarkoitetaan ulommassa kyselyssä mainittua relaatiota.
- ▶ Näin ollen rivimuuttujat O ja B kalvolla 12 ovat tarpeettomia.

Alikysely FROM-osassa

- ▶ Myös kyselyn **FROM**-osaan voidaan kirjoittaa alikysely. Ulompi kysely kohdistuu tällöin alikyselyn tuloksena tulevaan relaatioon.
- ▶ Jos alikyselyn tulosrelaatioon halutaan viitata muualla kyselyssä, pitää käyttää rivimuuttujaa.
- ▶ Esimerkki: toinen tapa hakea niiden asiakkaiden nimet ja asiakasnumerot, jotka ovat tilanneet tuotteen T-77445.

```
SELECT name, custNo
FROM Customers, (SELECT custNo
                  FROM Orders AS O, BelongsTo AS B
                  WHERE O.orderNo = B.orderNo AND
                        productNo = 'T-77445'
                  ) AS C1
WHERE Customers.custNo = C1.custNo;
```


Liitosoperaatioiden kirjoittaminen käyttämällä SQL:n liitosoperaattoreita

- ▶ Aikaisemmin kurssilla on kirjoitettu kahden tai useamman relaation liitoksia kirjoittamalla liitettävät relaatiot **FROM**-osaan ja käytettävät liitosehdot **WHERE**-osaan.
- ▶ Toinen vaihtoehto on käyttää suoraan SQL:n liitosoperaattoreita **CROSS JOIN**, **JOIN**, **NATURAL JOIN** ja **OUTER JOIN**.
- ▶ **CROSS JOIN** tuottaa kahden relaation karteesisen tulon, joka on harvoin sellaisenaan käyttökelpoinen. Seuraavilla kalvoilla käydään läpi muita mainittuja liitosoperaatioita.

Operaattori JOIN

- ▶ Operaattorin **JOIN** avulla voidaan tehdä Theta-liitos. Esimerkiksi halutaan yhdistää tuotteet niiden valmistajiin

```
SELECT prodName, manufName  
FROM Products JOIN Manufacturers ON  
    manufID = ID;
```

Luonnollinen liitos

- ▶ Luonnollinen liitos voidaan tehdä operaation **NATURAL JOIN** avulla. Tulosrelaatioon otetaan kaikki ne alkuperäisten relaatioiden monikoiden yhdistelmät, joissa kaikilla samannimisillä attribuuteilla on sama arvo.
- ▶ Tulosrelaatiossa samannimiset attribuutit esiintyvät vain kerran (SQLitessä ne tulevat kuitenkin useaan kertaan).
- ▶ Esimerkki: haetaan asiakkaiden tiedot yhdessä heidän tilaustensa tietojen kanssa

```
SELECT *  
FROM Customers NATURAL JOIN Orders;
```

Ulkoliitokset

- ▶ Liitosta tehdessä alkuperäisissä relaatioissa on usein monikoita, joille ei löydy yhtään vastinparia toisesta liitokseen osallistuvasta relaatiosta.
- ▶ Normaalisti tällaiset alkuperäisen relaation monikot eivät näy mitenkään liitoksen tulosrelaatioissa. Joskus tällaisetkin monikot kuitenkin haluttaisiin tulosrelaatioon mukaan. Ne saadaan mukaan *ulkoliitoksella* (outerjoin).
- ▶ Jos monikolla ei ole vastinparia toisessa liitokseen osallistuvassa relaatioissa, on ulkoliitoksen tulosrelaatioissa tätä monikkoa vastaavassa monikossa toisen relaation attribuuttien arvona NULL.
- ▶ Esimerkki: halutaan liittää asiakkaiden ja tilausten tiedot niin, että tuloksessa on kaikki asiakkaat ja tilaukset, siis myös ne asiakkaat, jotka eivät ole tilanneet mitään, ja ne tilaukset, joiden tehnyttä asiakasta ei löydy tietokannasta.

Ulkoliitos, jatkoa

- ▶ Kysely (ei toimi SQLitessä):

```
SELECT *  
FROM Customers NATURAL FULL OUTER JOIN Orders;
```

- ▶ Jos tulosrelaatioon halutaan vain ilman vastinparia olevat asiakkaat, mutta ei tuntemattomien asiakkaiden tilauksia, tehdään *vasen ulkoliitos* (left-outerjoin, ei toimi SQLitessä näin):

```
SELECT *  
FROM Customers NATURAL LEFT OUTER JOIN Orders;
```

- ▶ Jos taas tulosrelaatioon halutaan vain ilman vastinparia olevat tilaukset, mutta ei asiakkaita, tehdään *oikea ulkoliitos* (right-outerjoin, ei toimi SQLitessä):

```
SELECT *  
FROM Customers NATURAL RIGHT OUTER JOIN Orders;
```

Ulkoliitos, jatkoa

- ▶ Ulkoliitoksissa voidaan käyttää myös joitain ehtoa, esimerkiksi (ei toimi SQLitessä):

```
SELECT *  
FROM Products FULL OUTER JOIN Manufacturers ON  
    manufID = ID;
```

Ulkoliitos, jatkoa

- ▶ Myös ehdon kanssa annettu ulkoliitos voidaan tehdä vasemman- tai oikeanpuoleisena versiona:

```
SELECT *  
FROM Products LEFT OUTER JOIN Manufacturers ON  
      manufID = ID;
```

sisältää myös ne tuotteet, joille ei löydy valmistajaa

ja (ei toimi SQLitessä)

```
SELECT *  
FROM Products RIGHT OUTER JOIN Manufacturers ON  
      manufID = ID;
```

sisältää myös ne valmistajat, joiden tuotteita ei löydy tietokannasta.

Ulkoliitos SQLitessä

- ▶ SQLitessä on toteutettu vain vasemmanpuoleinen ulkoliitos **LEFT OUTER JOIN**
- ▶ **FULL OUTER JOIN** ja **RIGHT OUTER JOIN** ja kaikki versiot **NATURAL OUTER JOIN**ista siis puuttuvat.
- ▶ **Huom:** **LEFT JOIN** tarkoittaa samaa kuin **LEFT OUTER JOIN** (myös muissa järjestelmissä kuin SQLitessä).

Välitehtävä 2

- ▶ Kirjoita seuraava kysely SQL:llä: listaa kaikki tietokannasta löytyvien tuotteiden tuotenumero ja nimi yhdessä niiden tilausten tilausnumeroiden kanssa, joihin tuotteet kuuluvat. Listaa myös ne tuotteet, joita ei ole tilattu lainkaan.

Ratkaisu välitehtävään

- ▶ Kirjoita seuraava kysely SQL:llä: listaa kaikki tietokannasta löytyvien tuotteiden tuotenumero ja nimi yhdessä niiden tilausten tilausnumeroiden kanssa, joihin tuotteet kuuluvat. Listaa myös ne tuotteet, joita ei ole tilattu lainkaan.

```
SELECT number, prodName, orderNo  
FROM Products LEFT OUTER JOIN BelongsTo ON  
    number = productNo;
```

Voidaan kirjoittaa myös

```
SELECT number, prodName, orderNo  
FROM Products LEFT JOIN BelongsTo ON  
    number = productNo;
```

Koosteoperaattorit

- ▶ Ongelma: halutaan laskea tilastoja jonkin relaation jonkin attribuutin arvosta, esimerkiksi kaikkien tuotteiden hintojen summa tai keskiarvo tai niiden asiakkaiden lukumäärä, jotka ovat syntyneet vuoden 1990 jälkeen.
- ▶ Näitä voidaan laskea *koosteoperaattoreiden* (aggregation operators) avulla.
- ▶ SQL tarjoaa koosteoperaattorit **SUM**, **AVG**, **MIN**, **MAX** ja **COUNT** summan, keskiarvon, minimin, maksimin ja lukumäärän laskemista varten.
- ▶ Operaattoreiden operandina voi yleensä olla jokin attribuutti, jolla on numeerinen arvo.
- ▶ Poikkeuksena **COUNT**, jolla voidaan laskea kyselyn tulokseen kuuluvien monikoiden lukumäärä.

Esimerkkejä koosteoperaattoreista

- ▶ Esimerkki: lasketaan kaikkien *Products*-relaatiossa olevien tuotteiden hintojen keskiarvo:

```
SELECT AVG(price)
FROM Products;
```

- ▶ Lasketaan kameroiden hintojen keskiarvo (kuvaus 'camera'):

```
SELECT AVG(price)
FROM Products
WHERE description = 'camera';
```

- ▶ Haetaan niiden asiakkaiden lukumäärä, jotka ovat syntyneet vuoden 1990 jälkeen:

```
SELECT COUNT(*)
FROM Customers
WHERE born > 1990;
```

Esimerkkejä koosteoperaattoreista, jatkuu

- ▶ Haetaan relaatiossa *Orders* olevien eri asiakkaiden lukumäärät. Yksi asiakas lasketaan mukaan vain yhteen kertaan, vaikka hänellä olisi useita tilauksia.

```
SELECT COUNT(DISTINCT custNo)
FROM Orders;
```

Ryhmittely

- ▶ Ongelma: halutaan laskea jonkin koosteoperaattorin arvo monikoilla, joilla on sama arvo jollain toisella attribuutilla, esim. tuotteiden hintojen keskiarvo erikseen eri valmistajien tuotteille.
- ▶ Koosteoperaattoreita käytettäessä monikoita voidaan ryhmitellä **GROUP BY**-operaattorin avulla. Tuotteiden hintojen keskiarvo valmistajittain saataisiin kyselyllä

```
SELECT manufID, AVG(price)
FROM Products
GROUP BY manufID;
```

- ▶ Jos kyselyyn halutaan mukaan vain kamerat, kirjoitetaan kysely:

```
SELECT manufID, AVG(price)
FROM Products
WHERE description = 'camera'
GROUP BY manufID;
```

Esimerkki kyselyn suorittamisesta

- Suoritetaan alla olevalle relaation instanssille seuraava kysely:

```
SELECT manufID, AVG(price)
FROM Products
GROUP BY manufID;
```

Relation Products

<i>number</i>	<i>prodName</i>	<i>description</i>	<i>price</i>	<i>manufID</i>
T-33441	Samsung Galaxy A5	cellphone	250.0	S123
R-55336	IPad Air 2	tablet	495.0	M554
T-77445	Superstar Track M	jacket	30.0	A432
S-65221	Brasserie 24	pan	33.50	F542
T-33442	NX 300 Smart Camera	camera	399	S123
T-33455	Sony Cyber-shot	camera	463.0	L711
P-48221	Teema 26	plate	16.00	F542
T-29783	40 Smart LED-TV	TV	519.0	L711
T-27511	55 Smart LED-TV	TV	799.90	S123

Esimerkki kyselyn suorittamisesta, jatkoa

- ▶ Jaetaan monikot ryhmiin **GROUP BY**-kohdassa annetun attribuutin mukaan:

Relation Products

<i>number</i>	<i>prodName</i>	<i>description</i>	<i>price</i>	<i>manufID</i>
T-33441	Samsung Galaxy A5	cellphone	250.0	S123
T-33442	NX 300 Smart Camera	camera	399	S123
T-27511	55 Smart LED-TV	TV	799.90	S123
R-55336	IPad Air 2	tablet	495.0	M554
T-77445	Superstar Track M	jacket	30.0	A432
S-65221	Brasserie 24	pan	33.50	F542
P-48221	Teema 26	plate	16.00	F542
T-33455	Sony Cyber-shot	camera	463.0	L711
T-29783	40 Smart LED-TV	TV	519.0	L711

Esimerkki kyselyn suorittamisesta, jatkoa

- Lasketaan haluttu keskiarvo jokaiselle ryhmälle erikseen

Relation Products

<i>number</i>	<i>prodName</i>	<i>description</i>	<i>price</i>	<i>manufID</i>	AVG
T-33441	Samsung Galaxy A5	cellphone	250.0	S123	
T-33442	NX 300 Smart Camera	camera	399	S123	
T-27511	55 Smart LED-TV	TV	799.90	S123	482.96
R-55336	IPad Air 2	tablet	495.0	M554	495.0
T-77445	Superstar Track M	jacket	30.0	A432	30.0
S-65221	Brasserie 24	pan	33.50	F542	
P-48221	Teema 26	plate	16.00	F542	24.75
T-33455	Sony Cyber-shot	camera	463.0	L711	
T-29783	40 Smart LED-TV	TV	519.0	L711	491.0

Esimerkki kyselyn suorittamisesta, jatkoa

- ▶ Listataan kyselyssä pyydyt attribootit ja koostefunktioiden arvot

Result

<i>manufID</i>	AVG(price)
S123	482.96
M554	495.0
A432	30.0
F542	24.75
L711	491.0

- ▶ Oikeasti kyselyn täsmällinen suoritus voi poiketa esitetystä, mutta lopputulos on sama.

Ryhmittely, jatkoa

- ▶ Ryhmittelyä voi käyttää myös silloin, kun kyselyssä on mukana useita relaatioita.
- ▶ Esimerkki: halutaan laskea kunkin tilauksen yhteishinta

```
SELECT orderNo, SUM(price * count)
FROM Products, BelongsTo
WHERE productNo = number
GROUP BY orderNo;
```

NULL-arvot koosteoperaattoreiden yhteydessä

- ▶ Jos relaation monikoilla on joidenkin attribuuttien arvona NULL, vaikuttavat ne seuraavasti koosteoperaatioiden tuloksiin:
 - ▶ NULL-arvoja ei oteta lainkaan huomioon koosteoperaatioiden arvoja laskettaessa. Ne eivät siis vaikuta esimerkiksi summaan tai keskiarvoon. Lukumäärää laskettaessa `COUNT(*)` kertoo aina relaatioon kuuluvien monikoiden määrä (jolloin mukaan lasketaan myös monikot, joiden jollain attribuutilla on arvo NULL), mutta `COUNT(A)` kertoo relaation niiden monikoiden lukumäärän, joilla attribuutilla A on eri arvo kuin NULL.
 - ▶ Ryhmittelyssä NULL katsotaan kuitenkin tavalliseksi arvoksi. Ryhmittely voidaan siis tehdä myös sellaisen attribuutin suhteen, jolla on osalla monikoista arvo NULL.
 - ▶ Jos koosteoperaattori kohdistetaan ryhmään, jossa ei ole yhtään monikkoa, tuloksena on arvo NULL. Poikkeuksena on operaatio `COUNT`, jonka tuloksena on tällöin 0.

HAVING-ehdot

- ▶ Jos ryhmälle halutaan antaa jokin ehto, jossa käytetään koosteoperaattoria, ei tällaista ehtoa voida kirjoittaa **WHERE**-osaan, koska **WHERE**-osassa ei voida käyttää koosteoperaattoreita..
- ▶ Esimerkikki: halutaan laskea eri valmistajien tuotteiden hintojen keskiarvot. Halutaan kuitenkin ottaa mukaan vain sellaiset valmistajat, joilla on vähintään yksi yli 100 euroa maksava tuote. Otetaan vastaukseen mukaan valmistajien tunnuksen ja hintojen keskiarvon lisäksi valmistajan nimi.
- ▶ Kysely:

```
SELECT manufID, manufName, AVG(price)
FROM Products, Manufacturers
WHERE id = manufID
GROUP BY manufID
HAVING MAX(price) > 100;
```

HAVING-ehdot, jatkoa

- ▶ Huomaa:
 - ▶ **HAVING**-osassa esiintyvän koostefunktion arvo lasketaan erikseen jokaiselle ryhmälle (esimerkissä jokaiselle valmistajan tunnukselle).
 - ▶ **HAVING**-osassa voi esiintyä minkä tahansa **FROM**-osassa annettujen relaatioiden koostefunktioita, mutta niiden lisäksi myös **GROUP BY**-osassa annettuja attribuutteja ilman koostefunktioita.