

Tietokantasuunnittelusta

- Tietokantasuunnittelun pääperiaatteena on **tiedon toiston välttäminen**.
- Tiedon toistumiseen liittyy monenlaisia ongelmia
 - toistuva tieto vie 'turhaa' tilaa
 - ylläpito muodostuu hankalaksi
kaikki 'kopiot' päivitettävä
 - ylläpito-operaatioilla voi olla odottamattomia sivuvaikutuksia.
 - Toisaalta toistosta on etuakin – tiedon haku saattaa nopeutua

1

Tietokantasuunnittelusta

- Esimerkki taulusta, joka ei käyttyä hyvin: EMP_DEPT:

E_no	E_name	E_bdate	D_no	D_name	D_location
1	M.Seppä	1.3.59	3	Myynti	Helsinki
2	D.Leivo	4.10.40	3	Myynti	Helsinki
3	K.Koivu	30.1.66	4	Hallinto	Lahti
4	B.Oja	2.5.65	4	Hallinto	Lahti
5	O.Itä	10.2.55	6	Tuotanto	Helsinki

Avain: E_no

Jos O.Itä poistetaan, häviää tieto tuotanto-osastosta

Toistettava jokaisen osaston t:n kohdalla

Jos Hallinto muuttaa Espooseen om muutettava useita rivejä

2

Ei tarpeetonta toistoa

E_no	E_name	E_bdate	D_no
1	M.Seppä	1.3.59	3
2	D.Leivo	4.10.40	3
3	K.Koivu	30.1.66	4
4	B.Oja	2.5.65	4
5	O.Itä	10.2.55	6

D_no	D_name	D_location
3	Myynti	Helsinki
4	Hallinto	Lahti
6	Tuotanto	Helsinki

Parempi ratkaisu

3

Tietokantasuunnittelusta

- Tietokannan suunnittelun vaiheita ovat
 - tietosisällön kartoitus
 - käsiteanalyysi
 - kartoittamalla attribuutit ja niiden väliset riippuvuudet
 - loogisten rakenteiden suunnittelu ja
 - teknisten rakenteiden suunnittelu.
- Loogisen rakenteen suunnittelun tavoite on sijoittaa yhteenkuuluvat tiedot samaan taulu**

4

Muunnos käsitekaaviosta relaatiokaavioon

- Käsiteanalyysi tuottaa tietosisällön määrittelyn käsitekaavion muodossa
- Käsitekaavio kuvaa tällöin järjestelmän **pysyväisluonteisia** (persistent) tietoja - tietojen on säilyttävä ohjelman suorituskertojen välillä

5

Muunnos käsitekaaviosta relaatiokaavioon

- Lähtökohtana **normalisoitu** käsitekaavio = **tiettyjä sääntöjä noudattava kaavio**
 - Yhteydet näkyviä, ei attributeiksi piilotettuja
 - Kukin asia esitetään vain kertaalleen
 - Ei johdettavissa / pääteltävissä olevaa tietoja
 - Yhteydet määritelty oikeiden osapuolten välillä

6

Yhteydet piilotettu

- Olioiden välinen kytkentä esitetään aina yhteytenä, sitä ei piiloteta ominaisuudeksi (attribuutiksi)

Kytkeä on piilotettuna kurssin attribuuttiin opettajan_nimi

7

Yhteyden piilotus

- Olioiden välinen kytkentä esitetään aina yhteytenä, sitä ei piiloteta ominaisuudeksi (attribuutiksi)

8

Asiat esitetään vain kertaalleen

- Samaa asiaa ei pidä esittää sekä yhteytenä että ominaisuutena

9

Ei esitetä johdettavissa olevaa tietoa

- tulisi selvittää mitkä tiedot ovat perustietoja ja mitkä voidaan päätellä (laskea) muiden tietojen perusteella
- attribuutin arvo tai yhteyden olemassaolo voi olla pääteltävissä

10

Ei esitetä johdettavissa olevaa tietoa

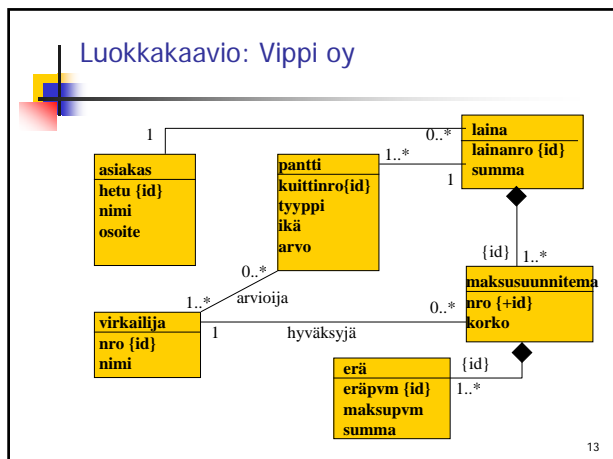
Kurssilla-yhteys on pääteltävissä, jos jokaisen kurssin osanottajan on oltava jossain ryhmässä

11

Muunnoksen perussäännöt:

- Kutakin tietokohdetta (tyyppiä) vastaa samanniminen taulu
- Kullakin kohteen yksiarvoisella attribuutilla on samanniminen vastinsarake kohdetta vastaavassa taulussa
 - Kohteen tunnistavia attribuutteja vastaavat sarakkeet kuuluvat kohdetta vastaavan taulun avaimiin

12



Kokoelma-attribuutit

- Kutakin **kokoelma-attribuuttia (moniarvoista)** vastaa taulu, jonka sarakkeina ovat
 - viiteavain kokoelmatyyppisen attribuutin sisältävää kohdetta vastaavaan tauluun,
 - sarake attribuutin arvoa varten.
- Taulun kaikki sarakkeet kuuluvat avaimen

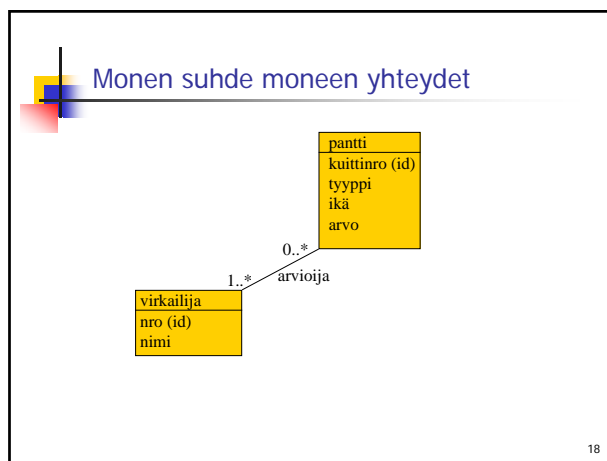
Kokoelma-attribuutit

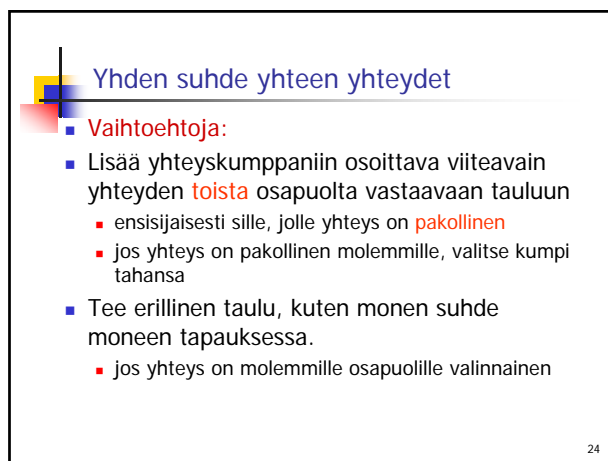
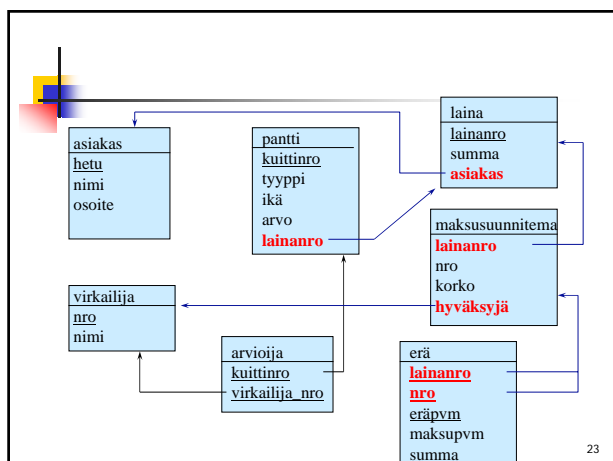
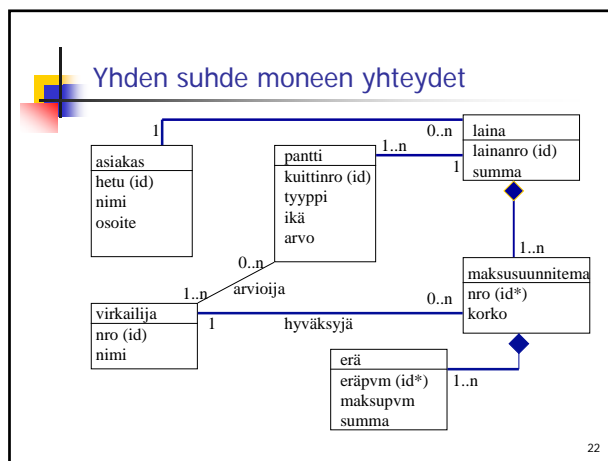
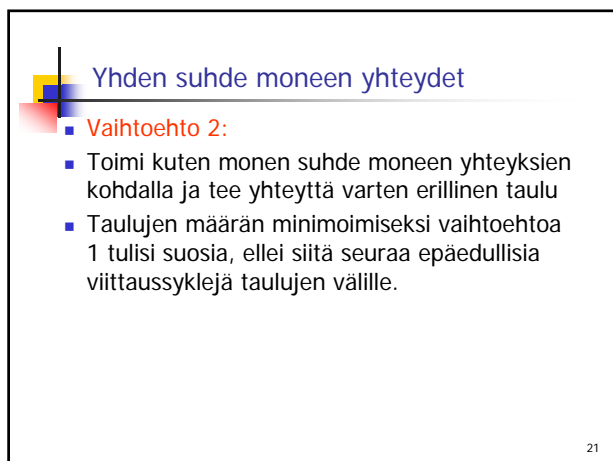
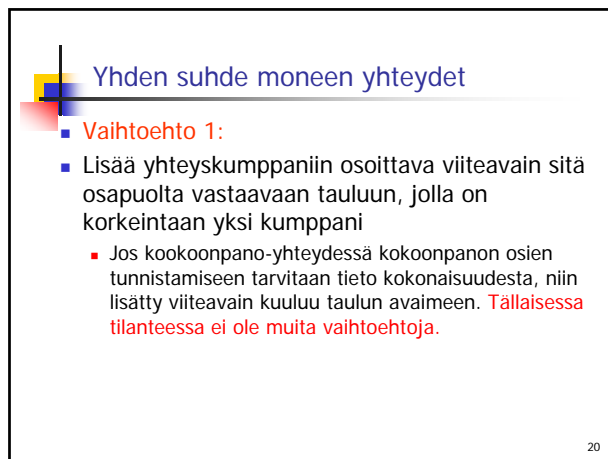
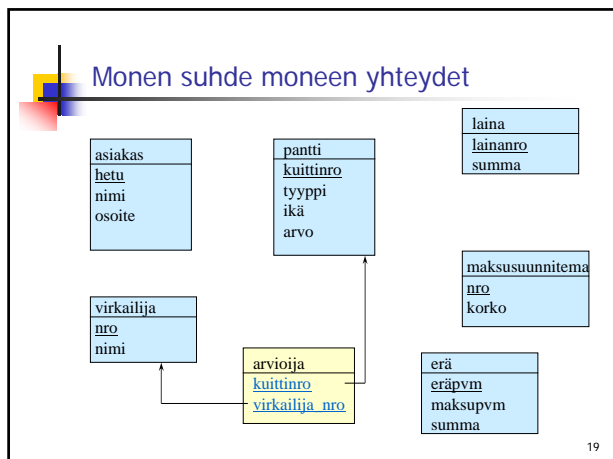
- Vippi oy:n esimerkissä ei ole kokoelma-attribuutteja

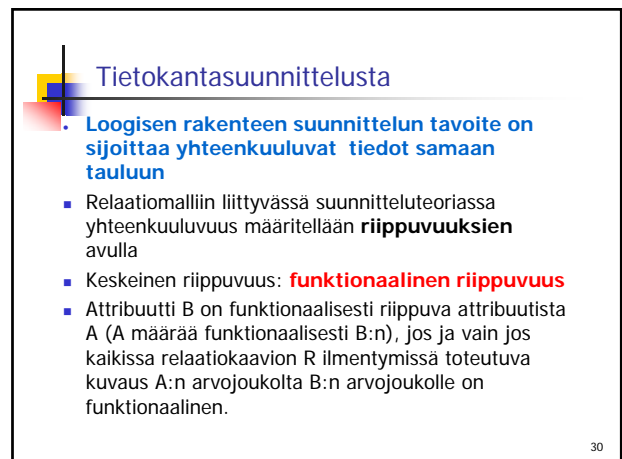
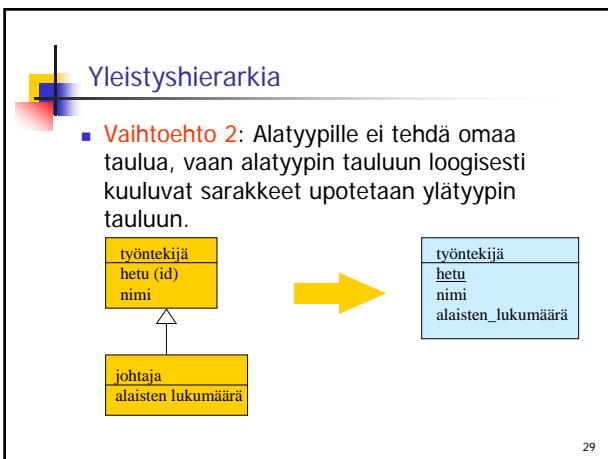
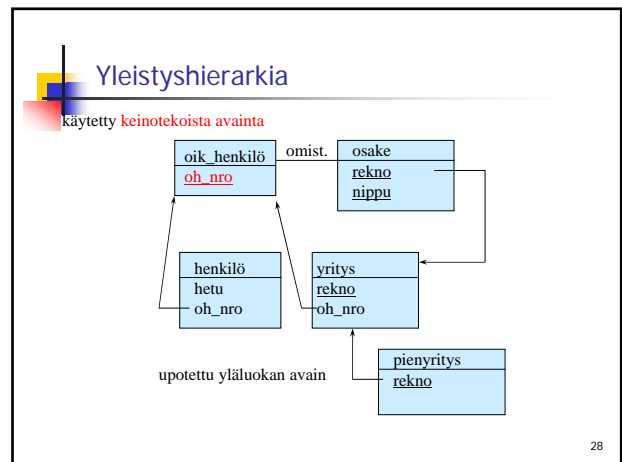
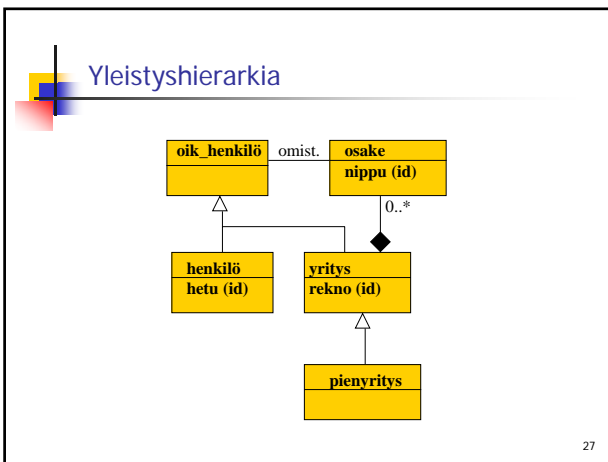
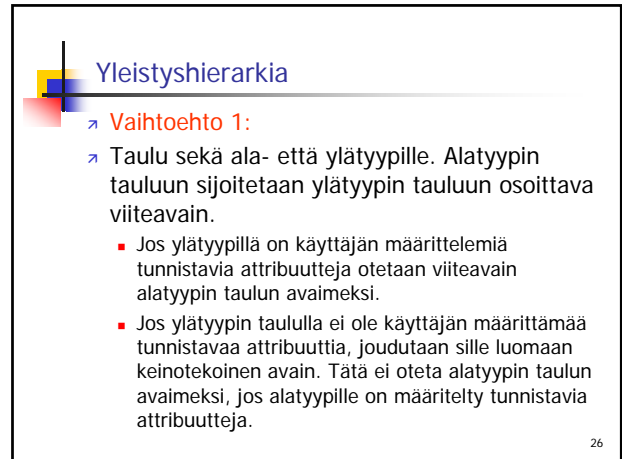
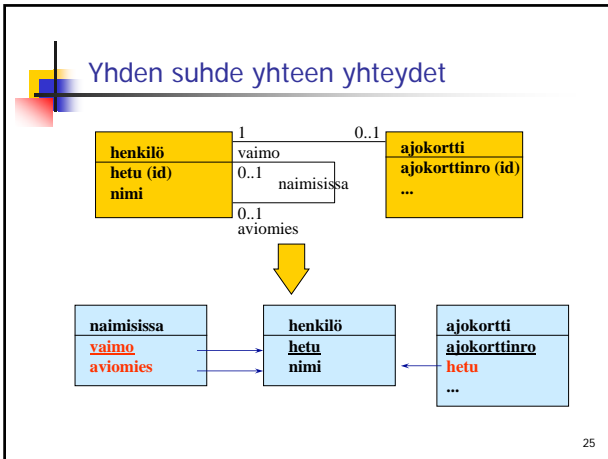
esim:

Monen suhde moneen yhteydet

- Kutakin **monen suhde moneen** yhteyttä vastaa taulu
 - Taulun nimi = yhteyden nimi
 - Taulun sarakkeina ovat yhteyden osapuoliin osoittavat viiteavaimet.
 - Taulun kaikki sarakkeet kuuluvat taulun pääavaimen
 - monen suhde moneen yhteys on yhteys, jossa kummankin osapuolen maksimiosallistumisrajoite on suurempi kuin 1.**







Tietokantasuunnittelusta

- Kuvaus $f: v(A) \rightarrow v(B)$ on funktionaalinen, jos kaikissa relaatiokaavion R ilmentymissä jokainen A:n arvo kuvautuu yhdelle B:n arvolle eli,
 - jos riveillä r ja s attribuutilla A on sama arvo ($r.A = s.A$), niin näillä riveillä täytyy myös B-attribuuteilla olla keskenään sama arvo ($r.B = s.B$).
 - Funktionaalinen riippuvuus tarkoittaa sitä, että attribuutin B arvo on yksikäsitteisesti selvitettävissä, kun tiedetään attribuutin A arvo. Selvittäminen voisi tapahtua kyselyllä
 - `select distinct B from R where A=a;` (tuloksena olisi enintään yksi rivi)

31

Tietokantasuunnittelusta

- Funktionaalista riippuvuutta, jossa A määrää B:n merkitään $A \rightarrow B$. Attribuuttia A kutsutaan määrääjäksi.
- Yksittäisen attribuutin A tilalla voi olla myös attribuuttiyhdistelmä. Tavoitteena on kuitenkin löytää yhdistelmät, joissa on minimaalinen määrä attribuutteja.
- Jos $A \rightarrow B$, voidaan määrääjään lisätä mikä tahansa attribuutti x ja pätee $Ax \rightarrow B$.
- Avain määrää funktionaalisesti kaikki relaatiokaavan attribuutit (= kun tietoa haetaan avaimen arviolla saadaan vastaukseksi enintään yksi rivi)
- Riippuvuudet ilmaisevat jonkin kuvattavaan todellisuuteen liittyvän säännön

32

Tietokantasuunnittelusta

- Tarkastellaan relaatiokaaviota
- Kurssilainen(
 - Kurssikoodi,
 - Hetu,
 - OpiskelijaNimi,
 - KurssiNimi,
 - TehtavaLkm).
- Oletetaan, että relaatio sisältää tietoja useista opiskelijoista ja useista kursseista ja kuvaa opiskelijoiden osallistumista kursseille.

33

Tietokantasuunnittelusta

- **Hetu \rightarrow OpiskelijaNimi**
Yhtä henkilötunnusta kohden on vain yksi Opiskelijanimi
Henkilötunnuksen perusteella on selvitettävissä opiskelijan nimi
- **Kurssikoodi \rightarrow KurssiNimi**
Yhtä kurssikoodia kohti on vain yksi kurssinimi
- **Hetu, Kurssikoodi \rightarrow TehtäväLkm**
 - Yhtä henkilötunnus & kurssikoodi yhdistelmää kohti on vain yksi tehtävälukumäärä = Henkilön tekemien tehtävien lukumäärä on kurssikohtainen
 - Yhtä henkilötunnusta kohti voi olla useita tehtävämääriä (opiskelija on monella kursilla)
 - Yhtä kurssikoodia kohti voi olla monta tehtävämäärää (kursilla on monta opiskelijaa)

34

Tietokantasuunnittelusta

- **Mitkä sarakkeet samaan relaatiokaavioon?**
- **Erlaisia kriteerejä**
- Yhteenkuuluvuussääntönä, ns. Boyce-Codd normaalimuodon sääntö:
 - relaatiokaavion R attribuutit kuuluvat yhteen, jos ja vain jos relaatiokaavioon R ei liity yhtään sellaista funktionaalista riippuvuutta, jossa määrääjä ei sisältäisi relaation avainta.
- Kurssilainen relaation avain on pari **Kurssikoodi, Hetu**
- => Kaikki attribuutit eivät BC-säännön mukaan kuulu yhteen koska
 - Hetu \rightarrow OpiskelijaNimi ja
 - Kurssikoodi \rightarrow KurssiNimi rikkovat BC-sääntöä

35

Tietokantasuunnittelusta

- Attribuutit on uudelleenjärjesteltävä relaatiokaavioiksi, joissa yhteenkuuluvuussäännöt ovat voimassa.
- päädytään relaatiokaavioihin
 - Opiskelija(Hetu, OpiskelijaNimi)
 - Kurssi(Kurssikoodi, KurssiNimi)
 - Osallistuminen(Hetu, Kurssikoodi, TehtäväLkm).

36

Jako Boyce-Codd normaalimuotoon

- Miten järjestellään:
 - Karsi riippuvuusjoukko minimaaliseksi – poista johdettavissa olevat riippuvuudet (materiaalissa sääntöjä)
 - Ryhmitä riippuvuudet yhteisen määrääjän (vasen puoli) perusteella.
 - Muodosta jokaista ryhmää kohti oma relaatio
 - Jos alkuperäisen kaavion yksikään avain ei sisälly muodostuneisiin relaatiokaavioihin tee sille oma kaavio
 - Anna relaatiokaavioille nimet

37

Taulurakenteen laadunvalvontaa

- Määrittele taulussa voimassaolevia funktionaalisia riippuvuuksia
- Määrää taulun avain
 - avaimen sisältyvät ainakin kaikki sarakkeet, jotka eivät esiinny missään funktionaalisessa riippuvuudessa vasemmalla puolella (= eivät riipu mistään)
- Jos löytyy yksikin riippuvuus, jossa määrääjä ei sisällä taulun avainta relaatio ei ole BCNF-muodossa ja on mietittävä sen jakamista

38

Tietokantasuunnittelusta

- **Tilauslomaketta analysoitaessa löydettiin seuraavat attribuutit:**
 - lomakenumero,
 - tilaajan tunnus,
 - tilaajan nimi,
 - tilaajan osoite,
 - tilaajan puhelinnumero,
 - toimitusosoite,
 - rivinumero,
 - tavarankoodi,
 - tavaranimi,
 - tilattu määrä, ja
 - tilauspäivä.

39

Tietokantasuunnittelusta

- lomakenumero → tilaajan tunnus (lomakkeella voidaan ilmoittaa vain yksi tilaaja)
- tilaajan tunnus → tilaajan nimi (tunnus identifioi tilaajan, joten sen perusteella saamme selville kaikki tilaajaan liittyvät tiedot)
- tilaajan tunnus → tilaajan osoite,
- tilaajan tunnus → tilaajan puhelinnumero,
- lomakenumero → toimitusosoite (lomakkeella voidaan tilata tavaroita vain yhteen paikkaan)
- lomakenumero → tilauspäivä (lomakkeeseen liittyy yksikäsitteinen tilauspäivä)

40

Tietokantasuunnittelusta

- tavarankoodi → tavaranimi (tavarankoodi on tavarantunniste, jonka kautta päästään kaikkiin tavarantietoihin)
- lomakenumero, rivinumero → tavarankoodi (lomakkeen rivillä voi ilmoittaa yhden tilattavan tavarankoodin)
- lomakenumero, rivinumero → tilattu määrä (lomakkeen rivillä voi ilmoittaa yhden tilattavan tavarankoodin määrän)

41

Tietokantasuunnittelusta

- Jako relaatioihin yhteisen määrääjän perusteella (jos riippuvuuksissa sama määrääjä, niin niissä esiintyvät attribuutit samaan relaatioon)
 - X(tilaajan_tunnus, tilaajan nimi, tilaajan osoite, tilaajan puhelinnumero)
 - Y(tavarankoodi, tavaranimi)
 - Z(lomakenumero, tilauspäivä, tilaajan_tunnus, toimitusosoite)
 - T(lomakenumero, rivinumero, tavarankoodi, tilattu määrä)

42

Tietokantasuunnittelusta

- Jos kaavioille löytyy kuvaava nimi, jako on onnistunut:
 - tilaaja(tilaajan tunnus, tilaajan nimi, tilaajan osoite, tilaajan puhelinnumero)
 - tavara(tavaran koodi, tavaran nimi)
 - tilaus(lomakenumero, tilauspäivä, tilaajan_tunnus, toimitusosoite)
 - tilausrivi(lomakenumero_rivinumero, tavaran koodi, tilattu määrä)

43

