



Kérdések:

1. Listázzuk ki azoknak az autóknak a rendszámát, amelyet Fodor Eszter valamikor megvásárolt!
2. Listázzuk ki az összes autómárkát, és a hozzájuk tartozó típusokat (azokat a márkákat is listázzuk, amelyekhez nem tartoznak típusok)
3. Színenként mennyi a legnagyobb első vásárlási ár? A listát rendezzük szín szerint csökkenő sorrendbe.
4. Melyek azok a színek, amelyekhez 5-nél kevesebb autó tartozik?
5. Melyek azok az autótípusok, amelyekhez 5-nél kevesebb autó tartozik?
6. Melyik a legdrágább első vásárlási árú autó?
7. Ki az utolsó tulajdonosa annak az autónak, amelyikhez a legdrágább autófelfértékelés tartozik?
8. Ki volt az autók előző tulajdonosa?
9. Melyik az az autótípus, amelyekhez a legtöbb autó tartozik?
10. Színenként melyik a legdrágább első vásárlási árú autó?

11. Tárolt eljárás létrehozása:

Egy bejövő tábla paraméterből (UDT): @autok (Oszlopai: SZ_AUTO táblával megegyező) beszúrja az SZ_AUTO táblába azokat a sorokat/vagy módosítja (Kulcs: AZON oszlop), amelyekhez tartozik megfelelő autótípus, illetve autómárka. Amelyekhez nem, azokat a sorokat a program ne töltsse be az adatbázisba, és dobjon egy figyelmeztető üzenetet.

12. Elméleti kérdés (elegendő szöveges válasz vagy tárolt eljárások vázlata):

Folyamatos adatszolgáltatással szükség lenne az adatbázis feltöltésére egy külső Applikáció segítségével (Pl.: Java, .NET, stb) Milyen tárolt eljárásokat hozna létre, milyen bemenő paraméterekkel, mentési logikával? Figyelembe véve, hogy több száz ezres adatsorok esetén is a lehető legoptimálisabb megoldást nyújtsa.

Bónusz feladat:

Adott az alábbi lekérdezés, célja, hogy visszaadja az összes szükséges rendeléssort rangsorolva (f_azon mező), ami alapján később sorszámot tép a program és új sorokat szúr be az adatbázisba. Néhány esetben mégis rossz adatot ad vissza a lekérdezés, és nem megfelelő mennyiségű sorszámot tép a program miért?

```
select *
from (
select r.riktyszam
, r.rendelszam
, r.ugyfelkod
, t.tetelssz
, t.raktar
, t.etk
, t.gyartas
, case when c.eladbonthat = '1' then t.rendmenny - t.trendmenny - t.kellmenny - t.addki
      else floor((t.rendmenny - t.trendmenny - t.kellmenny - t.addki)/c.kiszereles) * c.kiszereles
end kikuldheto
, dense_rank() over (order by r.riktyszam) f_azon
from rendelf r
join rendelt t on t.riktyszam = r.riktyszam
join cikk c on c.etk = t.etk
where r.status = 101
and t.rendmenny - t.trendmenny - t.kellmenny - t.addki > 0.01
and t.raktar = 86
and c.gyartasszam = '0'
and c.cikkkatkod > 2
and t.rendall = 9
and r.ugyfelkod not in (select ugyfelkod from ugyfucs where ugyfcskod = 952) -- Ismétlésből kizárt ügyfél
and exists (select 1 from wf_feladat
      where f_bizid = r.riktyszam and f_modul = 'R' and f_status < 6
      and személykod in (select személykod from szemszcs where szemcskod = 262)) -- Ismétlő csoport
) y
where kikuldheto > 0
```