

Adatbázis 2 pót ZH

A feladatsort figyelmesen olvasd el!

A pót ZH feladat leírása mellett található egy **auditalo.exe** állomány, amit első lépésként tölts le.

Hozz létre egy mappát az alábbi néven: **név_neptun_kód_adatb2_pot_zh**

Ebbe a létrehozott mappába helyezd el az exe fájlt, majd futtasd (kattints rá duplán).

Ha a Windows blokkolná a futtatást, akkor kattints a felugró ablakban a **További információ** linkre, majd a **Futtatás mindenképpen** opcióra, ezután megnyílik egy command window, ezt ne zárd be, a teljes ZH alatt futnia kell.

Létre fog jönni a mappában egy .git és egy audit mappa, ezeket ne töröld és ne módosíts bele!!!!

A ZH feladatokat úgy készítsd, hogy létrehozol minden feladatnak egy-egy sql kiterjesztésű fájlt (pl.: 1_feladat.sql, 2_feladat.sql) a korábban létrehozott mappában (amiben már fut az exe) és ezeket az sql fájlokat nyitod meg a developer-ben és ott írod a feladatok megoldását és ments folyamatosan.

Ha elkészültél minden feladattal, akkor a korábban létrehozott mappát csomagold zip formátumra.

A tömörített állománynak tartalmaznia kell a .git könyvtárat, az audit mappát és amit az auditáló létrehoz, illetve a megoldásokat tartalmazó sql fájlokat (az exe is maradjon benne).

Ha nem fut a ZH megkezdésétől végig az auditáló, akkor az 0 pontos ZH-t eredményez.

Ha a beadott tömörített állományban nem lesz ott a .git és audit mappa, akkor a ZH 0 pontos.

Ha a .git vagy az audit mappában bármilyen módosítás történik, akkor a ZH 0 pontos.

A ZH során sem egymást, sem semmilyen AI-t nem lehet használni, ezek ZH közbeni vagy utólagosan történő észrevétele esetén a ZH 0 pontos.

Órai anyag és internet használható a feladatok megoldása során, viszont teljes blokkok másolása esetén az adott feladat 0 pontot ér.

A szintaktikailag hibás és nem futtatható kód esetén a feladat 0 pontot ér.

Az invalid kód esetén a feladat szintén 0 pontot ér.

A ZH mellett található egy insert.sql, a táblák elkészítése után azzal feltölthető adatokkal minden tábla.

1. Hozz létre egy SAJAT_WEBSHOP sémát, a jelszó legyen *webshop_jelszo* és a sémát használd a fejlesztés további részében.

2. Az adatbázisban webshophoz kapcsolódó adatokat tárolunk.

Egy táblában tároljuk a vevők adatait, kötelezően a vevők azonosítóját, nevét, lakcímét, emailcímét és telefonszámát.

Egy táblában tároljuk termékre vonatkozó adatokat, kötelezően tároljuk a termék azonosítóját, megnevezését, árát (az ár 10 karakter hosszú lehet, 2 tizedes pontossággal) a és a raktárkészletet (a raktárkészlet csak szám lehet).

Egy táblában tároljuk rendelésekre vonatkozó információkat kötelezően a rendelések azonosítóját, vevő azonosítóját, a rendelés dátumát és a státuszát.

Egy táblában tároljuk a rendelések-tételek kapcsolatát, itt tároljuk a kapcsolat azonosítóját, a rendelés azonosítót, a termék azonosítót, a mennyiséget és az aktuális árat (az ár 10 karakter hosszú lehet, 2 tizedes pontossággal), (az ár eltérhet a termék árától, mivel lehet akciós vagy éppen kupont váltott be rá).

Egy vevő több rendelést is leadhat és egy rendeléshez több termék is tartozhat.

3. A táblák azonosítója egyértelműen azonosítson minden rekordot.

Ahol az egyik tábla adatait használjuk fel egy másikban, ott össze kell kapcsolni a két táblát.

A rendelés-tétel táblában a rendelés azonosítónak és a termék azonosítónak együtt egyedinek kell lennie.

4. A táblák azonosítója automatikusan kerüljön kitöltésre, ne kell azonosítót beszúrni.

A vevő tábla azonosítója 8000 azonosítóval induljon, a termék tábla azonosítója 12500-tól induljon, a rendelés tábla azonosítója 35000 azonosítóval induljon, a rendelés-tétel tábla azonosítója 10 azonosítóval induljon.

5. Készíts egy nézetet amely összefoglalja a vásárlók költési szokásait.

A nézet tartalmazza a vásárló nevét, rendelés dátumát, a termékek megnevezéseit vesszővel elválasztva (ABC sorrendben), az adott rendelés végösszegét.

6. Futtasd be a `seged.sql`-t.

A kapott táblát és a benne található adatokat használva készítsen egy eljárást, ami a termékeket tartja karban.

Ha már létezik a termék, akkor frissíteni kell a termék adatait, ha még nem létezik, akkor be kell szúrni.

Ha létezik, akkor a nevet, kategóriát és árat frissítse, a raktárkészletet pedig növelje meg a `beszallito_frissites` táblában található mennyiséggel.

Ha bármilyen probléma történik a folyamat során, akkor dobjon egy `migration_exc` exceptiont.

7. Készíts egy eljárást, amivel új terméket lehet rögzíteni.

A feladat részeként old meg, hogy a terméket csak ezen az egy eljáráson keresztül lehessen rögzíteni. (se grafikus nézetten a táblába beszúrva, se kézzel írt inserttel)

Ha mégis ilyen rögzítést próbálna meg valaki, akkor dobj egy `cannot_insert_product_exc` exceptiont.

8. Készíts egy eljárást, ami termék azonosító alapján adja vissza, hogy ki vásárolt a legtöbbet (darabszám alapján) az adott termékből.

Ha nem található a termék az azonosító alapján, akkor dobjon egy `product_not_found_exc` exceptiont.

9. Készíts egy típus, ami tartalmazz a termék nevét, az eladott darabszámot, és a bevételt.

Készíts a típushoz egy listát.

10. Az előzőleg elkészített listát használva készíts egy függvényt, ami egy adott kategóriára vonatkozóan visszaadja a termékek eladási statisztikáit.

11. Egy anonimusz blokkban írasd ki az elkészített függvény által visszaadott listát.