Tehokkaan tietokannan suunnittelu CASE-ympäristössä

Marja Kärmenniemi, IBM
marja_karmeniemi @ fi.ibm.com

CASE-välineiden on varmasti jotenkin muutettava tehokkaan tietokantaratkaisun suunnittelua, mutta miten?

Myös STST-kerhossa, jossa kehitetään menetelmiä tietokantojen jatkuvaan suunnitteluun, nousi tämä kysymys ilmaan. Millaista apua CASE-välineet voisivat tarjota vai saattaisivatko ne jopa estää hyvien päätösten tekemisen joidenkin teknisten rajoitusten vuoksi?

Vastauksen löytämiseksi vertasin suunnittelua käyttäen STST-mallia (Sytyke raportti, 1993) ja *Seer*Method* -menetelmää , jota *Seer*HPS* CASE-välineet tukevat. Kohdemallista johdetaan tietokantakaavion 0-versio, mikä CASE-välineellä tietysti syntyy pyydettäessä automaattisesti. Kohteista saadaan tauluja, ominaisuuksista sarakkeita, tunnisteista perusavaimet ja kohteiden välisistä yhteyksistä viiteavaimet sekä avaimia vastaavat hakemistot. Dokumentoitu tietokantakaavio, jossa on lisäksi loogisen ja fyysisen tietomallin välinen yhteys tallessa, on kuvauskannassa odottamassa muokkaamista.

Tavoitteena on tietysti luoda mahdollisimman hyvä looginen tietomalli jo suunnitteluvaiheessa, jotta taulujen muokkaamiseen olisi toteutuksessa mahdollisimman vähän tarvetta. Tietokeskeisessä olioperusteisessa Seer*Method -menetelmässä on pyrkimyksenä iteratiivisen suunnittelun avulla päätyä vankkaan loogiseen tietomalliin jo ennen tietokantakaavion generoimista. Tietoja tarkastellaan normalisoidusta kohdemallista muodostettuina kokonaisuuksina, Business Objecteina, joille tehdään elinkaarianalyysi ja joiden eheydestä huolehtivat toiminnot määritellään.

Liiketoimintatapahtumien suunnittelussa yhdistellään näistä toiminnoista tapahtumia. Yleisille ja/tai raskaille tapahtumille voidaan lisäksi tehdään tapahtumakohtainen kohdemalli, jonka perusteella tarvittaessa viritetään tietokantaratkaisua esim. tauluja denormalisoimalla. Päätökset jäävät edelleen tekijöille.

CASE-välineet tarjoavat päätöksenteon tueksi erilaisia matriiseja, joiden avulla mahdollisia ongelmakohtia voidaan etsiä; mutta eivät paljon muuta. Ne on suunniteltu sovelluskehitykseen, jossa sovelluksen kohdeympäristö voi vaihdella työasemista suurkoneiden tapahtumankäsittelyyn ja tiedonhallintajärjestelmissä on valinnanvaraa. Siten myös generoidut ratkaisut ovat niin tekniikka- kuin sovellusriippumattomia.

Usein myös kehitys- ja kohdeympäristö poikkeavat toisistaan enemmän kuin perinteisessä ohjelmoinnissa. Oman kohdeympäristön ja sovellusalueen tuntevat suunnittelijat tarvitaan tekemään tehokkuuspäätöksiä. CASE-välineiden tarjoamaa tukea ei kuitenkaan kannata vähek-syä, sillä aukoton dokumentointi parantaa päätösten laatua.

Hakemistotarpeet tulevat luultavasti useimmin esille vasta toteutusvaiheessa, kun ollaan kirjoittamassa SQL-kutsuja. Saman CASE-välineen tietokantakaavio on selkeä paikka dokumentoida nämä tarpeet. Kaikki kuvauskannan tiedot ovat välittömästi myös toisten käytössä. Hyväksi havaittu menettely “dokumentoida” hakemistot testikannan hakemistotauluihin luomalla ne, ei aina välttämättä toimi, jos kehitysympäristö ja sovelluksen testausympäristö ovat erilliset. Hakemistot eivät synny itsestään perinteisillä menetelmillä, eikä CASE-väline tuo tähän muutosta. Sama pätee tehokkaaseen käsittelyyn. Tietokantahaut pitää toteuttaa. Seer*HPS-ssä tämä tehdään SQL-kielellä, jonka kirjoittamiseen on tarjolla syntaksista huolehtiva apuri, mutta itse tapahtumat eli käyttäjät määräävät tietotarpeet, jotka on toteutettava tehokkaalla tavalla.

Mikään ei sittenkään muutu, ainakaan paljon.

CASE-välineet lisäävät sovelluskehityksen teknistä laatua, jos suunnittelijat tekevät hyviä päätöksiä. Uuden välineen käyttöönotto voi myös viedä huomion tietokantaratkaisun tehokkuudesta, kun on paljon muutakin ajateltavaa ja 0-versio saadaan automaattisesti. Se on vain versio nolla..

Seer*Method ja Seer*HPS ovat Seer Technologies, Inc. tavaramerkkejä