

SQL tietokantakieli

- n Relatiomalliin pohjautuvat tietokannat
 - n Perustana relatiomallin mukainen näkemys tietojen rakenteesta ja käsittelystä
 - n Tietokantojen rakenteen määrittelyssä ja käsittelymahdollisuuksissa on paljon piirteitä, joihin abstrakti relatiomalli ei ota kantaa, esim.
 - n Miten tietokantakaavio käytännössä esitetään
 - n Millaista tietoa tietokantaan voi tallettaa
 - n Miten esitetään tietokantaan kohdistuvat kyselyt
 - n Miten tietokantaa ylläpidetään
 - n Miten hallitaan tietokannan käyttöoikeuksia
 - n Miten hallitaan samanaikaisia tietojen käyttöä
 - n Miten kytketään ohjelmat tietokantaan
 - n Useita toteutuksia, joissa keskeisenä osana SQL-tietokantakieli

1

SQL tietokantakieli

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

2

SQL tietokantakieli

- n SQL on standardoitu
- n viimeisin standardi vuodelta 2003
- n murteita – mutta kuitenkin yhteinen ydin
 - n Vaihto toteutuksesta toiseen ei välttämättä ole kuitenkaan yksinkertaista

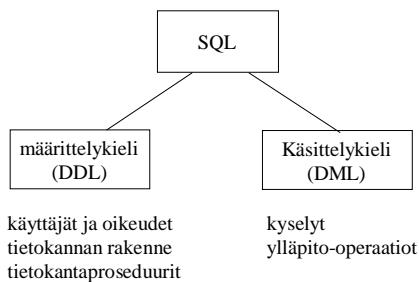
3

SQL-tietokanta

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

4

SQL



5

SQL

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

6

SQL tiedonmäärittelykieli

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

7

SQL taulun luonti

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

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

8

Taulun määrittely

```
create table Ordered (  
  OrderId      integer not null,  
  WhenMade     timestamp(0) 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  
    (CustomerId) references Customer  
);
```

9

Taulun määrittely

```
create table Ordered (  
  OrderId      integer not null,  
  WhenMade     timestamp(0) 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  
    (CustomerId) references Customer  
);
```

Aikaleima, ei sekunnin osia

vaihtuvamittainen Merkkijono Maksimipituus 20

kokonaisluku

desimaaliluku kokonaispituus 6, desimaaliosa 2

10

Taulun määrittely

```
create table Ordered (  
  OrderId      integer not null,  
  WhenMade     timestamp(0) 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  
    (CustomerId) references Customer  
);
```

pakollinen tieto

avain

viiteavain

11

SQL tiedonmäärittely

- n Aikoja
 - n Date päiväys
 - n Time kellonaika
 - n Timestamp(x) päiväys ja kellonaika
 - n (sekunnin osat x desimaalin tarkkuudella)
 - n Interval aikaero
- n Aikoja voidaan verrata ja niillä voi laskea
 - Olkoon *this_day* date-tyyppinen,
 - this_day* + interval 3 day
 - on kolmen päivän päästä
- n Huom: Harjoittelukannassa oleva date-tietotyyppi käyttäytyy kuten timestamp(0) = Oraclen vanha tapa. Päiväkseen voi lisätä aikaa desimaalilukuina, joissa kokonaisosa kertoo vuorokausien lukumäärän esim *this_day*+0.5 on 12 tunnin päästä *this_day* arvosta.

12

SQL tiedonmäärittely

- n Viiteavainmäärittelyyn voidaan liittää toimintasääntö: mitä tehdään operaation rikkoessa viite-ehyden

```
foreign key (sarakeet) references taulu [(sarakeet2)]  
[ 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 on tarkoituksenmukaista

13

SQL -kysely

Kyselyn yleisrakenne:

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

Kysely tuottaa nimettömän tulostaulun.

14

SQL-kysely

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

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

15

SQL -kysely

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

Diagram: A box around the select clause is labeled 'melkein projektio' (almost projection). A box around the where clause is labeled 'valinta' (selection).

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

16

SQL-kysely

- n Tulostietomäärittelyn elementeille lasketaan normaalitapauksessa arvo jokaista valintaehdot täyttävää riviä (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

17

SQL -kysely

- n 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

18

SQL-kysely

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

19

SQL -kyselyt

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

AND	true	false	unknown
true	true	false	unknown
false	false	false	false
unknown	unknown	false	unknown

NOT	
true	false
false	true
unknown	unknown

20

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

Sarake = null: tuottaa aina tuloksen unknown

21

SQL -kyselyt

- n Tietotyyppistä riippuen sarakearvoilla voi tulostietomäärittelyissä ja ehdoissa suorittaa laskentaoperaatioita (+, -, *, /, merkkijonoille katenaatio || (peräkkäin laittaminen))
 - n pituus*paino, etunimi||' '|sukunimi
- n Merkkijonovakiot 'hipsuihin', numeeriset ilman
- n Päiväykset muodossa date '2005-11-24'
- n On mahdollista käyttää myös skalaarifunktioita – esim. merkkijonon pituus length(Sarake), funktiovalikoima vaihtelee

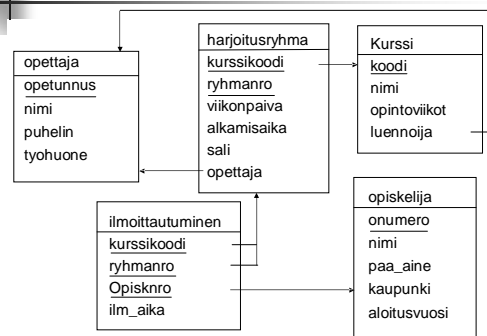
22

SQL -kyselyt

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

23

SQL- kyselyt esimerkkitaulu



24

SQL -kyselyt

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

25

SQL -kyselyt

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

26

Liitokset SQL:ssä

- n Kyselyn from-osassa voi olla useita tauluja
- n Kaikki ne taulut, joiden dataa halutaan mukaan tulokseen on annettava from-osassa
- n 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

27

Liitokset SQL:ssä

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

as ei käy Oraclessa

28

Liitokset SQL:ssä

- n Jos sama taulu esiintyy from -osassa useaan kertaan, on taulun esiintymät erotettava käyttämällä aliasta
- n 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`
- n ehto `A.koodi<B.koodi` estää saman parin toistumisen eri järjestyksissä

29

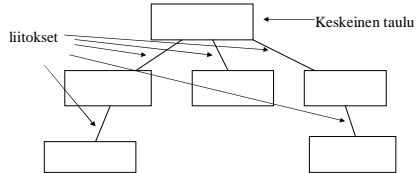
Liitokset SQL:ssä

- n Tyypillinen virhe liitoksissa on jättää jokin liitosehto pois, jolloin tuloksen rivijoukko tulee huomattavasti suuremmaksi kuin pitäisi
- n 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.

30

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.



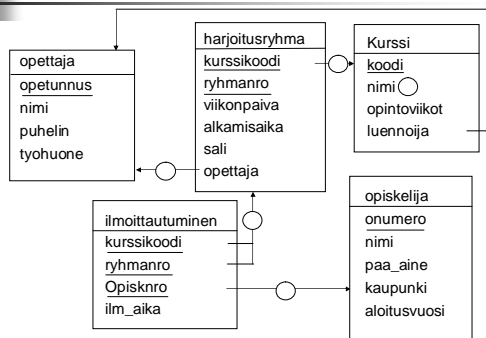
31

Liitokset SQL:ssä

- Laadi raportti kurssin Java-ohjelmointi harjoitusryhmistä
- Mitä halutaan tulokseen:
 - Ryhmän numero (taulussa harjoitusryhmä)
 - Ohjaajan nimi (taulussa opettaja)
 - kokoontumispaiva (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

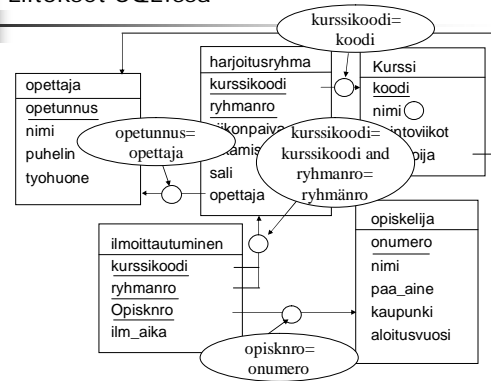
32

Liitokset SQL:ssä



33

Liitokset SQL:ssä



34

Liitokset SQL:ssä

```

select H.ryhmanro rno, 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;
    
```

35