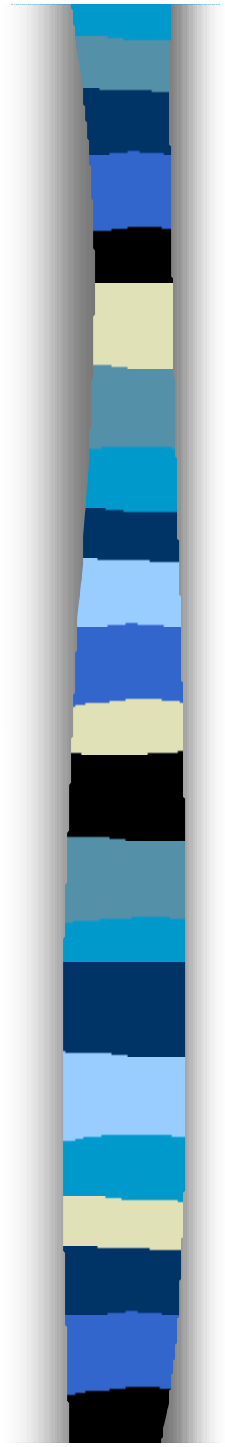
SQL tietokantakieli

- 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



SQL

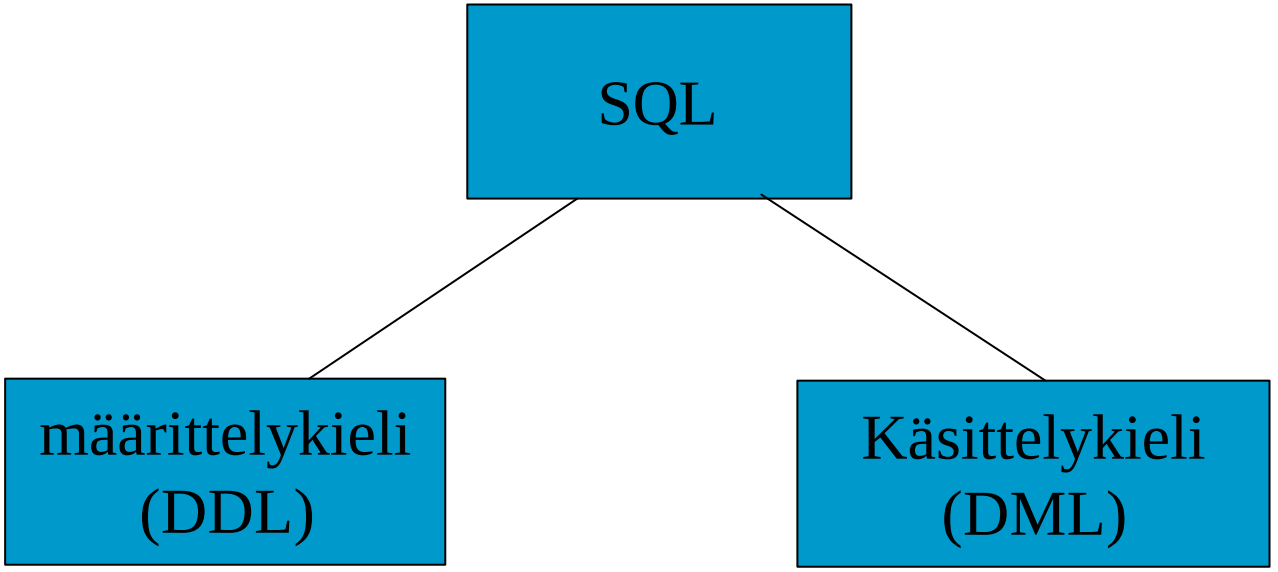
- SQL on standardoitu
- viimeisin standardi vuodelta 1999
- toteutukset noudattavat enimmäkseen vuoden -92 standardia ja sitäkin vain osittain
- murteita - yhteinen suppeahko ydin

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)

SQL



käyttäjät ja oikeudet
tietokannan rakenne
tietokantaproseduurit

kyselyt
ylläpito-operaatiot

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'

SQL tiedonmäärittelykieli

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

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* …]

Table definition

```
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
);
```

```
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
);
```

päiväys
(tässä on kyseessä
Oracle, joten timestamp)

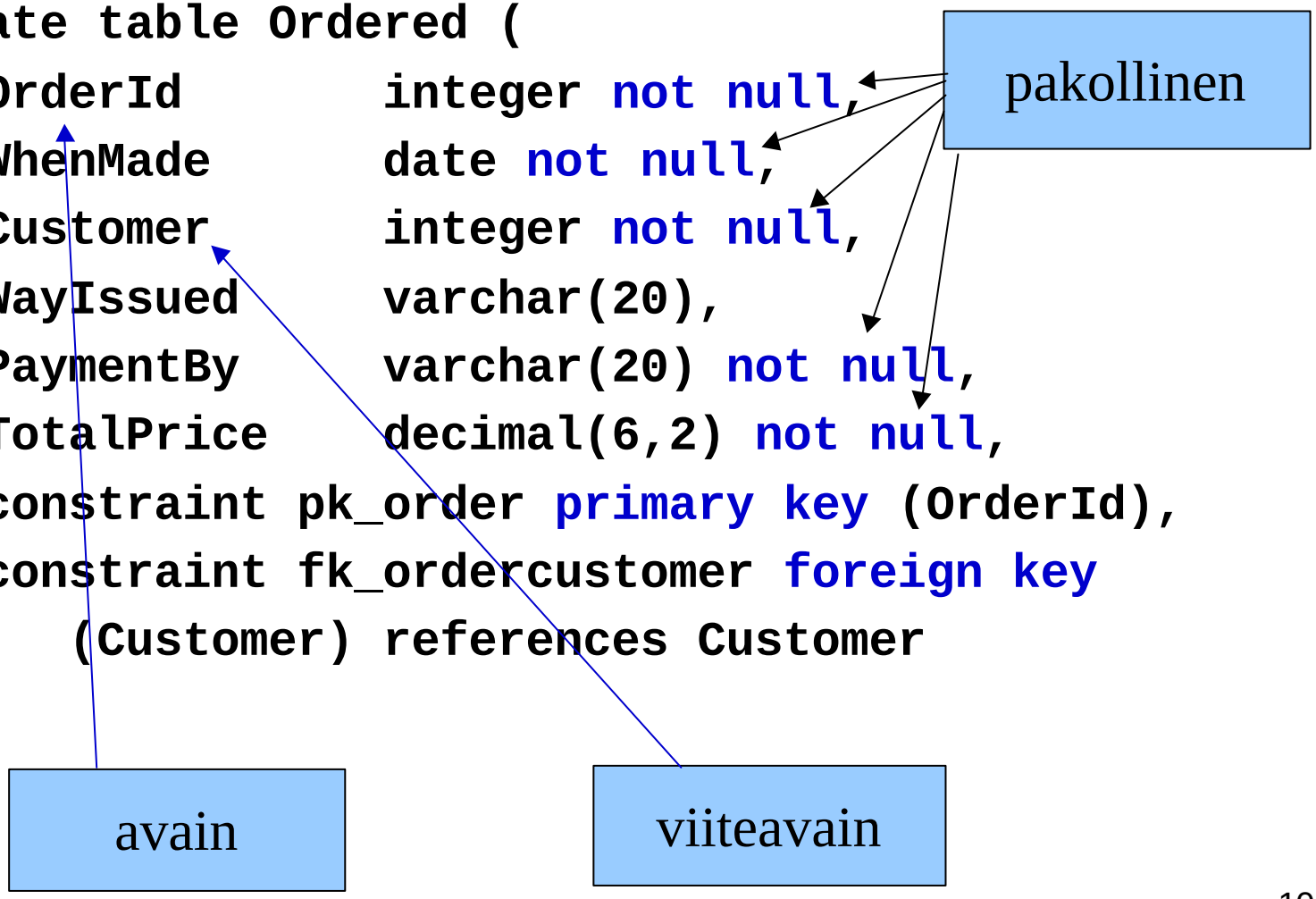
vaihtuvamittainen
merkkijono

kokonaisluku

desimaaliluku
kokonaispituus 6,
desimaaliosa 2

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
);
```



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

SQL tiedonmäärittely

- Viiteavainmäärittelyyn voidaan liittää toimintasääntö, mitä tehdään operaation rikkoessa viite-eheyden

  foreing key (*sarakkeet*) references *taulu* [(*sarakkeet2*)]

  [on delete {restrict | cascade |nullify}]

  [on update {restrict | cascade |nullify}]

  viitteen kohde katoaa:

  restrict estää rikkovan operaation (oletus)

  cascade vyöryttää, poistaa tai muuttaa viittaavat rivit

  nullify tyhjentää viittaukset

SQL kysely

Kyselyn yleisrakenne:

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

Kysely tuottaa nimettömän tulostaulun.

SQL-kysely

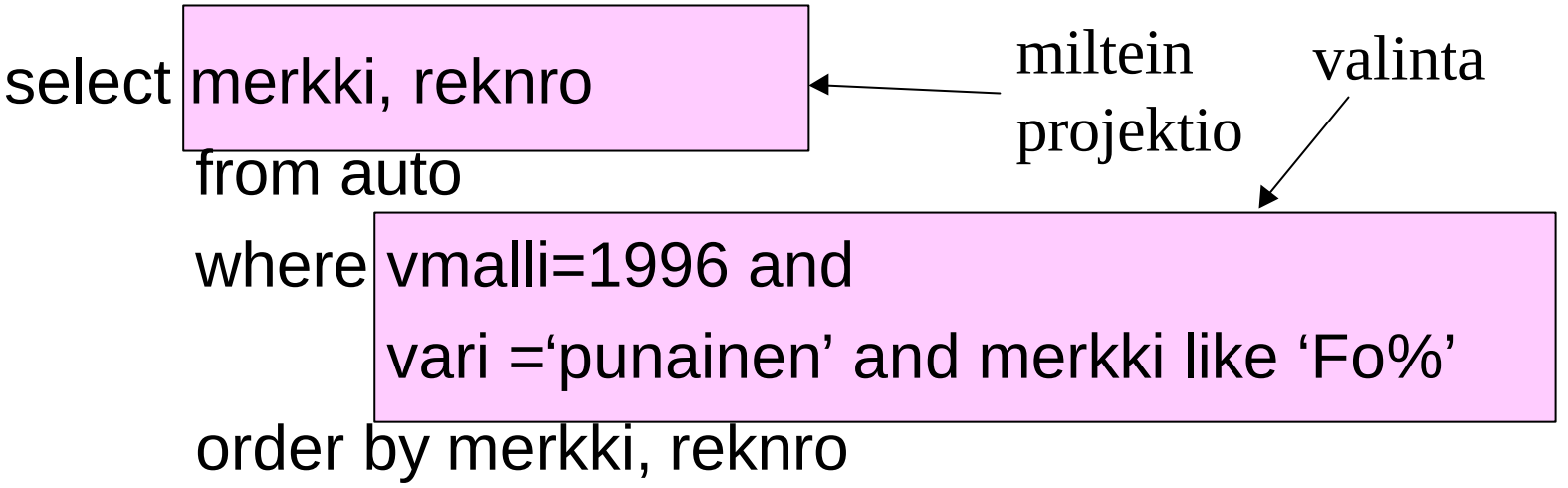
Taulu: AUTO(rekno, merkki, vmalli, vari)

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

SQL kysely



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

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 on 3 punaista vuoden 1996 Fordia tulee merkki ‘Ford’ tulostauluun 3 kertaa.

Toimii siis toisin kuin relaatioalgebran projektio, joka poistaa tuplat

SQL-kysely

ABC-123	Ford	1966	punainen
BCA-222	BMW	1966	musta
CCC-777	Ford	1971	punainen
CIA-105	Ford	1966	punainen
FBI-106	Ford	1966	punainen

SQL-kysely

ABC-123	Ford	1966	punainen
BCA-222	BMW	1966	musta
CCC-777	Ford	1971	punainen
CIA-105	Ford	1966	punainen
FBI-106	Ford	1966	punainen

SQL-kysely

ABC-123	Ford	1966	punainen
BCA-222	BMW	1966	musta
CCC-777	Ford	1971	punainen
CIA-105	Ford	1966	punainen
FBI-106	Ford	1966	punainen

Kyselyn tulos:

Ford
Ford
Ford

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
```

Kyselyn tulos: Ford

SQL kysely

- Tulostietomäärittelyssä *-merkki

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

Kyselyn tulos:

ABC-123	Ford	1966	punainen
CIA-105	Ford	1966	punainen
FBI-106	Ford	1966	punainen

SQL-kysely

- Kyselyn ehto-osassa voidaan verrata saraketta tai vakiota tai funktion arvoa
  - sarakkeeseen, vakioon, funktion arvoon
    - asiakasnro>99, length(nimi)>=10
    - length(etunimi) = length(sukunimi), 1=2, 1<2
    - hinta>alehinta
  - arvojoukkoon
    - luku in (1,2,3), luku between 1 and 3
  - maskiin
    - nimi like ’P%’
- Voidaan myös tutkia sarakkeen tyhjyyttä
  - puhnro is null, puhnro is not null

SQL-kysely

- Jos vertailun toisena osapuolena on tyhjäarvo on tulos ’tuntematon’. Rivi tulee valituksi tulokseen vain jos ehdon arvo on ’tosi’ (true).

A	B
1	3
2	NULL
3	2
4	5

SQL-kysely

- Jos vertailun toisena osapuolena on tyhjäarvo on tulos ’tuntematon’. Rivi tulee valituksi tulokseen vain jos ehdon arvo on ’tosi’ (true).

A	B	
1	3	true
2	NULL	unknown
3	2	false
4	6	true

Kyselyn ehto-osassa: WHERE A<B

SQL kyselyt

- Totuusarvot tosi (true) ja epätosi (false) käyttäytyvät loogisissa lausekkeissa kuten ohjelmointikielten yhteydessä
- Kolmas totuusarvo ’tuntematon’ käyttäytyy seuraavasti (ehto1 and ehto2 , NOT ehto, ehto1 or ehto2):

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

NOT	
true	false
false	true
unknown	unknown

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 sarakkessa on tyhjäarvo, muuten false

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

SQL kyselyt

- Tietotyypistä riippuen sarakearvoilla voi tulostietomäärittelyissä ja ehdoissa suorittaa laskentaoperaatioita (merkkijonoille katenaatio || (perään laittaminen))
  - alehinta < 0.5*hinta, pvm+3
  - etunimi ||' '||sukunimi --> 'Olli Opiskelija'
- Merkkijonovakiot 'hipsuihin', numeeriset ilman
  - etunimi='Olli' and sukunimi='Opiskelija'
  - ika > 20

SQL kyselyt

- On mahdollista käyttää myös skalaarifunktioita – esim. merkkijonon pituus
  - length(Sarake)
- funktiovalikoima vaihtelee

SQL -kyselyt

- Kyselyn from osassa yksi tai useampi taulu (tai alikysely)
  - FROM Auto
  - FROM Auto, Omistaja
  - FROM (select …) -- tähän palataan myöhemmin
- Jos tauluja on vain yksi on kyseessä valinta
  - Select reknro from Auto where vmalli=2003;
  - Select nimi from Omistaja where reknro like ’ABC-123’;
  - Select nimi from Omistaja where reknro=’ABC-123’

SQL -kyselyt

- Jos tauluja on monta on kyseessä ristitulo ellei ehto-osassa ole liitosehtoa (hyvin harvoin halutaan tulokseksi ristitulo)
  - select * from opiskelija, suoritus; ????
- Jos tauluja on monta ja ehto-osassa on liitosehto, on kyseessä liitos – muista siis liitosehto
  - select * from opiskelija, suoritus where opiskelija.hetu=suoritus.hetu;
  - select * from opiskelija, suoritus, kurssi where opiskelija.hetu=suoritus.hetu and suoritus.kkoodi=kurssi.kkoodi;

SQL-kysely

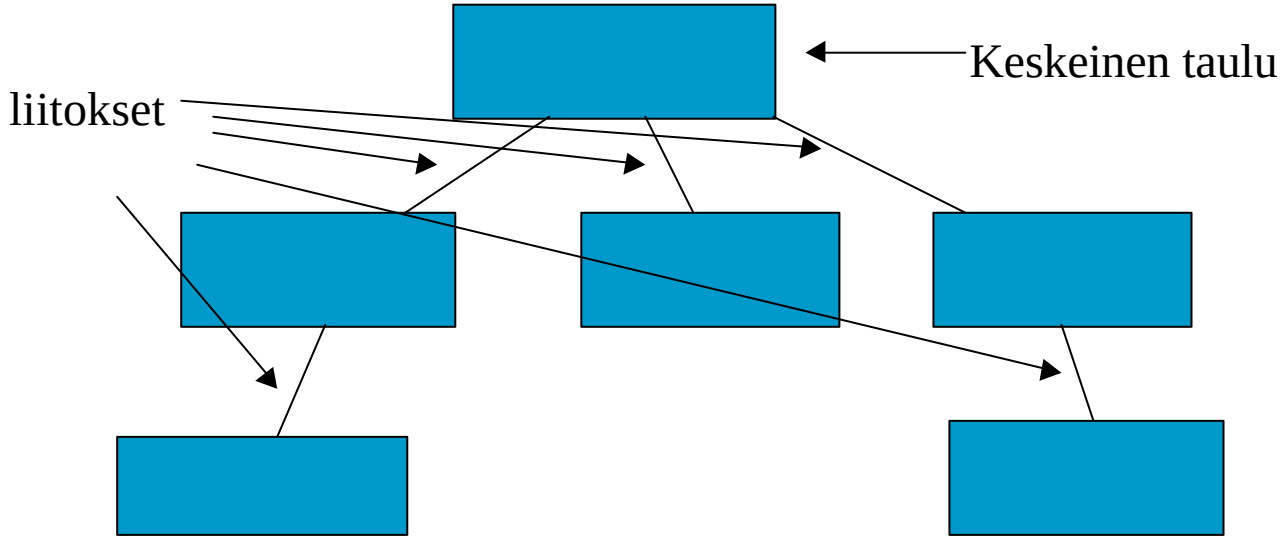
- Tyypillinen virhe liitoksissa on jättää jokin liitosehto pois, jolloin tuloksen rivijoukko tulee huomattavasti suuremmaksi kuin pitäisi
- jos from-osassa on n kpl liitettäviä tauluja tarvitaan vähintään n-1 liitosehtoa. Taulujen liittäminen voi perustua useaan sarakkeeseen, jolloin ehtolausekkeessa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

SQL-kysely

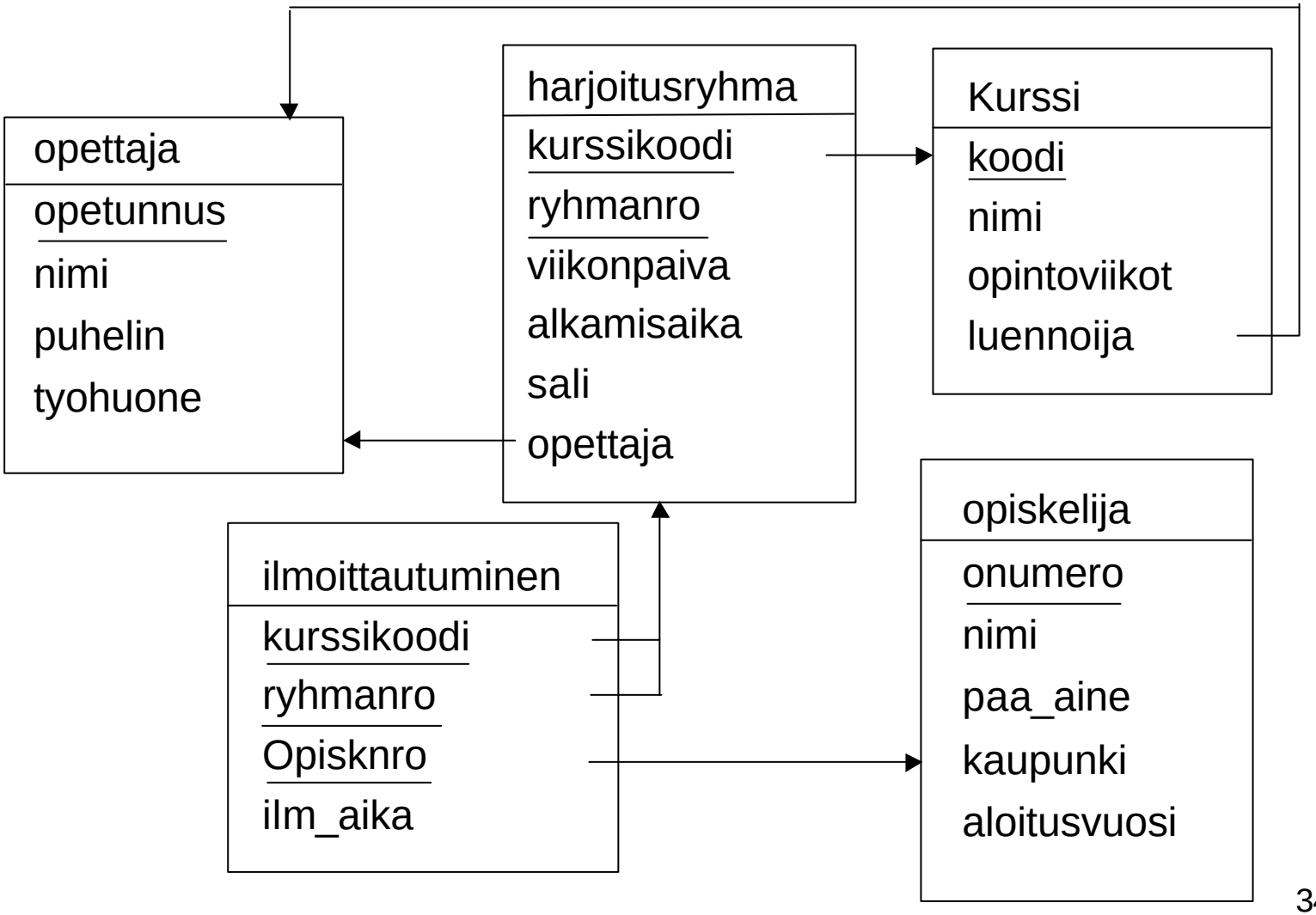
- Tyypillinen virhe liitoksissa on jättää jokin liitosehto pois, jolloin tuloksen rivijoukko tulee huomattavasti suuremmaksi kuin pitäisi
- jos from-osassa on n kpl liitettäviä tauluja tarvitaan vähintään n-1 liitosehtoa. Taulujen liittäminen voi perustua useaan sarakkeeseen, jolloin ehtolausekkeessa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

SQL-kysely

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



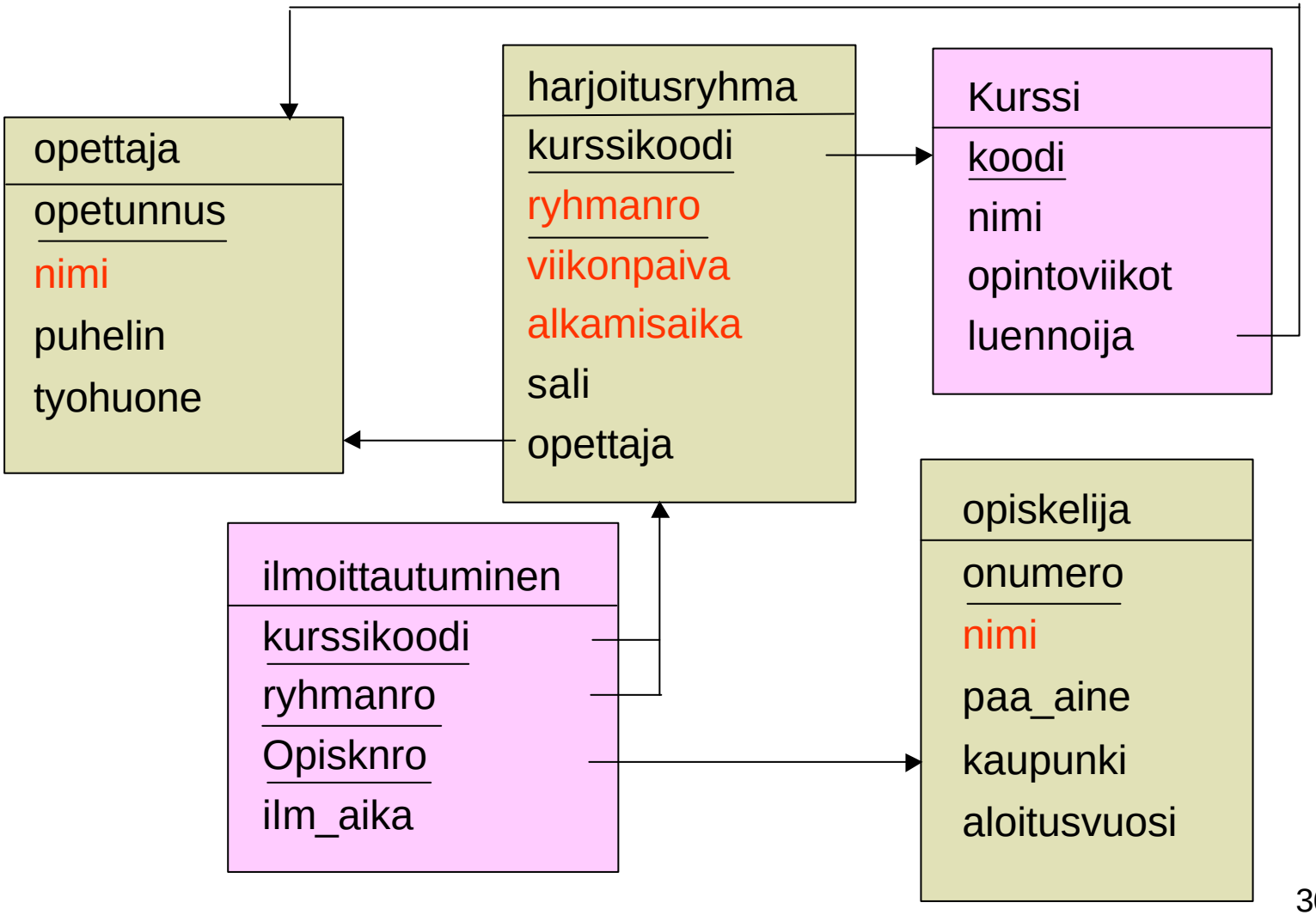
Kyselyn rakentaminen



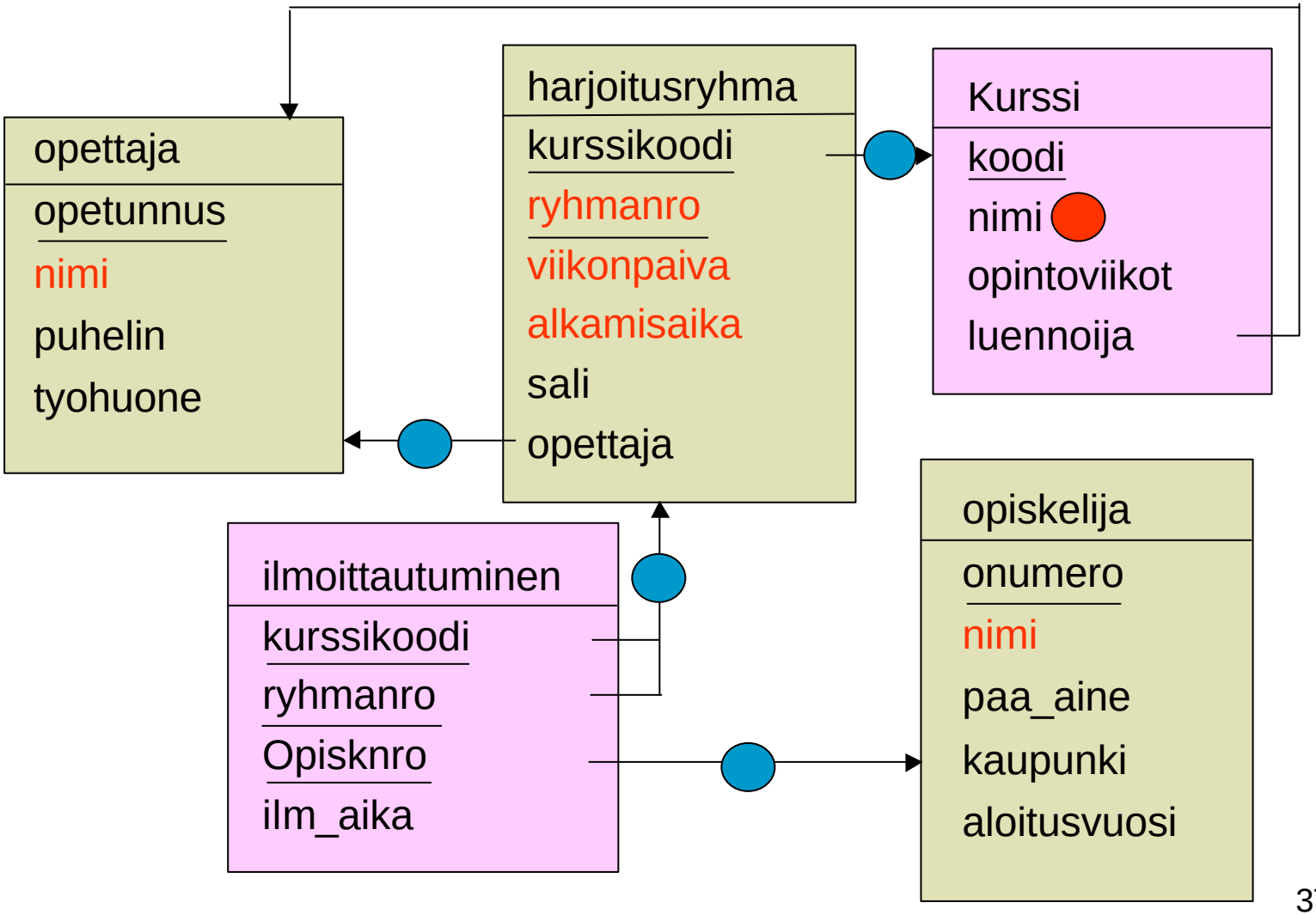
Kyselyn rakentaminen

- 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

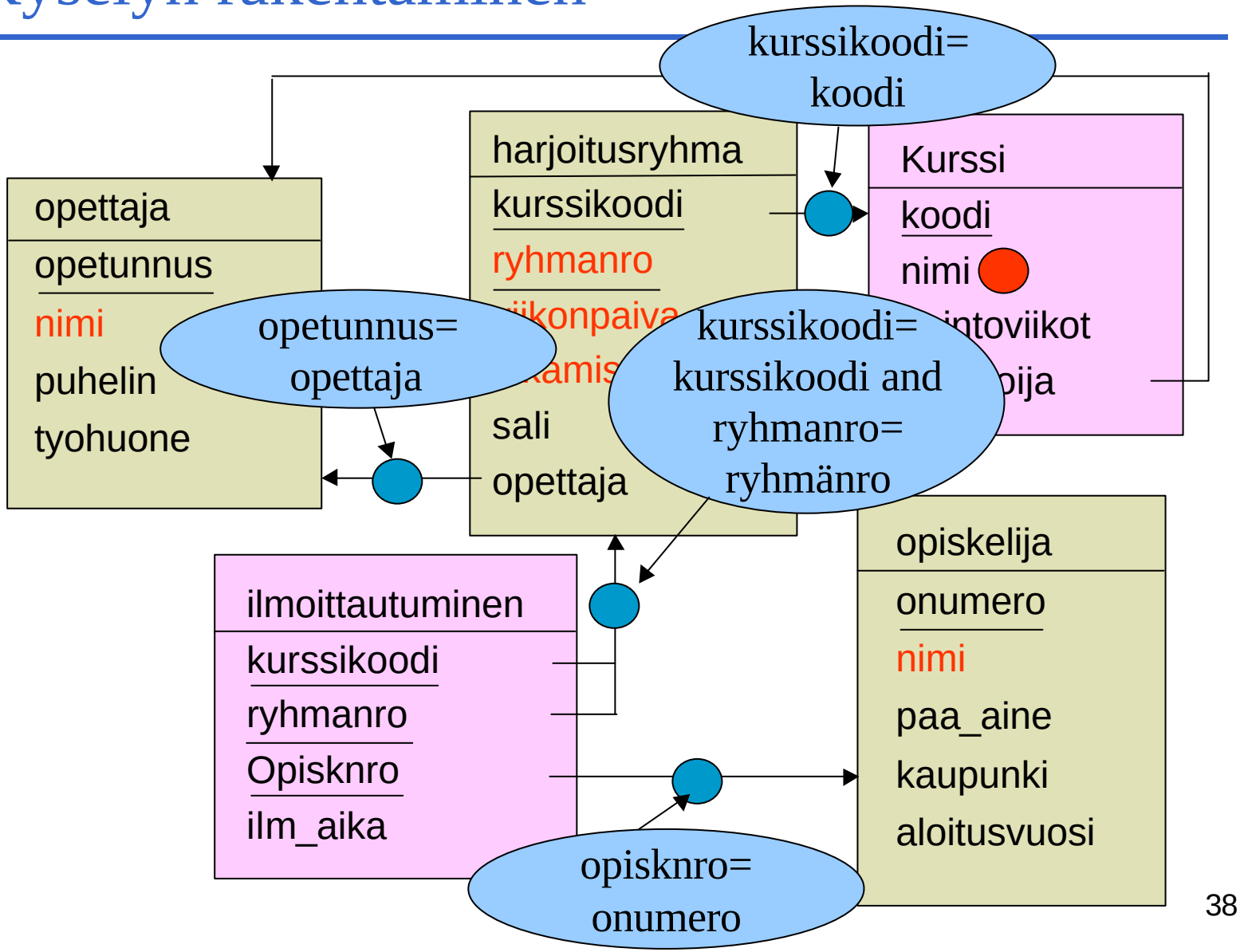
Kyselyn rakentaminen



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

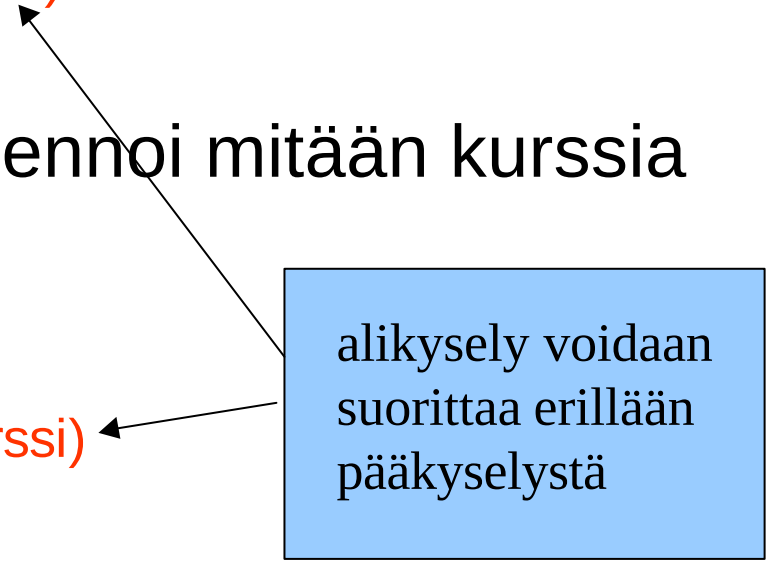
Alikyselyt

- 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;
```



Alikyselyt

- Luennoivat opettajat, kytketyllä alikyselyllä

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