

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

Tietokantasuunnittelusta

Esimerkki taulusta, joka ei käyttydy 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

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

Tietokantasuunnittelusta

- Tietokannan suunnittelun vaiheita ovat
 - tietosisällön kartoitus
 - käsitemallina
 - loogisten rakenteiden suunnittelu ja
 - tekniisten rakenteiden suunnittelu
- Verkkomateriaalissa on käsitelty tietokannan loogisen rakenteen (taulurakenteen) suunnittelua käsitemallin pohjalta. Seuraavassa käsitellään riippuvuuksiin perustuvaa taulurakenteen suunnittelua.

Tietokantasuunnittelusta

- Loogisen rakenteen suunnittelun tavoite on sijoittaa yhteenkuuluvat tiedot samaan tauluun
- Relaatiomalliin liittyvässä suunnitteluteoriassa yhteenkuuluvuus määritellään riippuvuuksien avulla
- Keskeinen riippuvuus: funktionaalinen riippuvuus
- Attribuutti B on funktionaalisesti riippuva attribuutista A (A määrää funktionaalisesti B:n), jos ja vain jos kaikissa relaatiokaavion R ilmentymissä toteutuva kuvaus A:n arvojoukolta B:n arvojoukolle on funktionaalinen.

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)

Tietokantojen perusteet, s 2005
Riippuvuus pohjainen suunnittelu

Tietokantasuunnittelusta

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

7

Tietokantasuunnittelusta

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

8

Tietokantasuunnittelusta

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

9

Tietokantasuunnittelusta

- n Mitkä sarakkeet samaan relaatiokaavioon?
- n Erilaisia kriteerejä
- n 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.
- n Kurssilainen relaation avain on pari Kurssikoodi, Hetu
- n => Kaikki attribuutit eivät BC-säännön mukaan kuulu yhteen koska
 - n Hetu \twoheadrightarrow OpiskelijaNimi ja
 - n Kurssikoodi \twoheadrightarrow KurssiNimi rikkovat BC-sääntöä

10

Tietokantasuunnittelusta

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

11

Jako Boyce-Codd normaalimuotoon

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

12

Tietokantasuunnittelusta

- n Tilauslomaketta analysoidessa löydettiin seuraavat attribuutit:
- n lomakenumero,
- n tilaajan tunnus,
- n tilaajan nimi,
- n tilaajan osoite,
- n tilaajan puhelinnumero,
- n toimitusosoite,
- n rivinumero,
- n tavaran koodi,
- n tavaran nimi,
- n tilattu määrä, ja
- n tilauspäivä.

13

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)

14

Tietokantasuunnittelusta

- tavaran koodi → tavaran nimi (tavaran koodi on tavaran tunniste, jonka kautta päästään kaikkiin tavaran tietoihin)
- lomakenumero, rivinumero → tavaran koodi (lomakkeen rivillä voi ilmoittaa yhden tilattavan tavaran koodin)
- lomakenumero, rivinumero → tilattu määrä (lomakkeen rivillä voi ilmoittaa yhden tilattavan tavaran määrän)

15

Tietokantasuunnittelusta

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

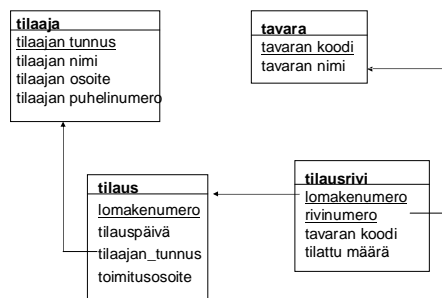
16

Tietokantasuunnittelusta

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

17

Tietokantasuunnittelusta



18

Tietokantasuunnittelusta

- n Riippuvuusanalyysillä voi tarkistaa tuottiko muunnos käsitteellistä relaatiomalliin kelvollisen tuloksen.
- n Erityisesti kannattaa tarkistaa taulut, joihin muunnoksen tuloksena on tullut useita viiteavaimia.