

Adatbázis-kezelés D08 – Forma-1

1.

Az adatbázist a könyvben részletesen bemutatott megoldásnak megfelelően helyi menük alkalmazásával VAGY a

CREATE DATABASE FORMA1 paranccsal hozzuk létre.

A txt fájlok importálását HeidiSQL kliens esetében a könyvben részletesen bemutatott mintának megfelelően végezzük el.

Felhívjuk a figyelmet, hogy a megoldásban található kód nem egyben futtatható, a feladatkiírásnak megfelelő részletekben kell lefuttatni (pl. külön-külön létrehozva a táblákat, futtatva a lekérdezéseket).

```
CREATE TABLE csapatok (  
    id INT PRIMARY KEY,  
    nev VARCHAR(30) NOT NULL,  
    nemzetiseg VARCHAR(25) NOT NULL  
)
```

```
CREATE TABLE palyak (  
    id INT PRIMARY KEY,  
    nev VARCHAR(40) NOT NULL,  
    hely VARCHAR(30) NOT NULL,  
    orszag VARCHAR(20) NOT NULL  
)
```

```
CREATE TABLE versenyzok (  
    id INT PRIMARY KEY,  
    keresztnev VARCHAR(40) NOT NULL,  
    vezeteknev VARCHAR(40) NOT NULL,  
    szuldatum DATE NOT NULL,  
    nemzetiseg VARCHAR(20) NOT NULL  
)
```

```
CREATE TABLE versenyek (  
    id INT PRIMARY KEY,  
    ev INT NOT NULL,  
    hanyadik INT NOT NULL,  
    palyaid INT NOT NULL,  
    nev VARCHAR(30) NOT NULL,  
    datum DATE NOT NULL,  
    ido VARCHAR(20) NULL,  
    FOREIGN KEY (palyaid) REFERENCES palyak(id)  
)
```

```
CREATE TABLE eredmenyek (  
    id INT PRIMARY KEY,  
    versenyid INT NOT NULL,  
    soforid INT NOT NULL,  
    csapatid INT NOT NULL,  
    helyezes INT,
```

```
pont INT DEFAULT(0),
korok INT,
FOREIGN KEY (soforid) REFERENCES versenyzok(id),
FOREIGN KEY (csapatid) REFERENCES csapatok(id),
FOREIGN KEY (versenyid) REFERENCES versenyek(id)
)
```

2.

```
SELECT * FROM versenyzok
WHERE versenyzok.szuldatum>'1980-01-01'
ORDER BY vezeteknev, keresztnév
```

3.

```
SELECT versenyek.ev, count(versenyek.id) AS 'futamok száma' FROM versenyek
GROUP BY versenyek.ev
```

4.

```
SELECT versenyek.hanyadik, palyak.nev, eredmenyek.pont FROM versenyzok
JOIN eredmenyek ON eredmenyek.soforid=versenyzok.id
JOIN versenyek ON eredmenyek.versenyid=versenyek.id
JOIN palyak ON versenyek.palyaid=palyak.id
WHERE versenyzok.vezeteknev LIKE 'Hamilton' AND versenyek.ev=2008
```

5.

```
SELECT SUM(eredmenyek.pont) AS 'össz pontszám', csapatok.nev, versenyek.ev FROM
eredmenyek
JOIN csapatok ON eredmenyek.csapatid=csapatok.id
JOIN versenyek ON eredmenyek.versenyid=versenyek.id
WHERE versenyek.ev=2015 AND csapatok.nev LIKE 'Ferrari'
```

6.

```
SELECT versenyzok.keresztnév, versenyzok.vezeteknev, sum(eredmenyek.pont) AS
'összpontszám', AVG(eredmenyek.korok) AS 'átlagos körszám' FROM versenyzok
JOIN eredmenyek ON eredmenyek.soforid=versenyzok.id
JOIN versenyek ON versenyek.id=eredmenyek.versenyid
WHERE versenyek.ev=2017
GROUP BY versenyzok.keresztnév, versenyzok.vezeteknev
```

7.

```
SELECT versenyzok.keresztnév, versenyzok.vezeteknev, COUNT(versenyek.id) AS
'indulások száma' FROM versenyzok
JOIN eredmenyek ON eredmenyek.soforid=versenyzok.id
JOIN versenyek ON eredmenyek.versenyid=versenyek.id
GROUP BY versenyzok.keresztnév, versenyzok.vezeteknev
ORDER BY COUNT(versenyek.id) DESC
```

8.

```
SELECT count(palyak.id) 'verseny db', palyak.nev FROM palyak
JOIN versenyek ON palyak.id=versenyek.palyaid
GROUP BY palyak.nev
```

ORDER BY COUNT(palyak.id) desc

9.

```
SELECT versenyzok.keresztnev, versenyzok.vezeteknev, count(versenyek.id) AS 'versenydb',  
SUM(eredmenyek.pont) AS 'összes pontszám' FROM versenyzok  
JOIN eredmények ON eredmények.soforid=versenyzok.id  
JOIN versenyek ON versenyek.id=eredmények.versenyid  
WHERE versenyek.ev>2000 AND versenyek.ev<2020  
GROUP BY versenyzok.keresztnev, versenyzok.vezeteknev  
ORDER BY COUNT(versenyek.id) DESC
```