

Tietokantojen perusteet, K 2004

SQL tietokantakieli

- SQL:llä voidaan...
 - määritellä ja muokata tietokantaa ja sen käyttöoikeuksia
 - virittää tietokannan talletusrakenteita
 - hakea tietoa tietokannasta
 - näytölle tai tiedostoon
 - sovellusohjelman käyttöön
 - tehdä päivityksiä tietokantaan (muuttaa dataa)
 - vuorovaikutteisesti
 - sovellusohjelman kautta

Tietokantojen perusteet, K 2004

SQL

- SQL on standardoitu
- viimeisin standardi vuodelta 1999
- toteutukset noudattavat enimmäkseen vuoden -92 standardia ja sitäkin vain osittain
- murteita - yhteinen suppeahko ydin

Tietokantojen perusteet, K 2004

SQL-tietokanta

- SQL-tietokanta muodostuu yhden tai useamman kaavion (schema) määrittelemistä tauluista (table)
- Kullakin kaaviolla on omistaja, joka omistaa myös kaavion määrittelemät taulut. Taulu muodostuu riveistä (row)
- Taulu vastaa relaatiomallin relaatiota, mutta
 - sallii etenkin kyselyiden tuloksissa samanlaisen rivin toistumisen (duplikaatit)matemattisesti monijoukko (multiset)

Tietokantojen perusteet, K 2004

SQL

```
graph TD; SQL[SQL] --- DDL[määrittelykieli (DDL)]; SQL --- DML[Käsittelykieli (DML)];
```

käyttäjät ja oikeudet
tietokannan rakenne
tietokantaproseduurit

kyselyt
ylläpito-operaatiot

Tietokantojen perusteet, K 2004

SQL

- SQL-kielessä avainsanat, taulu- käyttäjä- ja sarakenimet voi kirjoittaa joko suur- tai pienaakkosina eli
select merkki ≡ SELECT MerKKI
- Tietokannassa olevan datan suhteen kieli on kuitenkin herkkä kirjainmuodolle eli
 - Merkki='Ford' on eri kuin Merkki='FORD'
 - joissain järjestelmissä tätä käyttäytymistä voidaan säätää

Tietokantojen perusteet, K 2004

SQL tiedonmäärittelykieli

- Tiedonmäärittelykielessä lauseita tietokantaelementtien {user, role, schema, table, domain, procedure, function, trigger, ...} luontiin, muokkaukseen ja poistoon
 - create -luo
 - alter - muokkaa
 - drop - poistaa

Tietokantojen perusteet, K 2004

SQL taulun luonti

- create table määrittelee taulun rakenteen
- create table *tablename* (
column definition 1, ...,
column definition n
[, *constraint 1*, ...])

sarakemäärittely ::=
column_name datatype [not null]
[default *value*] [*column constraint ...*]

Tietokantojen perusteet, K 2004

Taulun määrittely

```
create table Ordered (
  OrderId      integer not null,
  WhenMade     date not null,
  Customer     integer not null,
  WayIssued    varchar(20),
  PaymentBy    varchar(20) not null,
  TotalPrice   decimal(6,2) not null,
  constraint pk_order primary key (OrderId),
  constraint fk_ordercustomer foreign key
    (Customer) references Customer
);
```

Tietokantojen perusteet, K 2004

Taulun määrittely

```
create table Ordered (
  OrderId      integer not null,
  WhenMade     date not null,
  Customer     integer not null,
  WayIssued    varchar(20),
  PaymentBy    varchar(20) not null,
  TotalPrice   decimal(6,2) not null,
  constraint pk_order primary key (OrderId),
  constraint fk_ordercustomer foreign key
    (Customer) references Customer
);
```

Annotations:

- OrderId: päiväys (tässä on kyseessä Oracle, joten timestamp)
- Customer: vaihtuvamittainen merkkijono
- TotalPrice: kokonaisluku
- TotalPrice: desimaaliluku kokonaispituus 6, desimaaliosia 2

Tietokantojen perusteet, K 2004

Taulun määrittely

```
create table Ordered (
  OrderId      integer not null,
  WhenMade     date not null,
  Customer     integer not null,
  WayIssued    varchar(20),
  PaymentBy    varchar(20) not null,
  TotalPrice   decimal(6,2) not null,
  constraint pk_order primary key (OrderId),
  constraint fk_ordercustomer foreign key
    (Customer) references Customer
);
```

Annotations:

- OrderId: avain
- CustomerId: viiteavain
- WayIssued, PaymentBy, TotalPrice: pakollinen tieto

Tietokantojen perusteet, K 2004

SQL tiedonmäärittely

- Aikoja
 - Date päiväys
 - Time kellonaika
 - Timestamp päiväys ja kellonaika (Oraclella Date on oikeastaan Timestamp)
 - Interval aikaero
- Aikoja voidaan verrata ja niillä voi laskea
this_day date,
this_day + 3 on kolmen päivän päästä

Tietokantojen perusteet, K 2004

SQL tiedonmäärittely

- Viiteavainmäärittelyyn voidaan liittää toimintasääntö, mitä tehdään operaation rikkoessa viite-ehyden
foreign key (*sarakkeet*) references *taulu* [(*sarakkeet2*)]
[on delete {restrict | cascade | nullify}]
[on update {restrict | cascade | nullify}]
- Kun viitteen kohde katoaa:
restrict estää rikkovan operaation (oletusarvo)
cascade vyöryttää= poistaa tai muuttaa viittaavat rivit
nullify tyhjentää viittaukset
valinta sen perusteella mikä tarkoituksenmukaista

Tietokantojen perusteet, K 2004

SQL -kysely

Kyselyn yleisrakenne:

```
select tulostietomäärittely
from taulut
[where valintaehdot]
[group by ryhmitystekijät]
[having ryhmärajoitteet]
[order by järjestyserusta]
```

Kysely tuottaa nimettömän tulostaulun.

Tietokantojen perusteet, K 2004

SQL-kysely

```
select merkki, reknro
from auto
where vmalli=1996 and
      vari ='punainen' and merkki like 'Fo%'
order by merkki, reknro
```

- Vuoden 1996 mallia olevien punaisten merkiltään 'Fo'-alkuisten autojen merkki ja rekisterinumero merkin ja saman merkin sisällä rekisterinumeron mukaan järjestettynä

Tietokantojen perusteet, K 2004

SQL -kysely

```
select merkki, reknro
from auto
where vmalli=1996 and
      vari ='punainen' and merkki like 'Fo%'
order by merkki, reknro
```

melkein projektio valinta

- Vuoden 1996 mallia olevien punaisten merkiltään 'Fo'-alkuisten autojen merkki ja rekisterinumero merkin ja saman merkin sisällä rekisterinumeron mukaan järjestettynä

Tietokantojen perusteet, K 2004

SQL-kysely

- Tulostietomäärittelyn elementeille lasketaan normaalitapauksessa arvo jokaista valintaehdot täyttävää riviyhdistelmää kohden

```
select merkki
from auto
where vmalli=1996 and
      vari ='punainen' and merkki like 'Fo%'
order by merkki
```

Jos taulussa auto olisi 100 punaista vuoden 1996 Fordia tulisi merkki 'Ford' tulostauluun 100 kertaa.

Toimii siis toisin kuin relaatioalgebran projektio, joka poistaa tuplat

Tietokantojen perusteet, K 2004

SQL -kysely

- Projektion kaltainen toistuvien arvojen karsinta saadaan aikaan liittämällä tulostietomäärittelyn alkuun avainsana **distinct**

```
select distinct merkki
from auto
where vmalli=1996 and
      vari ='punainen' and merkki like 'Fo%'
order by merkki
```

Nyt Ford tulisi tulokseen vain kerran

Tietokantojen perusteet, K 2004

SQL-kysely

- Kyselyn ehto-osassa voidaan verrata saraketta, vakiota tai funktion arvoa
 - sarakeeseen, vakioon, funktion arvoon
 - arvojoukkoon
 - maskiin
- Voidaan myös tutkia sarakkeen tyhjyyttä
- Jos vertailun toisena osapuolena on tyhjäarvo on tulos 'tuntematon'. Rivi tulee valituksi tulokseen vain jos ehdon arvo on 'tosi' (true).

Tietokantojen perusteet, K 2004

SQL -kyselyt

- Totuusarvot tosi (true) ja epätosi (false) käyttäytyvät loogisissa lausekkeissa kuten ohjelmointikielten yhteydessä
- Kolmas totuusarvo 'tuntematon' käyttäytyy seuraavasti

AND	true	false	unknown	NOT	
true	true	false	unknown	true	false
false	false	false	false	false	true
unknown	unknown	false	unknown	unknown	unknown

Tietokantojen perusteet, K 2004

SQL-kyselyt

OR	true	false	unknown
true	true	true	true
false	true	false	unknown
unknown	true	unknown	unknown

Sarake is null: tuottaa tuloksen true, jos sarakkeessa on tyhjäarvo, muuten false

Sarake is not null: tuottaa tuloksen false, jos sarakkeessa on tyhjäarvo, muuten true

Tietokantojen perusteet, K 2004

SQL -kyselyt

- Tietotyyppistä riippuen sarakearvoilla voi tulostietomäärittelyissä ja ehdoissa suorittaa laskentaoperaatioita (merkkijonoille katenaatio || (perään laittaminen)
 - pituus*paino, etunimi||' '||sukunimi
- Merkkijonovakiot 'hipsuihin', numeeriset ilman
- On mahdollista käyttää myös skalaarifunktioita – esim. merkkijonon pituus
- length(Sarake), funktiovalikoima vaihtelee

Tietokantojen perusteet, K 2004

SQL -kyselyt

- Kyselyn from -osassa yksi tai useampi taulu (tai alikysely)
- Jos tauluja on vain yksi, on kyseessä valinta
- Jos tauluja on monta, on kyseessä ristitulo ellei ehto-osassa ole liitosehtoa (hyvin harvoin halutaan tulokseksi ristitulo)
- Jos tauluja on monta ja ehto-osassa on liitosehto, on kyseessä liitos – muista siis liitosehto

Tietokantojen perusteet, K 2004

SQL- kyselyt esimerkikanta

```

    erDiagram
        opettaja ||--o{ harjoitusryhma : "opettaja"
        harjoitusryhma ||--o{ Kurssi : "kurssikoodi"
        ilmoittautuminen ||--o{ harjoitusryhma : "kurssikoodi"
        ilmoittautuminen ||--o{ harjoitusryhma : "ryhmanro"
        ilmoittautuminen ||--o{ harjoitusryhma : "Opisknro"
        ilmoittautuminen ||--o{ harjoitusryhma : "ilm_aika"
        ilmoittautuminen ||--o{ opiskelija : "Opisknro"
        ilmoittautuminen ||--o{ opiskelija : "aloitusvuosi"
    
```

Tietokantojen perusteet, K 2004

SQL -kyselyt

- Opettajien nimet:
select nimi from opettaja;
- Opiskelijoiden pääaineet
select distinct paa_aine from opiskelija;
- Tietojenkäsittelytieteen pääaineopiskelijoiden nimet
select nimi from opiskelija where paa_aine = 'TKT';
- Essoossa asuvat matematiikan opiskelijat
select * from opiskelija where paa_aine='MAT' and kaupunki='Espoo';

Tietokantojen perusteet, K 2004

SQL -kyselyt

- Opiskelijat joiden sukunimi on Tele
 - select * from opiskelija where nimi like 'Tele %';
 - (esimerkkitaulussa nimet ovat muodossa sukunimi+'space'+etunimi)
- Opiskelijat, joiden etunimi alkaa L:llä
 - select * from opiskelija where nimi like '% L%';

Tietokantojen perusteet, K 2004

Liitokset SQL:ssä

- Kyselyn from-osassa voi olla useita tauluja
- Kaikki ne taulut, joiden dataa halutaan mukaan tulokseen on annettava from-osassa
- Arto Wiklan luennoimat kurssit


```

            Select kurssi.nimi
            from kurssi, opettaja
            where opettaja.nimi='Arto Wikla' and
            kurssi.luennoija=opettaja.opetunnus;
            
```

täytyy täsmentää koska sama sarake kahdessa taulussa

liitosehto

Tietokantojen perusteet, K 2004

Liitokset SQL:ssä

- Tauluille voidaan from-osassa antaa tilapäinen kyselyn sisäinen nimi (alias, correlation name)
 - from taulu [AS] alias
 - liitettävillä tauluilla on usein samannimisiä sarakkeita, joten taulunimeä on käytettävä tarkenteena - alias voi olla lyhenne, joka vähentää kirjoitusvaivaa

as ei Oraclessa

Tietokantojen perusteet, K 2004

Liitokset SQL:ssä

- Jos sama taulu esiintyy from -osassa useaan kertaan, on taulun esiintymät erotettava käyttämällä aliasta
- Esim.: Kurssiparit, joilla on sama luennoija


```

            select A.nimi, B.nimi
            from kurssi A, kurssi B
            where A.luennoija=B.luennoija and A.koodi<B.koodi
            order by A.nimi, B.nimi
            
```
- ehto A.koodi<B.koodi estää saman parin toistumisen eri järjestyksessä

Tietokantojen perusteet, K 2004

Liitokset SQL:ssä

- Tyypillinen virhe liitoksissa on jättää jokin liitosehto pois, jolloin tuloksen rivijoukko tulee huomattavasti suuremmaksi kuin pitäisi
- jos from-osassa on n kpl liitettäviä tauluja tarvitaan vähintään n-1 liitosehtoa. Taulujen liittäminen voi perustua useaan sarakkeeseen, jolloin ehtolausekkeessa tarvittavien alkeisehtojen määrä voi moninkertaistua.

Tietokantojen perusteet, K 2004

Liitokset SQL:ssä

- Yleensä kyselyt rakentuvat siten, että niissä on jokin keskeinen taulu, johon muita liitetään. Voi olla, ettei tuosta keskeisestä taulusta tule mitään dataa tulokseksi.

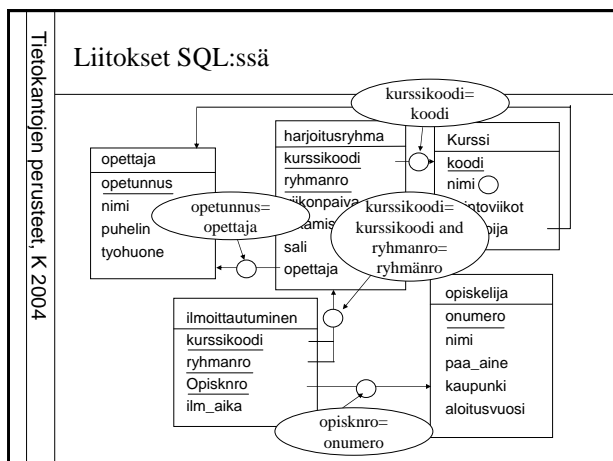
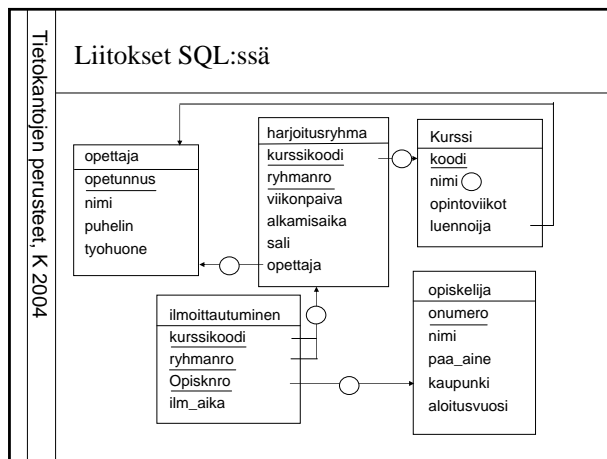
Keskeinen taulu

liitokset

Tietokantojen perusteet, K 2004

Liitokset SQL:ssä

- Laadi raportti kurssin Java-ohjelmointi harjoitusryhmistä
- Mitä halutaan tulokseen:
 - Ryhmän numero (taulussa harjoitusryhmä)
 - Ohjaajan nimi (taulussa opettaja)
 - kokoontumispäivä (taulussa harjoitusryhmä)
 - alkamisaika (taulussa harjoitusryhmä)
 - opiskelijan nimi (taulussa opiskelija)
- Taulut opettaja, harjoitusryhmä ja opiskelija on välttämättä otettava kyselyn from osaan
- Taulu ilmoittautuminen tarvitaan opiskelijoiden kytkemiseksi ryhmiin ja taulu kurssi, jotta saataisiin selville Java ohjelmoinnin kurssikoodi



Tietokantojen perusteet, K 2004

Liitokset SQL:ssä

```

select H.ryhmanro mo, Ope.nimi ope, H.viikonpaiva,
H.alkamisaika, O. Nimi opiskelija
from Harjoitusryhma H, opettaja Ope, opiskelija O,
ilmoittautuminen I, kurssi K
where
H.kurssikoodi=K.koodi and
I.kurssikoodi=H.kurssikoodi and
I.ryhmanro=H.ryhmanro and
Ope.Opetunnus=H. Opettaja and
I.Opisknro=O.onumero and K.nimi='Java ohjelmointi'
order by H.ryhmanro, O.nimi;
    
```