

Datu-base banatuak diseinatzeko betebeharren aurreproiektua eta hainbat irizpide

Ambrosio Goikoetxea

Software Engineering Center (SWEC), MITRE Corporation
11493 Sunset Hills Road, Mail Stop W624
Reston, Virginia 20190, (703) 883-6272, agoico@mitre.org

Laburpena: Lan honen helburua bikoitza da: (1) ingurune banatuko datu-baseen arkitectura diseinatzeko esparru metodologiko bat aurkeztea, eta (2) Defense Information Systems Agency (DISA) erakundeari datu-baseen segmentazioan laguntzea eta jarraibideak ematea Defentsa Departamentuaren sistema eta programetan erabili ahal izateko. Lehenik, datu-base handi eta bakun bat segmentu anizkun txikiagoetan zatitzeko diseinu-aukerak aztertu dira. Bigarrenik, datu-iturri anizkunak aztertu eta haietatik hautatutako azpimultzo bat zehaztu da, betebehar eta irizpide asko betetzen dituen datu-base konposatu bakar bat eratzeko. Hemen erabilitako egiturazko hurbilpena zero-bat programazio matematiko (PM) ez-lineal batena da, eta sistemaren betebehar eta murriztapenen multzo bat ematen du posible diren diseinuen aukerak aztertzeke.

Gako-hitzak: Datu-baseen diseinua, datu-base banatuen ingurunea, datu-base birtu-
len diseinua, datu-baseen segmentuen diseinua, DII COE, datu-baseen diseinuaren op-
timizazioa.

SARRERA

Gaur egun, datu-iturri bakar batean (esate baterako, datu-base batean) ez baizik eta Defentsa Departamentuaren barneko ataletako programa eta sistema batzuetan/askotan banaturiko datu-iturri askotako hainbat datu-multzo atzitu behar dituzten informazio-sistemak gero eta gehiago diseinatzen ari dira. Lan honetan ondorengo galderak aztertzen dira:

- Zer aukera ditu datu-baseen diseinatzaileak eskuragarri dauden **datu-iturri anizkoitzak** aztertzerakoan? Datu-iturrian multzo «hobetsi» bat identifikatzea komeni ote da, eta zein da hori egiteko erarik egokiena?
- Datu-baseen segmentuak kontratistek diseinatzen badituzte eta DISAren DII COE programara bidaltzen badira, zein **jarraibide eta**

irizpide erabili beharko lirateke datu-basea diseinatzean eta «segmentatzean»?

- **Segmentuak nola integratu** behar dira sistema bat osatzeko? Segmentazio-prozesuak segmentu txiki asko sortzea edo segmentu handi gutxi batzuk sortzea sustatu beharko luke, ondoren sistema bat eratzeko integratuak izan daitezkeen?

Galdera horien jatorria DISAren Defense Information Infrastructure (DII) Common Operating Environment, Integration and Runtime Specification (I&RTS) dokumentuan (4.0 bertsioa, 1999ko urrikoa) dago. Testu horren zati batzuk aztertuko ditugu, bi arazo espezifiko azalduko ditugu eta bi arazo horietarako irtenbideak bilatuko ditugu.

Datu-baseen segmentuen garapena. Datu-baseen segmentazioaren arduradunek datuen betebeharrak betetzeko erabil ditzaketen jarraibide orokorrak biltzen ditu DII COE dokumentuak:

- Dauden datu-biltegiak runtime eran erabili (horretarako sarbide-eskubideak lortu behar dira). Datu-biltegiak dauden bezala erabil daitezke, edota datu-biltegiaren jabearekin aldaketak negoziatu daitezke.
- COE Data Emporium-eko datu-baseen segmentuak berrikusi, berriro erabili ahal izateko. Adibidez, Emporium-ak Erakundearen ordezkari-tza arrunta duen datu-baseen segmentu bat baldin bada, segmentu hori erabili beharko litzateke.
- Betebehar berriak asetzeko segmentuak eguneratuta edukitzen saiatu. Eskuragarri baldin badaude, antzeko enpresa-segmentuak (hobetsiak) edo interes komunekoa segmentuak erabili.
- Dauden segmentuak luzatzeko, segmentu gehiago sortu, betebeharrak asetu ahal izateko.
- Dagoen eskema ahalik eta gehien berrerabiliz eta handituz, datu-segmentu berri bat sortu, betebeharrak asetu ahal izateko.

Datu-base baten egiturazko edukiak zehazteko, beste faktore batzuk ere gomendatzen dira:

- Unitate gisa zein taula erabil daitezkeen.
- Esparru funtzional bat sostengatzeko zein taula erabil daitezkeen.
- Datu-iturriak zeintzuk diren, eta
- Datu-baseen objektu-menpekotasunak zeintzuk diren.

«Datu-baseen segmentu anizkoitz konpartituen abantaila ondokoa da: segmentuak bikortsuagoak dira eta, horri esker, datu-zerbitzari konpartitu bat konfiguratu daiteke misio-aplikazioak sostengatzeko, behar ez diren datuak eduki beharrik gabe eta, gainera, konfigurazio-gai berezi gisa erabil

daitezke. Segmentu anizkoitz konpartituek, ordea, desabantaila badute: adibidez, *gako arrotzen murriztapenek* sortutako datu-baseen objektu-menpekotasunen kudeaketa. Segmentu-barruko menpekotasun horiek segmentuen instalazioa eta segmentuen kentzea ere zailtzen dute.» Jarraibide horiek, alabaina, ez diote diseinatzaileari berak hobetsitako datu-azpimultzo bat aukeratzeko edo gako arrotzen menpekotasunak minimizatzen eta datu-base bat segmentuetan zatitzeko modu edo mekanismo espezifikorik eskaintzen.

ARAZOEN AURKEZPENA

Lan honek datu-base banatuen diseinuak dituen bi arazo espezifiko aztertzen ditu:

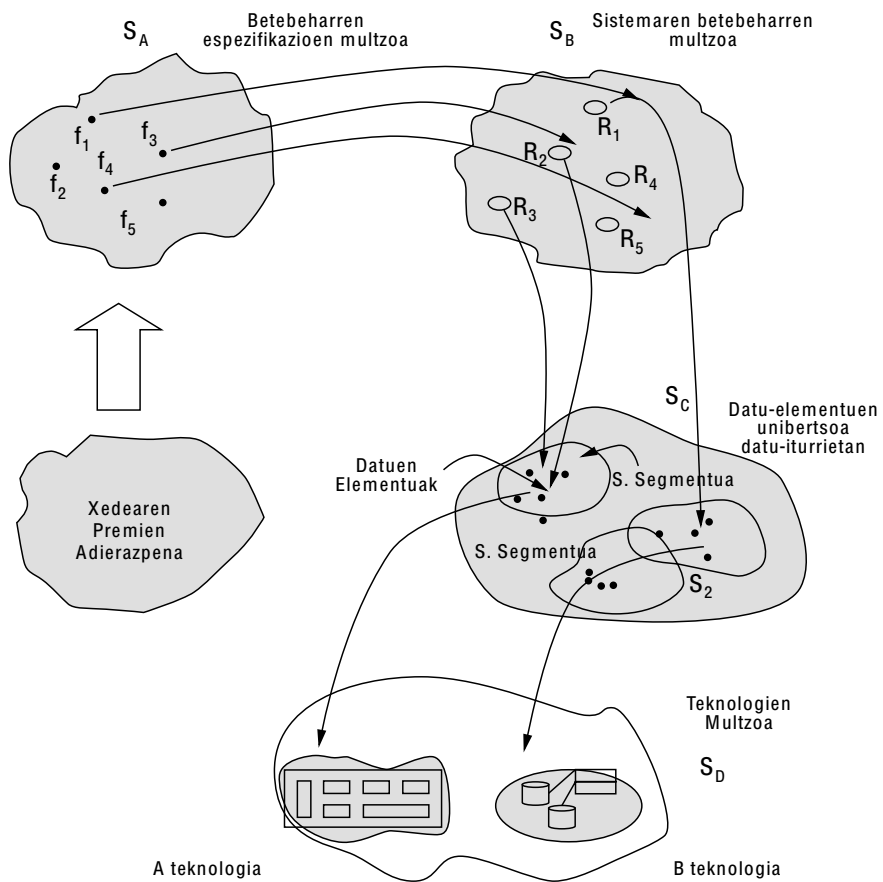
1. arazoa: Datu-base bat datu-base segmentuetan (hau da, taula-azpimultzoetan) zatitzeko erabakia emanik, zein diseinu-aukera ditu segmentazioaren arduradunak? Eta zein da errentagarritasuna kostuen, datuen banaketaren, sistemaren interportabilitatearen, sistemaren eraginkortasunaren eta beste irizpide batzuen aldetik? Diseinatzailea datu-base handi batekin hasten da, eta datu-base edo segmentu txikietan banatzen du; hau da, «batetik askotara» doan datu-baseen segmentazio-arazo bat da.

2. arazoa: hainbat datu-iturri alternatibo emanik, nola aukeratu du datu-basearen diseinatzaileak datu-iturrien azpimultzorik hoberena? «Datu-base birtual eta integratu» baten edo datu-iturri guztietan galderak egiteko aukera ematen duen interfaze-sistema bat eratzea izan daiteke sistemaren diseinuaren parte bat. Baldin galderen eraginkortasuna (betebehar ez funtzionala alegia) garrantzitsua bada, diseinuaren soluzioa datu-iturri multzo jakin bat identifikatzea izan daiteke. Baldin sistemaren garapenaren kostuak txikiak izateak lehentasuna badu, diseinuaren soluzioak datu-iturri multzo ezberdina identifika dezake, eta diseinu horrek ez du halabeharrez galderen eraginkortasun handia ekarriko. Diseinatzailea datu-base multzo handi batetik hasten da, eta datu-base bakoitzetik zenbait zati aukeratuz datu-base bakar eta konposatua osatzen du, hau da, «askotatik batera» doan datu-baseen segmentazio-arazoa da.

HURBILPEN METODOLOGIKOA

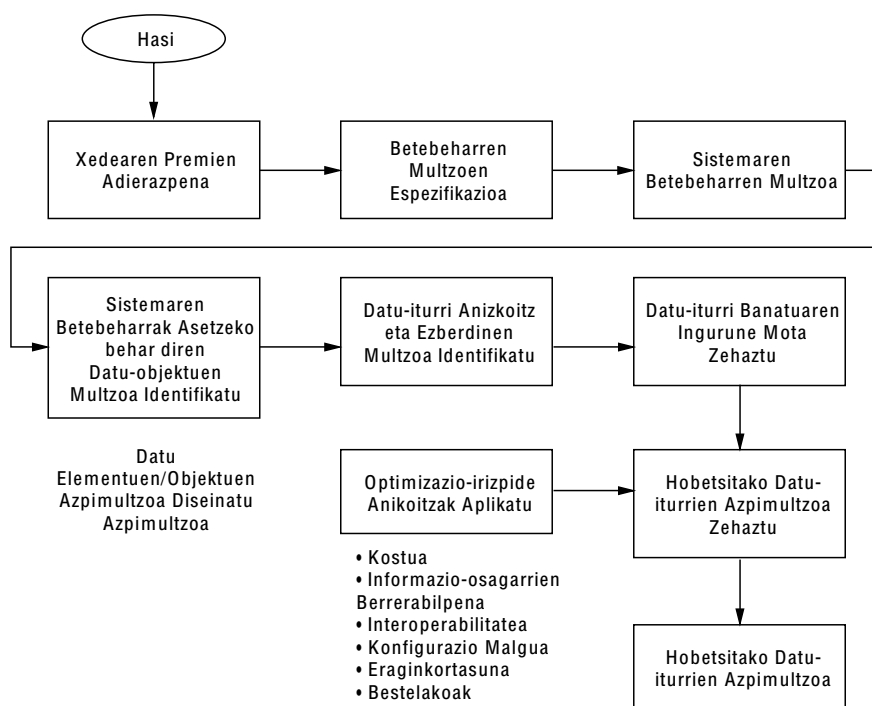
Xedearen betebeharrek datu-iturrien hautaketa gidatzen dute; datu-baseen diseinuan dauden aukerek hainbat soluzio islatzen dituzte «soluzio-espazio» batean (alegia, datu-baseen sistema handi eta bakarra; hain-

bat datu-base txiki; gune bakarra; gune anitz, datu-base banatuen ingurunea, etab.). Bestalde, garapen-kostuek eta betebeharrak ez funtzionalek (alegia, sistemaren eraginkortasuna, segurtasuna, fidagarritasuna, etab.) «sistema hobetsia» identifikatzen laguntzen dute, **1. eta 2. irudietan** adierazten den legez.



1. irudia. Datu-elementuen eta sistemaren betebeharren eskema datu-base bantuen ingurune batean.

Datu-baseen diseinuari buruzko gorabeherak eta datu-base bantuek diseinatzen ari diren teknologien aurkezpenerako, ikus Ozsu eta Valdiriez (1991) eta Simon (1995).



2. irudia. Datu-base banatuen ingurune batean datu-baseak diseinatzeko esparru orokorra.

1. arazoa: Bakar batetik askotarako datu-baseen segmentazioa

Segmentu anizkoitz konpartituek duten desabantaila da, besteak beste, gako arrotzen murriztapenek sortutako datu-baseen objektu-menpekotasunak kudeatzea. Segmentu-barruko menpekotasun horiek segmentuen instalazioa zailtzen dute, eta segmentuen kentzea ere bai. (DII COE).

Hemen proposatzen dugun hurbilpena, datu-basea (alegia, S taula-multzoa) segmentu anizkuntan deskonposatzea da (S_i datu-base segmentua S -ko taulen azpimultzo batez osatuta dago, beraz $S_i \in S$), hortaz taula bat segmentu bakar batean agertzen da, eta segmentu guztien batura S multzoa da, 3. irudian azaltzen den bezala.

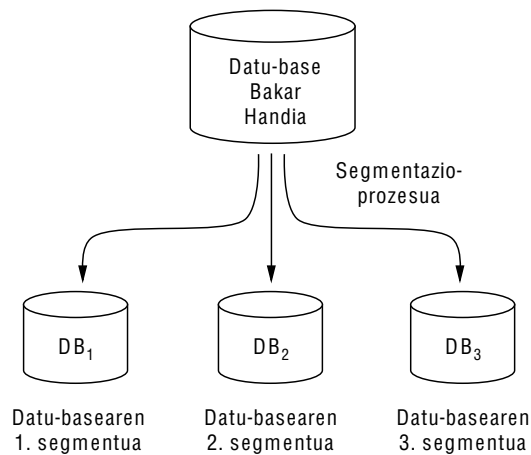
Ondoko oharrak egingo ditugu:

- a) Har dezagun 4. irudiko segmentatu beharreko datu-basea. 4. irudiko 4 taulako segmentu hautagai batek gako arrotzen 6 menpekotasun ditu, hau da, segmentu-mugak 6 aldiz «igarotzea»; gezi batek bi tau-

len arteko erlazioaren bukaerako muturra seinalatzen du; beste 4 taulako segmentuak ere 6 menpekotasun ditu, eta 3 taulako segmentuak aldiz 8; muga-igarotze guztiak batuz guztira gako arrotzen 10 menpekotasun ditugu.

- b) «Segmentazioaren diseinuaren arazoaren» formulazioak, bi-bitako menpekotasun posibleen kopurua (alegia, gutxienez gako arrotz bat dago bi taulen artean) *koefiziente binomialak* emandako konbinaketa-kopurua dela erakusten digu. 4. irudiko datu-basearen gehieneko zenbaki posiblea 55 da, hau da:

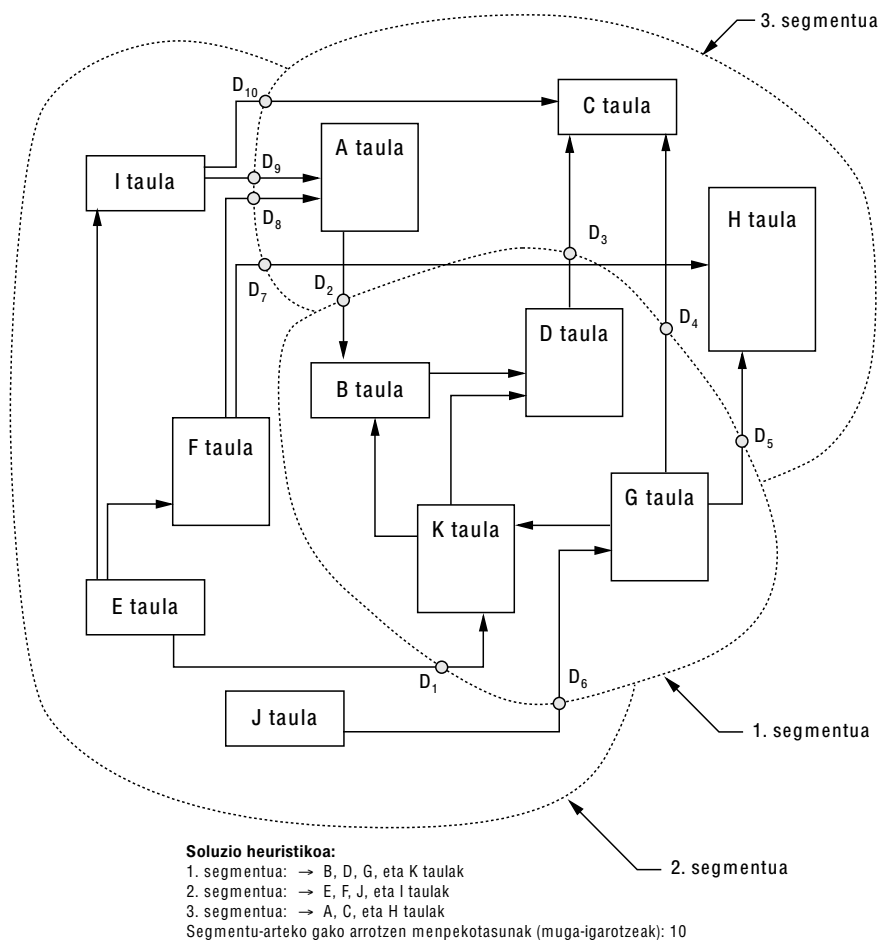
$$\binom{11}{2} = \frac{11!}{9!2!} = \frac{(11)(10)}{2} = 55$$



3. irudia. Datu-base handi baten deskonposizioa datu-baseen segmentu anizkoietan.

Hala ere, menpekotasunak dituzten taula-pareen N_1 kopurua 15 da soilik:

- | | |
|-----|-----|
| A-B | D-K |
| A-I | E-F |
| A-F | E-I |
| B-K | F-H |
| B-D | H-G |
| C-I | I-C |
| C-D | I-E |
| C-G | |



4. irudia. Datu-baseen segmentu hautagaiak gako arrotzen menpekotasunekin.

c) N_2 «segmentu menpekotasunen» kopurua 10 da, «taula-menpekotasunen» kopurua baino txikiagoa, alegia

$$N_2 < N_1$$

d) 2 segmentuk bakoitzeko 4 taula izateko eta segmentu batek 3 taula izateko segmentuen kopuru posiblea:

$$\frac{11!}{[(4! \cdot 4!) \cdot 2!] \cdot 3!} = 5.775 \text{ segmentazio-diseinu posible}$$

Hau da, 11 taula ditugunez, taulok (11!) eratan antola daitezke, baina {4, 4, 3} taulako 3 segmentu izan nahi ditugu hurrenez hurren (alegia, 4 taulako 2 segmentu eta 3 taulako segmentu 1), eta segmentu horien barruko taulen ordenak ez du axola; beraz, (11!) zati (4!4!3!) egiten badugu; ondoren, 4 taulako 2 segmentuen arteko ordenak axola ez duenez, zati (2!) egiten dugu. Horrela, 5.775 segmentazio-diseinu posible sortzen dira; diseinu bakoitzak 2 segmentu eta 4 taula ditu segmentu bakoitzeko, eta 3 taulako segmentu bat. Oso segmentu-diseinu kopuru handia, beraz.

Formulazio matematikoa:

Izan bedi $X_{ji} = 1$, i taula j segmentukoa baldin bada, 0 i taula j segmentukoa ez baldin bada, eta $i = A, B, \dots, K, L$, eta $j = 1, 2$, eta 3 izateko, **3. irudian** azaltzen den legez.

Beraz, optimizazioaren arazoa horrela adieraz daiteke: Minimizatu:

$$\sum_{i,j,k,l} X_{ij} X_{kl} \cdot \delta_{jk}$$

non $\delta_{jk} = 1$, baldin j eta k taulen artean menpekotasunik baldin badago; $\delta_{jk} = 0$, baldin j eta k taulen artean menpekotasunik ez badago; $k, j = A, B, C, \dots, L$, segmentuen izenak, baina $k \neq j$; $i, l = 1, 2$, eta 3 ere bai, segmentuen izenak, baina $i \neq l$, mugaturik:

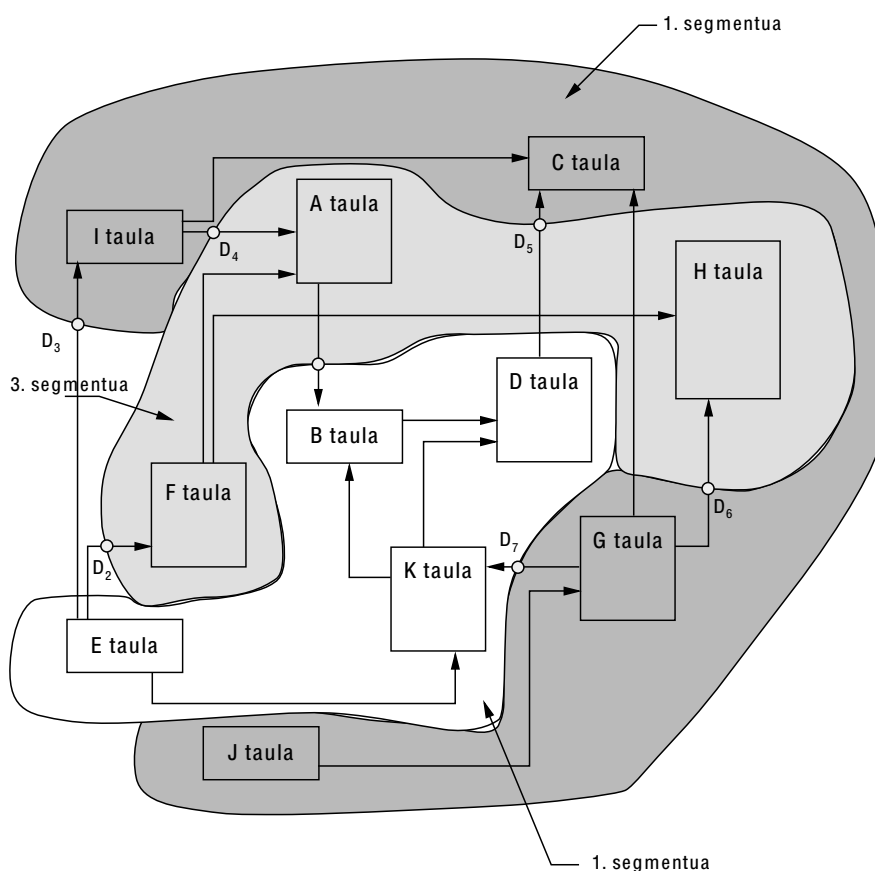
$X_{1A} + X_{1B} + X_{1C} + \dots + X_{1k} = 4$, 1. segmentuan 4 taula soilik behar izateko;
 $X_{2A} + X_{2B} + X_{2C} + \dots + X_{2k} = 4$, 2. segmentuan 4 taula soilik behar izateko;
 $X_{3A} + X_{3B} + X_{3C} + \dots + X_{3k} = 3$, 3. segmentuan 3 taula soilik behar izateko;
 $X_{1A} + X_{2A} + X_{3A} = 1$, A taula segmentu bakar batena izateko (1, 2, edo 3. segmentua);
 $X_{1B} + X_{2B} + X_{3B} = 1$, B taula segmentu bakar batena izateko;
 $X_{1C} + X_{2C} + X_{3C} = 1$, C taula segmentu bakar batena izateko; eta gauza bera beste taula guztietarako; halaber, eta $X_{ij} = 1$ edo 0 i eta j guztietarako.

Soluzioa:

Arazo ez-lineal, bitar horretarako soluziorik hoberena Microsoft Excelen Solver programazio matematikoko ahalmena erabiliz lortu da:

$$\begin{array}{lll} X_{1C} = 1 & X_{1G} = 1 & X_{1I} = 1 \\ X_{1J} = 1 & X_{2B} = 1 & X_{2K} = 1 \\ X_{2D} = 1 & X_{2E} = 1 & X_{1K} = 1 \\ X_{3A} = 1 & X_{3F} = 1 & X_{3H} = 1 \end{array}$$

Eta beste aldagai guztiak zero dira, hau da, ez dute hautatzen. **5. irudian** soluzio horren adierazpen grafikoa agertzen da. Adierazpen matematiko horrek beste muga batzuk ere har ditzake, hala nola entitate ahulak, negozio-arauak edo erabiltzaileak segmentu berean jarri nahi dituen taula-multzoak.



Soluziorik onena:

1. segmentua: → C, G, I, eta J taulak

2. segmentua: → B, D, E, eta K taulak

3. segmentua: → A, F, eta H taulak



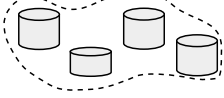
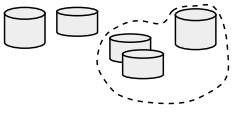
Segmentu-arteko gako arrotzen menpekotasunak (muga-igarotzeak): 7

5. irudia. Gako arrotzen menpekotasun-kopururik txikienak dakartzan datu-baseen segmentaziorik onena.

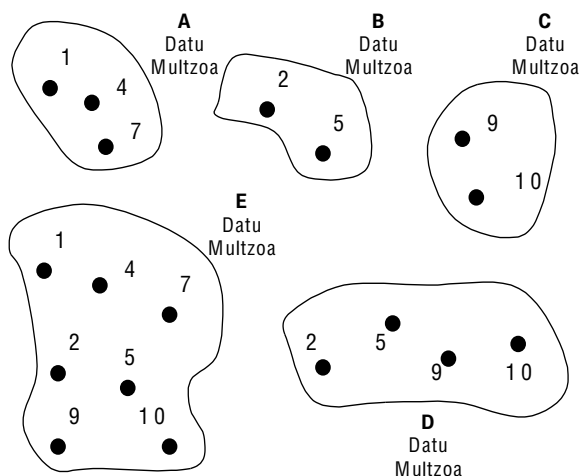
2. arazoa: Askotatik bakar baterako Datu-baseen Segmentazioa

Datu-basearen diseinatzailea berari interesatzen zaizkion taulak eta datu-elementuak gordetzen dituzten datu-base anizkoitzak bilatzen hasten denean, zenbait egoera gerta daitezke, **1. taulan** erakusten den bezala. Kasurik sinpleenean, 1. kasuan, hobetsitako datu-elementu edo objektu guztiak (alegia, datuen elementuak, taulak, gordetako prozedurak, etab.) datu-iturri bakar batean kokatzen dira; horregatik, diseinu-estrategia datu-iturri bakarra erabiltzea da eta, beraz, ez dago erabakitze- aukerarik. 2. kasuan nahi diren objektuak datu-iturri anizkoitz eta ezberdinetan kokaturik daude, eta datu-objektuak ez dira datu-iturrietan banatzen. Azkenik, 4. kasuan hobetsitako objektuak datu-iturri ezberdin eta anizkoitzetan kokaturik daude, baina zenbait datu-objektu datu-iturrietan zehar konpartiturik daude. Ondoren, 4. kasuko erabakitze-arazoa aztertuko dugu.

1. taula. Datu-elementuen banaketaren ereduak datu-iturri anizkoitzetan.

| Kasua | Eskuragarri dauden datu-elementuen banaketa | Arkitektura-estrategia | Optimizazio-irizpidea |
|-------|--|---|---|
| 1 | Nahi diren datu-elementuak/ objektuak datu-iturri bakar batean daude. |  Datu-iturri bakarra hautatu | Bat ere ez. Ez dago erabakitze- aukerarik. |
| 2 | Nahi diren datu-objektuak datu-iturri batean daude, eta kopia zehatzak hainbat gunitan. |  Datu-iturri bakarra hautatu. | Sistema garatzeko kosturik txikienak eskaintzen dituen gunea aukeratu. Beste irizpide batzuk: eraginkortasuna. |
| 3 | Nahi diren datu-objektuak datu-iturri anizkoitz eta ezberdinetan daude; datu-objektuak ez dira datu-iturrietan zehar banatzen. |  | Datu-iturrien multzo osoa aukeratu. Ez dago erabakitze- aukerarik. |
| 4 | Nahi diren datu-objektuak datu-iturri anizkoitz eta ezberdinetan daude; datu-objektuak datu-iturrietan zehar banatzen dira. |  Datu-iturrien azpimulzoa aukeratu. | Datu-iturrien azpimulzoa aukeratu optimizazio-irizpidearen bitartez: — Eraginkortasuna. — Kostua. — Bestelakoak. |

Hobetsitako datu-iturrien azpimulzoaren hautaketa. Ondoren, 4. kasuko adibidea aztertuko dugu; guztira hobetsitako 7 datu-elementu daude, {1,2,4,5,7,9,10}, 5 datu-iturritan banaturik, **6. irudian** azaltzen den legez.



6. irudia. Datu-iturri anizkoitz alternatiboak.

N datu-iturriren «azpimultzo hobetsi» bat hautatzeko aukera sortzen denez, erabakitze-une bat gertatzen da; azpimultzoak ondoko irizpideak betetzen ditu:

1. nahi diren datu-elementu guztiak azpimultzo hobetsi horretan adierazita daude,
2. 2. irudiko «datu-elementuen diseinu-azpimultzoak» datu-elementu guztiak emateko behar diren datu-iturrien kopuru txikiena dauka,
3. irizpide baten optimizazioaren emaitza da (adibidez, diseinuaren kostua minimizatzearena, kontsultaren erantzun agregatua minimizatzearena, etab.), eta
4. Pareto soluzio bat da (hau da, menperatu gabeko soluzio bat Pareto espazio n-dimentsional batean; Goicoechea et al., 1992, 1982).

Arazo horretatik, hiru soluzio posible daudela ikus dezakegu:

1. soluzioa: Multzoak aukeratu: A + B + C
2. soluzioa: Multzoak aukeratu: A + D
3. soluzioa: Multzoa aukeratu: E

Hau da, 1. soluzioak hiru data-iturriz, A, B eta Cz, osaturiko azpimultzo bat dakar; 2. soluzioak A eta D datu-iturriez osaturiko azpimultzoa dakar; eta 3. soluzioak E datu-iturriaz bakarrik osaturiko azpimultzoa dakar. Hiru soluzioetako bakoitzak datu-elementuen diseinu-multzo osoa, {1, 2, 4, 5, 7, 9, 10}, hartzen duela ikus dezakegu.

Mundu errealeko arazo baten kasuan, datu-iturri alternatiboen kopurua oso handia izan daiteke, eta soluzio posibleen zerrendatzea ez da praktikoa.

Programazio matematikoa (PM) izan daiteke diseinu-arazo hori konpontze-ko bidea, ondoren azaltzen dugun bezala.

Irizpide anizkoitza

Datu-base banatuen diseinuan irizpide anizkoitzak ondokoa barne har dezake:

- Kostua.
- Eraginkortasuna (erantzun-denbora, bestelakoak).
- Datu-iturrien berrerabilpena/banaketa.
- Konfigurazioaren malgutasuna.

Goicoechea et al.-ek (1992, 1982) deskribatu dute irizpide anizkoitzen erabilera ingeniariartzako diseinu-arazoetarako eta negozioetako erabakitze-arazoetarako.

Diseinu-aldagaiak: Izan bedi

- X_{ij} = j datu-elementua i datu-iturrian,
- X_{A1} = 1 datu-elementua A datu-iturrian
- X_{A4} = 4 datu-elementua A datu-iturrian
- X_{A7} = 7 datu-elementua A datu-iturrian
- X_{B2} = 2 datu-elementua B datu-iturrian
- X_{B5} = 5 datu-elementua B datu-iturrian
- X_{C9} = 9 datu-elementua C datu-iturrian
- X_{C10} = 10 datu-elementua C datu-iturrian
- X_{D2} = 2 datu-elementua D datu-iturrian
- X_{D5} = 5 datu-elementua D datu-iturrian
- X_{D9} = 9 datu-elementua D datu-iturrian
- X_{D10} = 10 datu-elementua D datu-iturrian
- X_{E1} = 1 datu-elementua E datu-iturrian
- X_{E2} = 2 datu-elementua E datu-iturrian
- X_{E4} = 4 datu-elementua E datu-iturrian
- X_{E5} = 5 datu-elementua E datu-iturrian
- X_{E7} = 7 datu-elementua E datu-iturrian
- X_{E9} = 9 datu-elementua E datu-iturrian
- X_{E10} = 10 datu-elementua E datu-iturrian

Erabakitze-arazoa. Orain erabakitze-arazoa programazio matematikoaren bidez (PM) formulatuko dugu. «Datu-elementuen azpimultzo aukeratu» datu-elementu bakoitzari erabakitze-aldagai bat ematen zaio, 1en (hau da, datu-elementu hori eta haren datu-iturria aukeratu) eta 0ren (hau da, datu-elementu hori eta haren datu-iturria ez aukeratu) balio posibleen multzo batekin. PMaren teknikak kostuak minimizatzen dituen datu-elementu eta datu-iturri baten konbinazioa bilatzen du, *soluziorik onena* alegia.

2. taula. Kostua eta erantzun-denborak.

| Diseinuaren aldagaia | C ₁ koefizientea | |
|-------------------------|-----------------------------|---------------------------------|
| | Kostua (\$, Dolarrak) | Erantzun-denbora (Segundoak) |
| X _{A1} | 2 | 4 |
| X _{A4} | 2 | 4 |
| X _{A7} | 3 | 3 |
| X _{B2} | 10 | 3 |
| X _{B5} | 12 | 3 |
| X _{C9} | 1 | 2 |
| X _{C10} | 1 | 2 |
| X _{D2} | 3 | 3 |
| X _{D5} | 3 | 3 |
| X _{D9} | 3 | 3 |
| X _{D10} | 3 | 3 |
| X _{E1} | 2 | 15 |
| X _{E2} | 2 | 20 |
| X _{E4} | 1 | 15 |
| X _{E5} | 2 | 20 |
| X _{E7} | 2 | 20 |
| X _{E9} | 2 | 15 |
| X _{E10} | 1 | 15 |

Adibide horretarako kostuaren eta erantzun-denboraren parametroak **2. taulan** adierazten dira.

Sistema osoaren kostua minimizatzeko funtzioa:

$$F_{\text{kostua}} = C_1 X_{A1} + C_2 X_{A4} + C_3 X_{A7} + C_4 X_{B2} + \dots + C_{19} X_{E10}$$

Ondoko mugen menpe:

$$X_{A1} + X_{A4} + X_{A7} + X_{B2} + \dots + X_{E10} = 7 \text{ nahi diren datu-elementuen guztizko kopurua 7 da;}$$

$$(X_{A1} + X_{A4} + X_{A7})/3 = 0,1 \text{ muga bitarra, hau da, A datu-iturriko datu-elementu guztiak edo bat ere ez aukeratu;}$$

$$(X_{B2} + X_{B5})/2 = 0,1 \text{ muga bitarra, hau da, B datu-iturriko datu-elementu guztiak edo bat ere ez aukeratu;}$$

$$(X_{C9} + X_{C10})/2 = 0,1 \text{ muga bitarra, hau da, C datu-iturriko datu-elementu guztiak edo bat ere ez aukeratu;}$$

$$X_{A1}, X_{A4}, X_{A7}, \dots, X_{E10} = 0,1 \text{ erabakitze-aldagai guztiak bitarrak dira;}$$

Arazo horren soluzioa Excel-ek duen Solver programazio matematikoa-
ren (PM) bitartez lortu dugu:

$$\begin{array}{ll} X_{A1} = 1 & X_{D2} = 1 \\ X_{A4} = 1 & X_{D5} = 1 \\ X_{A7} = 1 & X_{D9} = 1 \\ & X_{D10} = 1 \end{array}$$

eta beste aldagai guztiei zero esleitzen zaie; horrela A eta D datu-iturriak hautaturiko azpimultzo gisa identifika ditzakegu, eta hori lehenago azaldu-
riko 2. soluzioari dagokio.

Hala eta guztiz ere, diseinu-irizpidea aukeratu behar diren datu-iturri guztietan erantzun-denbora agregatua minimizatzea baldin bada (hau da, eraginkortasuna maximizatzea), soluzioa A + B + C datu-iturriek dakarte; hori 2. taula aztertuz froga daiteke, eta lehenago azalduko 1. soluzioarekin bat egiten du.

AIPAMENAK

Egileak Dr. Robert Daniels eta Francis B. Driscoll Jr. jaunari, MITRE Corporation-i, Software Engineering Center-i eta Department W098-ri eskerrak eman nahi dizkie, baita Euskal Herriko Unibertsitateko EKAIAko zuzentzaile eta editoreei ere, lan hau berraztertzeagatik, zuzenketa teknikoak egiteagatik eta lanaren aurkezpena hobetzeko emandako aholku eskuzabal eta baliagarriengatik.

BIBLIOGRAFIA

- [1] *Defense Information Infrastructure (DII) Common Operating Environment (COE), Integration and Runtime Specification (I&RTS)*, Defense Information Systems Agency (DISA), Report, Revision 4.0, 1999ko urria.
- [2] *Principles of Distributed Database Systems*, by M.T. OZSU and P. VALDIRIEZ, Prentice Hall, 1991.
- [3] *Multiple Criteria Decision Making (MCDM), Applications in Industry, Business, and Government*, A. GOICOECHEA, L. DUCKSTEIN, eta S. ZIONTS (editoreak), Proceedings of the IX-th International Conference on MCDM, Fairfax, Virginia, 1992ko abuztuak 5-8.
- [4] *Multi-objective Decision Analysis with Engineering and Business Applications*, A. GOICOECHEA, D.R. HANSON, eta L. DUCKSTEIN, John Wiley and Sons Publishers, 560 or., New York, 1982.
- [5] *Strategic Database Technology: Management for the Year 2000*, A.R. SIMON, Morgan Kaufmann Publishers, 1995.