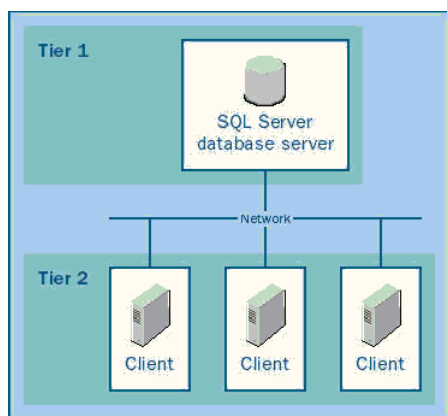


Višeslojna arhitektura

Emir Sadiković

Iako je decembar idealan za trošenje novca, nastavimo priču o uštedama koje možete postići za svoju organizaciju, odnosno, svoga klijenta, upotrebom besplatnih programa. U prošlom broju smo opisali sistem za upravljanje relacionim bazama podataka (RDBMS) Microsoft Desktop Engine (MSDE) i videli da moć koju poseduje SQL Server nije rezervisana samo za one koji su u stanju da izdvoje značajne količine novca.

Kada pominjemo SQL Server i produkciono okruženje, podrazumeva se da govorimo o najmanje dva sloja sistema, odnosno o višeslojnoj arhitekturi. U opštem slučaju, govorimo o klijent/server arhitekturi.



Slika 1

Kao što se vidi na slici 1, klijentski deo sistema je odvojen od same baze podataka. Razdvojenost sistema ne mora biti samo logička, već se slojevi mogu i fizički odvojiti. Drugim rečima, u mrežnom okruženju, na serveru je smeštena sama baza podataka, a na radnim stanicama se nalazi klijentski deo, koji se još naziva aplikacija (Windows ili Web). Podrazumeva se da se aplikacija razvija u nekom od mnogobrojnih raspoloživih alata. To može biti Visual Basic 6.0, VB.NET, C#, C++ ili bilo koji drugi alat koji se može pronaći na tržištu.

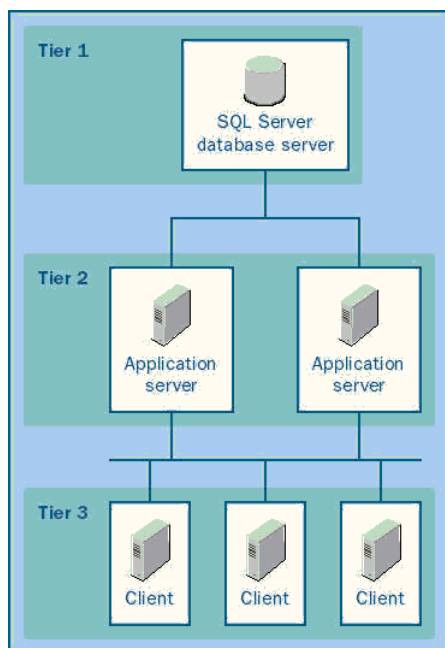
Budimo otvoreni, odluka o tome koji će se alat upotrebiti za izradu aplikacije, obično zavisi od okolnosti u kojima se dati programer obučavao. Drugim rečima, radiće se u onom alatu koji je programeru poznat.

U opštoj podeli, arhitektura sistema se sastoji iz tri sloja. Sloj koji je direktno izložen korisniku se naziva prezentacioni nivo (Presentation layer). On sadrži korisnički interfejs (GUI). Drugi sloj sadrži poslovnu logiku sistema. To znači da su sva pravila koja važe za



OMEGA

neki poslovni sistem inkorporirana u poslovni sloj (Business layer). I najzad, sloj podataka je sama baza podataka (Data layer).



Sika 2. Višeslojna arhitektura

Vodite računa o tome da poslovna logika može, ali ne mora biti izdvojena kao logička ili fizička celina u sistemu. Ona se može uključiti bilo u klijentski deo, bilo u samu bazu podataka. Ukoliko je pridružena aplikaciji, sva poslovna logika će biti na klijentskoj strani. Takav slučaj se naziva “debeli klijent/tanki server” (Fat client/Thin server). Suštinski, to znači da će te poslovnu logiku implementirati u programskom jeziku, odnosno alatu u kome ste razvijali klijentski deo.

Ako poslovnu logiku želite da pridružite samoj bazi podataka, to je naravno moguće samo pomoću programskog jezika Transact SQL (naravno ako govorimo o SQL Serveru ili Microsoft Desktop Engine-u). Način da to učinite je upotreba uskladištenih procedura (stored procedures), funkcija, trigeri (triggers), pravila (rules), podrazumevanih vrednosti (defaults), extended stored procedures i ostalih elemenata jezika koje vam stoje na raspolaganju.

Odluka o tome gde će te smestiti poslovnu logiku zavisi od velikog broja varijabli. Nekada su one veoma značajne, kao recimo usko grlo u mreži, koje vas primorava da smanjite ukupan protok informacija i smestite poslovnu logiku na samom serveru. Nekada su sigurnosni zahtevi ti koji određuju mesto poslovnoj logici sistema.

A opet, nekada su razlozi krajnje banalni, kao što je recimo bolje poznavanje jednog programskog jezika od drugog i slično.

Omega magazin



O M E G A

I u ovom slučaju besplatno shvatite uslovno, jer za dobar razvojni alat mora da se plati, a ako ste već kupili Office paket, onda ga bar iskoristite najviše što možete. U svakom slučaju vežbajte. Ne može da škodi.

Omega magazin

Džordža Vašingtona 40, Beograd, Jugoslavija, tel/faks +381 11 30-30-666,
www.omegamagazin.com