

2. Ismertesse az adatbázisok szerver oldali programozásának koncepcióját, előnyeit, hátrányait! Milyen programozási lehetőségei, sajátosságai vannak a Microsoft SQL Servernek?

DIA 3 4oldal

A feladatokat, illetve műveleteket, amiket az adatrétegben hoznánk létre normálist esteben azokat az adatbázis szerver oldalán hozzuk létre. Lényege, hogy eljárásokat tárolunk az adatbázisban, amiket meghívva valamilyen műveleti eredményt kaphatunk. Ezek az eljárások nagyon hasonlítanak egy normál metódusra, itt is megtalálhatóak az eljárás paraméterei, változók, elágazások és hiba jelzések.

DIA 3 8-tol 9-ig

Ezeknek az előnye abban rejlik, hogy a végén az adatbázis felel a konzisztenciáért, ami véget megváltozik az adatbázis szerepe végül, hiszen innentől adatforrást és szolgáltatást is nyújt a rendszer felé. Ezen kívül ez az eljárás hozzáad a rendszerhez egy biztonsági faktort, mert innentől ezek a műveletek egy zárt futtató környezetben lesznek és felesleges vagy káros adat nem kerülhet ki az adatbázisból. Persze ha eljárás az adatbázisban lesz meghívva akkor a teljesítmény is növelve lesz hiszen sokkal kevesebb lesz a hálózati forgalom. A programozás is sokkal könnyeb lesz egy ilyen megoldásnak a beiktatásával, hiszen egy ilyen eljárás az adatbázisban van eltárolva így akár több komponens is képes meghívni ezt az eljárást, ez segíthet mind programozás során, mint karbantartás közben hiszen ez a kód rész csak az adatbázisban létezik, így javítás esetén elég egy konkrét helyen javítani.

DIA 3 10oldal

Persze ezeknek meg van a maga árny oldalai is. Ezek a megoldások nem szabványosok, mivel platformfüggő nyelvi elemek és megoldások lesznek benne, hiszen adatbázisból sem csak egy fajta létezik. Ezen kívül interpretált és növeli a szerver terhelését, amik nehezen vagy egyáltalán nem kezelhető problémákká fejlődhetnek. Illetve ez a megoldás nehezen vagy sehogy sem skálázható, amit későbbi tervezési megoldásoknál alaposan figyelembe kell vennie az embernek.

DIA 3 11oldal-tol

A Microsoft SQL szervernek a specifikus nyelvét erre Transact-SQL nyelv-nek nevezzük vagy T-SQL-nek. Itt is megtalálhatóak a változók, utasítási blokkok, ciklusok, strukturált hibakezelés, illetve új konstrukciók.

Itt általában egy tárolt eljárásnak nincs visszatérési értéke viszont nem kizárt így képes akár az ember visszatérési értéket is megadni.

A függvényeknek ezzel ellentétben van visszatérési értéke, de ezt csak olvashatja az adatbázis és nem írhatja őket.

Illetve megtalálhatóak még a trigger-ek, ezekből kétfajtát különböztetünk meg van a DML esemény trigger-ek ahol rendszer szintű események által/ért hozzuk létre őket. Megtalálhatóak még a Instead of trigger-ek, amik egy speciális DML trigger-ek, ezeket azért hozzuk létre, hogy nézetek adatmódosításával foglalkozzon és sémához kötődik. Ezeket DDL trigger-nek nevezzük.

Trigger: Valami esemény történik az adatbázisban, ami által meghívódik az eltárolt speciális eljárás, ami ehhez képest frissíti az adatokat és nézeteket. DML esemény trigger-ek rendszeresemény által általában naplóznak, míg a DDL trigger-ek amik nézetek adatmódosításával foglalkoznak.

Eljárás: Egy kód rész, ami csinál valamit és lehet akár visszatérési értéke

Függvény: Egy kód rész, ami .NET Assembly-ben írható és van visszatérési értéke, viszont az adatbázist csak olvashatja és nem szerkesztheti. Így tökéletes lekérdezések megfogalmazására véleményem szerint.