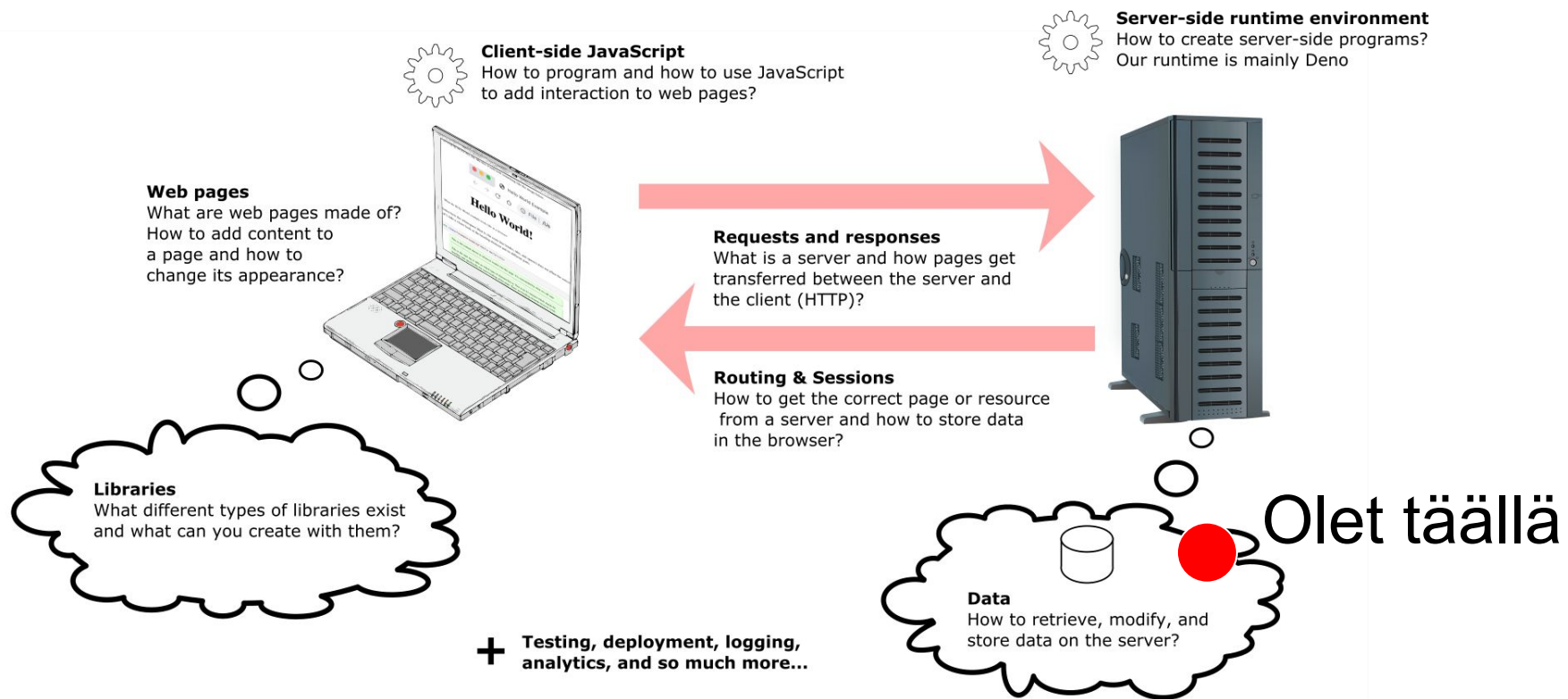


# Päivä 3 - Tietokannat

2021-10-29

AaltoPRO - Websovelluskehitys

# Web-sovellukset korkealla tasolla



# Päivä 3

- 9-12 Aamupäivä
  - Tietokannanhallintajärjestelmät ja tietokannat, tietokantakyselyt yhteen tietokantatauluun
  - Kahvitauko
  - Tietokantakyselyt useampaan tietokantatauluun, tietokantakyselyt sovelluksessa
  - Yhteenveto
- 12:00 - 13:00 Lounas
- 13:00 - 16:00 Iltapäivä
  - Tietokannan hahmottelu: Käsitemanalyysi, tietokantakaavio, tietokannan luominen
  - Kahvitauko
  - SQL-kyselyn tulosten ryhmittely, kyselyiden tehokkuudesta
  - Yhteenveto

# Resursseja

- Tietokantojen perusteet, HY
  - <https://tietokantojen-perusteet-19.mooc.fi/>
- Web Software Development (WSD) -kurssimateriaali
  - <https://wsd.cs.aalto.fi/>
  - Tietokantoihin liittyvä osa (luku 7)
- W3C SQL Tutorial
  - <https://www.w3schools.com/sql/>
- PostgreSQL:n dokumentaatio
  - <https://www.postgresql.org/docs/>

# Tietokannanhallintajärjestelmät ja tietokannat

# Tietokannanhallintajärjestelmä ja tietokanta

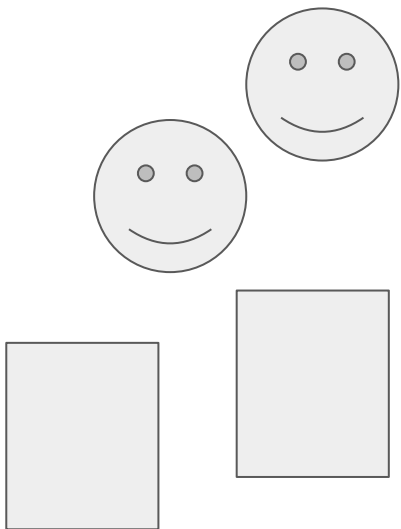
- Tietokannanhallintajärjestelmä on ohjelmisto, jolla käsitellään tietokantaan säilöttyä tietoa
- Tietokanta on tiettyyn aihepiiriin liittyvien tietojen kokoelma, jota tietokannanhallintajärjestelmä hallinnoi
- Arkikielessä käsitteet tietokanta ja tietokannanhallintajärjestelmä esiintyvät synonyymeinä
- Mm. PostgreSQL, MySQL, MariaDB, MongoDB, ...



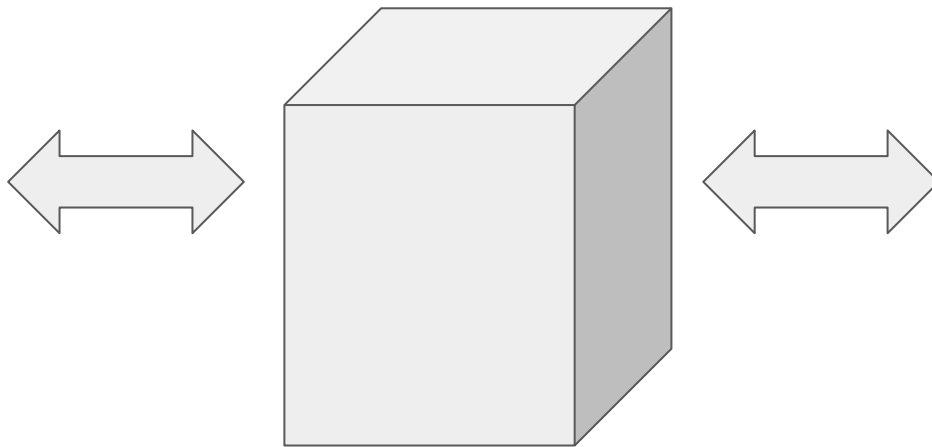
PostgreSQL



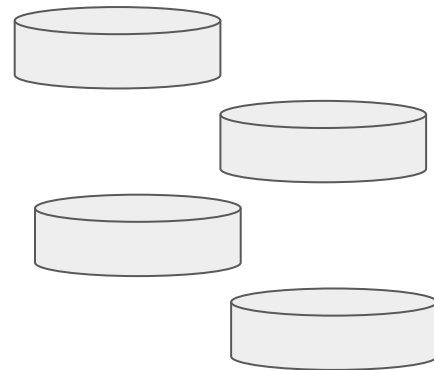
Käyttäjät ja  
sovellukset



Tietokannanhallintajärjestelmä



Tietokannat



# Tänään pääpaino relaatiotietokannoissa

- Relaatiotietokannoissa (esim. PostgreSQL, MySQL, MariaDB, ...) tieto “tauluissa” (relaatiot)



# Tänään pääpaino relaatiotietokannoissa

- Relaatiotietokannoissa (esim. PostgreSQL, MySQL, MariaDB, ...) tieto “tauluissa” (relaatiot)

orders:

<b><u>id</u></b>	<b><i>contact</i></b>
1	Arto
2	Lassi
3	...

order\_items:

<b><u>id</u></b>	<b><u>order_id</u></b>	<b><i>item</i></b>
1	1	Coffee
2	1	Coffee
3	2	Tea

# Tänään pääpaino relaatiotietokannoissa

- Relaatiotietokannoissa (esim. PostgreSQL, MySQL, MariaDB, ...) tieto “tauluissa” (relaatiot)
- Yhteydet taulujen välillä toteutettu pää- ja viiteavaimilla

orders:

<b><u>id</u></b>	<b><i>contact</i></b>
1	Arto
2	Lassi
3	...

order\_items:

<b><u>id</u></b>	<b><u>order_id</u></b>	<b><i>item</i></b>
1	1	Coffee
2	1	Coffee
3	2	Tea

# Tänään pääpaino relaatiotietokannoissa

- Relaatiotietokannoissa (esim. PostgreSQL, MySQL, MariaDB, ...) tieto “tauluissa” (relaatiot)
- Yhteydet taulujen välillä toteutettu pää- ja viiteavaimilla

Pääavain yksilöi taulun rivin -- esimerkiksi orders-taulussa olevan id-sarakkeen arvo on uniikki jokaiselle kyseisen taulun riville

orders:

<u><i>id</i></u>	<i>contact</i>
1	Arto
2	Lassi
3	...

order\_items:

<u><i>id</i></u>	<u><i>order_id</i></u>	<i>item</i>
1	1	Coffee
2	1	Coffee
3	2	Tea

# Tänään pääpaino relaatiotietokannoissa

- Relaatiotietokannoissa (esim. PostgreSQL, MySQL, MariaDB, ...) tieto “tauluissa” (relaatiot)
- Yhteydet taulujen välillä toteutettu pää- ja viiteavaimilla

Pääavain yksilöi taulun rivin -- esimerkiksi orders-taulussa olevan id-sarakkeen arvo on uniikki jokaiselle kyseisen taulun riville

Viiteavain yksilöi toisen taulun rivin -- esimerkiksi order\_items -taulun sarakkeen order\_id arvo vastaa aina jotain orders-taulun id-sarakkeen arvoa

orders:

<u>id</u>	contact
1	Arto
2	Lassi
3	...

order\_items:

<u>id</u>	<u>order_id</u>	item
1	1	Coffee
2	1	Coffee
3	2	Tea

# Tänään pääpaino relaatiotietokannoissa

- Relaatiotietokannoissa (esim. PostgreSQL, MySQL, MariaDB, ...) tieto “tauluissa” (relaatiot)
- Yhteydet taulujen välillä toteutettu pää- ja viiteavaimilla

Pääavain yksilöi taulun rivin -- esimerkiksi orders-taulussa olevan id-sarakkeen arvo on uniikki jokaiselle kyseisen taulun riville

Viiteavain yksilöi toisen taulun rivin -- esimerkiksi order\_items -taulun sarakkeen order\_id arvo vastaa aina jotain orders-taulun id-sarakkeen arvoa

orders:

<u>id</u>	contact
1	Arto
2	Lassi
3	...

order\_items:

<u>id</u>	<u>order_id</u>	item
1	1	Coffee
2	1	Coffee
3	2	Tea



# Tänään pääpaino relaatiotietokannoissa

- Relaatiotietokannoissa (esim. PostgreSQL, MySQL, MariaDB, ...) tieto “tauluissa” (relaatiot)
- Yhteydet taulujen välillä toteutettu pää- ja viiteavaimilla

Pääavain yksilöi taulun rivin -- esimerkiksi orders-taulussa olevan id-sarakkeen arvo on uniikki jokaiselle kyseisen taulun riville

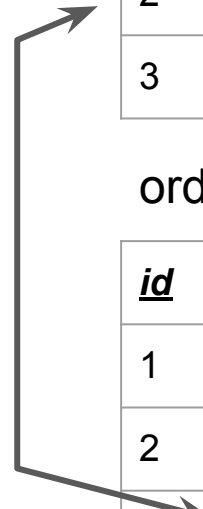
Viiteavain yksilöi toisen taulun rivin -- esimerkiksi order\_items -taulun sarakkeen order\_id arvo vastaa aina jotain orders-taulun id-sarakkeen arvoa

orders:

<u>id</u>	contact
1	Arto
2	Lassi
3	...

order\_items:

<u>id</u>	<u>order_id</u>	item
1	1	Coffee
2	1	Coffee
3	2	Tea



# Tiedon hakeminen SQL-kielellä

- SQL on relaatiotietokantojen käsittelyyn tarkoitettu kieli
- Tarjoaa mahdollisuudet mm. tiedon hakemiseen, päivittämiseen, poistamiseen, ...
- Esimerkkejä:

```
SELECT * FROM orders
```

```
SELECT COUNT(*) FROM orders
```

```
SELECT COUNT(*) FROM orders  
WHERE contact = 'Arto'
```

```
SELECT contact FROM orders
```

```
SELECT * FROM order_items WHERE order_id = 2
```

orders:

<b><u>id</u></b>	<b><i>contact</i></b>
1	Arto
2	Lassi
3	...

order\_items:

<b><u>id</u></b>	<b><u>order_id</u></b>	<b><i>item</i></b>
1	1	Coffee
2	1	Coffee
3	2	Tea

Demo: project-03-database-practice



## Hands-on

- Lataa projekti project-03-database-practice
- Käynnistä komennolla docker-compose up
- Mene osoitteeseen <http://localhost:4000>
  - Järjestelmä: PostgreSQL
  - Palvelin: database-practice
  - Käyttäjätunnus: username
  - Salasana: password
  - Database: database
- Kun kirjautuminen onnistunut, valitse vasemmasta laidasta “SQL Command”, kirjoita avautuneeseen laatikkoon “SELECT 1”, ja paina “Execute”.

<b>System</b>	PostgreSQL ▾
<b>Server</b>	database-practice
<b>Username</b>	username
<b>Password</b>	.....
<b>Database</b>	database

☐ Permanent login

# Hands-on, kyselyharjoittelua

- Hae kaikki käyttäjät
- Hae kaikki käyttäjät, joiden maa on Kiina (China)
- Kenen IP-osoite on “127.109.241.45”?
- Kuinka monta käyttäjää on ruotsista?
- Mikä on tilin keskisaldo?
  - *SELECT AVG(balance) FROM ...*
- Millä tilillä on suurin saldo?
  - *SELECT ... ORDER BY ...*

Kahvitauko?

# Taulujen yhdistäminen

- Useamman taulun yhdistäminen tapahtuu JOIN-avainsanalla
- Avainsanalle kerrotaan minkä sarakkeiden perusteella yhdistäminen tapahtuu (yleensä pääavain ja viiteavain)
- Esimerkiksi:

```
SELECT * FROM orders
  JOIN order_items
    ON orders.id = order_items.order_id
 WHERE orders.contact = 'Lassi'
```

orders:

<b><u>id</u></b>	<b><i>contact</i></b>
1	Arto
2	Lassi
3	...

order\_items:

<b><u>id</u></b>	<b><u>order_id</u></b>	<b><i>item</i></b>
1	1	Coffee
2	1	Coffee
3	2	Tea

# Hands-on, kyselyharjoittelua

- Mitkä tilit käyttäjällä 'Arlan Dreinan' on?
- Mikä on käyttäjän 'Arlan Dreinan' tilien summa?
  - *SELECT SUM(balance) FROM ...*
- Mikä on kiinalaisten tilien keskisaldo?
- Mikä on ruotsalaisten tilien keskisaldo?
- Kenen tilillä on suurin saldo?
  - *... ORDER BY ... LIMIT ...*

# Hands-on, tietokantakyselyt projektiin

- Projektin tietokantakyselyiden toteuttaminen, kts.
  - userService.js
  - accountService.js
- Uusia ominaisuuksia (mikäli aikaa)
  - Käyttäjän tilien saldojen summa käyttäjän sivulle
  - Valikko
  - Tyylittelyä

Lounas

Tietokannan hahmottelu: käsiteanalyysi,  
tietokantakaavio, SQL-kielinen tietokannan  
luominen



# Käsiteanalyysi

- Käsitteiden ja niiden yhteyksien tunnistaminen annetusta kuvauksesta.
- Askeleet:
  1. Tunnista käsitteet -- etsi substantiiveja ja ilmiöitä; rajaa pois epäoleelliset
  2. Tunnista käsitteiden väliset yhteydet -- etsi verbejä, yhteyksiä, lausahduksia
  3. Tunnista osallistumisrajoitteet -- etsi adjektiiveja ja määreitä
  4. Tunnista attribuutit -- tarkastele käsitteitä ja mieti niiden olemassaoloriippuvaisuuksia (esim. "henkilön nimi riippuu henkilöstä, nimen täytyy olla attribuutti")
  5. Yleistä ja eriytä käsitteitä -- tunnista käsitteistä ylliluokkia ja aliluokkia "onko käsite toisen käsitteen erikoistapaus"
  6. Tunnettujen attribuuttien lisääminen (esim. oman kokemuksen pohjalta tai asiakkaalta saatujen tietojen perusteella)

# Käsiteanalyysi

“Opiskelijalla on opintosuorituksia.  
Jokainen opintosuoritus sisältää  
tiedon kurssista, päivämäärästä sekä  
arvosanasta.”

# Käsiteanalyysi

“**Opiskelijalla on opintosuorituksia.**

Jokainen **opintosuoritus** sisältää  
tiedon **kurssista, päivämäärästä**  
sekä **arvosanasta.**”

1. Käsitteiden tunnistaminen

# Käsiteanalyysi

“**Opiskelijalla on opintosuorituksia.**  
Jokainen **opintosuoritus** sisältää  
tiedon **kurssista, päivämäärästä**  
sekä **arvosanasta.**”

1. Käsitteiden tunnistaminen  
opiskelija, opintosuoritus, kurssi,  
päivämäärä, arvosana

# Käsiteanalyysi

“**Opiskelijalla** on **opintosuorituksia**.  
Jokainen **opintosuoritus** sisältää  
tiedon **kurssista**, **päivämäärästä**  
sekä **arvosanasta**.”

1. Käsitteiden tunnistaminen  
opiskelija, opintosuoritus, kurssi,  
päivämäärä, arvosana

opiskelija

opintosuoritus

kurssi

päivämäärä

arvosana

# Käsiteanalyysi

“Opiskelijalla on opintosuorituksia.  
Jokainen opintosuoritus sisältää  
tiedon kurssista, päivämäärästä sekä  
arvosanasta.”

1. Käsitteiden tunnistaminen  
opiskelija, opintosuoritus, kurssi,  
päivämäärä, arvosana
2. Yhteyksien tunnistaminen

opiskelija

opintosuoritus

kurssi

päivämäärä

arvosana

# Käsiteanalyysi

“Opiskelijalla **on** opintosuorituksia.  
Jokainen opintosuoritus **sisältää**  
tiedon kurssista, päivämäärästä sekä  
arvosanasta.”

1. Käsitteiden tunnistaminen  
opiskelija, opintosuoritus, kurssi,  
päivämäärä, arvosana
2. Yhteyksien tunnistaminen

opiskelija

opintosuoritus

kurssi

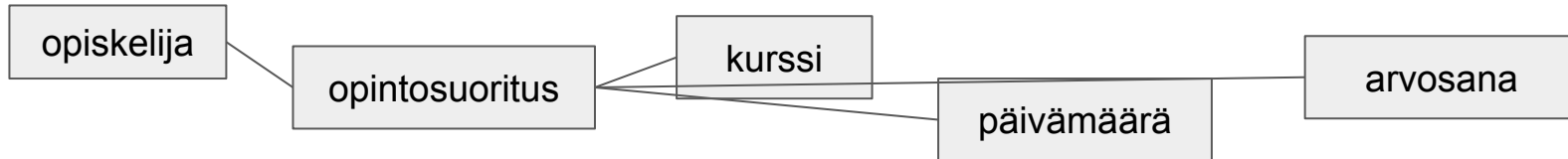
päivämäärä

arvosana

# Käsiteanalyysi

“Opiskelijalla **on** opintosuorituksia.  
Jokainen opintosuoritus **sisältää**  
tiedon kurssista, päivämäärästä sekä  
arvosanasta.”

1. Käsitteiden tunnistaminen  
opiskelija, opintosuoritus, kurssi,  
päivämäärä, arvosana
2. Yhteyksien tunnistaminen

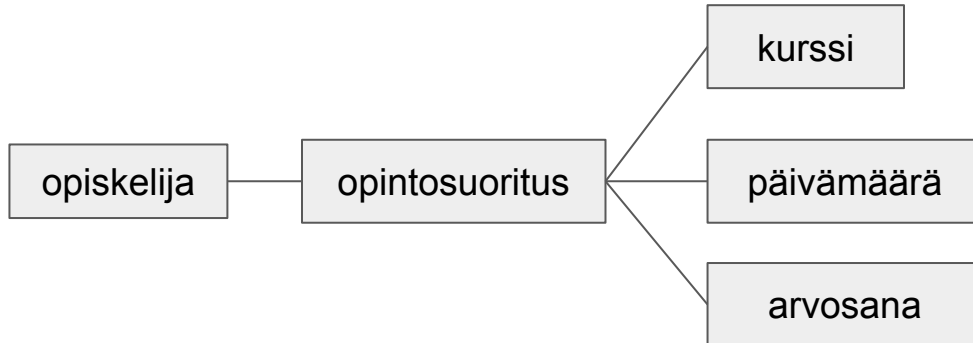




# Käsiteanalyysi

“Opiskelijalla **on** opintosuorituksia.  
Jokainen opintosuoritus **sisältää**  
tiedon kurssista, päivämäärästä sekä  
arvosanasta.”

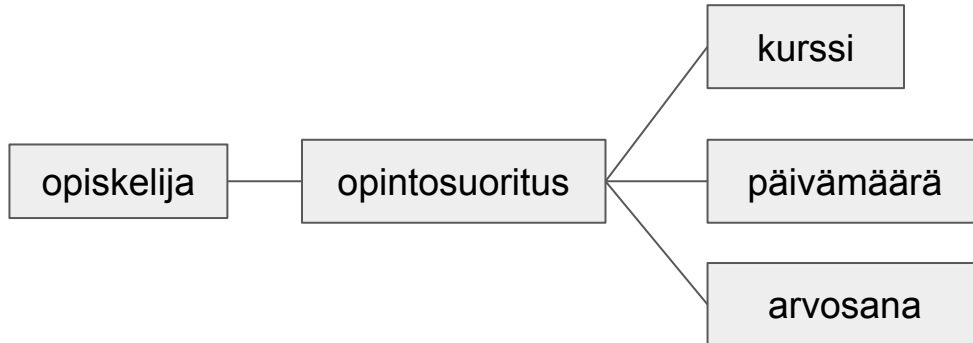
1. Käsitteiden tunnistaminen  
opiskelija, opintosuoritus, kurssi,  
päivämäärä, arvosana
2. Yhteyksien tunnistaminen



# Käsiteanalyysi

“Opiskelijalla on opintosuorituksia.  
Jokainen opintosuoritus sisältää  
tiedon kurssista, päivämäärästä sekä  
arvosanasta.”

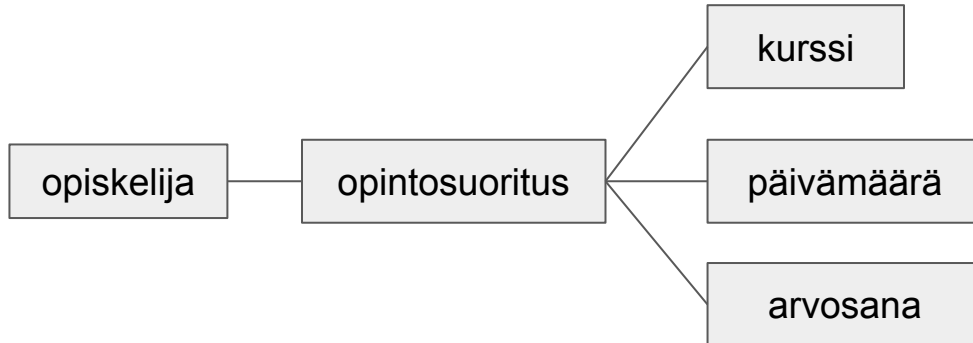
1. Käsitteiden tunnistaminen  
opiskelija, opintosuoritus, kurssi,  
päivämäärä, arvosana
2. Yhteyksien tunnistaminen
3. Osallistumisrajoitteiden tunnistaminen



# Käsiteanalyysi

“Opiskelijalla on opintosuoritu**ksia**.  
Jokainen opintosuoritus sisältää  
tiedon kurssi**sta**, päivämäärä**stä** sekä  
arvosana**sta**.”

1. Käsitteiden tunnistaminen  
opiskelija, opintosuoritus, kurssi,  
päivämäärä, arvosana
2. Yhteyksien tunnistaminen
3. Osallistumisrajoitteiden tunnistaminen



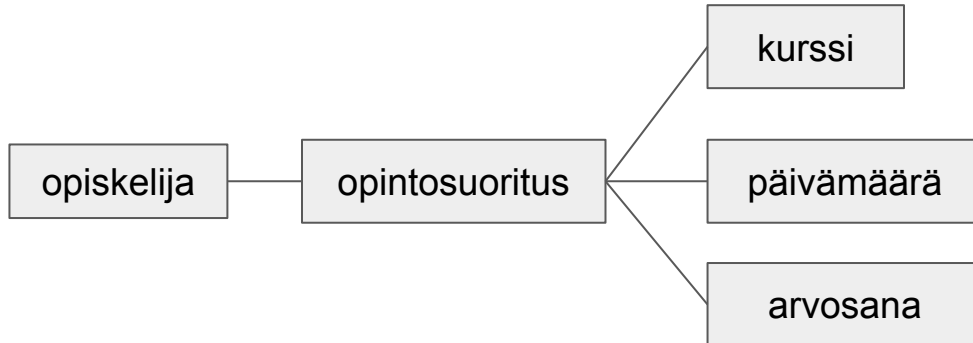
# Käsiteanalyysi

monta



“Opiskelijalla on opintosuorituksia.  
Jokainen opintosuoritus sisältää  
tiedon kurssista, päivämäärästä sekä  
arvosanasta.”

1. Käsitteiden tunnistaminen  
opiskelija, opintosuoritus, kurssi,  
päivämäärä, arvosana
2. Yhteyksien tunnistaminen
3. Osallistumisrajoitteiden tunnistaminen



# Käsiteanalyysi

“Opiskelijalla on opintosuorituksia.  
Jokainen opintosuoritus sisältää  
tiedon kurssista, päivämäärästä sekä  
arvosanasta.”

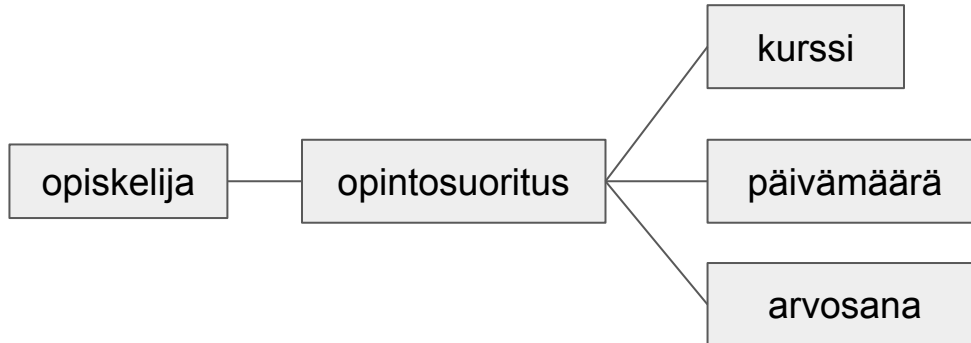
monta



yksi

yksi

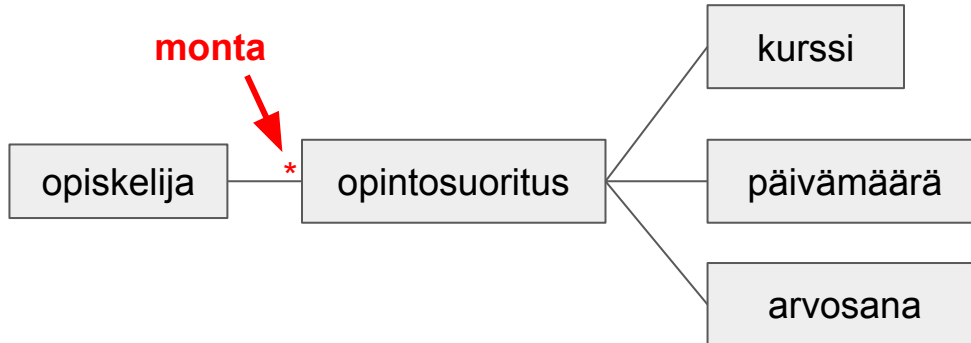
yksi



1. Käsitteiden tunnistaminen  
opiskelija, opintosuoritus, kurssi,  
päivämäärä, arvosana
2. Yhteyksien tunnistaminen
3. Osallistumisrajoitteiden tunnistaminen

# Käsiteanalyysi

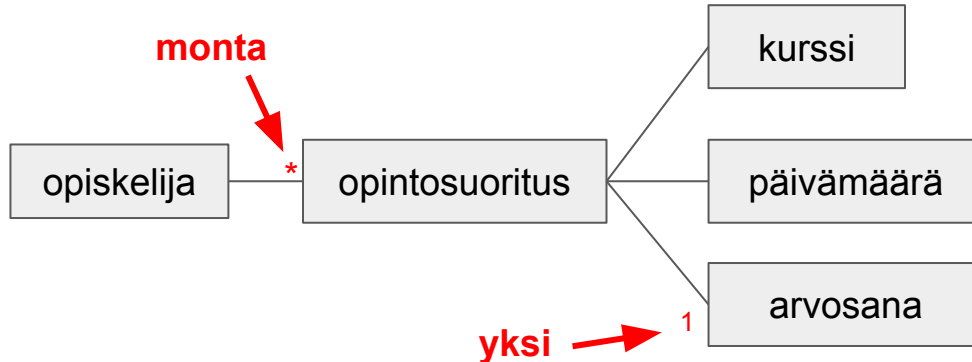
“Opiskelijalla on opintosuorituksia.  
Jokainen opintosuoritus sisältää  
tiedon kurssista, päivämäärästä sekä  
arvosanasta.”



1. Käsitteiden tunnistaminen  
opiskelija, opintosuoritus, kurssi,  
päivämäärä, arvosana
2. Yhteyksien tunnistaminen
3. Osallistumisrajoitteiden tunnistaminen

# Käsiteanalyysi

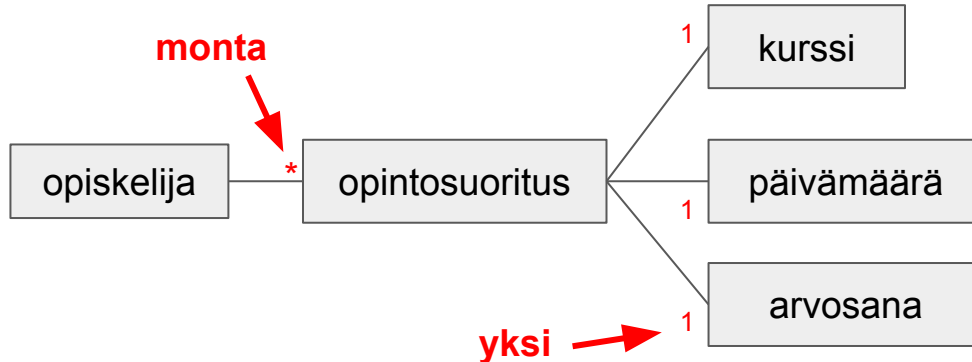
“Opiskelijalla on opintosuorituksia.  
Jokainen opintosuoritus sisältää  
tiedon kurssista, päivämäärästä sekä  
arvosanasta.”



1. Käsitteiden tunnistaminen  
opiskelija, opintosuoritus, kurssi,  
päivämäärä, arvosana
2. Yhteyksien tunnistaminen
3. Osallistumisrajoitteiden tunnistaminen

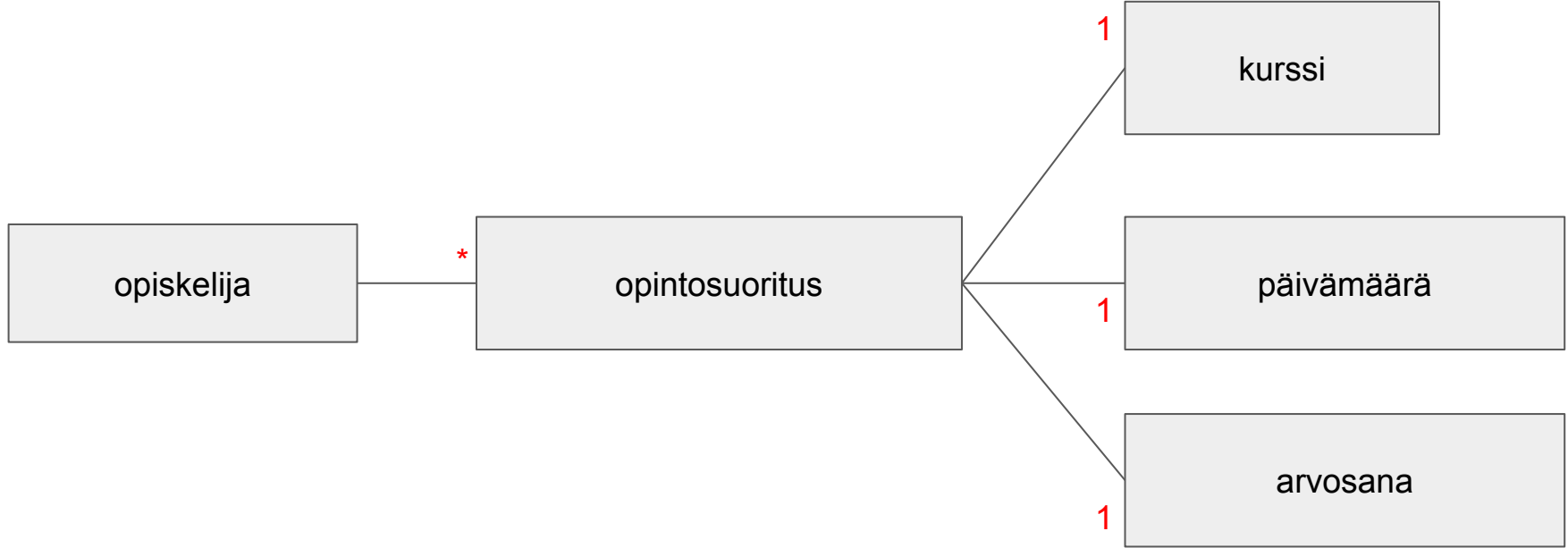
# Käsiteanalyysi

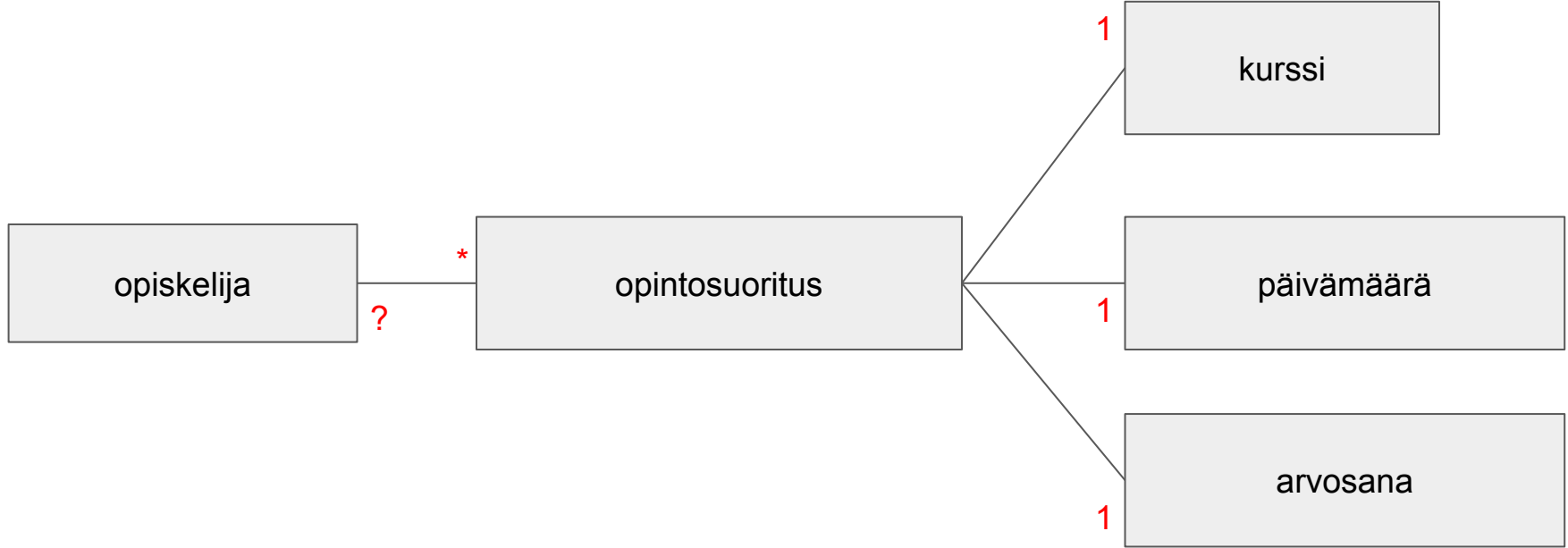
“Opiskelijalla on opintosuorituksia.  
Jokainen opintosuoritus sisältää  
tiedon kurssista, päivämäärästä sekä  
arvosanasta.”

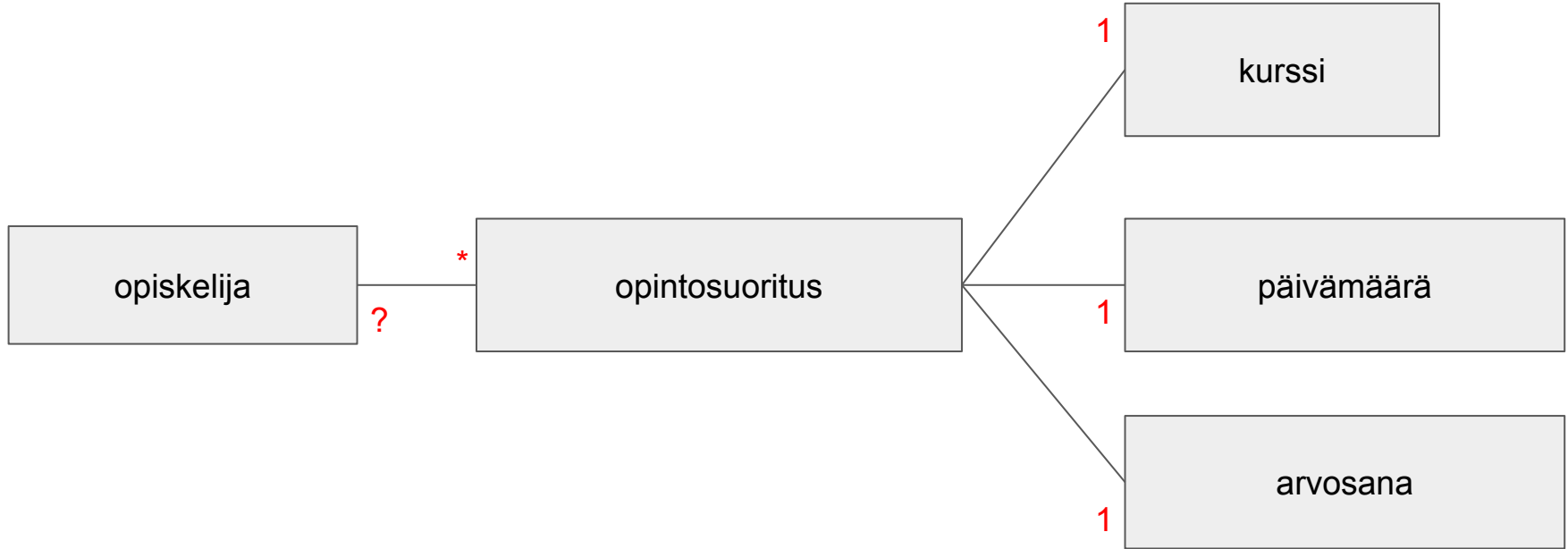


1. Käsitteiden tunnistaminen  
opiskelija, opintosuoritus, kurssi,  
päivämäärä, arvosana
2. Yhteyksien tunnistaminen
3. Osallistumisrajoitteiden tunnistaminen





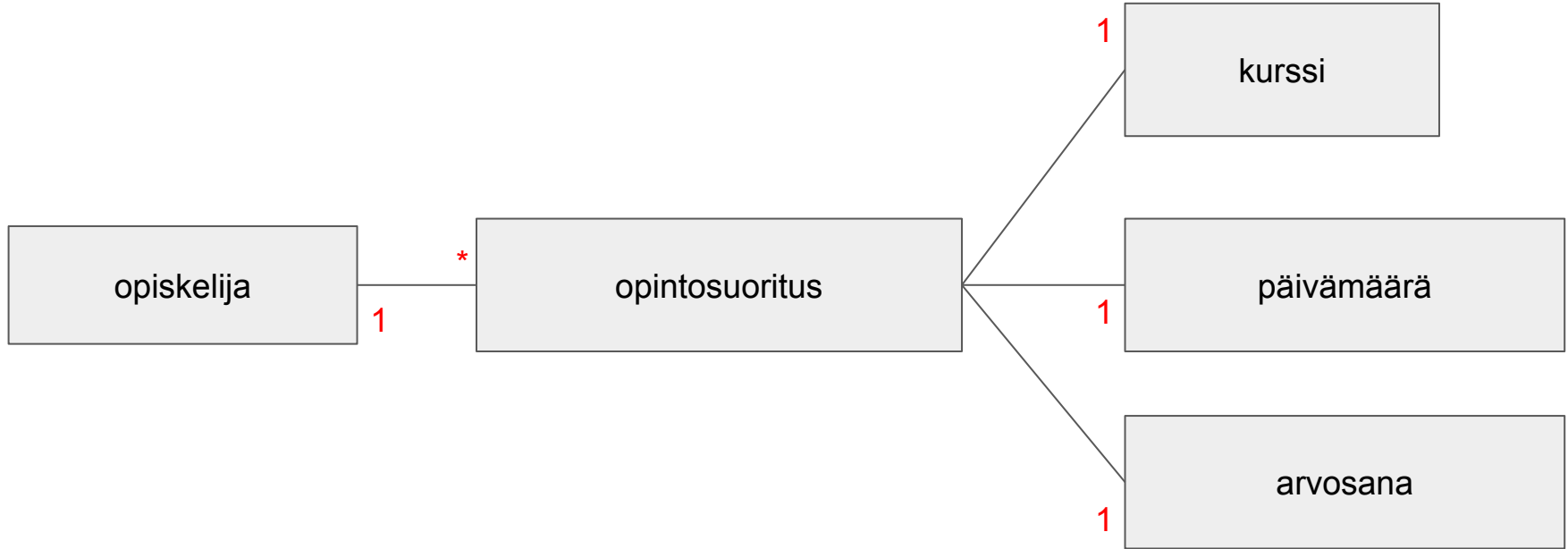




Vaihtoehdot:

1 = “opintosuoritukseen liittyy aina yksi opiskelija”

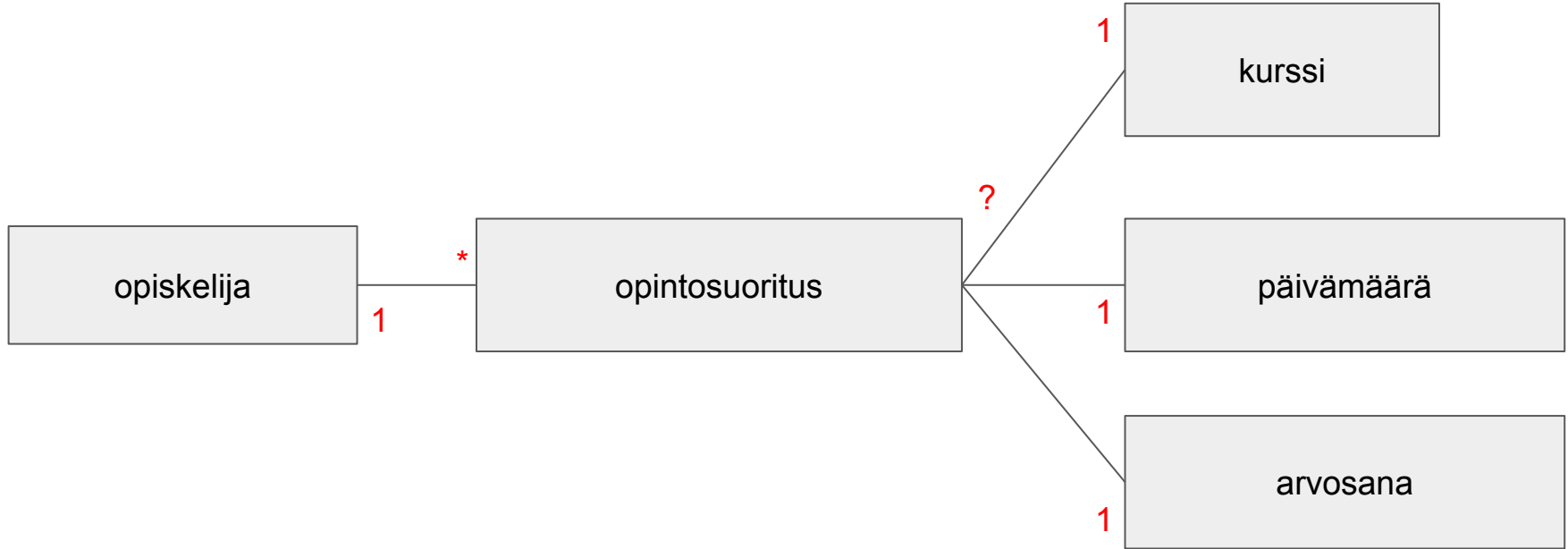
\* = “opintosuoritukseen liittyy yksi tai useampi opiskelija”

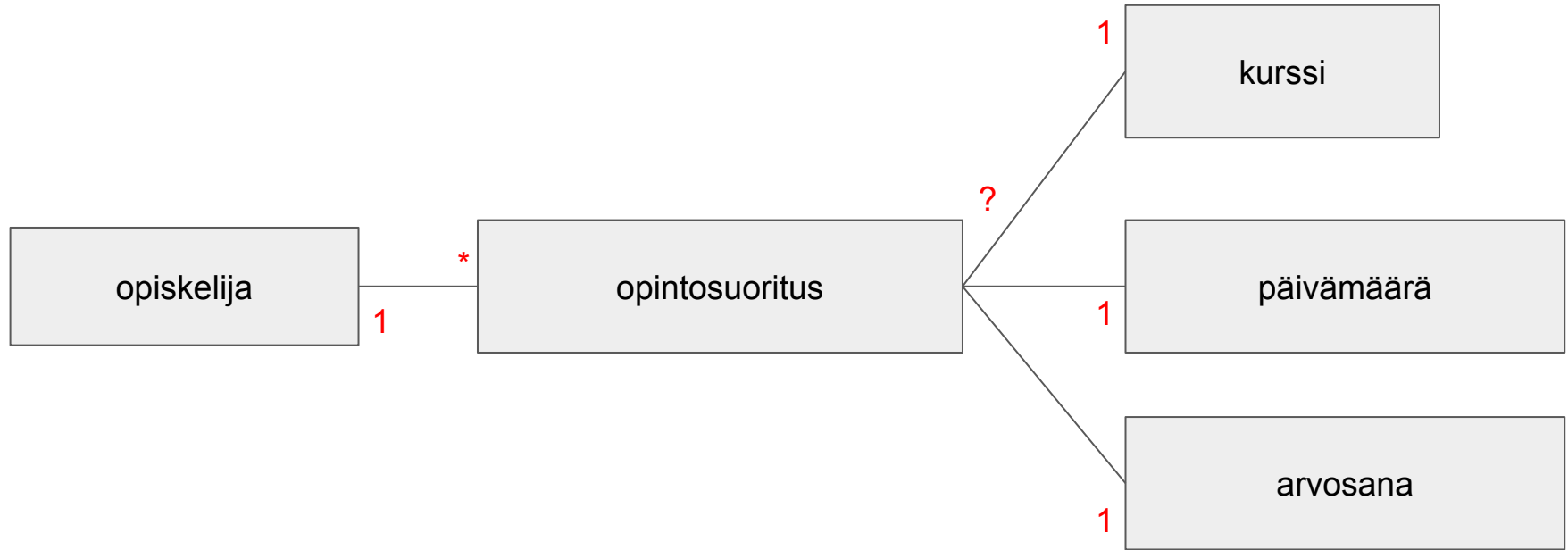


Vaihtoehdot:

1 = “opintosuoritukseen liittyy aina yksi opiskelija”

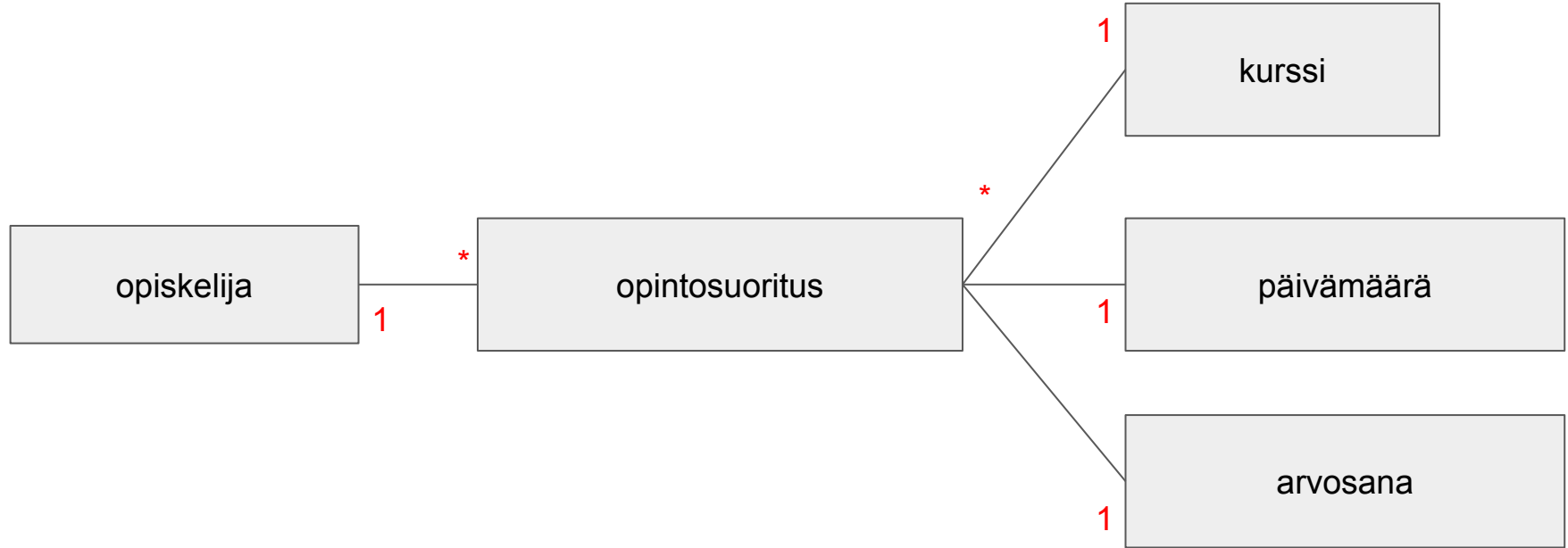
\* = “opintosuoritukseen liittyy yksi tai useampi opiskelija”





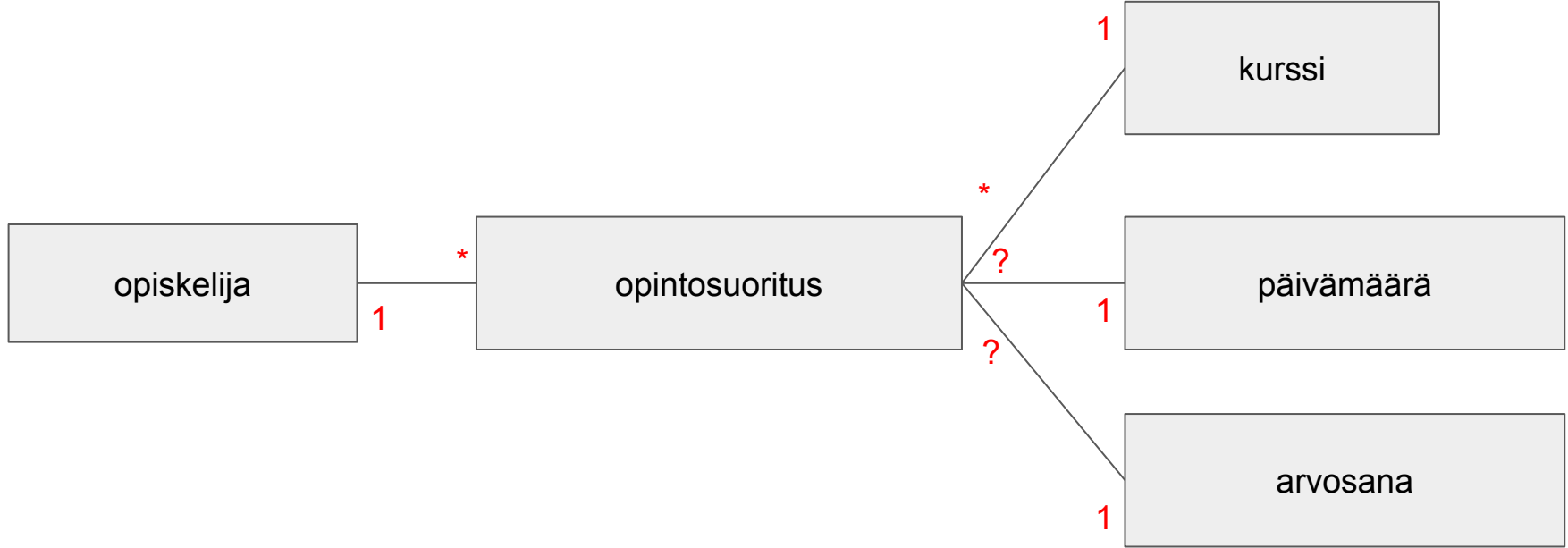
Vaihtoehdot:

- 1 = “kurssiin liittyy aina yksi opintosuoritus (= kurssin voi suorittaa korkeintaan kerran)”
- \* = “kurssiin liittyy yksi tai useampi opiskelija (= kurssin voi suorittaa monta kertaa)”

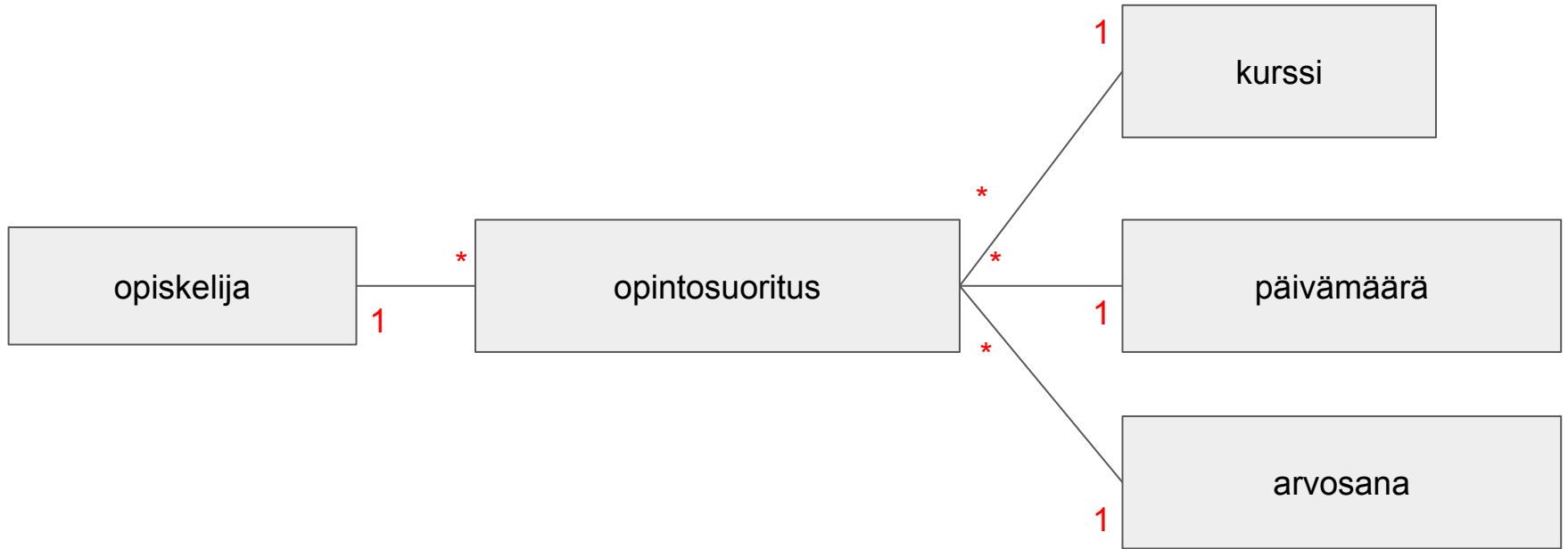


Vaihtoehdot:

- 1 = “kurssiin liittyy aina yksi opintosuoritus (= kurssin voi suorittaa korkeintaan kerran)”
- \* = “kurssiin liittyy yksi tai useampi opiskelija (= kurssin voi suorittaa monta kertaa)”







# Käsiteanalyysi

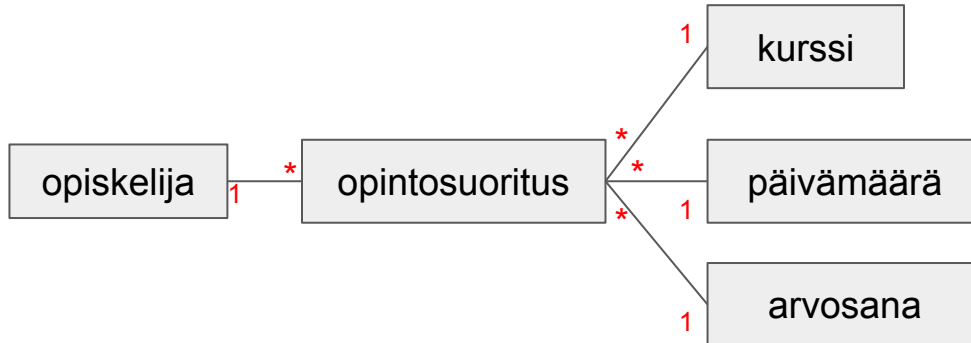
“Opiskelijalla on opintosuorituksia.  
Jokainen opintosuoritus sisältää  
tiedon kurssista, päivämäärästä sekä  
arvosanasta.”

monta

yksi

yksi

yksi

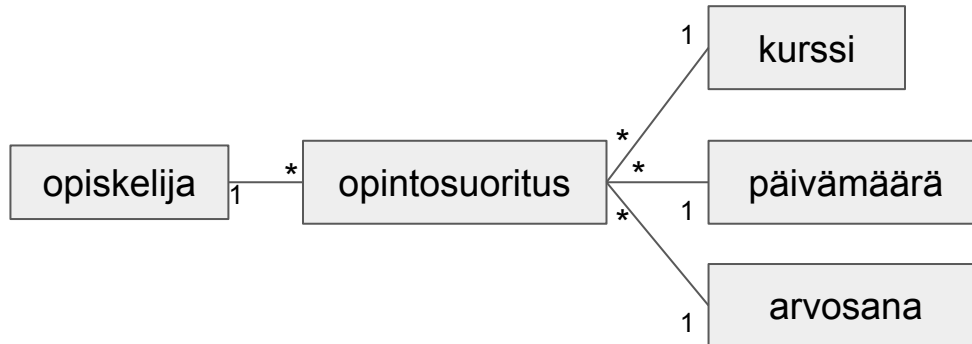


1. Käsitteiden tunnistaminen  
opiskelija, opintosuoritus, kurssi,  
päivämäärä, arvosana
2. Yhteyksien tunnistaminen
3. Osallistumisrajoitteiden tunnistaminen

# Käsiteanalyysi

“Opiskelijalla on opintosuorituksia.  
Jokainen opintosuoritus sisältää  
tiedon kurssista, päivämäärästä sekä  
arvosanasta.”

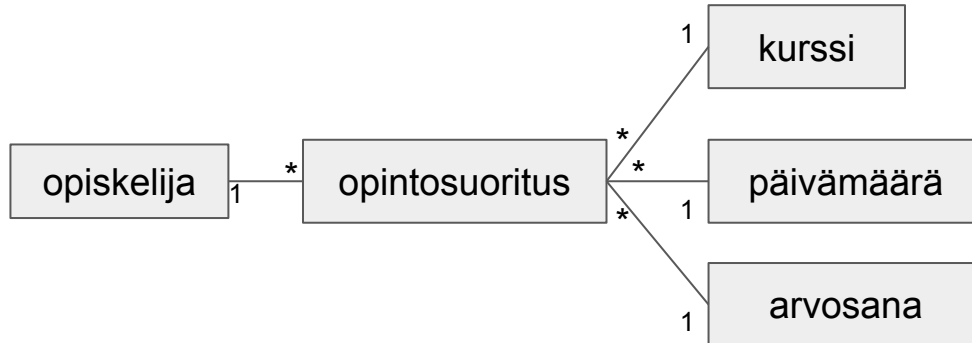
1. Käsitteiden tunnistaminen  
opiskelija, opintosuoritus, kurssi,  
päivämäärä, arvosana
2. Yhteyksien tunnistaminen
3. Osallistumisrajoitteiden tunnistaminen
4. Attribuuttien tunnistaminen



# Käsiteanalyysi

“Opiskelijalla on opintosuorituksia.  
Jokainen opintosuoritus **sisältää**  
tiedon kurssista, päivämäärästä sekä  
arvosanasta.”

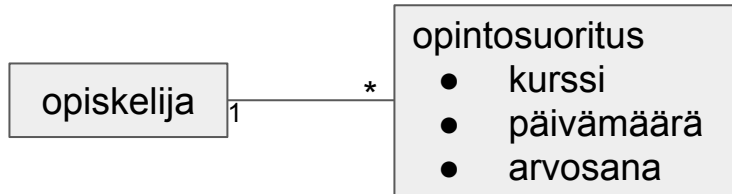
1. Käsitteiden tunnistaminen  
opiskelija, opintosuoritus, kurssi,  
päivämäärä, arvosana
2. Yhteyksien tunnistaminen
3. Osallistumisrajoitteiden tunnistaminen
4. Attribuuttien tunnistaminen



# Käsiteanalyysi

“Opiskelijalla on opintosuorituksia.  
Jokainen opintosuoritus **sisältää**  
tiedon kurssista, päivämäärästä sekä  
arvosanasta.”

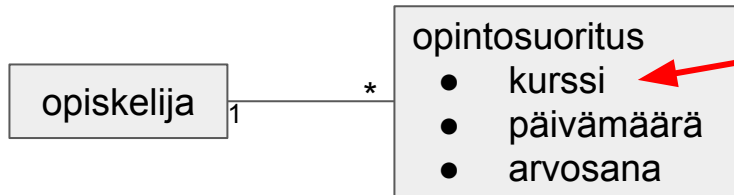
1. Käsitteiden tunnistaminen  
opiskelija, opintosuoritus, kurssi,  
päivämäärä, arvosana
2. Yhteyksien tunnistaminen
3. Osallistumisrajoitteiden tunnistaminen
4. Attribuuttien tunnistaminen



# Käsiteanalyysi

“Opiskelijalla on opintosuorituksia.  
Jokainen opintosuoritus **sisältää**  
tiedon kurssista, päivämäärästä sekä  
arvosanasta.”

1. Käsitteiden tunnistaminen  
opiskelija, opintosuoritus, kurssi,  
päivämäärä, arvosana
2. Yhteyksien tunnistaminen
3. Osallistumisrajoitteiden tunnistaminen
4. Attribuuttien tunnistaminen

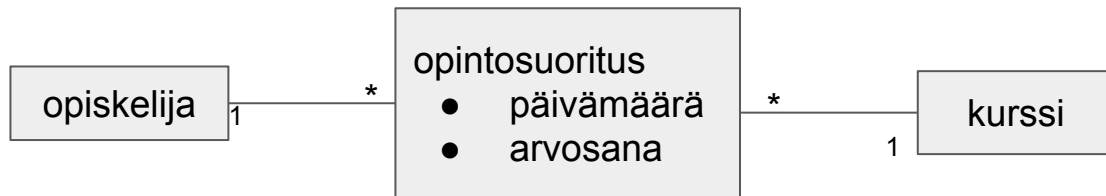


Nyt jokaisessa opintosuorituksessa tietty kurssi,  
kurssin nimi kirjoitettaisiin esim. merkkijonona

# Käsiteanalyysi

“Opiskelijalla on opintosuorituksia.  
Jokainen opintosuoritus **sisältää**  
tiedon kurssista, päivämäärästä sekä  
arvosanasta.”

1. Käsitteiden tunnistaminen  
opiskelija, opintosuoritus, kurssi,  
päivämäärä, arvosana
2. Yhteyksien tunnistaminen
3. Osallistumisrajoitteiden tunnistaminen
4. Attribuuttien tunnistaminen

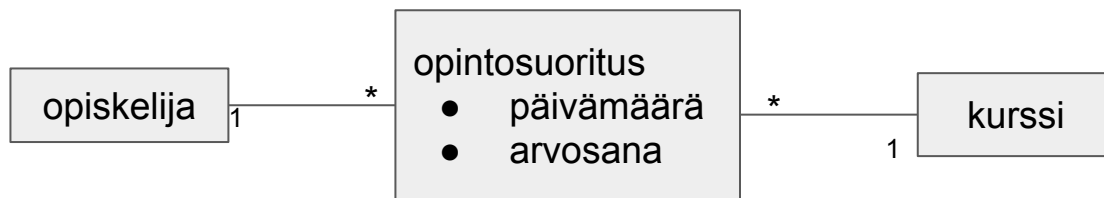


... toinen vaihtoehto...

# Käsiteanalyysi

“Opiskelijalla on opintosuorituksia.  
Jokainen opintosuoritus sisältää  
tiedon kurssista, päivämäärästä sekä  
arvosanasta.”

1. Käsitteiden tunnistaminen  
opiskelija, opintosuoritus, kurssi,  
päivämäärä, arvosana
2. Yhteyksien tunnistaminen
3. Osallistumisrajoitteiden tunnistaminen
4. Attribuuttien tunnistaminen
5. Yleistä ja eriyttä käsitteitä

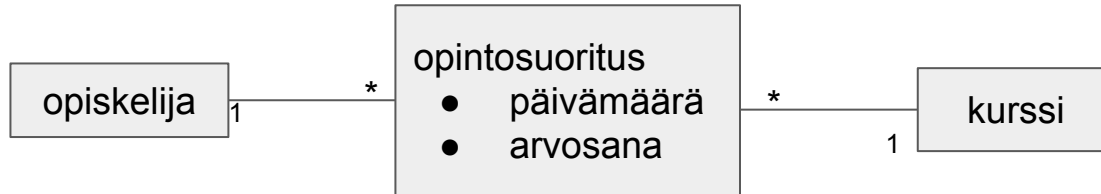




# Käsiteanalyysi

“Opiskelijalla on opintosuorituksia.  
Jokainen opintosuoritus sisältää  
tiedon kurssista, päivämäärästä sekä  
arvosanasta.”

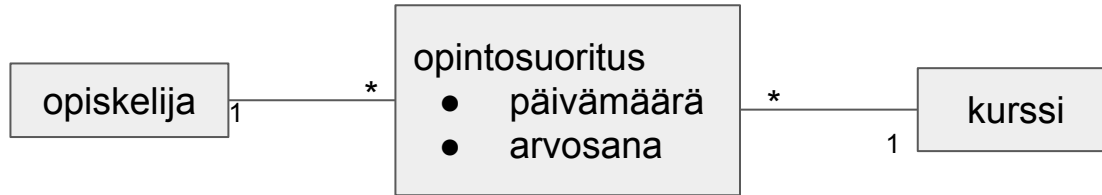
1. Käsitteiden tunnistaminen  
opiskelija, opintosuoritus, kurssi,  
päivämäärä, arvosana
2. Yhteyksien tunnistaminen
3. Osallistumisrajoitteiden tunnistaminen
4. Attribuuttien tunnistaminen
5. Yleistä ja eriytä käsitteitä (ei tehtävää)



# Käsiteanalyysi

“Opiskelijalla on opintosuorituksia.  
Jokainen opintosuoritus sisältää  
tiedon kurssista, päivämäärästä sekä  
arvosanasta.”

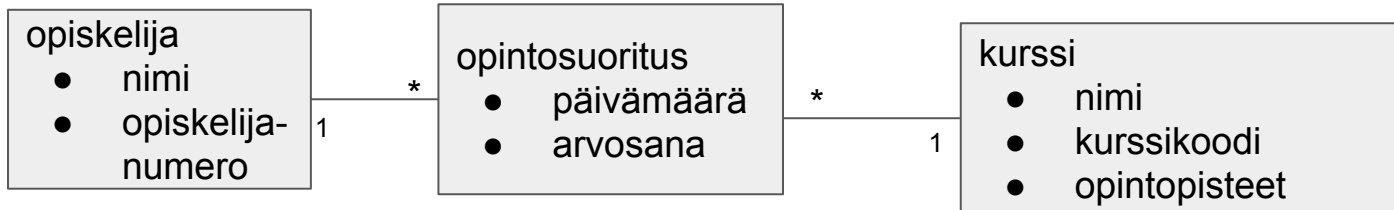
1. Käsitteiden tunnistaminen  
opiskelija, opintosuoritus, kurssi,  
päivämäärä, arvosana
2. Yhteyksien tunnistaminen
3. Osallistumisrajoitteiden tunnistaminen
4. Attribuuttien tunnistaminen
5. Yleistä ja eriytä käsitteitä (ei tehtävää)
6. Tunnettujen attribuuttien lisääminen



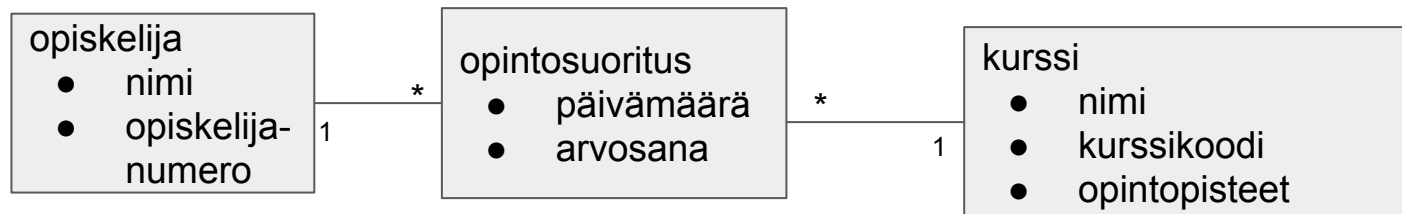
# Käsiteanalyysi

“Opiskelijalla on opintosuorituksia.  
Jokainen opintosuoritus sisältää  
tiedon kurssista, päivämäärästä sekä  
arvosanasta.”

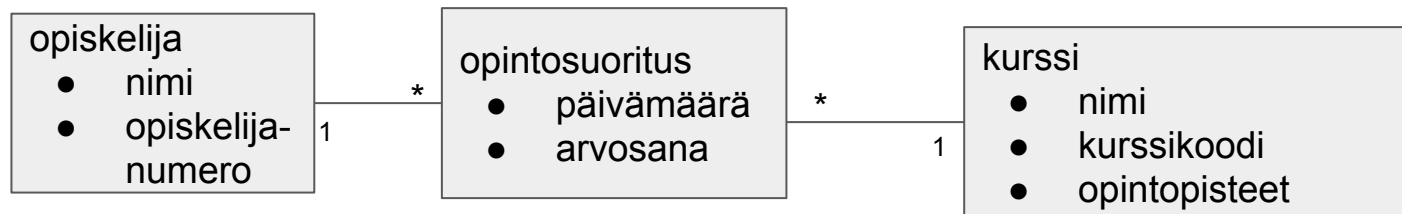
1. Käsitteiden tunnistaminen  
opiskelija, opintosuoritus, kurssi,  
päivämäärä, arvosana
2. Yhteyksien tunnistaminen
3. Osallistumisrajoitteiden tunnistaminen
4. Attribuuttien tunnistaminen
5. Yleistä ja eriytä käsitteitä (ei tehtävää)
6. Tunnettujen attribuuttien lisääminen



# Lopputulos



# Lopputulos → tietokantakaavioksi

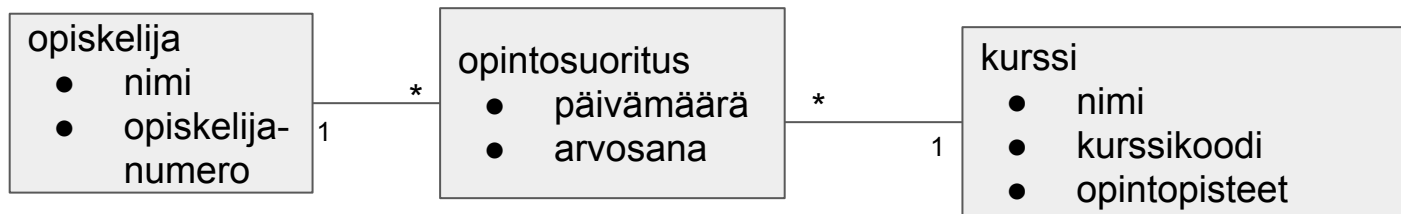


# Tietokantakaavio (relaatiokaavio)

- Tietokantaan luotavan tietokannan kuvaus (skeema).
- Askeleet:
  1. Määrittele käsitteille käsitteen yksilöivä pääavain “primary key”
  2. Muunna monesta moneen (\* -- \*) yhteydet yhdestä moneen -muotoisiksi lisäämällä liitostaulu
  3. Lisää "moneen"-yhteyden päädyssä olevaan tauluun viiteavain “foreign key”

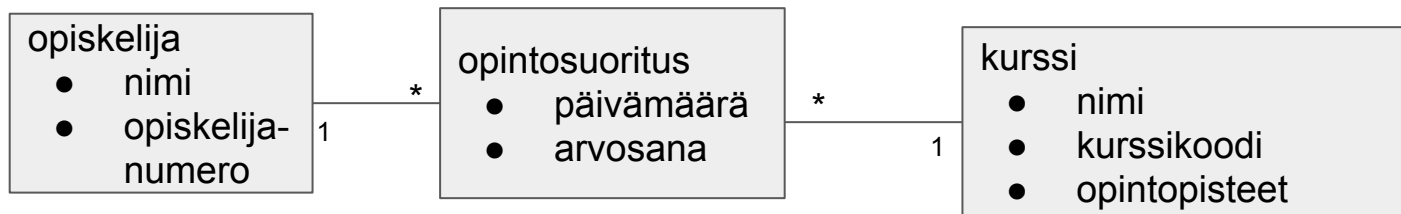
# Askel 1. Määrittele pääavaimet

## Lähtökohta



# Askel 1. Määrittele pääavaimet

## Lähtökohta

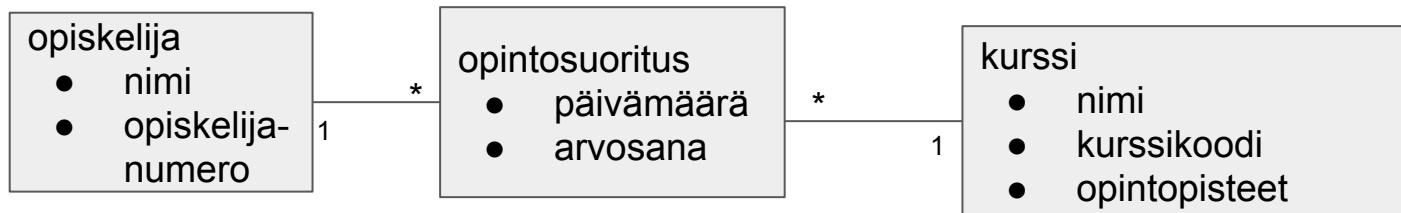


suoraviivainen tapa:  
jokaiselle käsitteelle  
uusi attribuutti "id"



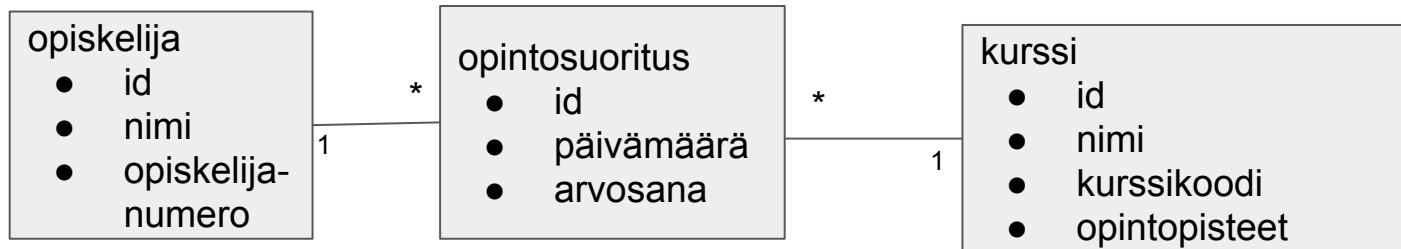
# Askel 1. Määrittele pääavaimet

## Lähtökohta



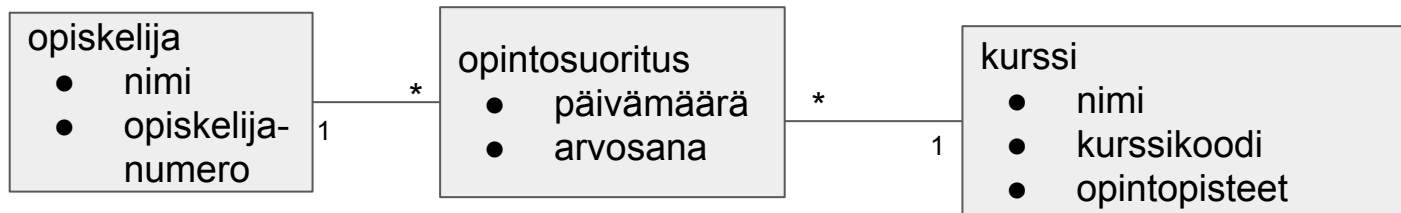
suoraviivainen tapa:  
jokaiselle käsitteelle  
uusi attribuutti "id"

## Lopputulos



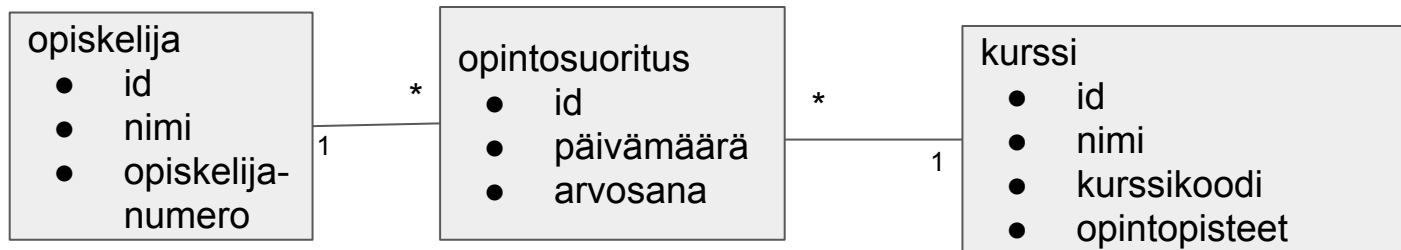
# Askel 1. Määrittele pääavaimet

## Lähtökohta



suoraviivainen tapa:  
jokaiselle käsitteelle  
uusi attribuutti "id"

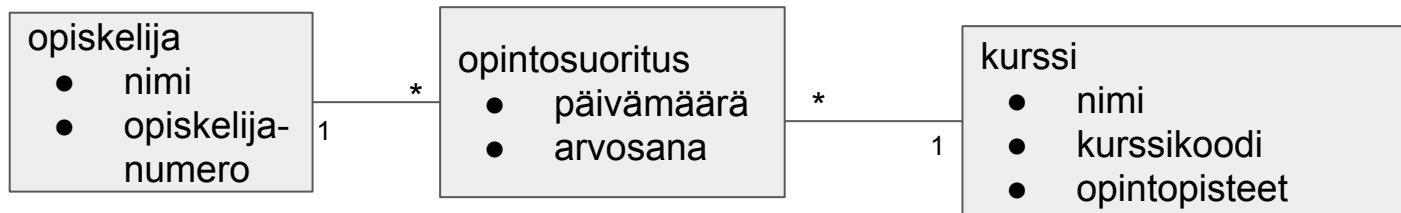
## Lopputulos



lisätään selkeyden  
vuoksi pääavaimille  
määre (pk)

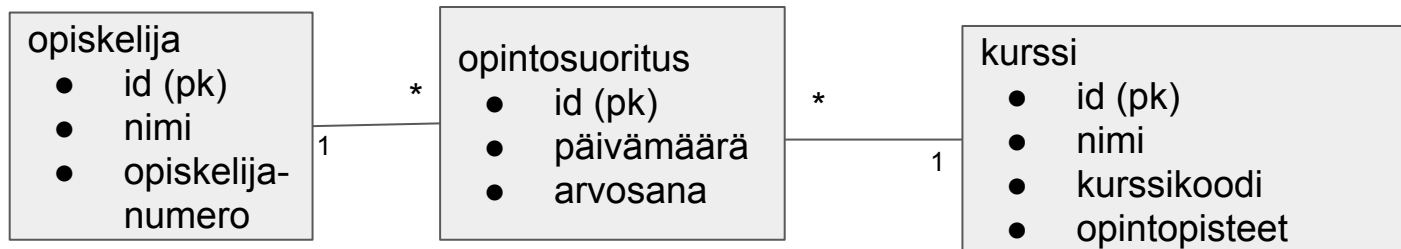
# Askel 1. Määrittele pääavaimet

## Lähtökohta



suoraviivainen tapa:  
jokaiselle käsitteelle  
uusi attribuutti "id"

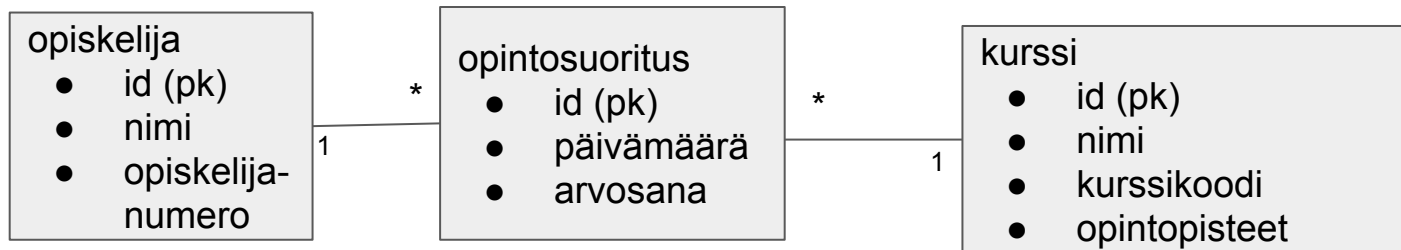
## Lopputulos



lisätään selkeyden  
vuoksi pääavaimille  
määre (pk)

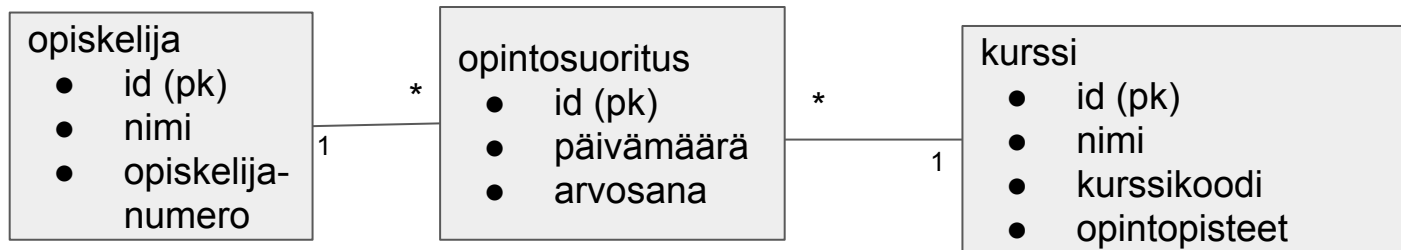
# Askel 2. Muunna monesta moneen -yhteydet

## Lähtökohta



# Askel 2. Muunna monesta moneen -yhteydet

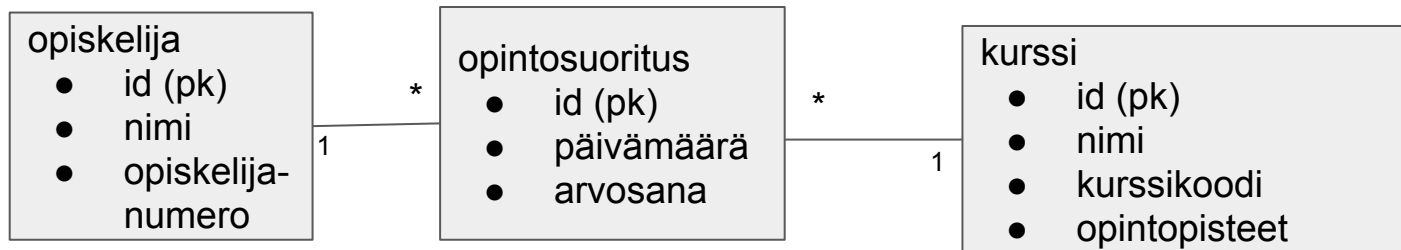
## Lähtökohta



ei monesta moneen  
-yhteyksiä (\* -- \*)

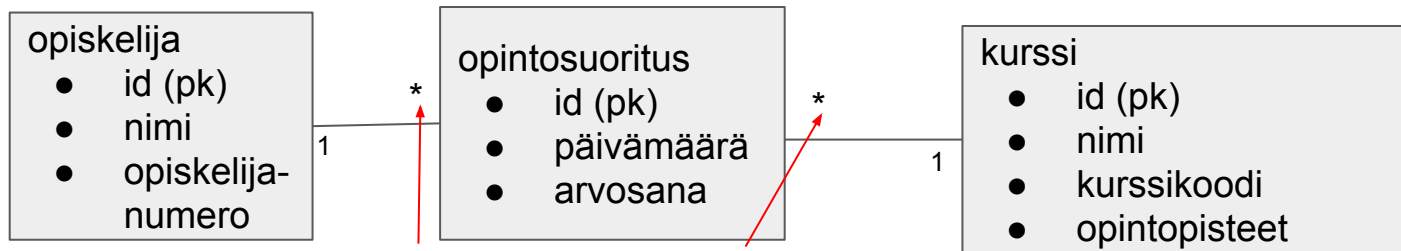
# Askel 3. Muunna monesta yhteen -yhteydet

## Lähtökohta



# Askel 3. Muunna monesta yhteen -yhteydet

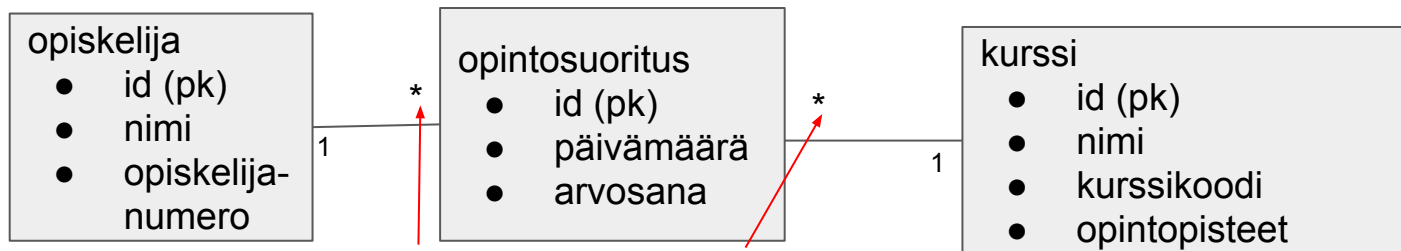
## Lähtökohta



monesta yhteen (tai yhdestä moneen)

# Askel 3. Muunna monesta yhteen -yhteydet

## Lähtökohta



monesta yhteen (tai yhdestä moneen)

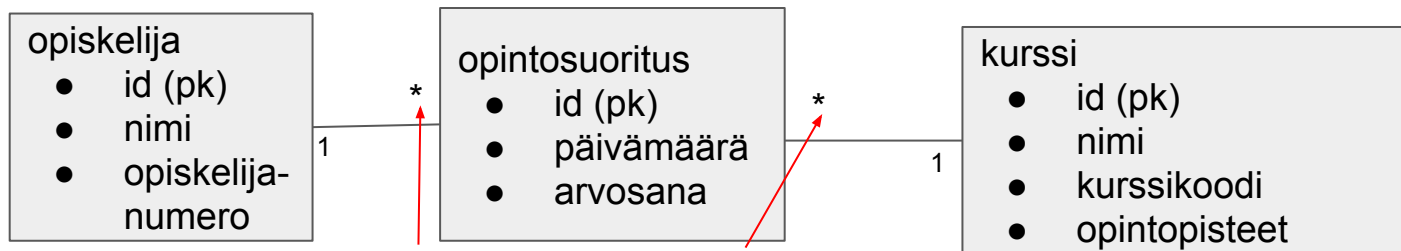
lisätään tähden  
päätyyn viiteavain

viiteavainten  
nimentä usein  
taulu\_id



# Askel 3. Muunna monesta yhteen -yhteydet

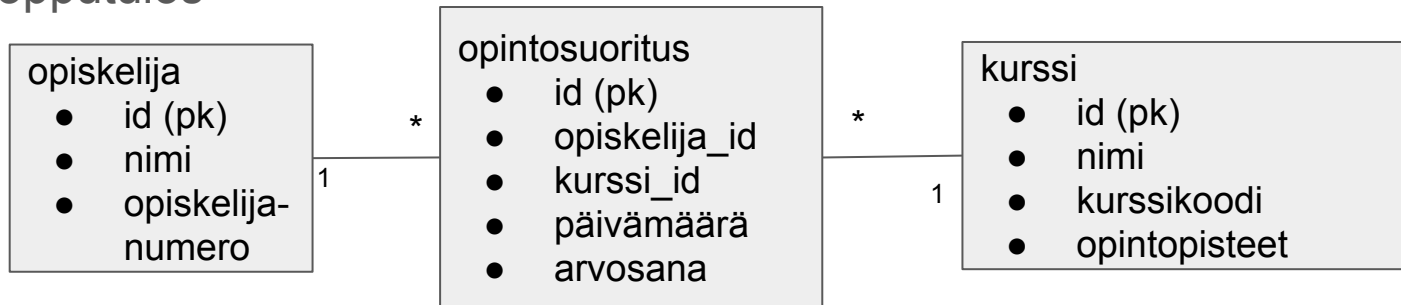
## Lähtökohta



monesta yhteen (tai yhdestä moneen)

lisätään tähden  
päätyyn viiteavain

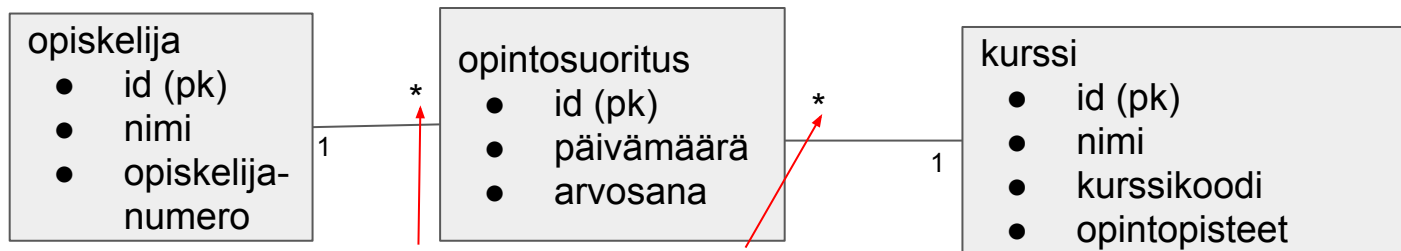
## Lopputulos



viiteavainten  
nimentä usein  
taulu\_id

# Askel 3. Muunna monesta yhteen -yhteydet

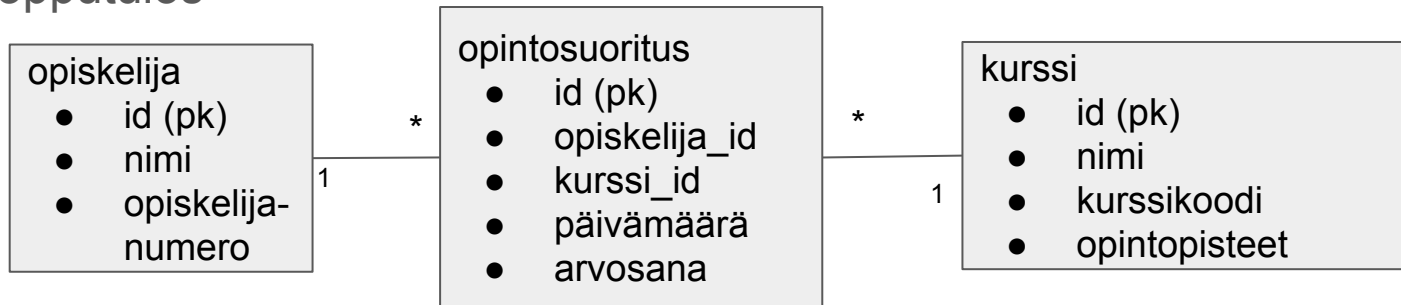
## Lähtökohta



monesta yhteen (tai yhdestä moneen)

lisätään tähden  
päätyyn viiteavain

## Lopputulos

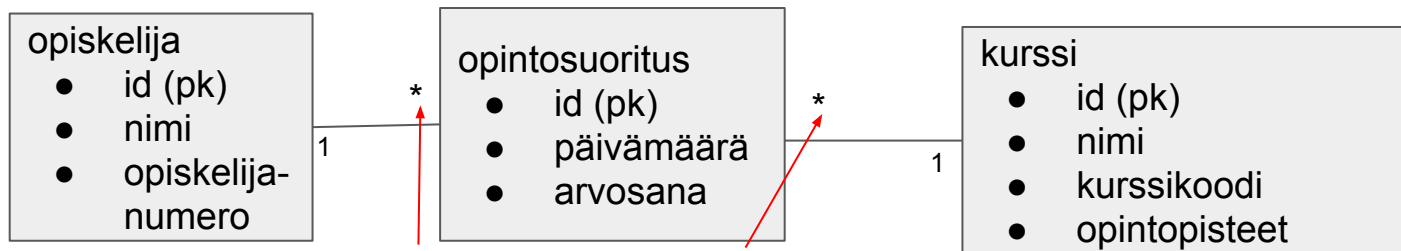


viiteavainten  
nimentä usein  
taulu\_id

lisätään vielä (fk)  
viiteavaimiin  
selkeyden vuoksi

# Askel 3. Muunna monesta yhteen -yhteydet

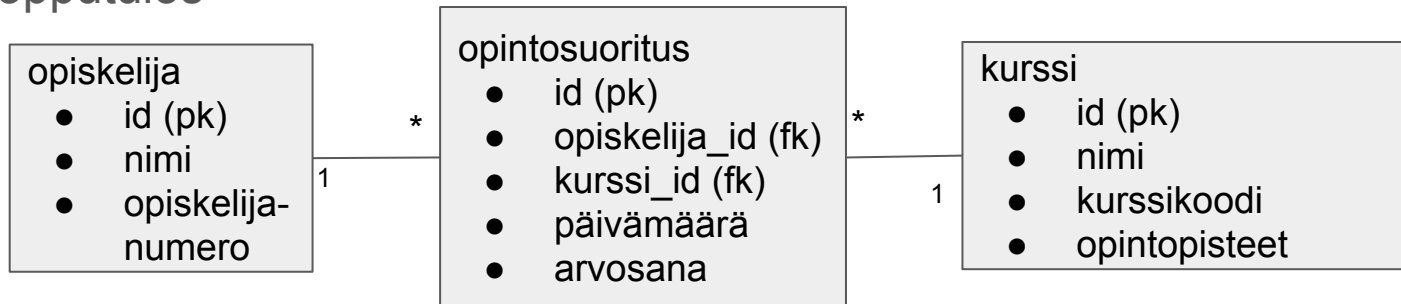
## Lähtökohta



monesta yhteen (tai yhdestä moneen)

lisätään tähden  
päätyyn viiteavain

## Lopputulos

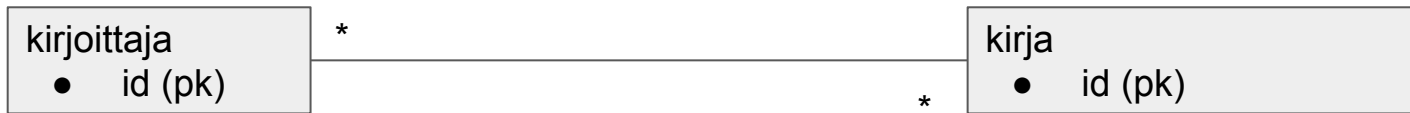


viiteavainten  
nimentä usein  
taulu\_id

lisätään vielä (fk)  
viiteavaimiin  
selkeyden vuoksi

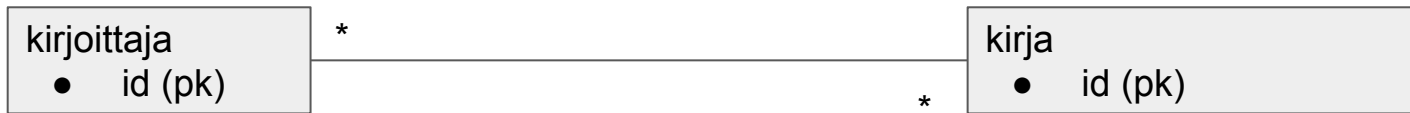
Lyhyt monesta moneen esimerkki

*“Kirjoittajalla voi olla monta kirjaa.  
Kirjalla voi olla monta kirjoittajaa.*



pääavaimet lisätty --  
ensimmäinen askel  
käyty jo läpi

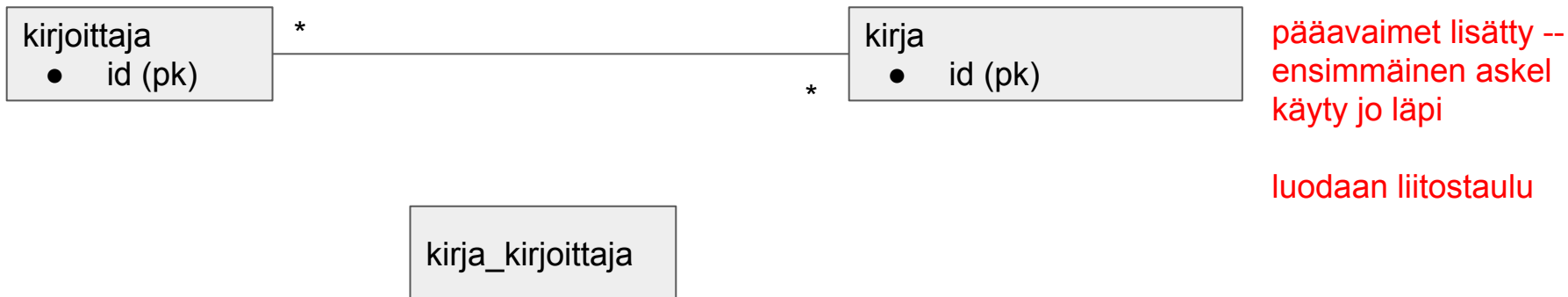
*“Kirjoittajalla voi olla monta kirjaa.  
Kirjalla voi olla monta kirjoittajaa.*



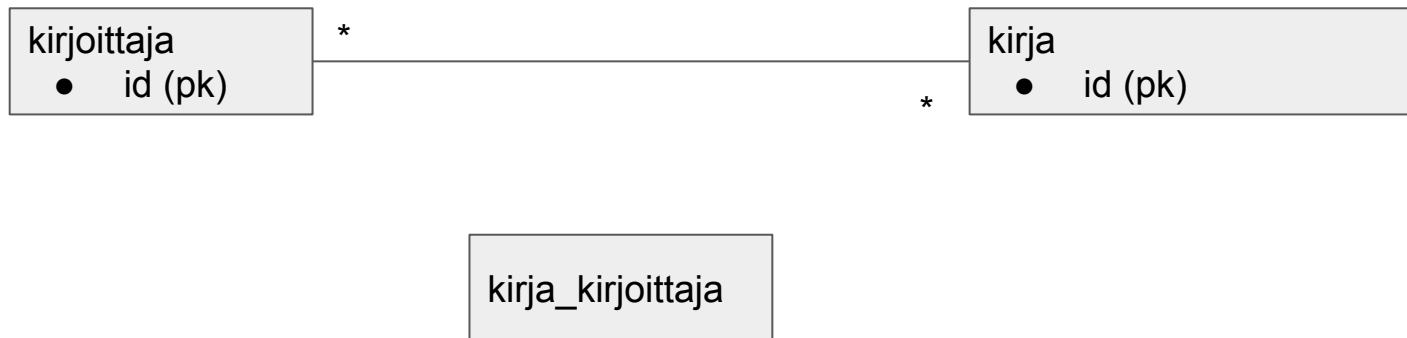
pääavaimet lisätty --  
ensimmäinen askel  
käyty jo läpi

luodaan liitostaulu

*“Kirjoittajalla voi olla monta kirjaa.  
Kirjalla voi olla monta kirjoittajaa.*



*“Kirjoittajalla voi olla monta kirjaa.  
Kirjalla voi olla monta kirjoittajaa.*



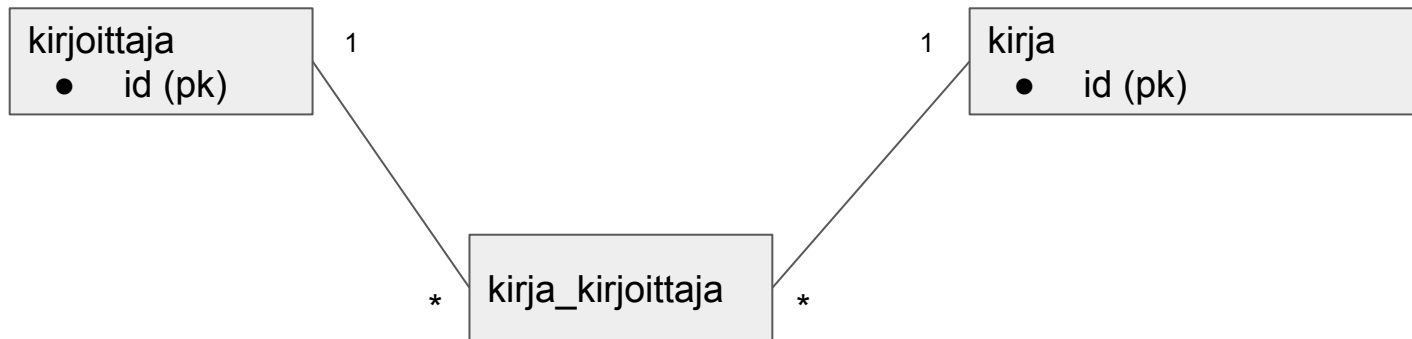
pääavaimet lisätty --  
ensimmäinen askel  
käyty jo läpi

luodaan liitostaulu

monesta moneen  
-yhteys muuttuu  
kahdeksi yhdestä  
moneen yhteydeksi



*“Kirjoittajalla voi olla monta kirjaa.  
Kirjalla voi olla monta kirjoittajaa.*

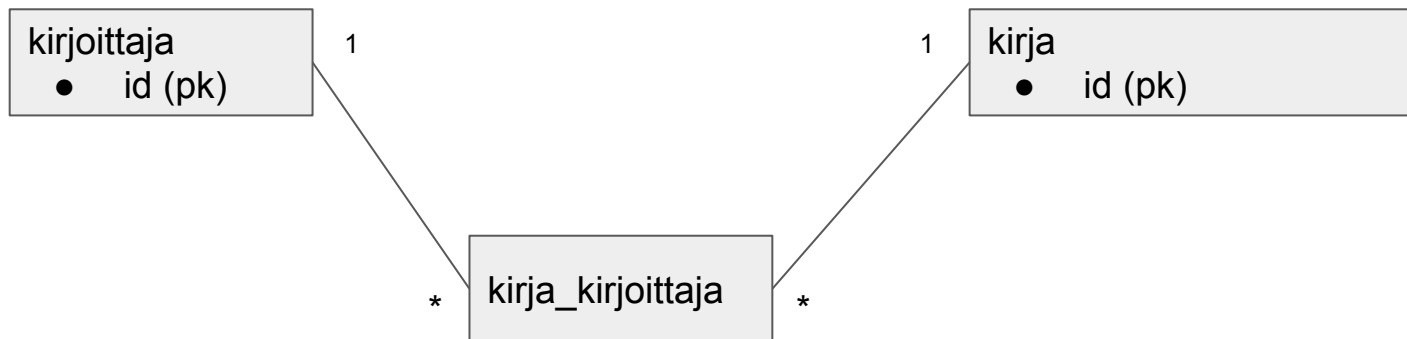


pääavaimet lisätty --  
ensimmäinen askel  
käyty jo läpi

luodaan liitostaulu

monesta moneen  
-yhteys muuttuu  
kahdeksi yhdestä  
moneen yhteydeksi

*“Kirjoittajalla voi olla monta kirjaa.  
Kirjalla voi olla monta kirjoittajaa.*



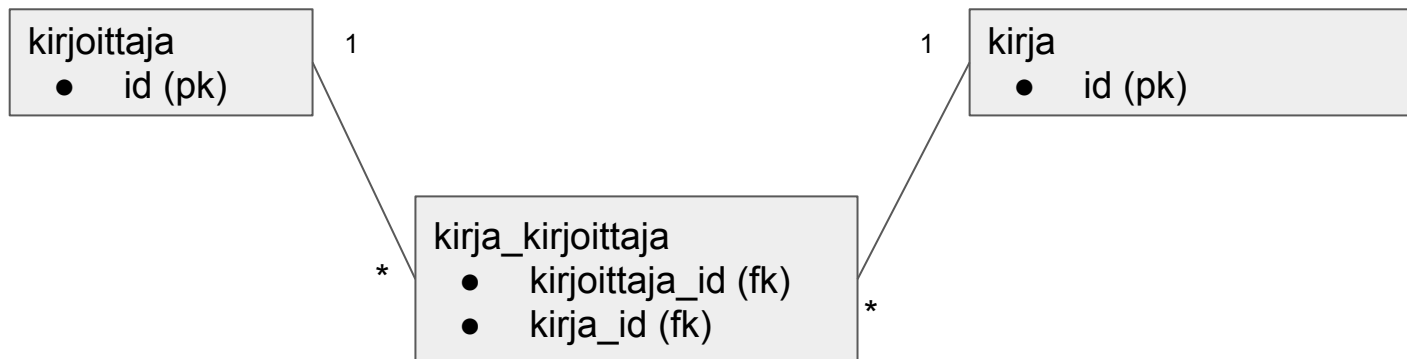
pääavaimet lisätty --  
ensimmäinen askel  
käyty jo läpi

luodaan liitostaulu

monesta moneen  
-yhteys muuttuu  
kahdeksi yhdestä  
moneen yhteydeksi

→ jatketaan  
purkamalla yhdestä  
moneen -yhteydet

*“Kirjoittajalla voi olla monta kirjaa.  
Kirjalla voi olla monta kirjoittajaa.*



pääavaimet lisätty --  
ensimmäinen askel  
käyty jo läpi

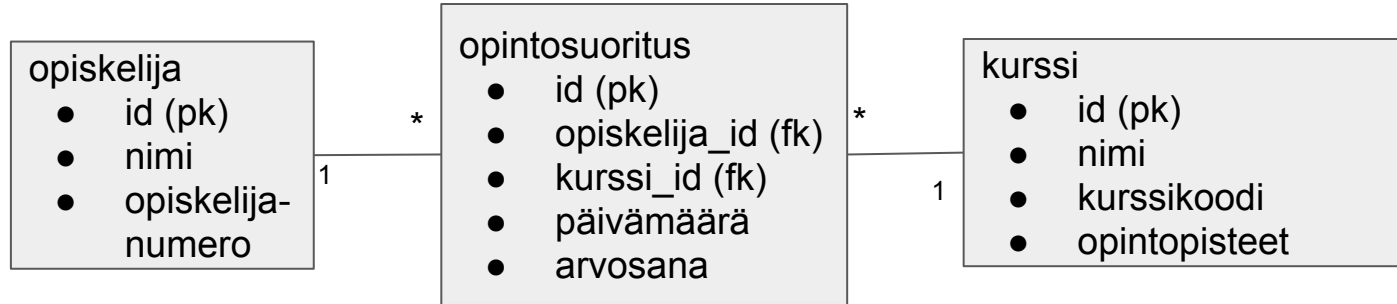
luodaan liitostaulu

monesta moneen  
-yhteys muuttuu  
kahdeksi yhdestä  
moneen yhteydeksi

→ jatketaan  
purkamalla yhdestä  
moneen -yhteydet

Tietokantataulut tietokantakaaviosta

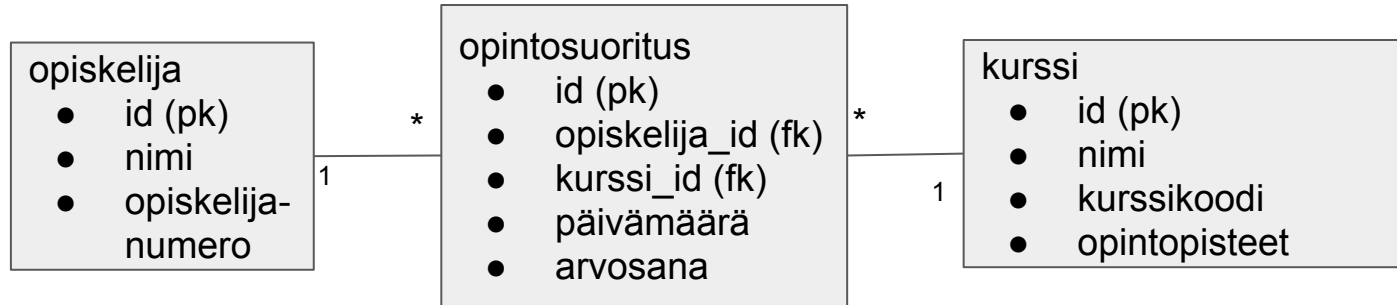
*Käytännössä kaavion kirjoitus  
SQL-kielen CREATE TABLE  
-lauseiksi*



# Käytännössä kaavion kirjoitus

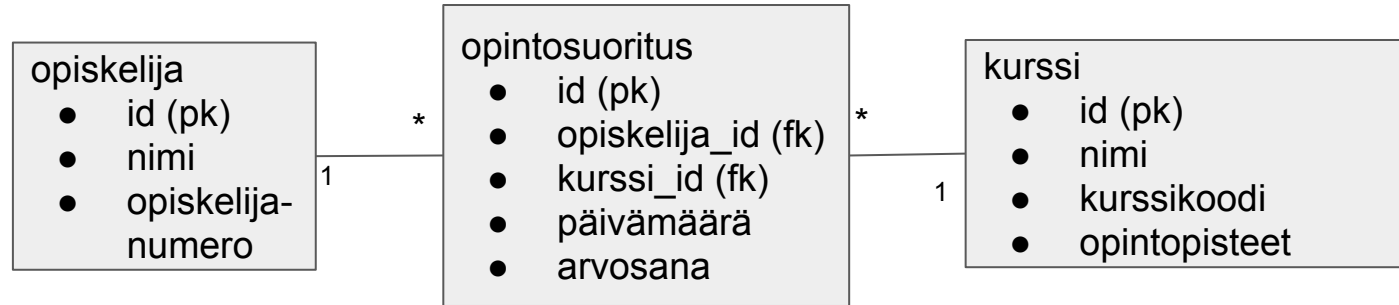
## SQL-kielen CREATE TABLE

### -lauseiksi



```
CREATE TABLE Opiskelija (  
    id SERIAL PRIMARY KEY,  
    nimi TEXT,  
    opiskelijanumero TEXT  
)
```

*Käytännössä kaavion kirjoitus  
SQL-kielen CREATE TABLE  
-lauseiksi*

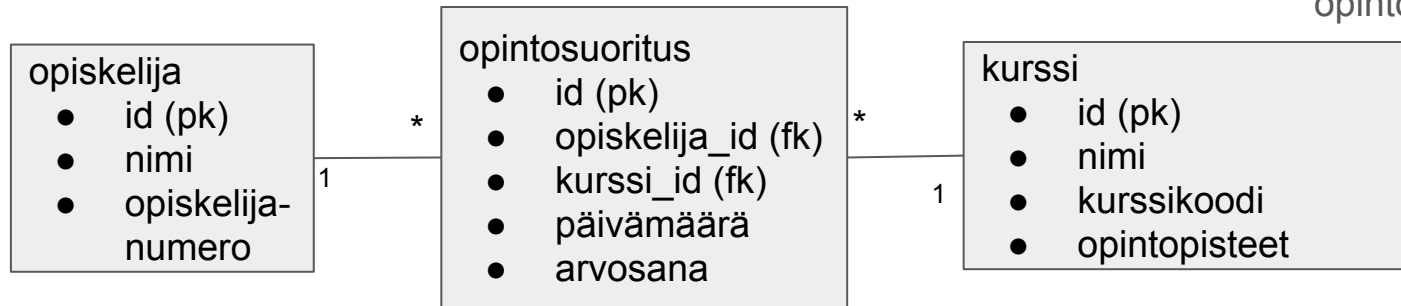


```
CREATE TABLE Opiskelija (  
    id SERIAL PRIMARY KEY,  
    nimi TEXT,  
    opiskelijanumero TEXT  
)
```

```
CREATE TABLE Opintosuoritus (  
    id SERIAL PRIMARY KEY,  
    opiskelija_id INTEGER REFERENCES opiskelija(id),  
    kurssi_id INTEGER REFERENCES kurssi(id),  
    paivamaara DATE,  
    arvosana INTEGER  
)
```

# Käytännössä kaavion kirjoitus SQL-kielen CREATE TABLE -lauseiksi

```
CREATE TABLE Kurssi (  
    id SERIAL PRIMARY KEY,  
    nimi TEXT,  
    opiskelijanumero TEXT,  
    opintopisteet INTEGER)
```



```
CREATE TABLE Opiskelija (  
    id SERIAL PRIMARY KEY,  
    nimi TEXT,  
    opiskelijanumero TEXT  
)
```

```
CREATE TABLE Opintosuoritus (  
    id SERIAL PRIMARY KEY,  
    opiskelija_id INTEGER REFERENCES opiskelija(id),  
    kurssi_id INTEGER REFERENCES kurssi(id),  
    paivamaara DATE,  
    arvosana INTEGER  
)
```



# Hands-on

- Tietokannan rakenteen hahmottelu - käsiteanalyysistä tietokannaksi:

*Olemme kehittämässä pikkujoulujen järjestämistä helpottavaa palvelua. Pikkujoulut järjestetään aina jossain osoitteessa ja pikkujouluihin tulee ilmoittautua. Ilmoittautumisen yhteydessä kerrotaan nimi ja puhelinnumero. Ilmoittautumisen yhteydessä kerrotaan myös mahdollisista ruokavalioon liittyvistä rajoitteista. Pikkujoulutapahtumaa luodessa käyttäjä voi valita tapahtumaan myös esiintyjiä. Palveluun lisätään esiintyjiä sitä mukaa kun saamme heitä mukaan.*

Kahvitauko

# Hahmotelman läpikäynti

Muutama sana kyselyiden tehokkuudesta

# Kyselyiden tehokkuudesta

- Tieto tallennetaan tietokantaan tietokannanhallintajärjestelmän määrittelemässä muodossa
- Kyselyn tehokkuuteen vaikuttavat mm. käytössä olevat hakurakenteet
- Tietokantaa ja kyselyiden tehokkuutta voidaan tarkastella erilaisilla työvälineillä (esim. kyselyihin liittyvä EXPLAIN-toiminnallisuus)

# Kyselyiden tehokkuudesta

- Kokeillaan kyselyä:

```
EXPLAIN SELECT users.* FROM users
```

```
  JOIN accounts ON users.id = accounts.user_id
```

```
  WHERE accounts.iban = 'GT71 6XGJ TNBH INI8 PXIN 6UXD ZPBI'
```

# Kyselyiden tehokkuudesta

- Kokeillaan kyselyä:

```
EXPLAIN SELECT users.* FROM users
```

```
  JOIN accounts ON users.id = accounts.user_id
```

```
 WHERE accounts.iban = 'GT71 6XGJ TNBH INI8 PXIN 6UXD ZPBI'
```

QUERY PLAN
Nested Loop (cost=0.28..94.31 rows=1 width=68)
-> Seq Scan on accounts (cost=0.00..86.00 rows=1 width=4)
Filter: ((iban)::text = 'GT71 6XGJ TNBH INI8 PXIN 6UXD ZPBI'::text)
-> Index Scan using users_pkey on users (cost=0.28..8.29 rows=1 width=68)
Index Cond: (id = accounts.user_id)

# Kyselyiden tehokkuudesta

- Kokeillaan kyselyä:

```
EXPLAIN SELECT users.* FROM users
```

```
JOIN accounts ON users.id = accounts.user_id
```

```
WHERE accounts.iban = 'GT71 6XGJ TNBH INI8 PXIN 6UXD ZPBI'
```

Käy taulu läpi  
rivi riviltä

QUERY PLAN	
Nested Loop (cost=0.28..94.31 rows=1 width=68)	
Seq Scan on accounts (cost=0.00..86.00 rows=1 width=4)	
Filter: ((iban)::text = 'GT71 6XGJ TNBH INI8 PXIN 6UXD ZPBI'::text)	
-> Index Scan using users_pkey on users (cost=0.28..8.29 rows=1 width=68)	
Index Cond: (id = accounts.user_id)	



# Kyselyiden tehokkuudesta

- Kokeillaan kyselyä:

```
EXPLAIN SELECT users.* FROM users
```

```
JOIN accounts ON users.id = accounts.user_id
```

```
WHERE accounts.iban = 'GT71 6XGJ TNBH INI8 PXIN 6UXD ZPBI'
```

Käy taulu läpi  
rivi riviltä

Etsi haku-  
rakenteesta

QUERY PLAN	
Nested Loop (cost=0.28..94.31 rows=1 width=68)	
Seq Scan on accounts (cost=0.00..86.00 rows=1 width=4)	
Filter: ((iban)::text = 'GT71 6XGJ TNBH INI8 PXIN 6UXD ZPBI'::text)	
Index Scan using users_pkey on users (cost=0.28..8.29 rows=1 width=68)	
Index Cond: (id = accounts.user_id)	

# Kyselyiden tehokkuudesta

- Tietokantataulun läpikäynti rivi riviltä ei yleensä ottaen ole kovin tehokasta.
  - Toisaalta, “Premature optimization is the root of all evil” - Dijkstra
- Hakurakenteet auttavat -- kokeillaan

```
CREATE INDEX ON accounts (iban)
```

# Kyselyiden tehokkuudesta

- Kokeillaan kyselyä uudestaan:

```
EXPLAIN SELECT users.* FROM users
```

```
  JOIN accounts ON users.id = accounts.user_id
```

```
 WHERE accounts.iban = 'GT71 6XGJ TNBH INI8 PXIN 6UXD ZPBI'
```

QUERY PLAN
Nested Loop (cost=0.56..16.61 rows=1 width=68)
-> Index Scan using accounts_iban_idx on accounts (cost=0.28..8.30 rows=1 width=4)
Index Cond: ((iban)::text = 'GT71 6XGJ TNBH INI8 PXIN 6UXD ZPBI'::text)
-> Index Scan using users_pkey on users (cost=0.28..8.29 rows=1 width=68)
Index Cond: (id = accounts.user_id)

# Kyselyiden tehokkuudesta

- Kokeillaan kyselyä uudestaan:

```
EXPLAIN SELECT users.* FROM users
```

```
JOIN accounts ON users.id = accounts.user_id
```

```
WHERE accounts.iban = 'GT71 6XGJ TNBH INI8 PXIN 6UXD ZPBI'
```

Etsi haku-  
rakenteesta



QUERY PLAN	
Nested Loop (cost=0.56..16.61 rows=1 width=68)	
Index Scan using accounts_iban_idx on accounts (cost=0.28..8.30 rows=1 width=4)	
Index Cond: ((iban)::text = 'GT71 6XGJ TNBH INI8 PXIN 6UXD ZPBI'::text)	
-> Index Scan using users_pkey on users (cost=0.28..8.29 rows=1 width=68)	
Index Cond: (id = accounts.user_id)	

# Kyselyiden tehokkuudesta

- Kokeillaan kyselyä uudestaan:

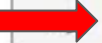

```
EXPLAIN SELECT users.* FROM users
```

```
JOIN accounts ON users.id = accounts.user_id
```

```
WHERE accounts.iban = 'GT71 6XGJ TNBH INI8 PXIN 6UXD ZPBI'
```

Etsi haku-  
rakenteesta

Etsi haku-  
rakenteesta

QUERY PLAN	
Nested Loop (cost=0.56..16.61 rows=1 width=68)	
	Index Scan using accounts_iban_idx on accounts (cost=0.28..8.30 rows=1 width=4) Index Cond: ((iban)::text = 'GT71 6XGJ TNBH INI8 PXIN 6UXD ZPBI'::text)
	Index Scan using users_pkey on users (cost=0.28..8.29 rows=1 width=68) Index Cond: (id = accounts.user_id)

# Hands-on

- Kokeile hakea käyttäjää sähköpostin perusteella
  - Lisää käyttäjän sähköpostille indeksi
  - Kokeile hakea käyttäjää sähköpostin perusteella
- 
- Kokeile hakea tilejä, joiden saldo on yli 90000
  - Lisää tilin saldolle indeksi
  - Kokeile hakea tilejä, joiden saldo on yli 90000

# SQL-kyselyn tulosten ryhmittely

# Tulosten ryhmittely

- Tulosten ryhmittely tapahtuu GROUP BY -avainsanalla
- Avainsanaa seuraa sarakkeet, joiden perusteella ryhmittely tapahtuu
- Sarakkeet esiintyvät (yleensä) myös näytettävien sarakkeiden listauksessa, joita seuraa ryhmään päätyville riveille tehtävä operaatio
- Esimerkiksi:

```
SELECT orders.contact, COUNT(*) FROM orders  
JOIN order_items  
ON orders.id = order_items.order_id  
GROUP BY orders.contact
```

orders:

<b><u>id</u></b>	<b><i>contact</i></b>
1	Arto
2	Lassi
3	...

order\_items:

<b><u>id</u></b>	<b><u>order_id</u></b>	<b><i>item</i></b>
1	1	Coffee
2	1	Coffee
3	2	Tea



# Hands-on

- Laske maakohtaiset tilien keskisaldot -- listauksessa tulee olla maa ja keskisaldo
- Laske maakohtaiset tilien saldojen summat
- Listaa maakohtaisten saldojen summat järjestyksessä suurimmasta pienimpään

# ORMit lyhyesti

Muutamia lakitekknisiä asioita

# Yhteenveto iltapäivästä

+ *seuraava lähipäivä lyhyesti*