**MINISTERUL EDUCAŢIEI AL REPUBLICII MOLDOVA**

****

**UNIVERSITATEA TEHNICĂ A MOLDOVEI**

**FACULTATEA CALCULATOARE INFORMATICĂ ŞI MICROELECTRONICĂ**

**CATEDRA INFORMATICA APLICATĂ**

**Lucrarea nr. 2**

**LA BD**

**Povestea:”Ursul pacalit de vulpe”**

**Numele :** *Gherman* *Catalin*

**Numărul :** *3*

**Grupa :** *MI-181*

**Pînă la data :** *…/…/……*

## Lucrarea efectuată # …

|  |  |  |
| --- | --- | --- |
| Nr | Sarcina | Puncte |
| 1 | *Structura,* |  |
| 2 | *Continutul,* |  |
| 3 | *Exemplul/exemple,* |  |
| 4 | *Utilitatea, domeniile de aplicabilitate.* |  |
| 5 | *Recomandări dacă studentul le consideră utile.* |  |
| Total | |  |

**Verificat / Nota :** *………………………….*

**Data :** *………………………….*

**Povestea:”Ursul pacalit de vulpe”**

1) Era odată intr-o poiana o vulpe vicleana, un urs brun, o caprioara sprintena, un arici posomarat si un lup mandru. Vulpea umblase o noapte întreagă după hrană ca carne, peste, cereale şi nu găsise nicăiri. Făcându-se ziua albă, vulpea iese la marginea drumului şi se culcă sub o tufă de maces pe langa cele de fistic, zmeura, coacaza si agris existente, gândindu-se ce să mai facă, ca să poată găsi ceva de mâncare.

Şăzând vulpea cu botul întins pe labele de dinainte, îi vine miros de peşte. Atunci ea rădică puţin capul şi, uitându-se la vale, în lungul drumului, zăreşte venind un car plin cu peste, fructe, legume si toate fiind acoperite cu paie tras de boi. – Bun! gândi vulpea. Iaca hrana ce-o aşteptam eu. Şi îndată iese de sub tufă şi se lungeşte în mijlocul drumului, ca şi cum ar fi fost moartă.

Carul apropiindu-se de vulpe, ţăranul ce mâna boii o vede şi, crezând că-i moartă cu adevărat, strigă la boi: Aho! Aho! Boii se opresc. Ţăranul vine spre vulpe, se uită la ea de aproape şi, văzând că nici nu suflă, zice: Bre! da’ cum naiba a murit vulpea asta aici?! Ti! ce frumoasă caţaveică am să fac nevestei mele din blana istui vulpoiu! Zicând aşa, apucă vulpea de după cap şi, târând-o până la car, se opinteşte ş-o aruncă deasupra peştelui. Apoi strigă la boi: "Hăis! Joian, cea! Bourean". Boii pornesc.

Ţăranul mergea pe lângă boi şi-i tot îndemna să meargă mai iute, ca s-ajungă degrabă acasă şi să ieie pelea vulpii. Însă, cum au pornit boii, vulpea a şi început cu picioarele a împinge peştele din car jos. Ţăranul mâna, carul scârţâia, şi peştele din car cădea.

După ce hoaţa de vulpe a aruncat o mulţime de peşte pe drum, bine…şor! sare şi ea din car şi, cu mare grabă, începe a strânge peştele de pe drum.

2) După ce l-a strâns grămadă, îl ia, îl duce la bizunia sa şi începe a mânca, că ta…re-i mai era foame!

Tocmai când începuse a mânca, iaca vine la dânsa ursul. – Bună masa, cumătră! Ti!!! da’ ce mai de peşte ai! Dă-mi şi mie, că ta…re! mi-i poftă! – Ia mai pune-ţi pofta-n cuiu, cumătre, că doar nu pentru gustul altuia m-am muncit eu. Dacă ţi-i aşa de poftă, du-te şi-ţi moaie coada-n baltă, ca mine, şi-i avea peşte să mănânci. – Învaţă-mă, te rog, cumătră, că eu nu ştiu cum se prinde peştele. Atunci vulpea rânji dinţii şi zise: Alei, cumătre! da’ nu ştii că nevoia te duce pe unde nu-ţi e voia şi te-nvaţă ce nici gândeşti? Ascultă, cumătre: vrei să mănânci peşte? Du-te desară la băltoaga cea din marginea pădurei deoarece e mai aproape decat cea din centru, vârâ-ţi coada-n apă şi stăi pe loc, fără să te mişti, până despre ziuă; atunci smunceşte vârtos spre mal şi ai să scoţi o mulţime de peşti: carasi, crapi, caracude, amuri poate îndoit şi-ntreit de cât am scos eu.

3) Ursul, nemaizicând nici o vorbă, aleargă-n fuga mare la băltoaga din marginea pădurei şi-şi vâră-n apă toată coada! În acea noapte începuse a bate un vânt răce, de îngheţa limba-n gură şi chiar cenuşa de sub foc. Îngheaţă zdravăn şi apa din băltoagă, şi prinde coada ursului ca într-un cleşte. De la o vreme, ursul, nemaiputând de durerea cozei şi de frig, smunceşte o dată din toată puterea. Şi, sărmanul urs, în loc să scoată peşte, rămâne făr’ de coadă!

4) ncepe el acum a mornăi cumplit ş-a sări în sus de durere; şi-nciudat pe vulpe că l-a amăgit, se duce s-o ucidă în bătaie. Dar şireata vulpe ştie cum să se ferească de mânia ursului. Ea ieşise din bizunie şi se vârâse în scorbura unui copac stramb din apropiere; şi când văzu pe urs că vine făr’ de coadă, începu a striga: – Hei cumătre! Dar ţi-au mâncat peştii coada, ori ai fost prea lacom ş-ai vrut să nu mai rămâie peşti în baltă?

5) Ursul, auzind că încă-l mai ie şi în râs, se înciudează şi mai tare şi se răpede iute spre copac; dar gura scorburei fiind strâmtă, ursul nu putea să încapă înlăuntru. Atunci el caută o creangă cu cârlig şi începe a cotrobăi prin scorbură, ca să scoată vulpea afară, şi să-i deie de cheltuială… Dar când apuca ursul de piciorul vulpei, ea striga: "Trage, nătărăule! mie nu-mi pasă, că tragi de copac…" Iar când anina cârligul de copac, ea striga: "Valeu, cumătre! nu trage, că-mi rupi piciorul!"

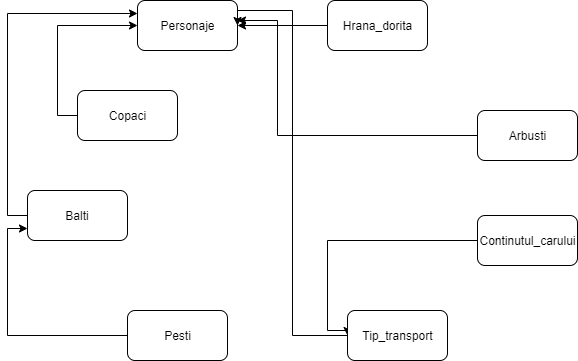
6) În zadar s-a năcăjit ursul, de-i curgeau sudorile, că tot n-a putut scoate vulpea din scorbura copacului. Şi iaca aşa a rămas ursul păcălit de vulpe!

**PASUL 1**, Elaborati si prezentati modelul conceptual al BD, utilizind paradigma Input -> Output.

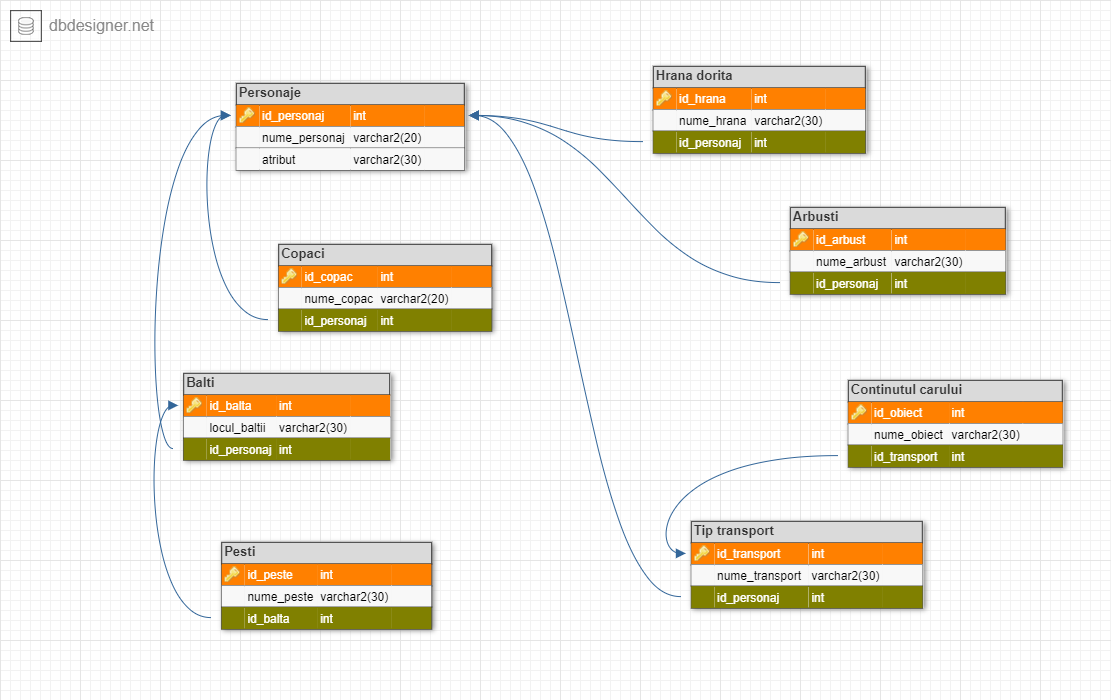
1. **Personaje:** „Vulpea vicleana”, „Urs brun”, „Caprioara sprintena”, „Arici veseli”, „Lup mandru”, „Taran priceput”;
2. **Hrana\_dorita:** „Carne”, „Peste”, „Cereale”;
3. **Arbusti:** „Coacaza”, „Agris”, „Fistic”, „Zmeura”, „Maces”;
4. **Tip\_transport:** „Car”, „Bolid”, „Tractor”;
5. **Continutul carului:** „Peste”, „Fructe”, „Legume”, „Paie”, „Vulpea”;
6. **Balti:** „Margine”, „Centru”
7. **Pesti:** „Carasi”, „Crapi”, „Caracunde”, „Amuri”;
8. **Copaci:** „Stramb”, „Drept”, „Ingust”, „Mic”.

**PASUL 2**, Elaborati si prezentati modelul Logic al BD, utilizind paradigma Input -> Output. Prezentați schema ER.

**Modelul conceptual:**



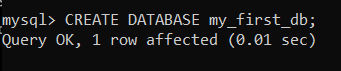
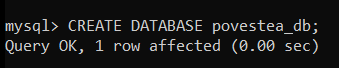
**ER:**

****

**PASUL 3**, Executati SQL for Beginners Part 1 pentru modelul Logic elaborat, în conformitate cu templat-ul propus in Anexa\_\_1 fisier atasat.

*Efectuati aceeași sarcina pentru cazul Proiectului Dvs, adică utilizati aceleasi comenzi MYSQL din fisierul mentionat pentru BD si tabelele concrete a Proiectului Dvs concret. (Vor fi executate aceleasi comenzi!!)*

CREATE DATABASE my\_first\_db;

DROP DATABASE my\_first\_db;

USE my\_first\_db

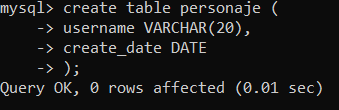
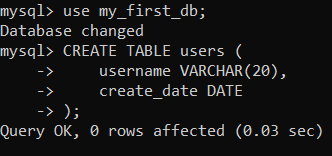
 

CREATE TABLE users (

    username VARCHAR(20),

    create\_date DATE

);



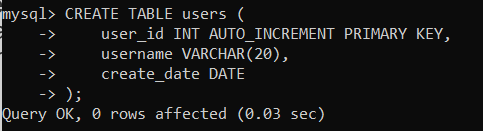
CREATE TABLE users (

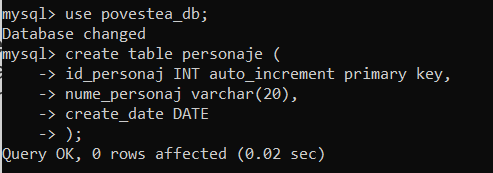
    user\_id INT AUTO\_INCREMENT PRIMARY KEY,

    username VARCHAR(20),

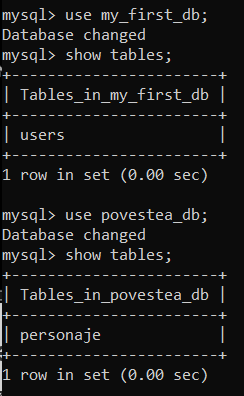
    create\_date DATE

);

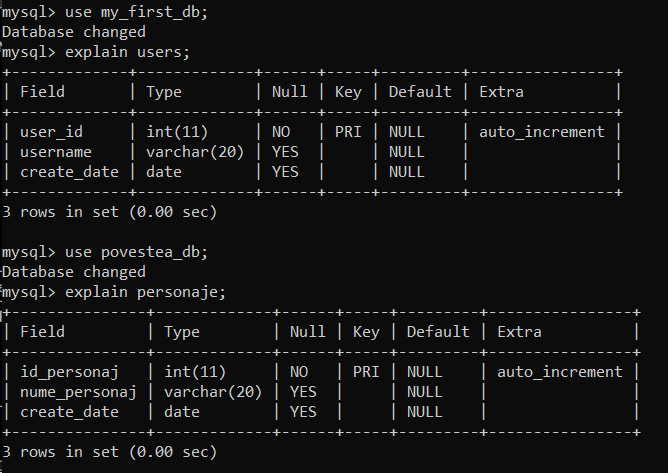




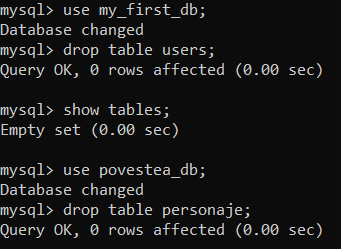
SHOW TABLES;



EXPALIN users;



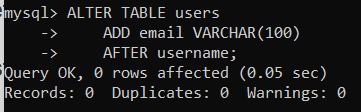
DROP TABLE users;

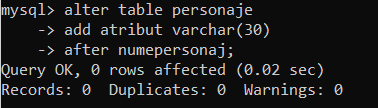


ALTER TABLE users

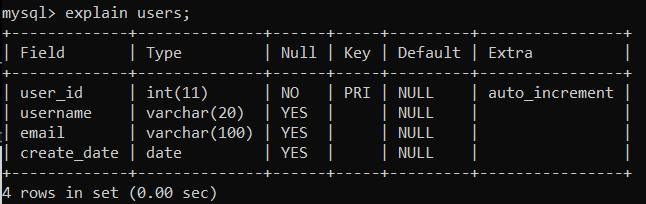
    ADD email VARCHAR(100)

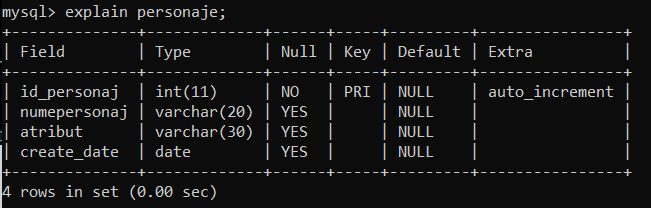
    AFTER username;



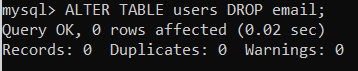


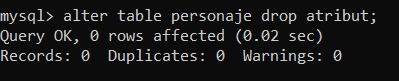
EXPLAIN users;





ALTER TABLE users DROP email;

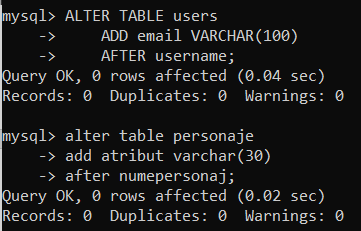




ALTER TABLE users

    ADD email VARCHAR(100)

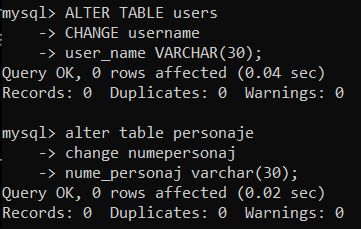
    AFTER username;



ALTER TABLE users

CHANGE username

user\_name VARCHAR(30);



INSERT INTO users VALUES (

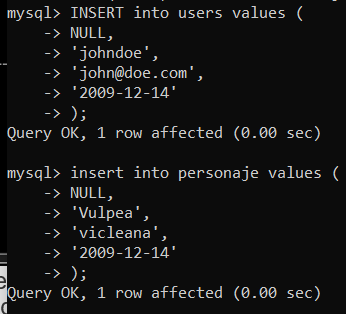
NULL,

`johndoe`,

`john@doe.com`,

`2009-12-14`

);

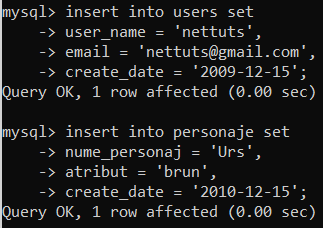


INSERT INTO users SET

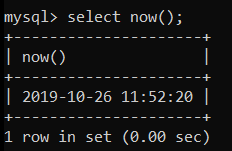
User\_name = ‘nettuts’,

Email = ‘nettuts@gmail.com’,

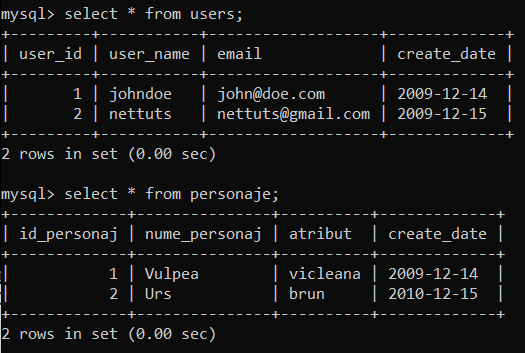
Create\_date = ‘2009-12-15’;



SELECT NOW();

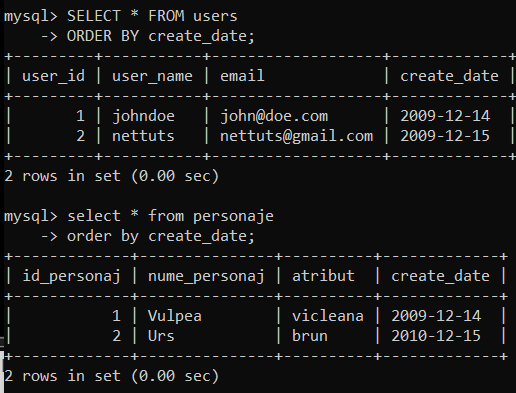


SELECT \* from users;



SELECT \* FROM users

ORDER BY create\_date;

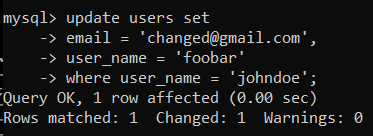


UPDATE users SET

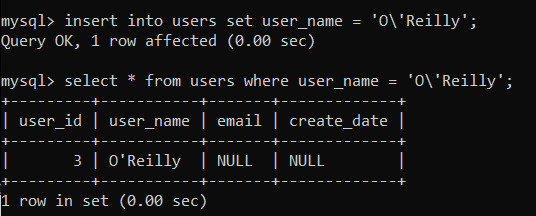
Email = ‘changed@gmail.com’,

User\_name = ‘foobar’

Where user\_name = ‘johndoe’;



INSERT INTO users SET user\_name = ‘O\’Reilly’;



**PASUL 4**, Executati SQL for Beginners Part 2 pentru modelul Logic elaborat, în conformitate cu templat-ul propus propus in Anexa\_\_2 fisier atasat.

*Efectuati aceeași sarcina pentru cazul Proiectului Dvs, adică utilizati aceleasi comenzi MYSQL din fisierul mentionat pentru BD si tabelele concrete a Proiectului Dvs concret. (Vor fi executate aceleasi comenzi!!)*

CREATE TABLE states (

  id INT AUTO\_INCREMENT,

  name VARCHAR(20),

  join\_year INT,

  population INT,

  PRIMARY KEY (id),

  UNIQUE (name),

  KEY (join\_year)

);

INSERT INTO states VALUES

(1, 'Alabama', 1819, 4661900),

(2, 'Alaska', 1959, 686293),

(3, 'Arizona', 1912, 6500180),

(4, 'Arkansas', 1836, 2855390),

(5, 'California', 1850, 36756666),

(6, 'Colorado', 1876, 4939456),

(7, 'Connecticut', 1788, 3501252),

(8, 'Delaware', 1787, 873092),

(9, 'Florida', 1845, 18328340),

(10, 'Georgia', 1788, 9685744),

(11, 'Hawaii', 1959, 1288198),

(12, 'Idaho', 1890, 1523816),

(13, 'Illinois', 1818, 12901563),

(14, 'Indiana', 1816, 6376792),

(15, 'Iowa', 1846, 3002555),

(16, 'Kansas', 1861, 2802134),

(17, 'Kentucky', 1792, 4269245),

(18, 'Louisiana', 1812, 4410796),

(19, 'Maine', 1820, 1316456),

(20, 'Maryland', 1788, 5633597),

(21, 'Massachusetts', 1788, 6497967),

(22, 'Michigan', 1837, 10003422),

(23, 'Minnesota', 1858, 5220393),

(24, 'Mississippi', 1817, 2938618),

(25, 'Missouri', 1821, 5911605),

(26, 'Montana', 1889, 967440),

(27, 'Nebraska', 1867, 1783432),

(28, 'Nevada', 1864, 2600167),

(29, 'New Hampshire', 1788, 1315809),

(30, 'New Jersey', 1787, 8682661),

(31, 'New Mexico', 1912, 1984356),

(32, 'New York', 1788, 19490297),

(33, 'North Carolina', 1789, 9222414),

(34, 'North Dakota', 1889, 641481),

(35, 'Ohio', 1803, 11485910),

(36, 'Oklahoma', 1907, 3642361),

(37, 'Oregon', 1859, 3790060),

(38, 'Pennsylvania', 1787, 12448279),

(39, 'Rhode Island', 1790, 1050788),

(40, 'South Carolina', 1788, 4479800),

(41, 'South Dakota', 1889, 804194),

(42, 'Tennessee', 1796, 6214888),

(43, 'Texas', 1845, 24326974),

(44, 'Utah', 1896, 2736424),

(45, 'Vermont', 1791, 621270),

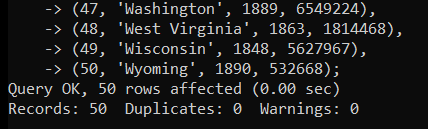
(46, 'Virginia', 1788, 7769089),

(47, 'Washington', 1889, 6549224),

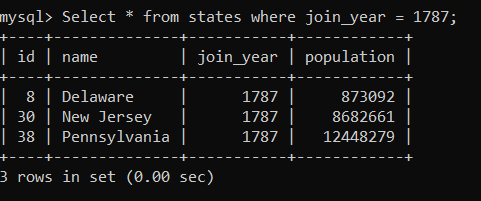
(48, 'West Virginia', 1863, 1814468),

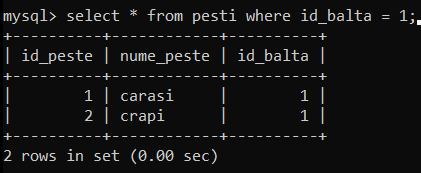
(49, 'Wisconsin', 1848, 5627967),

(50, 'Wyoming', 1890, 532668);

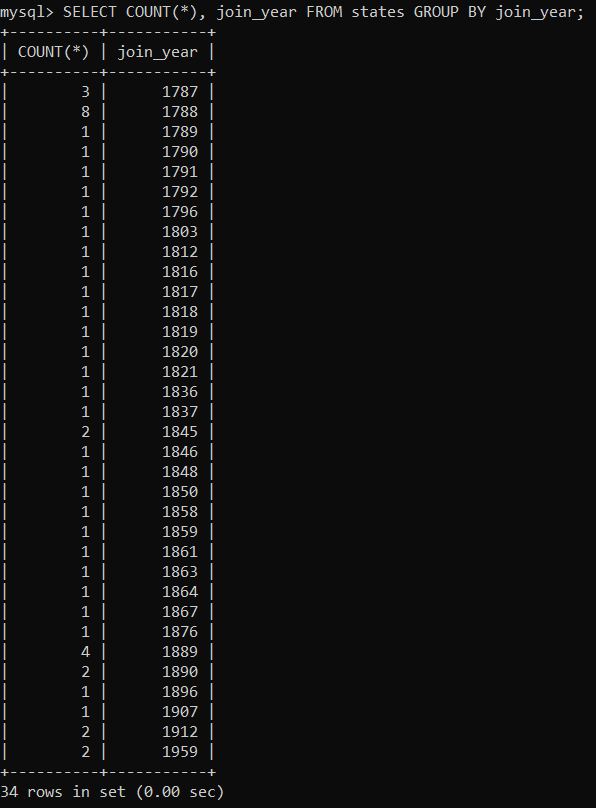


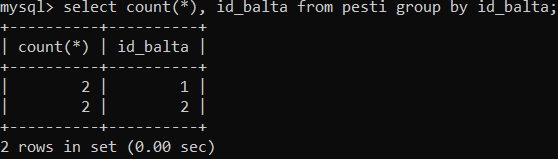
Select \* from states where join\_year = 1787;



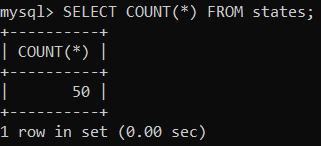


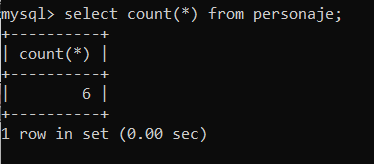
SELECT COUNT(\*), join\_year FROM states GROUP BY join\_year;



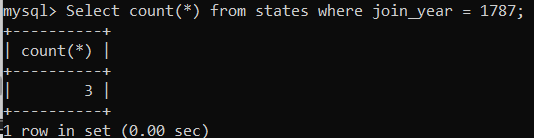


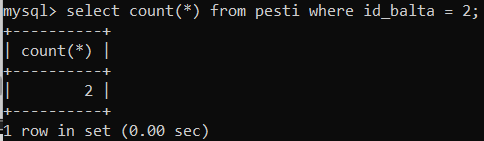
SELECT COUNT(\*) FROM states;



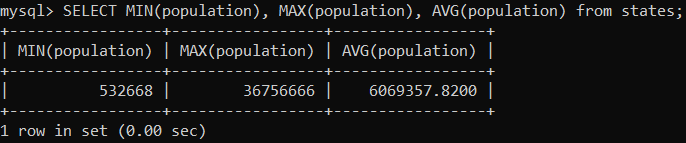


Select count(\*) from states where join\_year = 1787;



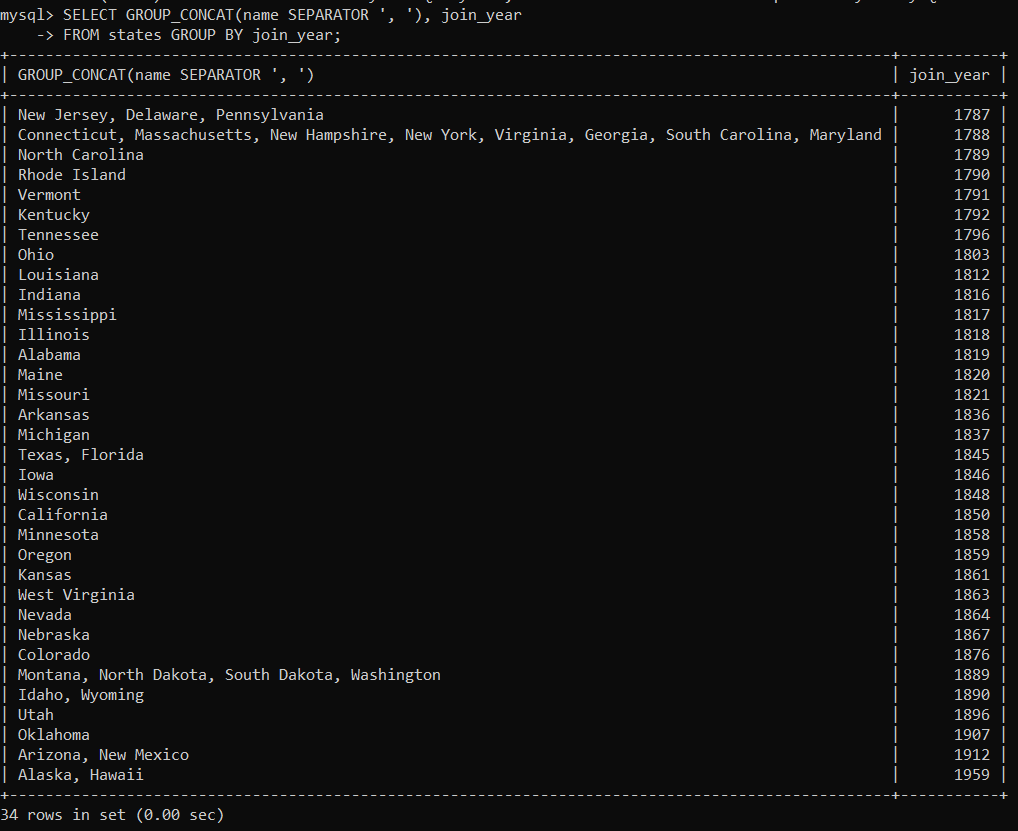


SELECT MIN(population), MAX(population), AVG(population) from states;

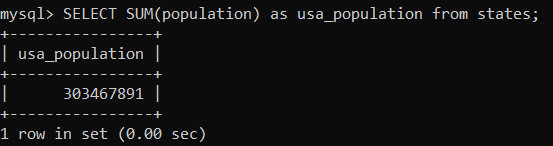


SELECT GROUP\_CONCAT(name SEPARATOR ', '), join\_year

FROM states GROUP BY join\_year;



SELECT SUM(population) as usa\_population from states;



SELECT

    SUM(

        IF(population > 5000000, 1, 0)

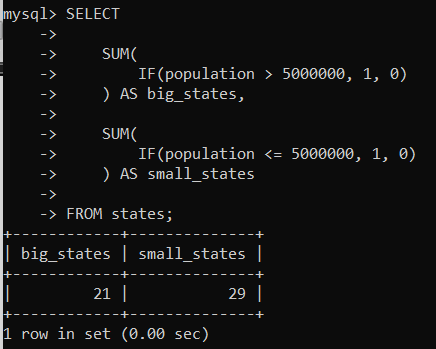
    ) AS big\_states,

    SUM(

        IF(population <= 5000000, 1, 0)

    ) AS small\_states

FROM states;



SELECT

COUNT(\*),

CASE

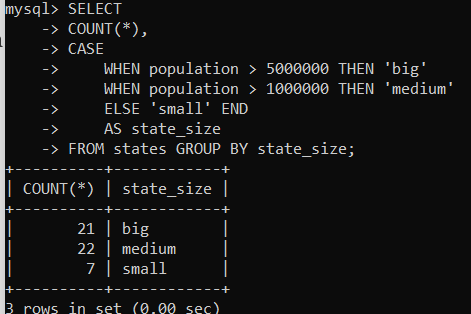
    WHEN population > 5000000 THEN 'big'

    WHEN population > 1000000 THEN 'medium'

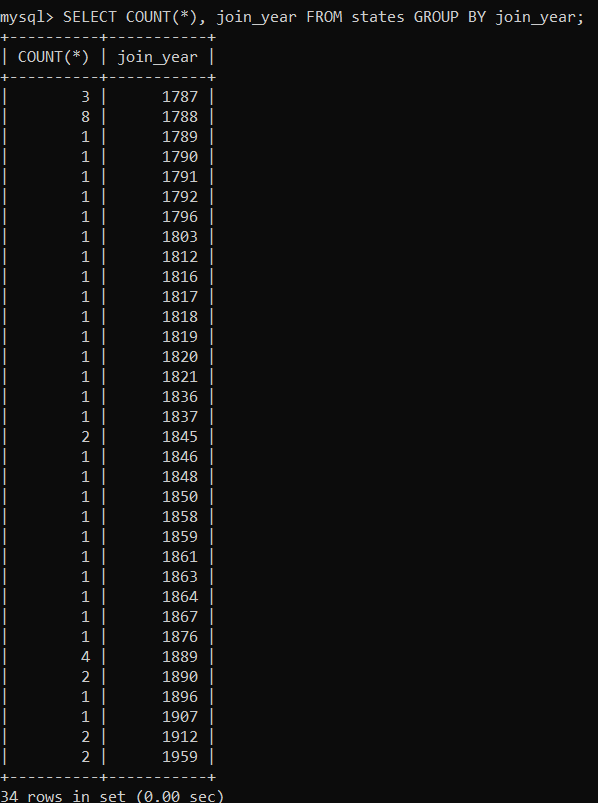
    ELSE 'small' END

    AS state\_size

FROM states GROUP BY state\_size;



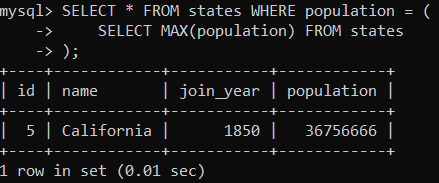
SELECT COUNT(\*), join\_year FROM states GROUP BY join\_year;



SELECT \* FROM states WHERE population = (

    SELECT MAX(population) FROM states

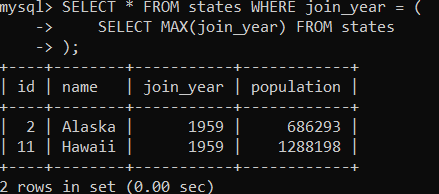
);



SELECT \* FROM states WHERE join\_year = (

    SELECT MAX(join\_year) FROM states

);



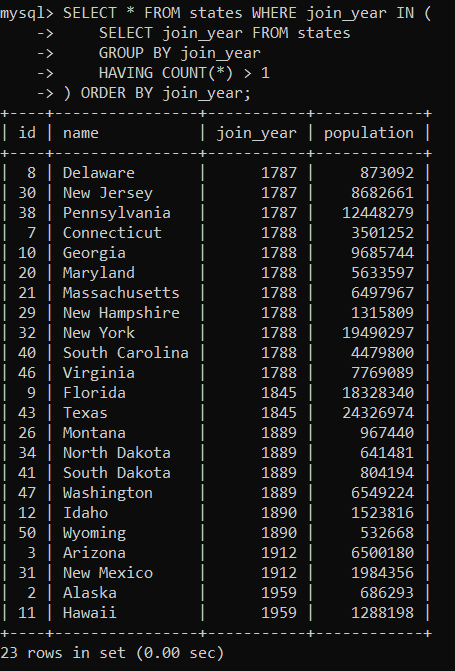
SELECT \* FROM states WHERE join\_year IN (

    SELECT join\_year FROM states

    GROUP BY join\_year

    HAVING COUNT(\*) > 1

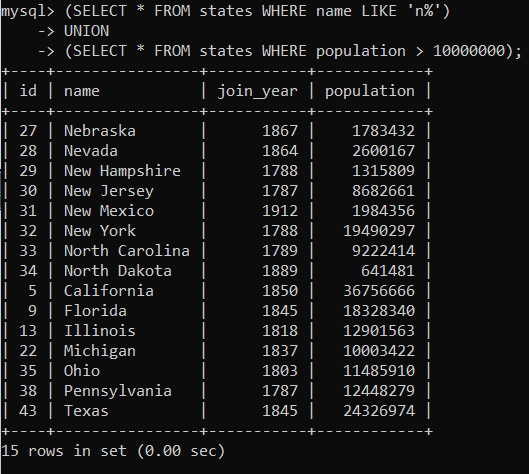
) ORDER BY join\_year;



(SELECT \* FROM states WHERE name LIKE 'n%')

UNION

(SELECT \* FROM states WHERE population > 10000000);



**PASUL 5**, Executati SQL for Beginners Part 3 pentru modelul Logic elaborat, în conformitate cu templat-ul propus in Anexa\_\_3 fisier atasat.

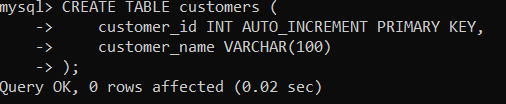
*Efectuati aceeași sarcina pentru cazul Proiectului Dvs, adică utilizati aceleasi comenzi MYSQL din fisierul mentionat pentru BD si tabelele concrete a Proiectului Dvs concret. (Vor fi executate aceleasi comenzi!!)*

**CREATE TABLE customers (**

**customer\_id INT AUTO\_INCREMENT PRIMARY KEY,**

**customer\_name VARCHAR(100)**

**);**



**CREATE TABLE orders (**

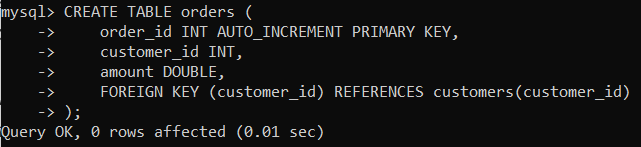
**order\_id INT AUTO\_INCREMENT PRIMARY KEY,**

**customer\_id INT,**

**amount DOUBLE,**

**FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id)**

**);**



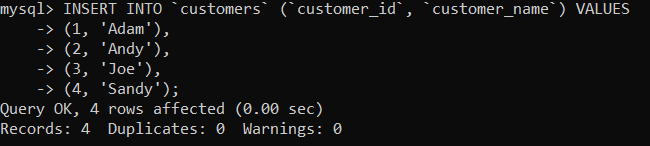
INSERT INTO `customers` (`customer\_id`, `customer\_name`) VALUES

(1, 'Adam'),

(2, 'Andy'),

(3, 'Joe'),

(4, 'Sandy');



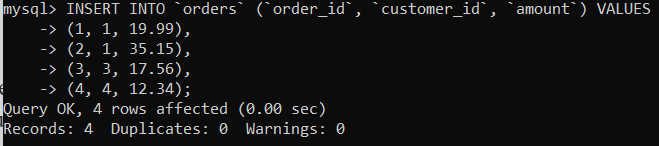
INSERT INTO `orders` (`order\_id`, `customer\_id`, `amount`) VALUES

(1, 1, 19.99),

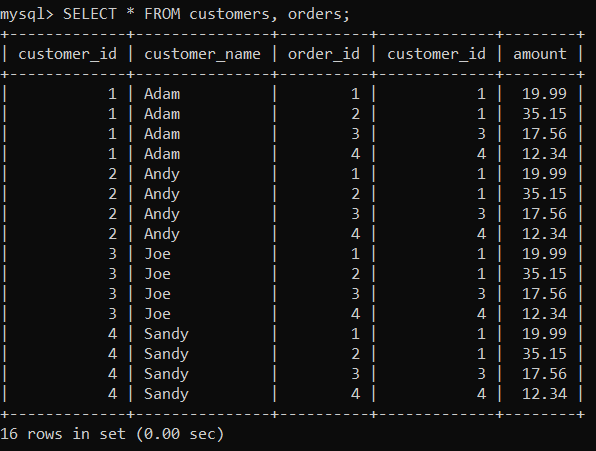
(2, 1, 35.15),

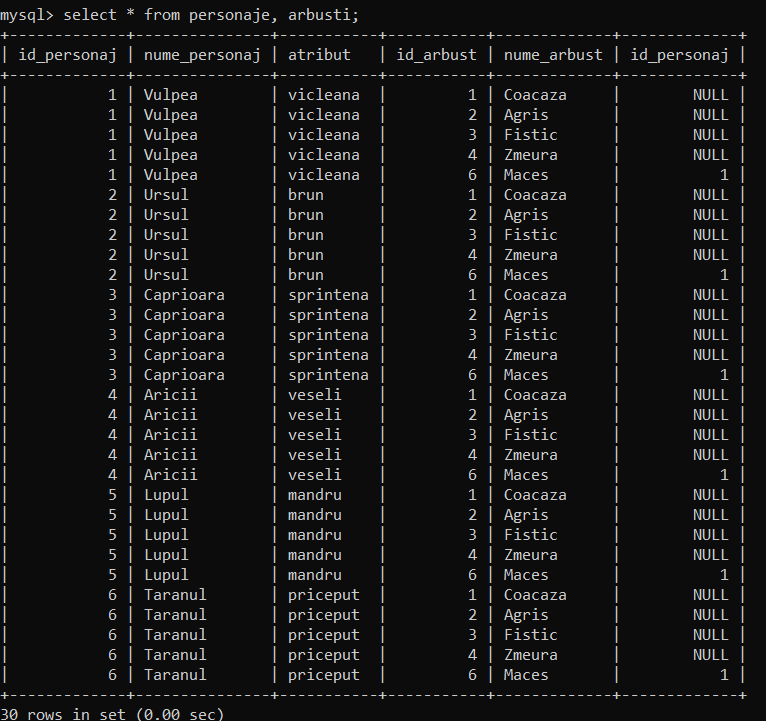
(3, 3, 17.56),

(4, 4, 12.34);

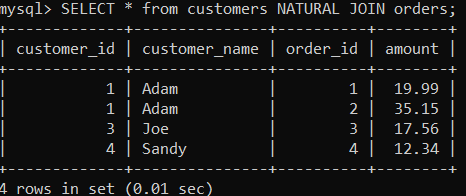


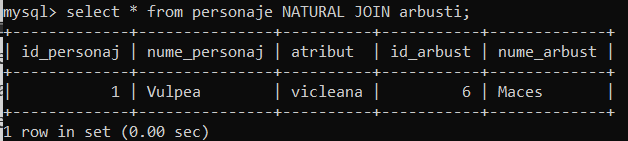
SELECT \* FROM customers, orders;





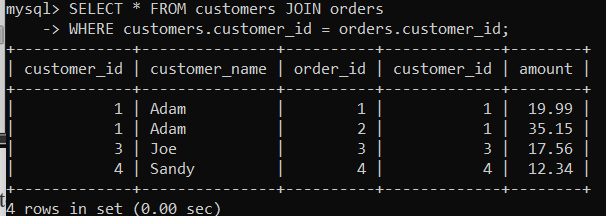
SELECT \* from customers NATURAL JOIN orders;

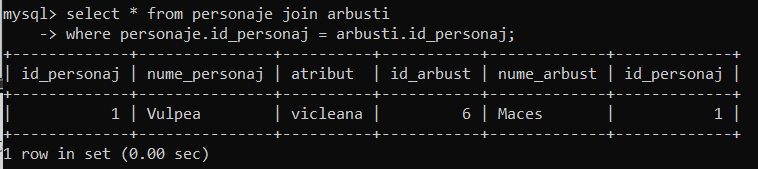




SELECT \* FROM customers JOIN orders

WHERE customers.customer\_id = orders.customer\_id;

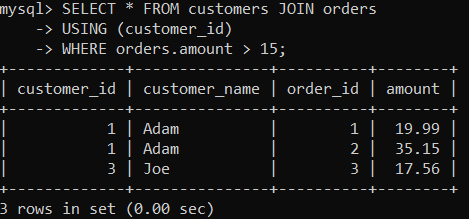


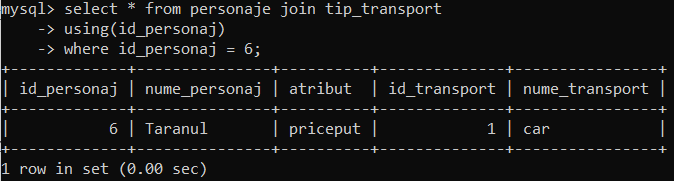


SELECT \* FROM customers JOIN orders

USING (customer\_id)

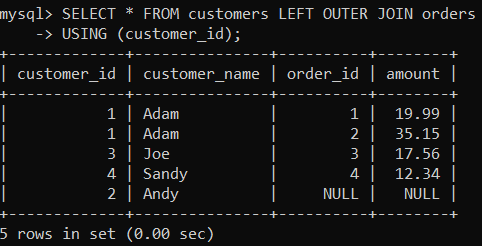
WHERE orders.amount > 15;

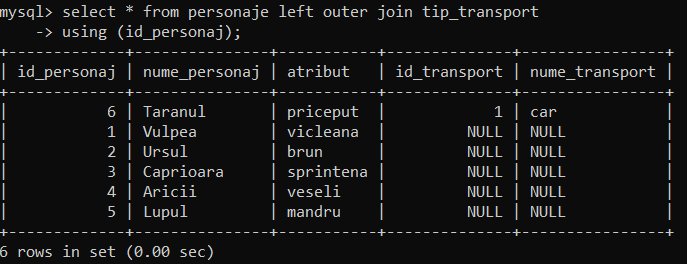




SELECT \* FROM customers LEFT OUTER JOIN orders

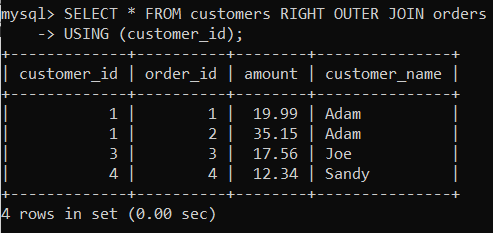
USING (customer\_id);

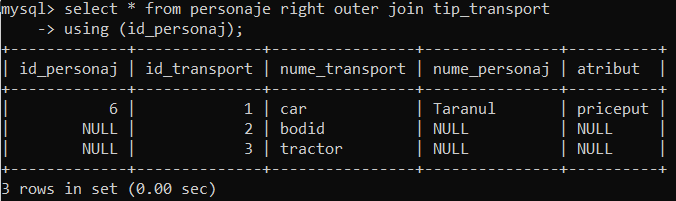




SELECT \* FROM customers RIGHT OUTER JOIN orders

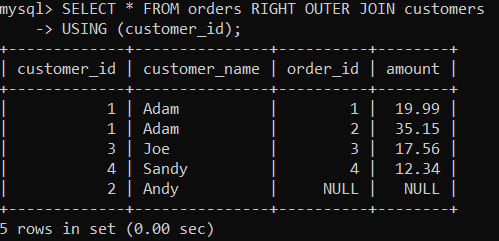
USING (customer\_id);

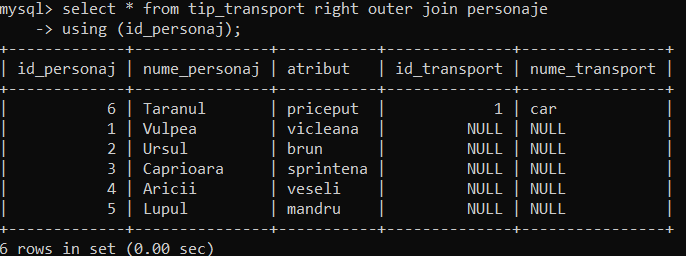




SELECT \* FROM orders RIGHT OUTER JOIN customers

USING (customer\_id);





**Concluzie:**

In urma realizarii acestui laborator am invat comenzile generale si proprietatile limbajului imens MYSQL. Si deci am inteles ca MySQL este un sistem de gestiune a bazelor de date relaționale, produs de compania suedeza MySQL AB și distribuit sub Licența Publică Generală GNU. Este cel mai popular SGBD open-source la ora actuală, fiind o componentă cheie a stivei LAMP (Linux, Apache, MySQL, PHP).

MySQL este componentă integrată a platformelor LAMP sau WAMP (Linux/Windows-Apache-MySQL-PHP/Perl/Python). Popularitatea sa ca aplicație web este strâns legată de cea a PHP-ului care este adesea combinat cu MySQL și denumit Duo-ul Dinamic. În multe cărți de specialitate este precizat faptul ca MySQL este mult mai ușor de învățat și folosit decât multe din aplicațiile de gestiune a bazelor de date, ca exemplu comanda de ieșire fiind una simplă și evidentă: „exit” sau „quit”.

Pentru a administra bazele de date MySQL se poate folosi modul linie de comandă sau, prin descărcare de pe internet, o interfață grafică: MySQL Administrator și MySQL Query Browser. Un alt instrument de management al acestor baze de date este aplicația gratuită, scrisă în PHP, phpMyAdmin.