1;#adatbázis létrehozása és karakter készlet meg a használandó abc sorrend be állítása (egy parancsba)

CREATE DATABASE magyar\_kgj

DEFAULT CHARACTER SET = uft8

DEFAULT COLLATE utf8\_hungarian\_ci;

5;#megváltoztatja a már létező adatbézis rendezését

ALTER DATABASE magyar\_kgj

DEFAULT COLLATE = latin2\_hungarian\_ci;

6;#törli az adatbázist mindenel ami benne van

DROP DATABASE magyar\_kgj;

7;#buli adatbázis

1,#utf8,magyar rendezés,

CREATE DATABASE buli\_kgj

DEFAULT CHARACTER SET = uft8

DEFAULT COLLATE utf8\_hungarian\_ci;

2,#elenörzés

SHOW VARIABLES LIKE "char%database";

SHOW VARIABLES LIKE "coll%database";

3,#tábla létrehozása

CREATE TABLE résztvevők

(

id INT NOT NULL PRIMARY KEY AUTO\_INCREMENT, #autómatikus növelés

név VARCHAR(20),

étel VARCHAR(30),

jön\_e CHAR(1),

dátuma DATE

);

4,#kiírja a tábla szerkezetétt

DESCRIBE résztvevők;

5,tábla belséjének megnézése;

SELECT \* FROM résztvevők;

6,#hozá adunk elemet a táblához #minden értéket sorba adunk meg

INSERT INTO résztvevők VALUES

(NULL,'Béla','husi','i',null); #ebbe az esetbe az első null az auto miat, és az utolsó mert nem rudjuk

INSERT INTO résztvevők (név,étel,jön\_e,dátuma)

VALUES('Patrik','sonka','i',date('2023-11-06'));

7,frissités

UPDATE résztvevők

SET dátuma='2023-11-06'

WHERE dátuma='2023-11-22';

8,#törlések

TRUNCATE TABLE;

DELETE FROM résztvevők

WHERE név='Sanyi';

DELETE FROM résztvevők

WHERE jön\_e='n';

DELETE FROM résztvevők

WHERE dátuma is null;

DELETE FROM résztvevők

WHERE id BETWEEN ;

DELETE FROM résztvevők

WHERE id BETWEEN 7 AND 10;

DROP TABLE résztvevők;

1. ISKOLA ADATBÁZIS - LÉTREHOZÁS

a., Hozzunk létre egy új adatbázist iskola néven (latin2 karakterkészlet, és azon belül a magyar nyelv

szabályai szerinti sorba rendezési beállításokkal. Ez legyen az adatbázison belül az alapértelmezett.

CREATE DATABASE iskola\_kgj

DEFAULT CHARACTER SET = latin2

DEFAULT COLLATE = latin2\_hungarian\_ci;

b., Készítsünk egy táblát a tanulók adatainak tárolásához:

tanulok (nev, osztaly, kor)

nev legfeljebb 40 karakteres szöveg, elsődleges kulcs

osztaly pontosan 3 karakter hosszúságú szöveg

kor egész szám

CREATE TABLE tanulok

(

nev VARCHAR(40) PRIMARY KEY,

osztaly CHAR(3),

kor INT DEFAULT 14

);

c., Állítsuk be a kor mezőnél az alapértelmezett értéket 14-re!

ALTER TABLE tanulok

ALTER kor SET DEFAULT 14;

Töltsük be az alábbi diákok adatait a táblába:

INSERT INTO tanulok VALUES

('Jánosi János', '10A', 14),

('Egyed Anna', '10B', 14),

('Bácskai Emese', '10A', 15),

('Balla Béla', '10B', 16),

('Egyem Károly', '10A', 14),

('Néma Levente', '11A', 15),

('Nagy Kata', '11A', 16),

('Nagy Ákos', '11B', 16),

('Aranyos Jolán', '11C', 17),

('Szalay Borbála', '11C', 16),

('Kósa Ferenc', '11B', 16),

('Kányi Máté', '11C', 17),

('Zoób Eszter', '11A', 15),

('Csernyik Éva', '12A', 17),

('Csóka Csaba', '12A', 17),

('Fóka Ferenc', '12A', 18);

2. ISKOLA ADATBÁZIS – LEKÉRDEZÉSEK1

Az iskola adatbázisban lévő tanulók táblában lekérdezések segítségével adjunk választ az alábbi kérdésekre:

a., Listázzuk ki a tanulók nevét és osztályát!

SELECT nev, osztaly

FROM tanulok;

//16-ot ír ki

b., Listázzuk ki az összes diák minden adatát a tanulók táblából!

SELECT \*

FROM tanulok;

//16-ot ír ki

c., Kik járnak a 11C osztályba? (csak név szükséges)

SELECT nev

FROM tanulok

WHERE osztaly = '11C';

//3 ilyen van

d., Írassuk ki a 16 éves tanulók nevét és osztályát!

SELECT nev, osztaly

FROM tanulok

WHERE kor = 16;

//5 ilyen tanuló van

e., Listázzuk ki a 14 és 17 év közötti tanulók nevét és életkorát!

SELECT nev, kor

FROM tanulok

WHERE kor BETWEEN 14 AND 17;

//15 tanuló ilyen tanuló van

f., Listázzuk ki a 16 alatti és 17 év fölötti tanulók nevét és életkorát (2 féle módon)!

×

SELECT nev, kor

FROM tanulok

WHERE kor < 16 OR kor > 17;

//7 ilyen van

SELECT nev, kor

FROM tanulok

WHERE kor NOT BETWEEN 16 AND 17;

SELECT nev, kor

FROM tanulok

WHERE kor NOT IN(16,17);

g., Listázzuk ki a 11A osztályba járó 15 éves tanulók nevét!

SELECT nev

FROM tanulok

WHERE kor = 15 AND osztaly = '11A';

//2 ilyen tanuló van

h., Adjuk meg a 11. évfolyam összes diákjának nevét!

×

SELECT nev

FROM tanulok

WHERE osztaly LIKE '11\_'; // \_ csak egy karaktert helyetesít

SELECT nev

FROM tanulok

WHERE osztaly LIKE '11%'; // % sok karaktert helyetesít

//8 ilyen tanuló van

i., Adjuk meg a 11. évfolyam 17 éves diákjainak nevét!

×

SELECT nev

FROM tanulok

WHERE osztaly LIKE '11\_' AND kor = 17;

//2 ilyen van

j., Listázzuk ki a tanulók nevét és osztályát, név szerint növekvő sorrendben!

SELECT nev, osztaly

FROM tanulok

ORDER BY nev ASC; //ASC = növekvő

k., Listázzuk ki a tanulók nevét és osztályát; osztály szerint növekvő, azon belül név szerint abc rendben!

SELECT nev, osztaly

FROM tanulok

ORDER BY osztaly,nev;

l., Adjuk meg a 10A tanulóit csökkenő abc rendben!

SELECT nev, osztaly

FROM tanulok

ORDER BY nev DESC; // DESC = fordított

m., Adjuk meg a 10. évfolyam tanulóit életkor szerint csökkenő sorrendben (név, osztály, kor)!

SELECT nev,osztaly,kor

FROM tanulok

WHERE osztaly LIKE '10%' //% több karakter ki töltése

ORDER BY kor DESC;

n., Adjuk meg az A osztályosok tanulóit évfolyam (növekvő), azon belül életkor szerint csökkenő, majd

névsor szerint ABC sorrendben (név, osztály, kor)!

SELECT nev,osztaly,kor

FROM tanulok

WHERE osztaly LIKE '%A'

ORDER BY osztaly ASC,kor DESC,nev ASC;

o., Listázzuk ki az intézmény összes osztályát növekvő sorrendben (de mindegyiket csak 1x)! Adjunk kétféle

megoldást!

SELECT DISTINCT osztaly

FROM tanulok

ORDER BY osztaly;

p., Listázzuk ki csökkenő sorrendben, hogy milyen életkorú diákok járnak az iskolába (2-féle módon is oldjuk

meg a feladatot)!

SELECT DISTINCT kor

FROM tanulok

ORDER BY kor DESC;

q., Listázzuk ki az osztályok szerinti életkorokat úgy, hogy osztály, azon belül kor szerint növekvő legyen a

sorrend! Adjunk 2-féle megoldást!

×

SELECT osztaly,kor

FROM tanulok

ORDER BY osztaly,kor;

r., Adjuk meg azok nevét, akik a névsorban Egyem Károly és Néma Levente között vannak!//edig

SELECT nev

FROM tanulok

WHERE nev BETWEEN 'Egyem Károly' AND 'Néma Levente';

3. ISKOLA ADATBÁZIS – LEKÉRDEZÉSEK2

a., Adjunk meg 3 diákot az iskolából (minden adat)!limit 3

SELECT \*

FROM tanulok

LIMIT 3;

b., Adjuk meg a névsor szerinti utolsó 5 diákot (minden adat)!

SELECT \*

FROM tanulok

ORDER BY nev DESC

LIMIT 5;

c., Adjunk meg a névsor szerinti 3. diáktól még 5 diákot (minden adat)!

×

SELECT \*

FROM tanulok

LIMIT 3,5;

d., Adjuk meg a legidősebb diák nevét!

SELECT nev

FROM tanulok

WHERE kor = (SELECT MAX(kor)FROM tanulok);

SELECT nev

from tanulok

ORDER by kor DESC

LIMIT 1;

e., Adjuk meg a legfiatalabb diák nevét!

SELECT nev

FROM tanulok

WHERE kor = (SELECT MIN(kor)FROM tanulok);

SELECT nev

from tanulok

ORDER by kor

LIMIT 1;

f., Jelenítsük meg a tanulók nevét és életkorát úgy, hogy a listában a megjelenített oszlop nevei: Tanuló

neve, Életkor legyenek!

SELECT nev as Tanulóneve, kor AS életkor

FROM tanulok;

order by nev

g., Hány tanuló jár az iskolába összesen? Létszám legyen az oszlop neve.

SELECT COUNT(\*) AS létszam

FROM tanulok;

h., Hány tanuló jár a 10. évfolyamra?

SELECT COUNT(\*) as letszam10evfoj

FROM tanulok

WHERE osztaly LIKE '10\_';

//5

i., Hány nagykorú tanuló jár az iskolába?

SELECT COUNT(\*) as nagykoru

FROM tanulok

WHERE kor >= 18;

j., Mennyi a tanulók átlagéletkora?

SELECT AVG(kor) as atlagkor

FROM tanulok;

//15.8125

k., Mennyi a 12A átlagéletkora?

SELECT AVG(kor) as atlagkor

FROM tanulok;

where osztaly = '12A';

l., Melyik évben születtek a 11. c osztály diákjai (Név, Szül.dátum)? A lista név szerinti abc sorrendben

jelenjen meg!

SELECT nev AS név,2023-kor AS Szüldátum

FROM tanulok

WHERE osztaly = '11C';

order by nev

m., Jelenítsük meg az összlétszámot, mellette az átlagéletkort!

SELECT COUNT(\*) AS 'létszam', AVG(kor) AS 'atlagkor'

FROM tanulok;

n., Adjuk meg, osztályonként hány tanuló van (Létszám)!

SELECT osztaly, COUNT(\*) AS 'létszám'

FROM tanulok

GROUP BY osztaly;

o., Adjuk meg a tanulók életkor szerinti megoszlását!

SELECT kor, COUNT(\*) AS 'létszám'

FROM tanulok

GROUP BY kor

;

p., Adjuk meg a 12A életkor szerinti megoszlását!

SELECT kor, COUNT(\*) AS 'létszám'

FROM tanulok

WHERE osztaly = '12A'

GROUP BY kor

;

q., Adjuk osztályonként az oda járók legmagasabb életkorát!

SELECT osztaly, MAX(kor) AS 'legidősebb'

FROM tanulok

GROUP BY osztaly

r., Adjuk meg az iskola tanulóinak életkor szerinti megoszlását úgy, hogy csak azok a kategóriák jelenjenek

meg, melyekbe pontosan 3 diák tartozik!

SELECT kor, COUNT(\*) as 'fő'

FROM tanulok

GROUP BY kor

HAVING 'fő' = 3; //utó feltétel

s., Adjuk meg az osztályok létszámát úgy, hogy csak azok az osztályok jelenjenek meg, amelyekbe 2-nél

többen járnak!

SELECT osztaly, COUNT(\*) as 'fő'

FROM tanulok

GROUP BY osztaly

HAVING 'fő' > 2;

4. ISKOLA ADATBÁZIS - ADATMÓDOSÍTÁS

Végezzük el az alábbi műveleteket:

a., Bácskai Emesét tegyük át a 10B-be!

UPDATE tanulok SET osztaly='10B'

WHERE nev = 'Bácskai Emesé';

b., Növeljük meg a tanulók életkorát 1 évvel!

UPDATE tanulok SET kor=kor + 1;

c., A 10B osztály nevét módosítsuk 10X-re!

UPDATE tanulok SET osztaly='10X'

WHERE ostaly = '10B';

d., Töröljük a 17 év feletti tanulókat!

DELETE FROM tanulok

WHERE kor > 17;

e., Töröljük a 11B osztály 16 évnél idősebb tanulóit!

DELETE FROM tanulok

WHERE osztaly = '11B' AND kor > 16;

f., Töröljük valamennyi tanulót! // ebben az eseten az öszeset

DELETE FROM tanulok

/\*

select count(nev) as letszam

avg(<kor>)

sum(<>)

MIN(<>)

MAX(<>)

select kor count(\*) as fő

from tanulok

group by kor

order by kor asc //lent van

having fő=3 / count(\*)=3

\*/

1.ADATBÁZIS, ÉS TÁBLÁK LÉTREHOZÁSA

a.

CREATE DATABASE autokereskedes

DEFAULT CHARACTER SET = utf8

DEFAULT COLLATE = utf8\_hungarian\_ci;

b.

CREATE TABLE Auto

(

Id int NOT NULL PRIMARY KEY,

Rendszam CHAR(7) NOT NULL,

Tipus VARCHAR(15),

Szin VARCHAR(6),

Gyartasiev int,

Ertek int

);

ellenörzés: DESCRIBE auto

CREATE TABLE Tulajdonos

(

TulajId INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

Autoazon INT,

Nev VARCHAR(30),

Született DATE,

Neme CHAR(1),

Elsotulaj BOOLEAN,

Varos VARCHAR(15)

);

ellenörzés: DESCRIBE tulajdonos

2.ADATOK BEIMPORTÁLÁSA

import

fromatum spec csak legfelső ; és 4. auto

3.LEKÉRDEZÉSEK

a.Adjuk meg a Skoda típusú autók rendszámát és gyártási évét!

SELECT Rendszam,Gyartasiev

FROM auto

WHERE Tipus = "Skoda"

b.Írassuk ki a rendszámokat, s mellé a tulajdonos nevét!

SELECT Rendszam,Nev

FROM auto INNER JOIN tulajdonos ON Id=Autoazon;

c.Jelenítsük meg valamennyi márkát, de csak egyszer

SELECT DISTINCT Tipus

FROM auto;

d.Hány piros Opel van? Az oszlop neve legyen: Piros Opelek száma

SELECT COUNT(\*) as PirosOpelekszáma

FROM auto

WHERE Tipus = "Opel" AND Szin = "Piros";

e.Mennyi a piros Opelek átlagértéke? Az oszlop neve legyen: Piros Opelek átlagértéke

SELECT AVG(Ertek) as PirosOpelekÁtlagértéke

FROM auto

WHERE Tipus = "Opel" AND Szin = "Piros";

f.Mennyi autótulajdonos van városonként? A lista a legtöbb tulajdonosú várossal kezdődjön és legalul

legyen a legkevesebb értékű város.

SELECT Varos,COUNT(\*) as Tulajdonosszam

FROM tulajdonos

GROUP BY Varos

ORDER BY Tulajdonosszam DESC;

g.Adjuk meg a budapesti tulajdonosú autók rendszámát, típusát, tulajdonosának nevét.

SELECT Rendszam,Tipus,Nev

FROM auto INNER JOIN tulajdonos ON Id = Autoazon

WHERE Varos = "Budapest";

h.Adjuk meg a legújabb és legdrágább autó rendszámát, típusát, tulajdonosának nevét és értékét.

SELECT Rendszam,Tipus,Nev,Ertek

FROM auto INNER JOIN tulajdonos on Id=Autoazon

WHERE Ertek = (SELECT MAX(Ertek) FROM auto) AND Gyartasiev = (SELECT MAX(Gyartasiev)FROM auto);

2.megoldás:

SELECT Rendszam,Tipus,Nev,Ertek

FROM auto INNER JOIN tulajdonos on Id=Autoazon

GROUP BY Gyartasiev DESC, Ertek DESC

LIMIT 1;

i.Keressük azon autók tulajdonosát, melyek rendszámában van „A” betű (írassuk ki a rendszámot és a

tulajdonos nevét is.)

SELECT Rendszam,Nev

FROM auto INNER JOIN tulajdonos on Id=Autoazon

WHERE Rendszam LIKE '%A%';

j.Keressük azon autók tulajdonosát, melyek rendszámában a második betű „A” (írassuk ki a rendszámot

és a tulajdonos nevét is.)

SELECT Rendszam,Nev

FROM auto INNER JOIN tulajdonos on Id=Autoazon

WHERE Rendszam LIKE '\_A%';

k.Keressük azon autók tulajdonosát, melyek rendszámában nincs „A” betű (írassuk ki a rendszámot és a

tulajdonos nevét is.)

SELECT Rendszam,Nev

FROM auto INNER JOIN tulajdonos on Id=Autoazon

WHERE Rendszam NOT LIKE '%A%';

l.Mit tudunk a LOL-003-as rendszámú autóról?

SELECT \*

FROM auto INNER JOIN tulajdonos on Id=Autoazon

WHERE Rendszam = 'LOL-003';

m.Melyek a 10 évnél nem régebbi autók? (Típus, tulajdonos, rendszám). A lista az autó típusa szerint

növekvő, azon belül pedig tulajdonos neve szerinti növekvő sorrendbe legyen rendezve.

SELECT Tipus,Nev,Rendszam

FROM auto INNER JOIN tulajdonos ON Id = Autoazon

WHERE 2024-Gyartasiev <= 10

ORDER BY Tipus, Nev;

n.Számoljuk meg, hány férfi és hány nő autótulajdonos van. A nők legyenek elöl.

SELECT Neme, COUNT(\*) as Autotulajdonosok

FROM tulajdonos

GROUP BY Neme

ORDER BY Neme DESC;

o.Írassuk ki a városokat (és az ott lakó tulajdonosok számát), melyben legalább két autótulajdonos lakik.

Nevezzük át az oszlopot Fő-re.

SELECT Varos,COUNT(\*) as Fő

FROM tulajdonos

GROUP BY Varos

HAVING Fő >= 2;

p.Írassuk ki a Károlyokat és autójuk típusát.

SELECT Tipus,Nev

FROM auto INNER JOIN tulajdonos on Id=Autoazon

WHERE Nev LIKE "Károly"

;

q.Hány autó van első tulajdonosnál? Nevezzük át az oszlopot

SElECT COUNT(\*) AS ELSŐTULAJ

FROM tulajdonos

WHERE Elsotulaj = 1

r.Hány budapesti első tulajdonos van? Az oszlopot is nevezzük át.

SElECT COUNT(\*) AS "ELSŐTULAJ-bp"

FROM tulajdonos

WHERE Elsotulaj = 1 AND Varos = "Budapest";

s.Készítsük el az autók szín szerinti eloszlását (milyen színű autóból hány db van).

SElECT Szin,COUNT(\*) AS db

FROM auto

Group by Szin;