

Online Könyvesbolt adatbázis rendszer kialakítása

Készítette: Fehér Krisztofer László

Neptun kód: HO3VMK

Adatbázis működése és tervezés:

Az adatbázis célja egy online könyvesbolt működésének támogatása, azáltal, hogy a vásárlási folyamatok során keletkező adatokat strukturáltan tárolja és kezeli. Az adatbázis alkalmas az egyes vásárlók és könyvek adatainak tárolására, valamint a vásárlók rendeléseinek nyilvántartására. A rendelések tételeit egy külön táblában helyeztem el, így egy rendeléshez több könyv is kapcsolható. Amikor egy rendelés státusza „Teljesített” lesz, az adatbázis automatikusan egy számlát generál az adott rendeléshez, a vásárló által vásárolt könyvek alapján.

Normálformák és táblák közötti kapcsolatok:

A táblák megfelelnek mind a 3 normálformának. Redundanciamentesen tárolnak adatokat, így az adatokhoz könnyen és rugalmasan is hozzá lehet férni. A táblák minden cellája atomi, hisz nem felosztható további részekre. A táblák minden egyes nem-kulcs attribútuma függ az adott tábla kulcs / összetett kulcs attribútumától, emellett tranzitív függőség sem áll fenn. A táblák közötti kapcsolatokat külső kulcsok biztosítják.

Táblák tartalma és a „Vasarlo” tábla particionálása:

„Konyvek” tábla:

Minden könyv egyedi azonosítóval rendelkezik, így könnyen beazonosítható. Rögzítettem az egyes könyvek címét, szerzőjét, árát és kiadásának dátumát is. A raktárkészletüket is nyilvántartottam, valamint az elérhetőségüket is. Ha egy könyv elfogyott és már nincs raktáron, az elérhetőségét „hamis” értékre állítottam, ezáltal már nem lesz megrendelhető az adott könyv. A tábla nem tartalmaz külső kulcsokat.

„Vasarlok” tábla:

Minden vásárló rendelkezik egy egyedi azonosítóval (ID) és névvel. A vásárlók e-mail címét is eltároltam, hogy könnyen kapcsolatba lehessen lépni velük. Rögzítettem továbbá a vásárlók címadatait – város, házszám, utca és irányítószám, hogy pontosan meg lehessen határozni, hová kell kiszállítani az egyes rendeléseket. A táblát a „varos” oszlop alapján particionáltam, így egyszerűbbé válik az egyes vásárlók szállítási adatainak lekérdezése. Az online könyvesbolt kizárólag belföldi szállítást támogat. A táblához nem kapcsolódik külső kulcs, viszont egy összetett kulcsa van a „vasarlo_id” és „varos” oszlopokra, hogy a vásárlók és rendeléseik könnyen nyomon követhetők legyenek.

„Rendelesek” tábla:

A rendeléseket a „rendeles_id” segítségével lehet nyomon követni. A tábla rendelkezik egy összetett külső kulccsal a „Vasarlok” táblából, vasarlo_id-varos. Így az egyes rendelések biztosítják, hogy az adott rendelés a megfelelő vásárlóhoz lesz kiszállítva és a megfelelő helyszínre. A rendelés állapotát is nyomon lehet követni, melynek egy „Enum” típust adtam (állapotok: 'Folyamatban', 'Teljesített', 'Visszavont'). A rendelés létrejöttének a dátuma is eltárolásra került. Az egyes rendelések tételeit, a „Tetelek” táblában tároltam el, hogy egy rendeléshez ne csak 1 könyv tartozzon.

„Tetelek” tábla:

Ebben a táblában az egyes rendelésekhez tartozó könyveket rögzítettem. A táblának a „rendeles_id” és a „konyv_id” az összetett kulcsa, amelyek külső kulcsként hivatkoznak a „Rendelesek” és a „Könyvek” táblákra. Ez lehetővé teszi, hogy egy rendeléshez több könyv is kapcsolódhasson, és azt is, hogy egy könyvből több példányt is lehessen vásárolni. Az egyes tételekhez tartozó mennyiségeket a „darabszam” oszlopban tároltam el.

„Szamlak” tábla:

Amikor egy rendelés teljesített státuszba kerül, automatikusan generálódik róla egy számla, amely a „Szamlak” táblában kerül rögzítésre. Ebben a táblában a „rendeles_id” külső kulcsként hivatkozik a „Rendelesek” táblára, és minden rendeléshez egyedi „szamla_id” generálódik. Továbbá, a táblában eltároltam a számla létrejöttének dátumát „szamla_teljesites_datuma” és az adott rendelés teljes összegét is.

Trigger a könyvek készletének és elérhetőségének az automatikus frissítésére:

Ez a trigger akkor hívódik meg, amikor a „Tetelek” táblába új tételt szűrnak be (INSERT utasítás). A trigger egy függvényt hív meg, amely több ellenőrzést és műveletet végez el. Először a függvény ellenőrzi, hogy a rendelt darabszám meghaladja-e a raktáron lévő mennyiséget. Ha igen, azaz több példány kerülne rendelésre, mint amennyi készleten van, akkor a művelet nem hajtódik végre, és a rendszer hibaüzenetet küld a felhasználónak. Ha elegendő készlet áll rendelkezésre, a függvény megvizsgálja, hogy az adott tételhez kapcsolódó rendelés státusza „Teljesített”-e. Amennyiben a rendelés nem „Teljesített”, a tétel hozzáadódik a „Tetelek” táblához anélkül, hogy a könyv készletét módosítaná. Ha viszont a rendelés „Teljesített” státuszú, akkor a függvény hozzáadja a rekordot a „Tetelek” táblához. Frissíti a „Könyvek” táblában az adott könyv készletét, csökkentve a raktáron lévő példányok számát a rendelt darabszámmal. Ellenőrzi a könyv elérhetőségét: ha a készlet darabszáma 0-ra csökken, az elérhetőséget logikai „0”-ra (nem elérhető) állítja, egyébként logikai „1”-en (elérhető) hagyja. Ez a megoldás azért fontos, mert ha egy vásárló úgy dönt, hogy visszavonja rendelését, a készletből ne legyen levonva az adott mennyiség, ezzel biztosítva az adatok helyességét és a raktárkészlet pontos nyilvántartását.

Tárolt eljárások:

Legjobban fogyó könyvek statisztikája:

A tárolt eljárás egy új táblát hoz létre, amelyben a „Tetelek” és „Könyvek” táblák alapján a legjobban eladott könyvek ranglistája kerül rögzítésre. Az eladásokat az egyes könyvekhez tartozó összesített tétel darabszám alapján kerül kiszámításra. Ez az eljárás lehetőséget nyújt arra, hogy gyorsan és egyszerűen megtekinthessük, mely könyvek a legnépszerűbbek a vásárlók körében.

Legaktívabb vásárlók statisztikája:

Ez a tárolt eljárás ugyanúgy működik mint ahogy a könyves, csak itt nem a könyvek tételei alapján kerül létrehozásra a statisztika, hanem a vásárlók rendelés mennyisége alapján. (Egy vásárló-ID hoz hány darab rendelés-ID tartozik.).

Számlák generálása:

A tárolt eljárás először ellenőrzi, hogy mely tételek vannak teljesítve a rendelés táblában. Amely rendelés teljesítve lett arra létrehoz egy számlát amelyet be is illeszt a „Szamlak” táblába. Csak annyi számla jön létre, amennyi rendelés teljesítve van a „Rendelesek” táblában.

Táblák feltöltése 10000 rekorddal:

A táblákat az „INSERT” utasítások után a „FROM generate_series” segítségével töltöttem fel 10000 rekorddal. Minden táblához más-más logikát alkalmaztam, és a feltöltött adatok típusa megegyezik az adott oszlopok típusával. Az egyetlen kivétel a „Szamlak” tábla, amelyet csak annyi rekordokkal töltöttem fel, amennyi rendelés már teljesült a „Rendelesek” táblában. Ezt a fentebb megemlített „Számlák generálása” tárolt eljárással valósítottam meg.

Lekérdezések Teljesítmény analízishez:

A teljesítmény analízishez egy lekérdezést írtam amely a JOIN utasítással másik 3 db. táblát is összekapcsol. Már a 10000 rekordos feltöltés előtt elkészítettem 1-2 lekérdezést, amelyekkel ellenőriztem az adatbázis működését és helyességét.