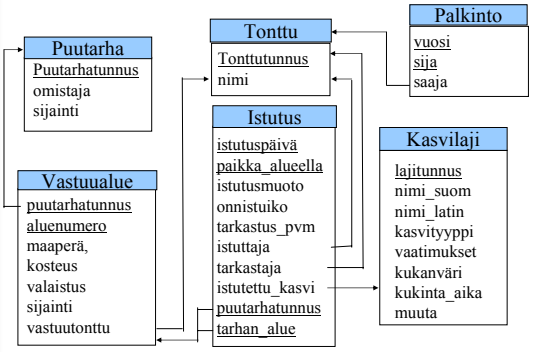
Harjoitustehtävä 1



Harjoitustehtävä 2

- create table Tonttu (
  tonttutunnus varchar(20) not null primary key,
  nimi varchar(40) not null
  );

  create table Puutarha(
  puutarhatunnus varchar(20) not null primary key,
  omistaja varchar(60),
  sijainti varchar(20) not null
  );
- create table Palkinto(
  vuosi int not null,
  sija int not null,
  saaja varchar(20) not null,
  primary key (vuosi, sija),
  foreign key (saaja) references tonttu
  );

Harjoitustehtävä 2

- create table Vastuualue (
  puutarhatunnus varchar(20) not null,
  aluenumero int not null,
  maapera varchar(60),
  kosteus varchar(20),
  valaistus varchar(20),
  sijainti varchar(20),
  vastuutonttu varchar(20),
  primary key (puutarhatunnus, aluenumero),
  foreign key (vastuutonttu) references tonttu,
  primary key (puutarhatunnus, aluenumero),
  foreign key (vastuutonttu) references tonttu,
  foreign key (puutarhatunnus) references puutarha on delete cascade
  );

Harjoitustehtävä 2

- create table Kasvilaji(
  lajitunnus varchar(30) not null primary key,
  nimi_suom varchar(60) not null,
  nimi_latin varchar(60) not null,
  kasvityyppi varchar(40) not null,
  vaatimukset varchar(1000),
  kukanvari varchar(40),
  kukinta_aika varchar(30),
  muuta varchar(1000)
  kukinta_aika varchar(30),
  muuta varchar(1000)
  );

Harjoitustehtävä 2

```
create table Istutus(
      istutuspaiva date not null,
      paikka_alueella varchar(20) not null,
      istutusmuoto varchar(20) not null,
      onnistuiko char,
      tarkastus_pvm date,
      istuttaja varchar(20)not null,
      tarkastaja varchar(20),
      istutettu_kasvi varchar(30) not null,
      puutarhatunnus varchar(20) not null,
      tarhan_alue int not null,
      primary key (puutarhatunnus, tarhan_alue, istutuspaiva,
         paikka_alueella),
      foreign key (istuttaja) references tonttu,
      foreign key (tarkastaja) references tonttu,
      foreign key (puutarhatunnus, tarhan_alue) references
        vastuualue  on delete cascade,
      foreign key (istutettu_kasvi) references kasvilaji);
```

SQL kysely

Kyselyn yleisrakenne:

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

Kysely tuottaa nimettömän tulostaulun.

SQL-kysely

- Tulostietomäärittelyn elementeille lasketaan normaalitapauksessa arvo jokaista valintaehdot täyttävää riviyhdistelmää kohden

SQL -kyselyt

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

Aliasten käyttö

- 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
```

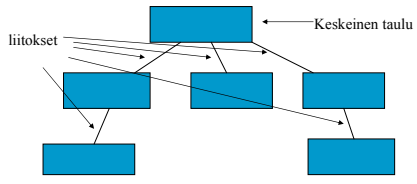
- ehto A.koodi<B.koodi estää saman parin toistumisen eri järjestyksessä

Kyselyn rakentaminen

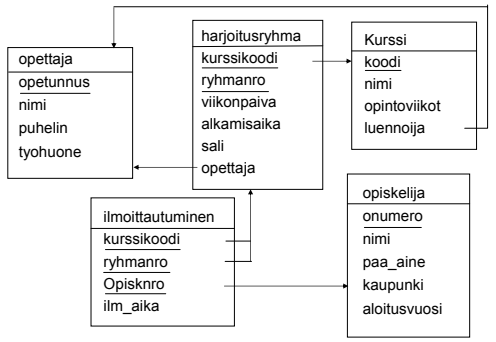
- Tyypillinen virhe liitoksissa on jättää jokin liitosehto pois, jolloin tuloksen rivijoukko tulee huomattavasti suuremmaksi kuin pitäisi
- jos from-osassa on n kpl liitettäviä tauluja tarvitaan vähintään n-1 liitosehtoa. Taulujen liittäminen voi perustua useaan sarakkeeseen, jolloin ehtolausekkeessa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

Kyselyn rakentaminen

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



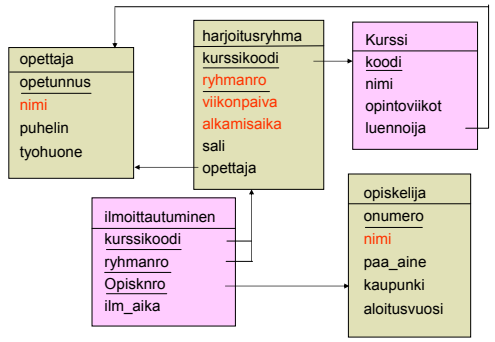
Kyselyn rakentaminen



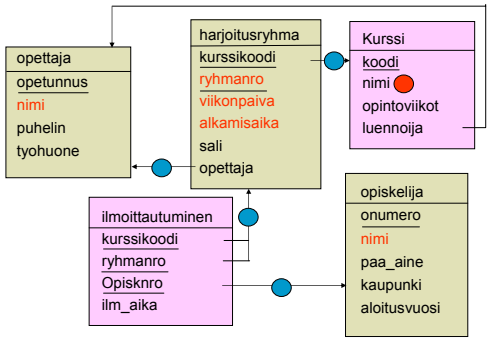
Kyselyn rakentaminen

- 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

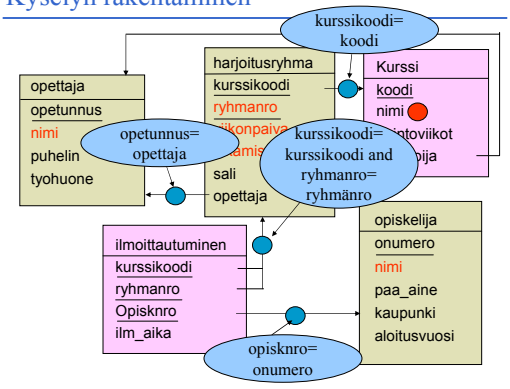
Kyselyn rakentaminen



Kyselyn rakentaminen



Kyselyn rakentaminen



Kyselyn rakentaminen

```
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;
```

Alikyselyt

- Alikyselyllä tarkoitetaan kyselyyn upotettua toista kyselyä. Upotettua kyselyä voidaan käyttää kyselyn from osassa mutta myös where osassa valintaehtojen operandina.
- Alikyselykin tuottaa tuloksenaan taulun
- Alikyselyiden käyttöön valintaehdoissa on omia predikaatteja ja lisätarkenteita, jotka määrittelevät, miten ehdon operandia sovelletaan alikyselyn tulokseen
- IN; NOT IN; tarkenteet any,some, all; EXISTS, NOT EXISTS

Alikyselyt

- Vuoden 92 standardissa from-osaan sallittiin normaalien taulujen myös alikyselyiden tulostaulut
- from (alikysely) [[as] alias [(sarakeluettelo)]]
  - alikysely on normaali kysely
  - sarakeluettelo uudelleennimeää alikyselyn tulossarakkeet
  - tästä rakenteesta on hyötyä, jos halutaan yhdistää yksityiskohtaista tietoa ja yhteenvetotietoa tai eri perustein laskettuja yhteenvetotietoja samalle riville. Yksityiskohtien kyselyssä rakennetta ei tarvita.
  - yleensä rakenteen käyttö vain sotkee asioita - VÄLTÄ

Alikyselyt

- Opettajat, jotka luennoivat jotain kurssia

```
select nimi from opettaja
where opetunnus in
  (select luennoija from kurssi)
order by nimi;
```

- Opettajat, jotka eivät luennoi mitään kurssia

```
select nimi from opettaja
where opetunnus not in
  (select luennoija from kurssi)
order by nimi;
```

alikysely voidaan suorittaa erillään pääkyselystä

Alikyselyt

- Luennoivat opettajat, kytketyllä alikyselyllä

```
select nimi from opettaja
where
exists  (select luennoija from kurssi
            where luennoija= opettaja.opetunnus)
order by nimi;
```

alikysely on evaluoitava erikseen jokaista opettaja-riviä kohti

Yhteenvetokyselyt

- SQL:ssä joukko yhteenvetofunktioita (aggregate function, koostefunktio)
  - AVG keskiarvo
  - MIN pienin arvo (minimi)
  - MAX suurin arvo (maksimi)
  - SUM summa
  - COUNT lukumäärä
  - Yhteenvetofunktioita käytettäessä tulosriviä ei muodostetakaan jokaisesta valintaehdon täyttävästä riviyhdistelmästä vaan, ellei ryhmittelyä ole määritelty, muodostetaan yksi tulosrivi koko aineistosta

Yhteenvetokyselyt

- Opiskelijoiden lukumäärä:
  - select count(*) from opiskelija;
- Count:n argumenttina voisi käyttää myös mitä tahansa vakiota, tulos olisi sama eli rivien lukumäärä
  - select count(1) from opiskelija;
- Jos parametrina annetaan sarake saadaan siinä olevien ei-tyhjien arvojen määrä
  - select count(onumero) from opiskelija;

Yhteenvetokyselyt

- Milloin pisimpään opiskelut helsinkiläinen opiskelija on aloittanut opintonsa?

```
select min(aloitusvuosi) from opiskelija
where kaupunki='Helsinki';
```

- Keskiarvoa, summaa, minimiä ja maksimia laskettaessa tyhjäarvot jätetään huomioimatta
- Kurssien keskimääräinen opintoviikkomäärä
  - select avg(opintoviikot) from kurssi;

Yhteenvetokyselyt

- Yhteenvedon laskenta voidaan rajata vain keskenään erilaisiin arvoihin (projektioon). Tällainen rajaus on yleensä järkevä vain lukumääriä laskettaessa
- Monellako eri paikkakunnalla opiskelijat asuvat?
  - select count(distinct kaupunki) from opiskelija;

Yhteenvetokyselyt

- Yhteenvetofunktion sisältävään kyselyyn ei voi ottaa mukaan dataa niiltä yksittäisiltä riveiltä, joilta funktio lasketaan
- Minkä kurssien opintoviikkomäärä on suurin?
- Ei siis ole mahdollista:

```
select nimi, max(opintoviikot)
from kurssi;
```

yksittäisen rivin dataa miltä riviltä nimi poimittaisiin?

Yhteenvetokyselyt

- Minkä kurssin opintoviikkomäärä on suurin? Toimivia vaihtoehtoja:

```
select nimi, opintoviikot from kurssi
where opintoviikot>=
ALL (select opintoviikot from kurssi);

Select nimi, opintoviikot from kurssi
where opintoviikot=
(select max(opintoviikot) from kurssi);

select nimi,maksi
from kurssi,
      (select max(opintoviikot) maksi from kurssi) as m
where opintoviikot =m.maksi;
```

Yhteenvetokyselyt -ryhmät

- Jos kyselyyn liitetään ryhmittelymääre (Group by) muuttuu tuolosrivien muodostusperiaate jälleen:
  - **muodostetaan yksi tulosrivi kutakin ryhmää kohti**
  - group by -määreessä luetellaan sarakkeet, joiden arvojen perusteella ryhmittely tehdään
  - kaikki ne rivit, joilla on sama arvo luetelluissa sarakkeissa muodostavat ryhmän
  - ryhmät muodostetaan sen jälkeen kun on ensin sovellettu where-ehtoa rivien karsintaan.

Yhteenvetokyselyt -ryhmät

Group by A

Taulu X

A	B	C	D
1	4	6	7
1	1	4	2
1	5	5	2
2	4	8	7
2	3	5	1
3	1	5	2
3	2	4	6

Select A, sum(B) from X
group by A;

A	B
1	10
2	7
3	3

Yhteenvetokyselyt -ryhmät

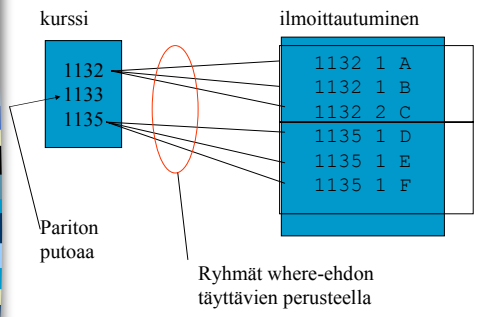
- Group by -lausetta käytettäessä tulostietoluettelossa voi olla yhteenvetofunktioiden lisäksi vain niitä sarakkeita,jotka esiintyvät group by -lauseessa.
- Kaikkien ryhmittelyyn käytettyjen sarakkeiden ei tarvitse olla mukana, mutta yleensä ne ovat

```
select koodi, nimi, ryhmanro, count(*)
from kurssi, ilmoittautuminen
where ilmoittautuminen.kurssikoodi=
          kurssi.koodi
group by koodi, nimi, ryhmanro;
```

Ei anna tyhjien ryhmien ilmoittautujamäärää

koska **nimi** on tulostietolistalla sen pitää olla myös ryhmittelytekijänä vaikka ilman sitä saataisiin samat ryhmät

Yhteenvetokyselyt -ryhmät



Yhteenvetokyselyt -ryhmät

```
select nimi, ryhmanro, count(*)
from kurssi, ilmoittautuminen
where ilmoittautuminen.kurssikoodi=kurssi.koodi
group by nimi, ryhmanro
union
(select nimi, ryhmanro, 0
from kurssi, harjoitusryhma H,
where koodi=H.kurssikoodi and
      (koodi,H.ryhmanro) not in
          (select kurssikoodi, ryhmanro
           from ilmoittautuminen));
```

vakioarvo

Yhteenvetokyselyt -ryhmät

- Ryhmäkohtaisen rivin mukaanottamista tulokseen voidaan rajoittaa having määreellä.
- Having-ehto toimii kuten where-ehto, mutta se perustuu ryhmäkohtaisesti lasketun yhteenvetofunktion arvoon
- Ryhmät, joihin on ilmoittautunut yli 20 opiskelijaa

```
select nimi, ryhmanro, count(*)
from kurssi, ilmoittautuminen
where ilmoittautuminen.kurssikoodi=kurssi.koodi
group by nimi, ryhmanro
having count(*) >20;
```

Yhteenvetokyselyt

- Yhteenvetofunktioilla voi laskea, mutta niitä ei voi ketjuttaa (eli toinen on toisen argumenttina)
- Millä kurseilla on suurin keskimääräinen ryhmäkoko, ei onnistu seuraavasti

```
select nimi, H.ryhmanro, max(avg(count(*)))
from kurssi, harjoitusryhma H, ilmoittautuminen I
where koodi=H.kurssikoodi and
      H.kurssikoodi=I.kurssikoodi and
      H.ryhmanro=I.ryhmanro
group by nimi, H.ryhmanro
      *
```

Ajettuna Oraclessa: VIRHE rivillä 1:
ORA-00937: tämä ei ole yhden ryhmän koostefunktio

Yhteenvetokyselyt -ryhmät

```
select koodi, nimi, opiskelijoita/ryhmia
  from kurssi,
 (select kurssikoodi, count(*) opiskelijoita
    from ilmoittautuminen
    group by kurssikoodi) as ilm,
 (select  kurssikoodi, count(*) ryhmia
    from harjoitusryhma
    group by kurssikoodi) as ryhm
 where kurssi.koodi= ilm.kurssikoodi and
    kurssi.koodi= ryhm.kurssikoodi and
    opiskelijoita/ryhmia = (XXXXX –seuraavalla kalvolla)
```

Yhteenvetokyselyt -ryhmät

- XXXXX=

```
select max(opiskelijoita/ryhmia)
  from
 (select kurssikoodi, count(*) opiskelijoita
    from ilmoittautuminen
    group by kurssikoodi) as i,
 (select  kurssikoodi, count(*) ryhmia
    from harjoitusryhma
    group by kurssikoodi) as ry
 where i.kurssikoodi = ry.kurssikoodi
```