CS-A1150 Tietokannat

26.2.2019

Oppimistavoitteet: tämän luennon jälkeen

- Tunnet SQL:n perusteet ja osaat tehdä yksinkertaisia SQL-kyselyitä, esimerkiksi
  - hakea relaatiosta halutun ehdon täyttävät monikot ja niistä halutut attribuutit (valinta ja projektio)
  - yhdistää kahden tai useamman relaation monikoita ja haetaan niistä halutun ehdon toteuttavat yhdistelmät (liitos)
  - järjestää kyselyn tulokset
  - tehdä erilaisia joukko-operaatioita vaativia kyselyitä.

SQL

- Tyypillisiä tietokantojen tarvitsemia toimenpiteitä
  - Uusien relaatioiden (taulujen) määrittelminen.
  - Tietueiden (monikoiden) lisääminen tietokantaan.
  - Tietueiden tietojen päivitys.
  - Kyselyjen tekeminen tietokannasta.
- Näiden toimenpiteiden tekemiseen relaatiotietokannoissa voi käyttää SQL-kieltä (Structured Query Language)
- Kieli kehitettiin 1970-luvulla, ja sen on nykyisin käytössä lähes kaikissa suurissa kaupallisissa tietokannan hallintajärjestelmissä.

Eri SQL-murteita

- ANSI SQL eli SQL1 vuodelta 1986
- SQL2
  - vuoden 1992 standardi
  - laajennettu SQL1:stä, uusina piirteinä esim. typpit DATE ja TIME, erilaisia uusia operaatioita atomisille tietotyypeille, uusia joukko- ja liitosoperaatioita jne.
- SQL-99 eli SQL:1999
  - Tunnettu myös SQL3-nimellä
  - Laajennettu SQL2:sta (esim. rekursio, laukaisimet ja oliot)
  - Standardia on myöhemmin laajenettu (SQL:2003, SQL:2006, SQL:2008, SQL:2011, SQL:2016).
- Ohjelmistotuottajien SQL-murteet ovat tyypillisesti SQL2:n laajennoksia SQL3:n piirteillä.
- Tällä kurssilla harjoitustehtävissä ja harjoitustyössä käytettävä SQLite toteuttaa lähes kaikki SQL2:n piirteet.

SQL:n kyselyt

- SQL:n kyselyt ovat deklaratiivisia: Ne kertovat, mihin kysymykseen halutaan vastaus, mutta eivät määrittele sitä, miten kysely käytännössä suoritetaan.
- Kyselyn käytännön suorittamisen suunnittelu jää tietokannan hallintajärjestelmän kyselynoptimoijan tehtäväksi.
- Kyselyn perusmuoto on

  SELECT <attribuuttilista>
  FROM <relaatiolista>
  WHERE ehto

- FROM-osa ilmaisee, mihin relaatioon/relaatioihin kysely kohdistuu (vrt. karteesinen tulo).
- SELECT-osa määrää, mitkä attribuutit tulostetaan tulokseen valituista monikoista (vrt. projektio).
- WHERE-osa ilmaisee ehdon, jonka kyselyn tulokseen valitun monikon pitää täyttää (vrt. valinta).

Esimerkkitietokanta

- Tämän luennon esimerkit käsittelevät verkkokaupan 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)`

- Oletetaan, että relaation `Products` sisältö on tällä hetkellä seuraava:

Relation `Products`

number	*prodName*	*description*	*price*	*manufID*
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

Esimerkkitietokannan selityksiä

- Relaatioon `Customers` on tallennettu verkkokaupan asiakkaiden tietoja. Attribuutteina ovat asiakkaan asiakasnumero, nimi, syntymävuosi, bonuspisteet, osoite ja sähköpostiosoite.
- Relaatioon `Products` on kerätty verkkokaupan tuotteiden tietoja. Attribuutteina ovat tuotenumero, tuotteen nimi, kuvaus (esim. 'camera'), hinta ja valmistajan yksilöivä tunnus.
- Relaatio `Manufacturers` sisältää valmistajien tietoja. Attribuutteina ovat valmistajan tunnus, nimi ja puhelinnumero.
- Relaatiossa `Orders` säilytetään tietoa tehdyistä tilauksista. Yksi tilaus voi sisältää monta tuotetta. Attribuutteina ovat tilausnumero, toimitustapa, tilauksen tila ja tilauksen tehneen asiakkaan asiakasnumero.
- Relaatio `BelongsTo` sisältää tiedon siitä, mikä tuote kuuluu mihinkin tilaukseen. Attribuutteina ovat tilauksen numero, yhden tilaukseen kuuluvan tuotteen numero ja tieto siitä, kuinka monta kappaletta ko. tuotetta sisältyy tilaukseen.

Yksinkertainen SQL-kysely

- Tarkastellaan ensimmäisenä kyselyä

```
SELECT description, price
FROM Products;
```

- Tulosrelaatiossa on samat tuotteet kuin alkuperäisessä relaatiossa, mutta niistä esitetään vain kaksi attribuuttia:

Result

description	*price*
cellphone	250.0
tablet	495.0
jacket	30.0
pan	33.50

Ehto mukaan SQL-kyselyyn

- Haetaan edellisestä `Products`-taulusta niiden tuotteiden nimet ja kuvaukset, joiden hinta on alle 35 euroa:

```
SELECT prodName, description
FROM Products
WHERE price < 35.0;
```

- Kysely siis käy läpi kaikki `Products`-relaation monikot, valitsee niistä annetun ehdon toteuttavat monikot ja tulostaa näistä SELECT-osassa luetellut attribuutit. (Kyselyä ei välttämättä suoriteta juuri tällä tavalla, mutta lopputulos on sama.)
- Tulos on nyt

Result

prodName	*description*
Superstar Track M	jacket
Brasserie 24	pan

Välitehtävä 1

- Kirjoita seuraava kysely SQL:llä: hae kaikkien niiden asiakkaiden asiakasnumerot ja nimet, joilla on vähintään 30 bonuspistettä.

Välitehtävä 1, ratkaisu

- *Hae kaikkien niiden asiakkaiden asiakasnumerot ja nimet, joilla on vähintään 30 bonuspistettä.*

```
SELECT custNo, name
FROM Customers
WHERE bonus >= 30;
```

Toinen esimerkki valinnasta

- Haetaan relaatiosta `Products` niiden tuotteiden nimet ja hinnat, joiden kuvaus on cellphone tai tablet ja joiden hinta on alle 400 euroa.

```
SELECT prodName, price
FROM Products
WHERE (description = 'cellphone' OR description = 'tablet') AND
          price < 400.0;
```

- Tulos

Result

prodName	*price*
Samsung Galaxy A5	250.0

Liitokset

- Liitos voidaan toteuttaa helposti kirjoittamalla kyselyn WHERE-osaan ehto, joka koskee kahden eri relaation attribuutteja
- Esimerkki: oletetaan, että relaation `Manufacturers` sisältö on seuraava:

Relation `Manufactures`

ID	*manufName*	*phone*
R122	Nike	09-8011
M554	Apple	09-5001
S123	Samsung	020-7300
L711	Sony	020-6500
F542	Fiskars	020-43910
A432	Adidas	09-70911

- Kirjoitetaan kysely "tulosta tietokannassa olevien tuotteiden nimet yhdessä kunkin tuotteen valmistajan nimen kanssa."

Liitokset, jatkuu

- *Tulosta tietokannassa olevien tuotteiden nimet yhdessä kunkin tuotteen valmistajan nimen kanssa.*

  SELECT prodName, manufName
  FROM Products, Manufacturers
  WHERE manufID = ID;

- Kysely voidaan ajatella suoritettavaksi seuraavasti:
  1. Muodostetaan FROM-osassa olevien relaatioiden karteesinen tulo: kaikki mahdolliset monikot, joiden ensimmäisenä osana on jokin relaation `Products` monikko ja toisena osana jokin relaation `Manufactures` monikko.
  2. Valitaan karteesisen tulon monikoista ne, jotka toteuttavat WHERE-osassa olevan ehdon.
  3. Otetaan tulokseen mukaan valituista monikoista vain ne attribuutit, jotka on lueteltu SELECT-osassa.
- Oikeasti kyselyä ei välttämättä suoriteta näin. Lopputulos on kuitenkin sama, ja SQL-kyselyn voi ymmärtää helposti ajattelemalla sitä näin.

Liitokset, jatkuu

- Esimerkki: muodostetaan karteesinen tulo:

number	*prodName*	*description*	*price*	*manufID*	*ID*	*manufName*	*phone*
T-33441	Samsung Galaxy A5	cellphone	250.0	S123	R122	Nike	09-8011
T-33441	Samsung Galaxy A5	cellphone	250.0	S123	M554	Apple	09-5001
T-33441	Samsung Galaxy A5	cellphone	250.0	S123	S123	Samsung	020-7300
T-33441	Samsung Galaxy A5	cellphone	250.0	S123	L711	Sony	020-6500
T-33441	Samsung Galaxy A5	cellphone	250.0	S123	F542	Fiskars	020-43910
T-33441	Samsung Galaxy A5	cellphone	250.0	S123	A432	Adidas	09-70911
R-55336	IPad Air 2	tablet	495.0	M554	R122	Nike	09-8011
R-55336	IPad Air 2	tablet	495.0	M554	M554	Apple	09-5001
R-55336	IPad Air 2	tablet	495.0	M554	S123	Samsung	020-7300
R-55336	IPad Air 2	tablet	495.0	M554	L711	Sony	020-6500
R-55336	IPad Air 2	tablet	495.0	M554	F542	Fiskars	020-43910
R-55336	IPad Air 2	tablet	495.0	M554	A432	Adidas	09-70911
T-77445	Superstar Track M	jacket	30.0	A432	R122	Nike	09-8011
T-77445	Superstar Track M	jacket	30.0	A432	M554	Apple	09-5001
T-77445	Superstar Track M	jacket	30.0	A432	S123	Samsung	020-7300
T-77445	Superstar Track M	jacket	30.0	A432	L711	Sony	020-6500
T-77445	Superstar Track M	jacket	30.0	A432	F542	Fiskars	020-43910
T-77445	Superstar Track M	jacket	30.0	A432	A432	Adidas	09-70911
S-65221	Brasserie 24	pan	33.50	F542	R122	Nike	09-8011
S-65221	Brasserie 24	pan	33.50	F542	M554	Apple	09-5001
S-65221	Brasserie 24	pan	33.50	F542	S123	Samsung	020-7300
S-65221	Brasserie 24	pan	33.50	F542	L711	Sony	020-6500
S-65221	Brasserie 24	pan	33.50	F542	F542	Fiskars	020-43910
S-65221	Brasserie 24	pan	33.50	F542	A432	Adidas	09-70911

Liitokset, jatkuu

- Valitaan ehdon täyttävät monikot:

number	*prodName*	*description*	*price*	*manufID*	*ID*	*manufName*	*phone*
T-33441	Samsung Galaxy A5	cellphone	250.0	S123	R122	Nike	09-8011
T-33441	Samsung Galaxy A5	cellphone	250.0	S123	M554	Apple	09-5001
T-33441	Samsung Galaxy A5	cellphone	250.0	S123	S123	Samsung	020-7300
T-33441	Samsung Galaxy A5	cellphone	250.0	S123	L711	Sony	020-6500
T-33441	Samsung Galaxy A5	cellphone	250.0	S123	F542	Fiskars	020-43910
T-33441	Samsung Galaxy A5	cellphone	250.0	S123	A432	Adidas	09-70911
R-55336	IPad Air 2	tablet	495.0	M554	R122	Nike	09-8011
R-55336	IPad Air 2	tablet	495.0	M554	M554	Apple	09-5001
R-55336	IPad Air 2	tablet	495.0	M554	S123	Samsung	020-7300
R-55336	IPad Air 2	tablet	495.0	M554	L711	Sony	020-6500
R-55336	IPad Air 2	tablet	495.0	M554	F542	Fiskars	020-43910
R-55336	IPad Air 2	tablet	495.0	M554	A432	Adidas	09-70911
T-77445	Superstar Track M	jacket	30.0	A432	R122	Nike	09-8011
T-77445	Superstar Track M	jacket	30.0	A432	M554	Apple	09-5001
T-77445	Superstar Track M	jacket	30.0	A432	S123	Samsung	020-7300
T-77445	Superstar Track M	jacket	30.0	A432	L711	Sony	020-6500
T-77445	Superstar Track M	jacket	30.0	A432	F542	Fiskars	020-43910
T-77445	Superstar Track M	jacket	30.0	A432	A432	Adidas	09-70911
S-65221	Brasserie 24	pan	33.50	F542	R122	Nike	09-8011
S-65221	Brasserie 24	pan	33.50	F542	M554	Apple	09-5001
S-65221	Brasserie 24	pan	33.50	F542	S123	Samsung	020-7300
S-65221	Brasserie 24	pan	33.50	F542	L711	Sony	020-6500
S-65221	Brasserie 24	pan	33.50	F542	F542	Fiskars	020-43910
S-65221	Brasserie 24	pan	33.50	F542	A432	Adidas	09-70911

Liitokset, jatkuu

- Valitaan ehdon täyttävistä monikoista pyydetyt attribuutit:

number	*prodName*	*description*	*price*	*manufID*	*ID*	*manufName*	*phone*
T-33441	Samsung Galaxy A5	cellphone	250.0	S123	S123	Samsung	020-7300
R-55336	IPad Air 2	tablet	495.0	M554	M554	Apple	09-5001
T-77445	Superstar Track M	jacket	30.0	A432	A432	Adidas	09-70911
S-65221	Brasserie 24	pan	33.50	F542	F542	Fiskars	020-43910

Result

prodName	*manufName*
Samsung Galaxy A5	Samsung
IPad Air 2	Apple
Superstar Track M	Adidas
Brasserie 24	Fiskars

Miten lähden tekemään SQL-kyselyä?

1. Mieti, mistä relaatioista tarvitset tietoa, jotta saisit halutun vastauksen.
   - Kirjoita näiden relaatioiden nimet FROM-osaan. Ota mukaan vain ne relaatiot, jotka ovat välttämättömiä.
2. Kuvittele, että sinulla on näiden relaatioiden monikoista kaikki mahdolliset yhdistelmät. Mitkä yhdistelmät otetaan mukaan vastaukseen?
   - Kirjoita tätä vastaava(t) ehto/ehdot WHERE-osaan. Jos ehtoja on useampi kuin yksi, tarvitset loogisia operaattoreita AND tai OR.
3. Mitä muita ehtoja kysely vaatii toteutettavaksi?
   - Lisää tätä vastaava ehto/ehdot WHERE-osaan. Yhdistä ehdot tilanteeseen sopivilla loogisilla operaattoreilla.
4. Mitä attribuutteja vastaukseen otetaan mukaan?
   - Kirjoita näiden attribuuttien nimet SELECT-osaan.

Toinen esimerkki liitoksesta

- Kysely: "tulosta niiden valmistajien nimet, joilla on tarjolla alle 300 euroa maksavia tuotteita, joiden kuvaus on cellphone".
- Kysely SQL:llä:

```
SELECT manufName
FROM Products, Manufacturers
WHERE manufID = ID AND description = 'cellphone'
            AND price < 300;
```

- Tulos esimerkkirelaatioilla:

Result

manufName
Samsung

Duplikaattien poistaminen

- Toisin kuin relaatioalgebrassa, SQL:ssä tulosrelaatiot eivät ole joukkoja (set), vaan monijoukkoja (bag), eli täsmälleen sama monikko voi esiintyä tuloksessa monta kertaa.
- Oletetaan, että relaatio `Customers` on seuraava

Relation `Customers`

number	*name*	*born*	*bonus*	*address*	*email*
112233	Teemu Teekkari	1995	55	Servinkuja 3	tteekkari@gmail.com
554422	Riina Raksalainen	1993	43	Otaranta 8	riinar@yahoo.com
37856	Antti Virta	1970	12	Aaltokatu 4	antti@hotmail.com
77233	Nina Teekkari	1991	20	Servinkuja 3	nite@hotmail.com

Duplikaattien poistaminen, jatkoa

- Suoritetaan sille kysely

```
SELECT address
FROM Customers
WHERE bonus > 15;
```

- Tulos on

Result

address
Servinkuja 3
Otaranta 8
Servinkuja 3

Duplikaattien poistaminen, jatkoa

- Jos duplikaatit halutaan poistaa, on SELECT-osaan lisättävä määre DISTINCT:

```
SELECT DISTINCT address
FROM Customers
WHERE bonus > 15;
```

- Tämän kyselyn vastaus on

Result

address
Servinkuja 3
Otaranta 8

Attribuuttien yksiselitteisyys

- Eri relaatioissa voi olla samannimisiä attribuutteja. Jos kysely kohdistuu tällaisiin relaatioihin, pitää yksikäsitteisesti määrätä, minkä relaation attribuutteja tarkoitetaan.
- Samannimiset attribuutit erotetaan toisistaan pistenotaatiolla: attribuutin alkuun liitetään sen relaation nimi. Esim. $R.A$ tarkoittaa relaation R attribuuttia A.
- Esimerkki: Halutaan löytää relaatioista

  `Customers(custNo, name, born, bonus, address, email)`
  `Orders(orderNo, deliver, status, custNo)`

  niiden asiakkaiden asiakasnumerot ja nimet, joilla on tilaus, jonka status on returned.

Attribuuttien yksiselitteisyys, jatkoa

- Kysely:

```
SELECT Customers.custNo, name
FROM Customers, Orders
WHERE Customers.custNo = Orders.custNo AND
          status = 'returned';
```

Rivimuuttujat

- Pistenotaation käyttö yksistään ei riitä, jos kyselyssä pitää verrata saman relaation rivejä keskenään.
- Esimerkki: halutaan tulostaa asiakkaat, jotka asuvat keskenään samassa osoitteessa. Tällöin on verrattava kahta saman `Customers`-relaation riviä keskenään.
- SQL:ssä voidaan käyttää rivimuuttujia (engl. tuple variables) viittaamaan relaatioihin relaatioalgebran uudelleennimeämisen tapaan. Samalle relaatiolle voi määritellä useita rivimuuttujia
- Rivimuuttujat määritellään FROM-osassa käyttämällä syntaksia

  FROM <relaation nimi> AS <rivimuuttujan nimi>

Rivimuuttujat, jatkoa

- Tulosta asiakkaat, jotka asuvat samassa osoitteessa keskenään.

```
SELECT C1.name, C2.name
FROM Customers AS C1, Customers AS C2
WHERE C1.address = C2.address AND C1.custNo < C2.custNo;
```

- Rivimuuttujan voi määritellä myös ilman AS-ilmausta:

```
SELECT C1.name, C2.name
FROM Customers C1, Customers C2
WHERE C1.address = C2.address AND C1.custNo < C2.custNo;
```

- Rivimuuttujia voidaan käyttää myös vähentämään kirjoitustyötä kyselyissä silloinkin, kun niiden käyttö ei ole varsinaisesti välttämätöntä.

Esimerkki kolmen relaation liitoksesta

- Halutaan niiden tilausten numerot ja tilat, joihin kuuluu tuote, jonka kuvaus on cellphone.
- Kysely

```
SELECT O.orderNo, status
FROM Orders AS O, Products AS P, BelongsTo AS B
WHERE O.orderNo = B.orderNo AND number = productNo AND
        description = 'cellphone';
```

Tulosrelaation rivien esitysjärjestyksen määrääminen

- Oletusarvoisesti kyselyn tulosrelaation rivit esitetään satunnaisessa järjestyksessä.
- Järjestys voi myös vaihdella kyselyn eri suorituskertojen välillä.
- Jos tulosrelaation rivit halutaan järjestää jonkin attribuutin mukaan, saadaan se aikaan lisäämällä kyselyyn määre

  ORDER BY <attribuuttilista>

- Tällöin tulosrelaatio järjestetään ensisijaisesti listan 1. attribuutin mukaan, toisijaisesti listan 2. attribuutin mukaan jne.
- Järjestys on oletusarvoisesti kasvava, mutta sen voi vaihtaa laskevaksi lisäämällä attribuutin jälkeen sanan DESC.

Esimerkki järjestyksen määräämisestä

- Tulostetaan asiakkaat, joilla on yli 30 bonuspistettä, bonuspisteiden määräämässä järjestyksessä (suurin ensin). Jos useammalla asiakkaalla on sama määrä bonuspisteitä, tulostetaan nämä syntymävuoden määräämässä järjestyksessä (pienin ensin).
- SELECT-osassa käytetty $*$ valitsee kyselyyn mukaan kaikki relaation attribuutit
- Kysely

```
SELECT *
FROM Customers
WHERE bonus > 30
ORDER BY bonus DESC, born;
```

WHERE-osasssa käytettävistä operaattoreista

- WHERE-osa voi sisältää vertailuoperaattoreita =, <> (erisuuruus), <, >, <= ja >=.
- Vertailtavat arvot voivat olla vakioita tai kyselyn FROM-osassa määriteltyjen relaatioiden attribuutteja.
- Vertailussa voidaan käyttää myös aritmeettisia operaattoreita +, -, * ja /. Esimerkiksi ehto

  ```
  (born - 1995) * (born -1995) < 100
  ```

  on tosi vuosille 1986–2004.
- Mahdollisia loogisia operaattoreita ovat AND, OR ja NOT.

Välitehtävä 2

- Kirjoita seuraava kysely: tulosta niiden asiakkaiden asiakasnumerot, joilla on useampi kuin yksi tilaus.
- Kyselyn pitää tulostaa asiakasnumerot järjestyksessä.

Välitehtävä 2, ratkaisu

- *Tulosta niiden asiakkaiden asiakasnumerot, joilla on useampi kuin yksi tilaus. Kyselyn pitää tulostaa asiakasnumerot järjestyksessä.*

```
SELECT O1.custNo
FROM Orders AS O1, Orders AS O2
WHERE O1.custNo = O2.custNo AND O1.orderNo <> O2.orderNo
ORDER BY O1.custNo;
```

- Vastauksessa voi olla duplikaatteja. Ne voidaan haluttaessa poistaa DISTINCT-ilmauksella.

Merkkijonojen vertailusta

- Myös merkkijonojen vertailuun voidaan käyttää operaattoreita =, <> (erisuuruus), <, >, <= ja >=. Tällöin tutkitaan merkkijonojen leksikografista järjestystä.
- Sen lisäksi merkkijonojen samankaltaisuutta voidaan tutkia LIKE-vertailuoperaattorin avulla.
- Vertailulauseke on muotoa

  s LIKE p

  s on merkkijono ja p rakennekaava (pattern), joka voi sisältää tavallisten merkkien lisäksi erikoismerkkejä % ja _.
- p:ssä esiintyvä % ilmaisee, että vastaavalla paikalla s:ssä voi esiintyä mikä tahansa nollan tai useamman merkin pituinen merkkijono.
- p:ssä esiintyvä _ ilmaisee, että vastaavalla paikalla s:ssä voi esiintyä mikä tahansa yksittäinen merkki.
- Lauseke s LIKE p on tosi, jos ja vain jos s toteuttaa rakennekaavan p.

Esimerkkejä merkkijonojen vertailuista

- Kysely

  ```
  SELECT prodName, description
  FROM Products
  WHERE description LIKE 'cell_____';
  ```

  etsii tuotteet, joiden kuvaus on yhdeksän merkin mittainen siten, että neljä ensimmäistä merkkiä ovat cell ja niiden jälkeen tulee täsmälleen viisi merkkiä, jotka voivat olla mitä tahansa.

- Kysely

  ```
  SELECT prodName, description
  FROM Products
  WHERE prodName LIKE '%Galaxy%';
  ```

  etsii tuotteet, joiden nimessä esiintyy jossain kohdassa peräkkäin kirjaimet Galaxy.

Aritmeettinen lauseke SELECT-osassa

- Esimerkki: halutaan tuotteiden nimet ja hinnat niin, että hinnat on muutettu dollareiksi kertomalla ne 1.2:llä
- Tällöin SELECT-osaan voidaan kirjoittaa kertolasku, jolla eurot voidaan muuttaa dollareiksi:

  SELECT prodName, price $*$ 1.2
  FROM Products;

  Tuloksen otsikko on tällöin

prodname	*price* * 1.2

- Jos haluaa kauniimman otsikon, kannattaa lauseke nimetä SELECT-osassa uudelleen:

  SELECT prodName, price $*$ 1.2 AS dollarprice
  FROM Products;

Yhdiste

- SQL-kyselyssä on mahdollista suorittaa joukko-operaatioita yhdiste, leikkaus ja erotus kyselyiden tuloksina syntyviin relaatioihin.
- Toisin kuin muissa SQL-kyselyissä, joukko-operaatioiden tulokset ovat aitoja joukkoja eli duplikaatit on poistettu. (Duplikaattien poiston voi kuitenkin estää kirjoittamalla kyselyyn sanan ALL joukko-operaation nimen jälkeen.)
- Esimerkki: halutaan esimerkkitietokannasta sekä kaikkien niiden asiakkaiden numerot, joiden tilauksen status on 'waiting', että kaikkien niiden asiakkaiden numerot, joiden osoitteessa esiintyy sana Servinkuja.
- ```
  SELECT CustNo
  FROM Orders
  WHERE status = 'waiting'
      UNION
  SELECT CustNo
  FROM Customers
  WHERE address LIKE '%Servinkuja%';
  ```
  ```

Leikkaus

- Esimerkki: halutaan niiden valmistajien nimet, joilta verkkokaupassa on tarjolla sekä matkapuhelimia että kameroita.

```
SELECT manufName
FROM Manufacturers, Products
WHERE manufID = ID and description = 'camera'
    INTERSECT
SELECT manufName
FROM Manufacturers, Products
WHERE manufID = ID and description = 'cellphone';
```

Erotus

- Joukkojen erotus esitetään SQL:ssä operaation EXCEPT avulla.
- Esimerkki: halutaan niiden valmistajien nimet, joilta verkkokaupassa on tarjolla kameroita, mutta ei matkapuhelimia.

```
SELECT manufName
FROM Manufacturers, Products
WHERE manufID = ID and description = 'camera'
    EXCEPT
SELECT manufName
FROM Manufacturers, Products
WHERE manufID = ID and description = 'cellphone';
```

Attribuuttien nimeäminen uudelleen joukko-operaatioiden yhteydessä

- Esimerkki: halutaan esimerkkitietokannasta sekä kaikkien A:lla alkavien asiakkaiden nimet että kaikki A:lla alkavien valmistajien nimet.
- Ongelma: `Customers`-relaatiossa nimeä kuvaavan attribuutin nimi on *name*, mutta `Manufacturers`-relaatiossa nimeä kuvaavan attribuutin nimi on *manufName*. Halutaan, että tulosrelaatiossa attribuutilla on sama nimi.
- Ongelma voidaan ratkaista nimeällä toisen kyselyn tuloksessa attribuutti uudelleen:

```
SELECT name
FROM Customers
WHERE name LIKE 'A%'
    UNION
SELECT manufName AS name
FROM Manufacturers
WHERE name LIKE 'A%';
```