

Adatbázis-kapcsolatok megvalósítása

A .NET Framework az adatbázisok kezelését az ADO.NET-ben definiált osztályokkal támogatja. Az adatbázisok elérésével kapcsolatban két alapvetően különböző technika lehetséges az ADO.NET-ben, melyek között az alapvető különbség az, hogy az alkalmazásunk folyamatos kapcsolatban áll-e az adatbázis-szerverünkkel, vagy sem. Az előbbi esetben kapcsolatorientált, az utóbbiban kapcsolatmentes vagy kapcsolat nélküli adatmodellről beszélünk.

Kapcsolatorientált adatmodell

A hagyományosnak mondható kapcsolatorientált adatmodell esetén az adatbázis-szerver és az alkalmazás között állandó kapcsolat van. Pontosabban fogalmazva: bármilyen adatbázissal kapcsolatos feladat merül is fel az alkalmazás futása közben, az alkalmazás a kérést rögtön továbbítja az adatbázis-szerver felé. Tehát, ha valamilyen adatbázisbeli adatra szükségünk van, akkor nyitnunk kell egy új kapcsolatot az adatbázis felé, el kell küldenünk neki a kérést, és meg kell várnunk a választ, végül pedig be kell zárnunk a kapcsolatot. Ezt a műveletsort kell ismételnünk minden egyes adatbázis-művelet esetén. Ez nem egy túlságosan hatékony megoldás.

Kapcsolatmentes adatmodell

Az előbbivel ellentétben a kapcsolatmentes adatmodell esetén az adatműveletek nagy részét lokálisan hajtja végre a program (egy memóriabeli másolaton), és csak akkor fordul az adatbázishoz, ha arra tényleg szükség van. A kiszolgáló és kliens között nincs folyamatos kapcsolat, a kiszolgálót csak akkor tartjuk fel, ha kérünk is tőle valamit, üresjáratban nem foglaljuk feleslegesen az erőforrásait. Ennek köszönhetően a kiszolgáló sok klienssel tud párhuzamosan foglalkozni.

A megvalósításhoz az ún. DataSet adatforrást használjuk. A DataSet nem más, mint egy memóriabeli másolata az adatbázisnak (pontosabban annak egy része). Ebbe beletartozik az összes tábla, a rajtuk értelmezett kapcsolatok, illetve a bennük tárolt adatok. Így nincs szükség „állandó” kapcsolatra az adatbázissal, hanem bőven elegendő, ha a memóriában szerkesztjük az adatokat, és a változások listáját csak bizonyos idő után küldjük fel kötegelten a szervernek. Így minimalizálni tudjuk a hálózati kommunikációt, illetve csökkenthetjük a szerver terheltségét is. A módszernek persze megvan az a hátránya, hogy nem feltétlenül a legfrissebb adattal dolgozik a kliens, vagyis inkonzisztencia léphet fel. Ez egy könyvtárháznál annyira nem nagy probléma, de egy banki rendszerben vagy a tőzsdén már eléggé nagy károkat okozhat.

Melyiket?

A kétféle modell használata előtt érdemes végiggondolni azt, hogy az adatbázisunk az alábbi két kategória közül melyikbe esik bele:

- ⇒ gyakori módosítások,
ritka lekérdezések;
rövid lefutású tranzakciók;
kevés párhuzamos elérés;
fontos a folyamatos adatfrissítés
⇒ ez esetben a **kapcsolatorientált** modell a megfelelőbb

- ⇒ gyakori lekérdezések,
ritka módosítások;
sok és hosszú ideig tartó párhuzamos adatelérés (pl. webes alkalmazások)
⇒ ez esetben a **kapcsolatmentes** modell a megfelelőbb