****

**MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII**

**AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Informatică şi Ingineria Sistemelor**

**RAPORT**

***Laborator nr.2***

***Disciplina: “Baze de date”***

**Efectuat:**

**Calancea Cătălin** MI-222

**Verificat:**

**Perebinos Mihail,** *a.u.*

**Chișinău – 20****23**

**Cuprins:**

[Tema: 3](#_Toc151574121)

[1.Proiectarea Input-Output: 3](#_Toc151574122)

[1.1 Modelul Conceptual 3](#_Toc151574123)

[1.2 Modelul Logic 4](#_Toc151574124)

[1.3 Machetul bazei de date 5](#_Toc151574125)

[2.Proiectarea Input-Output: 5](#_Toc151574126)

[**2.2 Elaborarea pasilor din SQL for Beginners:** 11](#_Toc151574127)

[3. Povestea modificata 27](#_Toc151574128)

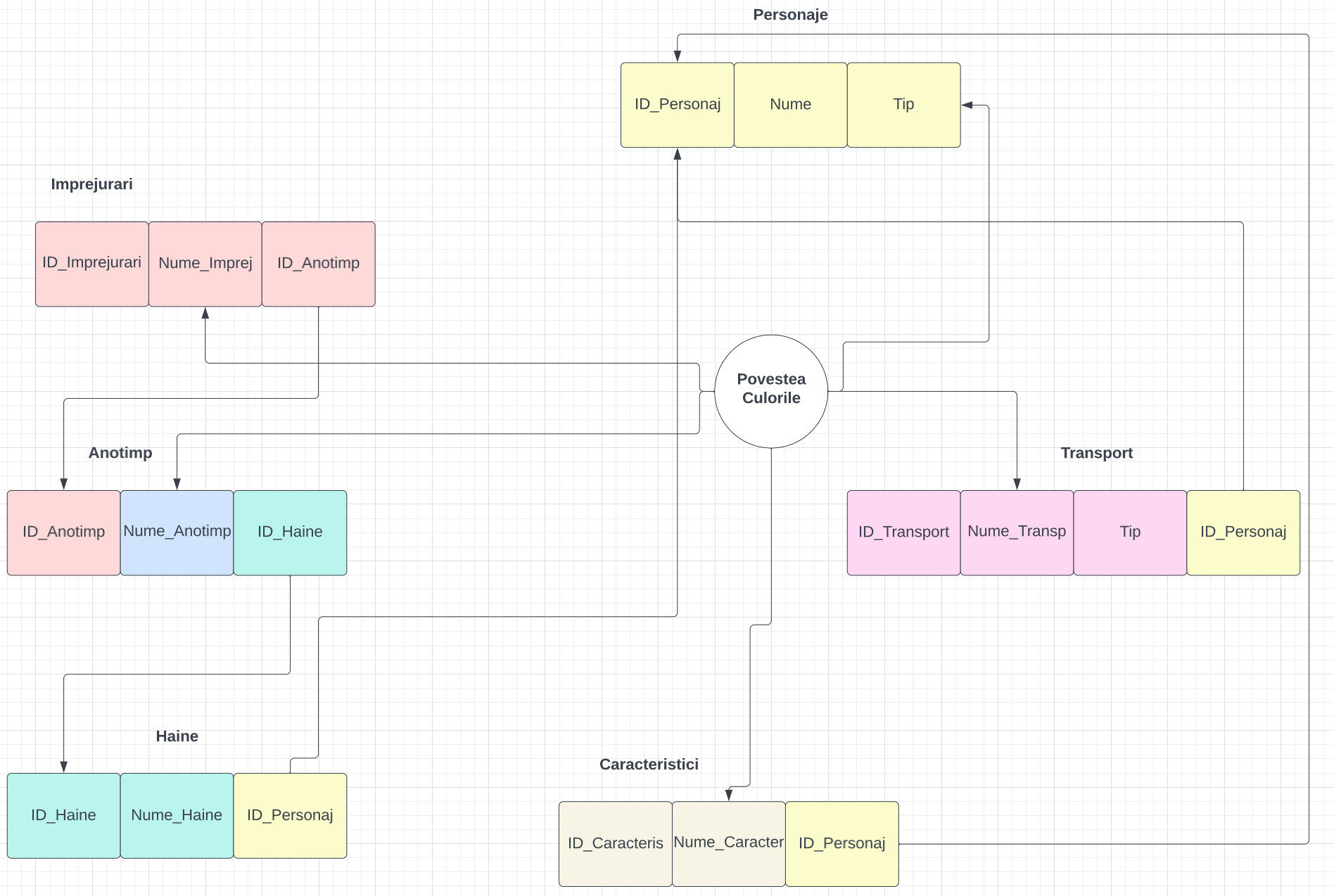
[Concluzie 28](#_Toc151574129)

Tema: Proiectarea Bazei de Date pentru proiectul primit la lucrarea de laborator nr 1

# 1.Proiectarea Input-Output:

## 1.1 Modelul Conceptual

## 1.2 Modelul Logic

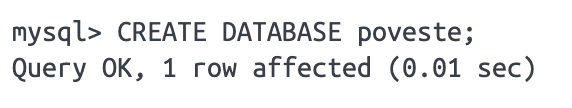


## 1.3 Machetul bazei de date

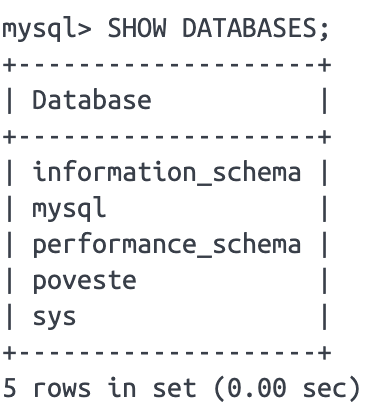
|  |  |  |  |
| --- | --- | --- | --- |
| **Personaje** | **ID\_Personaj** | INT | **PRIMARY KEY NOT NULL** |
| Nume\_Personaj | VARCHAR(20) |  |
| Tip\_Personaj | VARCHAR(10) |  |
| **Imprejurari** | ID\_Imprejurari | INT | PRIMARY KEY NOT NULL |
| Nume\_Imprejurari | VARCHAR(30) |  |
| **ID\_Anotimp** | INT | **FOREIGN KEY** |
| **Anotimp** | **ID\_Anotimp** | INT | **PRIMARY KEY NOT NULL** |
| Nume\_Anotimp | VARCHAR(30) |  |
| **ID\_Haine** | INT | **FOREIGN KEY** |
| **Haine** | **ID\_Haine** | INT | **PRIMARY KEY NOT NULL** |
| Nume\_Haine | VARCHAR(30) |  |
| **ID\_Personaj** | INT | **FOREIGN KEY** |
| **Caracteristici** | ID\_Caracteristici | INT | PRIMARY KEY NOT NULL |
| Nume\_Caracteristici | VARCHAR(30) |  |
| **ID\_Personaj** | INT | **FOREIGN KEY** |
| **Transport** | ID\_Transport | INT | PRIMARY KEY NOT NULL |
| Nume\_Transport | VARCHAR(30) |  |
| Tip\_Transport | VARCHAR(30) |  |
| **ID\_Personaj** | INT | **FOREIGN KEY** |

# 2.Proiectarea Input-Output:

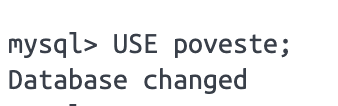
* Crearea bazei de date



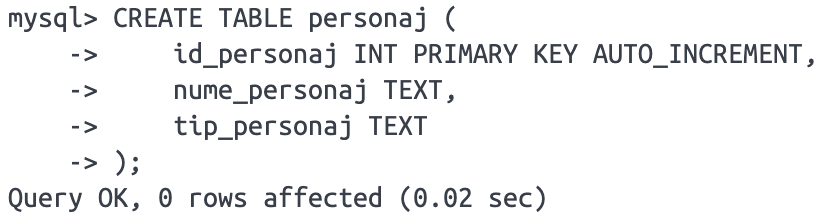
* Afisarea bazelor de date



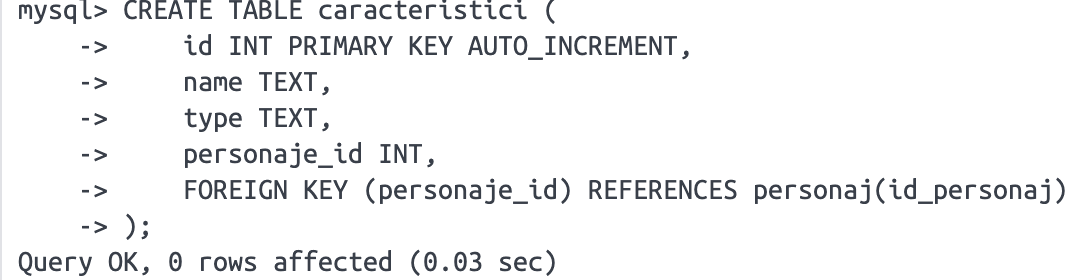
* Selectarea bazei de date



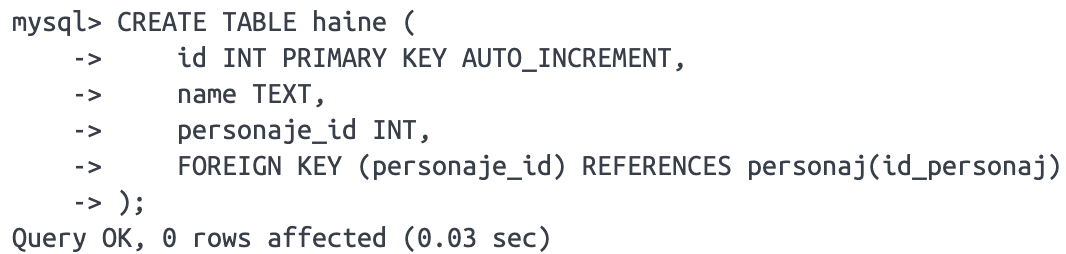
* Crearea tabelului personaj



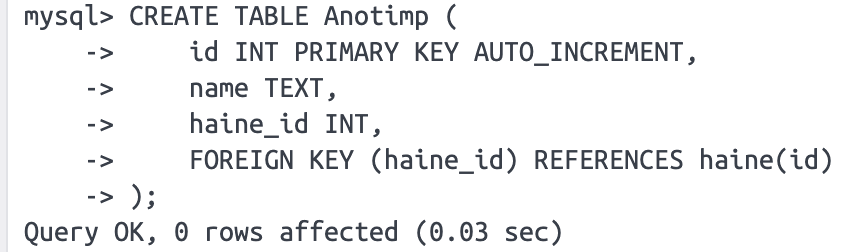
* Crearea tabelului caracterstici



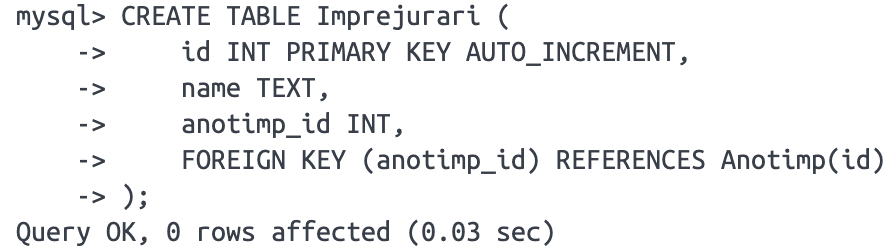
* Crearea tabelului haine



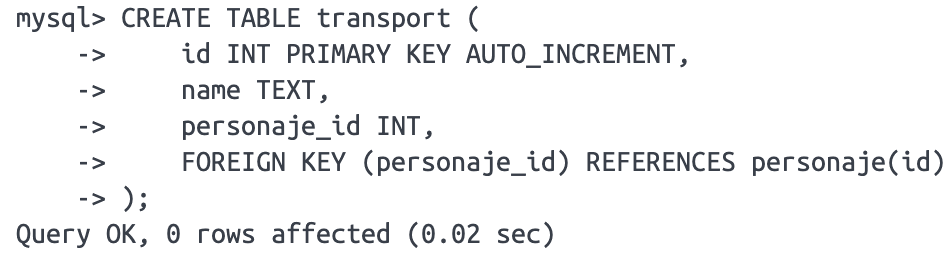
* Crearea tabelului anotimp



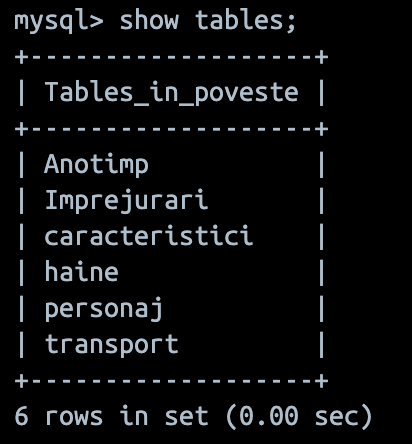
* Crearea tabelului imprejurari



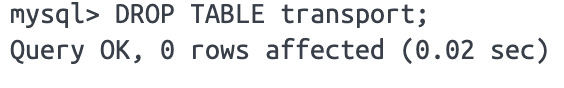
* Crearea tabelului transport



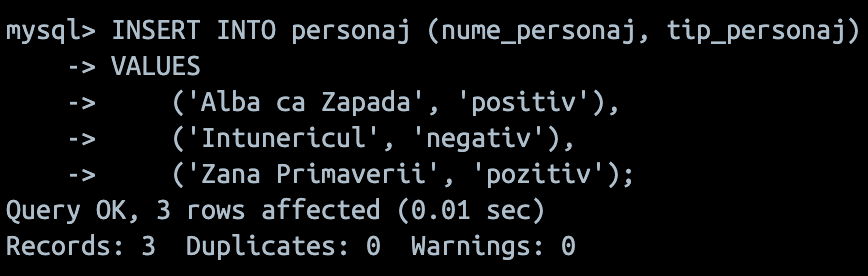
* Afisarea tabelelor;



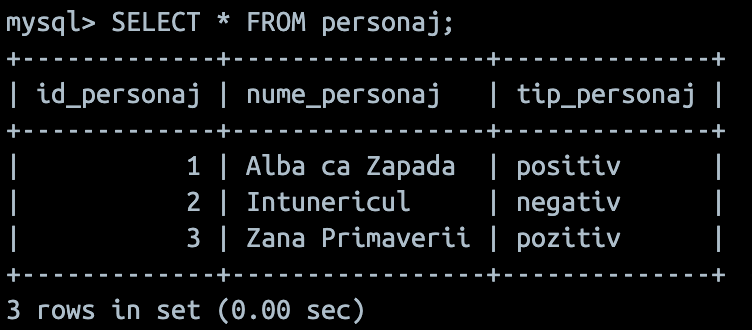
* Stergerea unui tabel



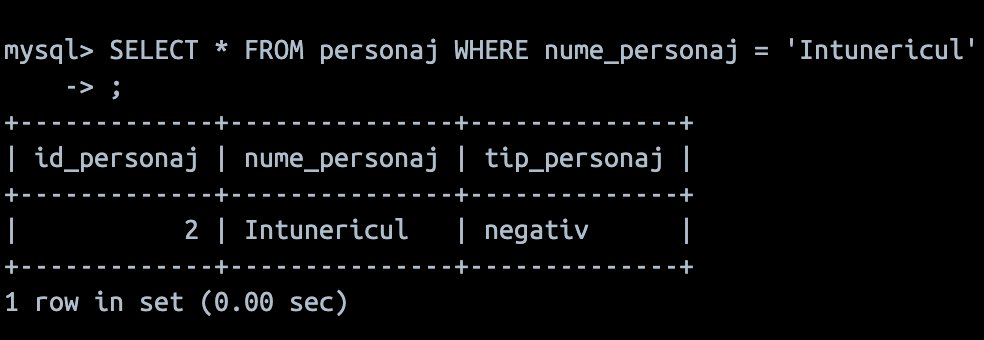
* Adaugarea valorilor in tabel

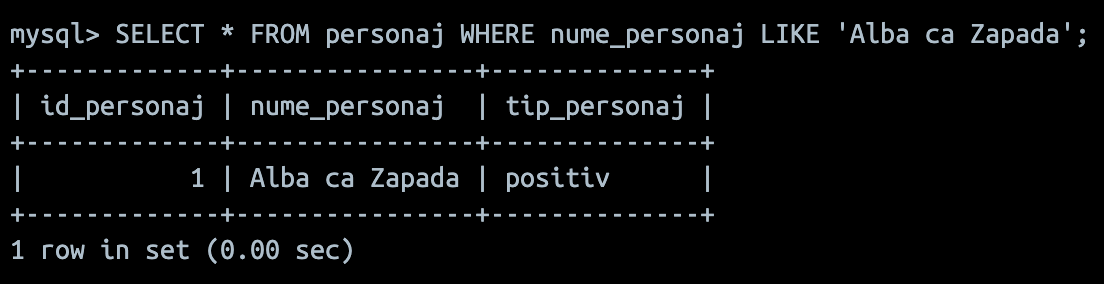


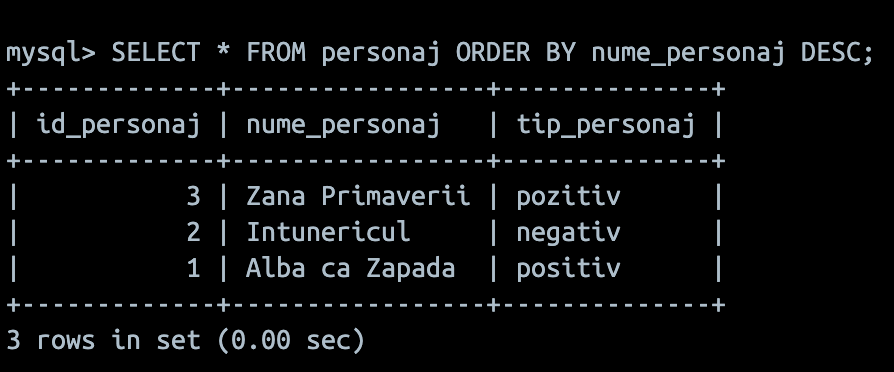
* Afisarea datelor din tabelul personaje



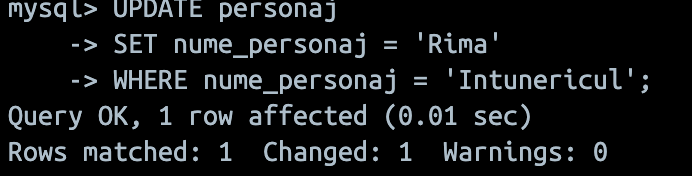
* Extragerea datelor



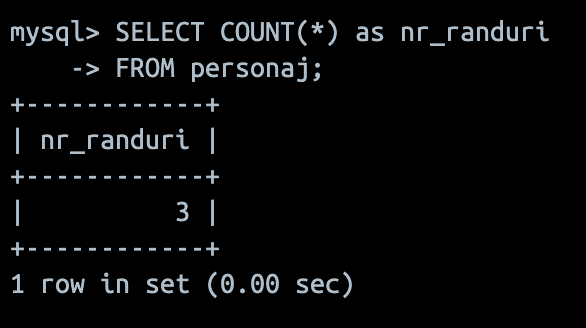




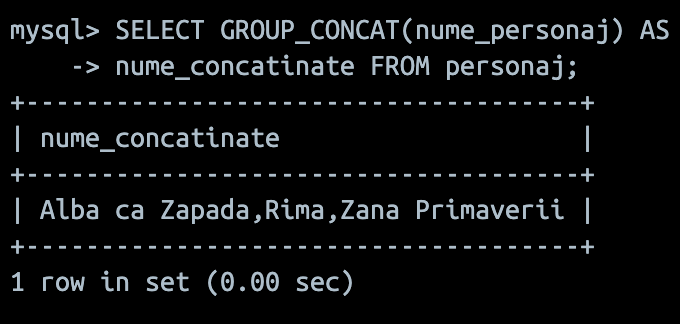
* Actualizarea unui camp din tabel



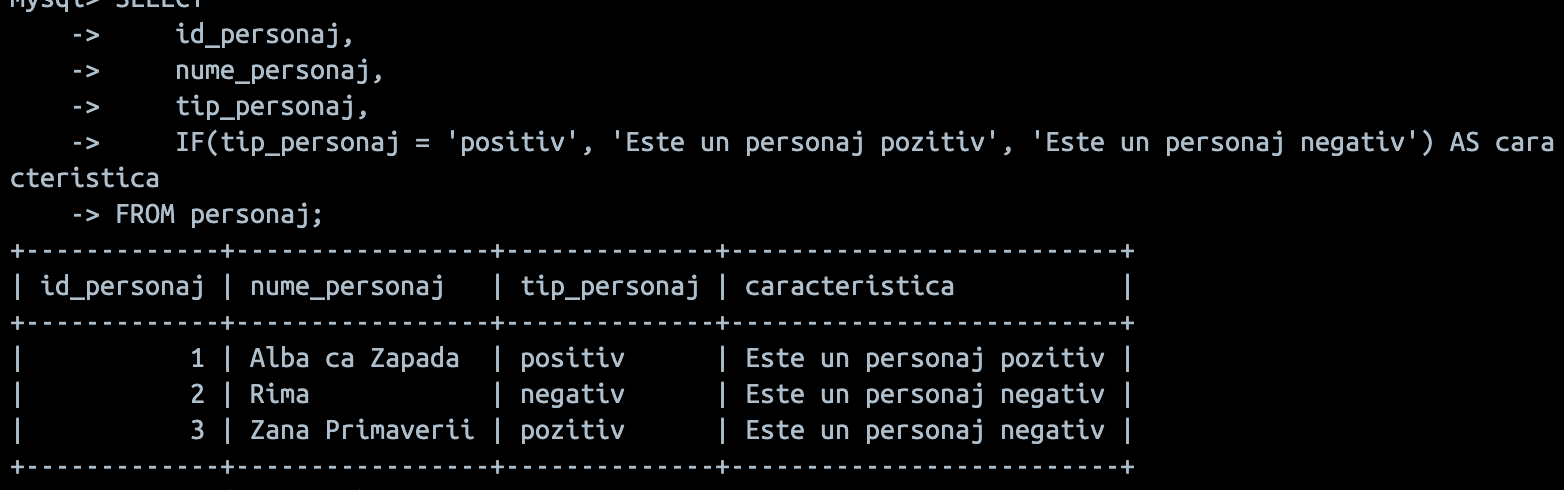
* Calcularea nr de campuri



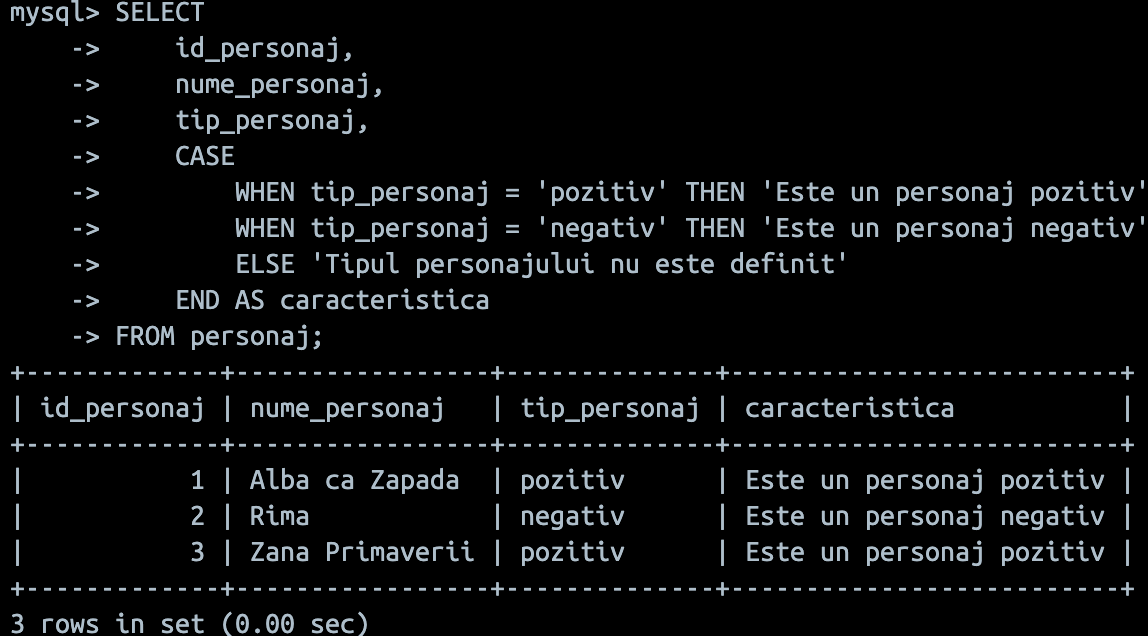
* Concatenarea



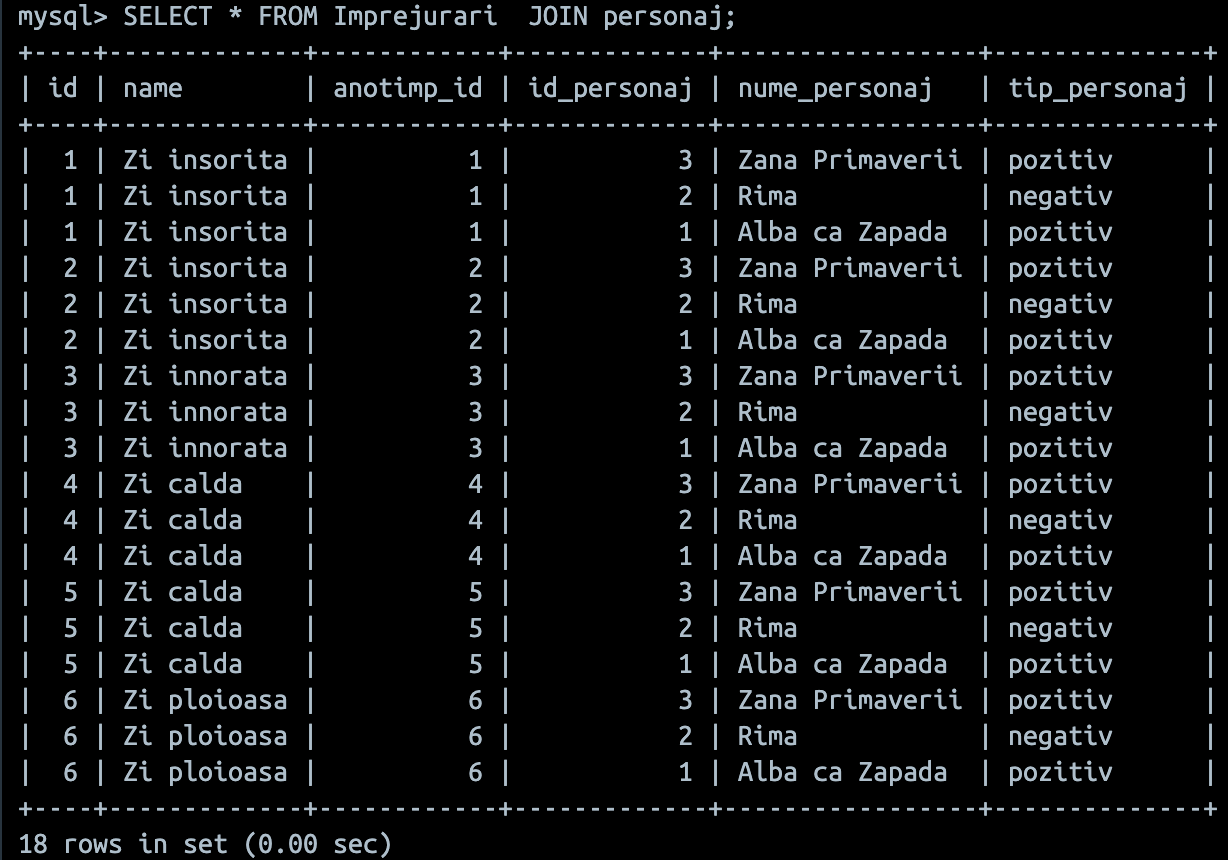
* IF cu true



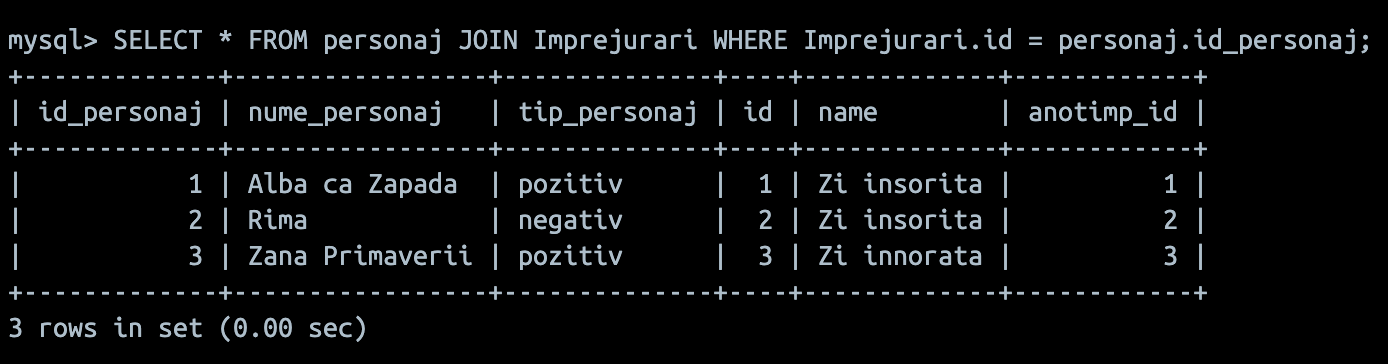
* CASE



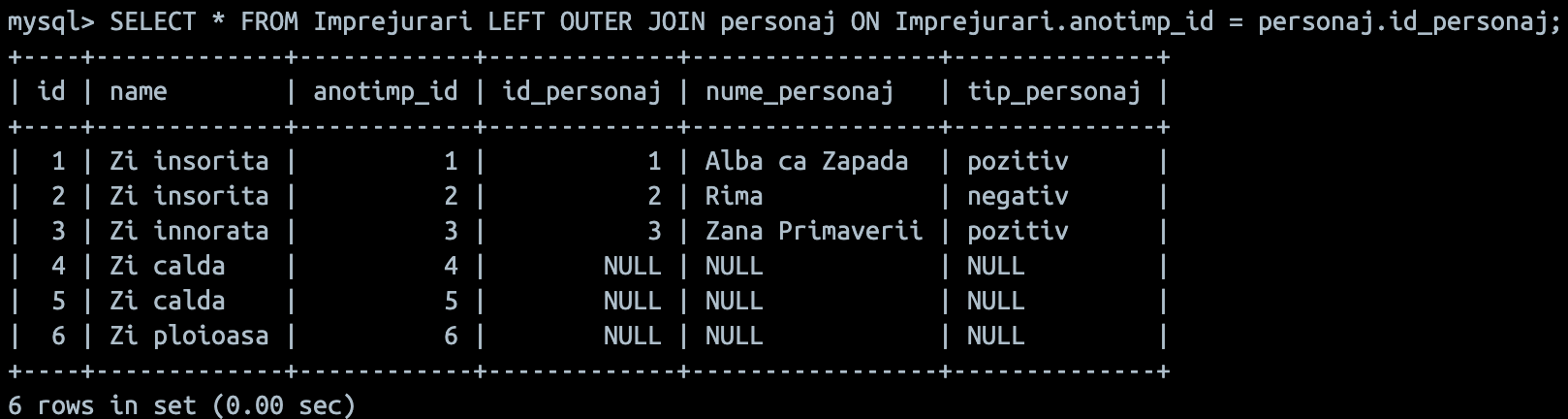
* JOIN



* Inner join



* Left outer join cu NULL



* Natural join

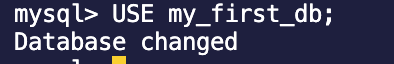


## **2.2 Elaborarea pasilor din SQL for Beginners:**

mysql> CREATE DATABASE my\_first\_db;



mysql> USE my\_first\_db;

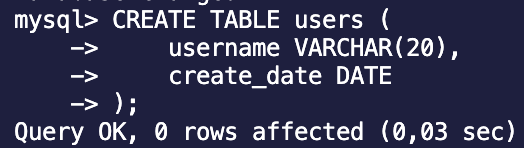


CREATE TABLE users (

    username VARCHAR(20),

    create\_date DATE

);



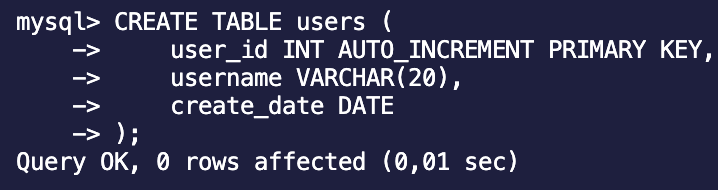
mysql> CREATE TABLE users (

-> user\_id INT AUTO\_INCREMENT PRIMARY KEY,

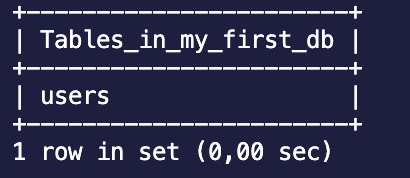
-> username VARCHAR(20),

-> create\_date DATE

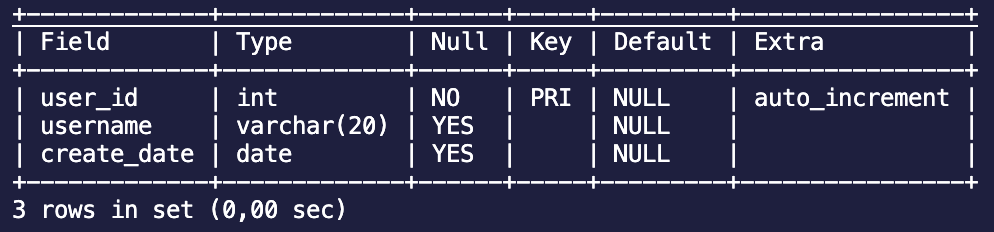
-> );



mysql> Show tables;



mysql> Explain users;

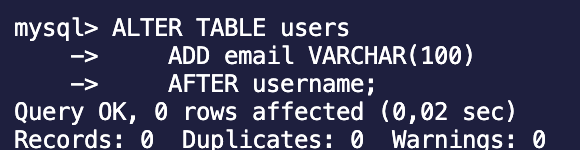


mysql> ALTER TABLE users

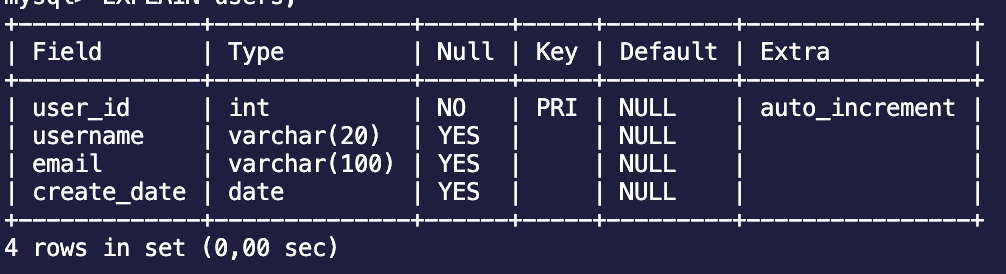
-> ADD email VARCHAR(100)

-> AFTER username;

Query OK, 0 rows affected (0,02 sec)



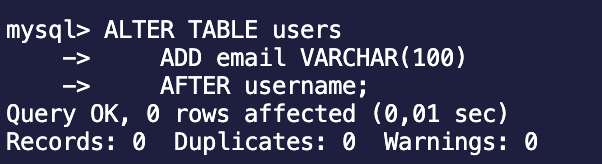
mysql> EXPLAIN users;



mysql> ALTER TABLE users DROP email;



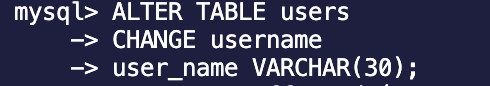
mysql> ALTER TABLE users



mysql> ALTER TABLE users

-> CHANGE username

-> user\_name VARCHAR(30);



INSERT INTO users VALUES (

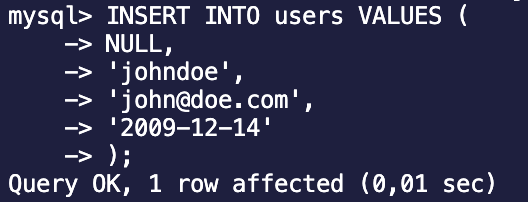
NULL,

'johndoe',

'john@doe.com',

'2009-12-14'

);

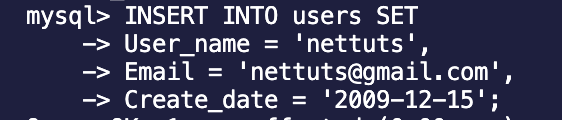


mysql> INSERT INTO users SET

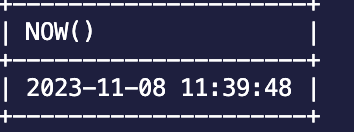
-> User\_name = 'nettuts',

-> Email = 'nettuts@gmail.com',

-> Create\_date = '2009-12-15';



mysql> SELECT NOW();



mysql> SELECT \* from users;



mysql> SELECT \* FROM users

-> ORDER BY create\_date;

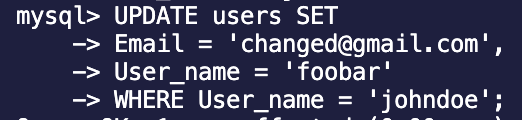


mysql> UPDATE users SET

-> Email = 'changed@gmail.com',

-> User\_name = 'foobar'

-> WHERE User\_name = 'johndoe';



mysql> INSERT INTO users SET user\_name = 'O''Reilly';



mysql> CREATE TABLE states (

-> id INT AUTO\_INCREMENT,

-> name VARCHAR(20),

-> join\_year INT,

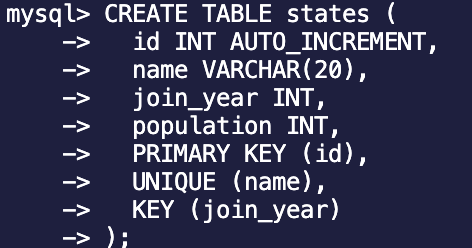
-> population INT,

-> PRIMARY KEY (id),

-> UNIQUE (name),

-> KEY (join\_year)

-> );



mysql> 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),

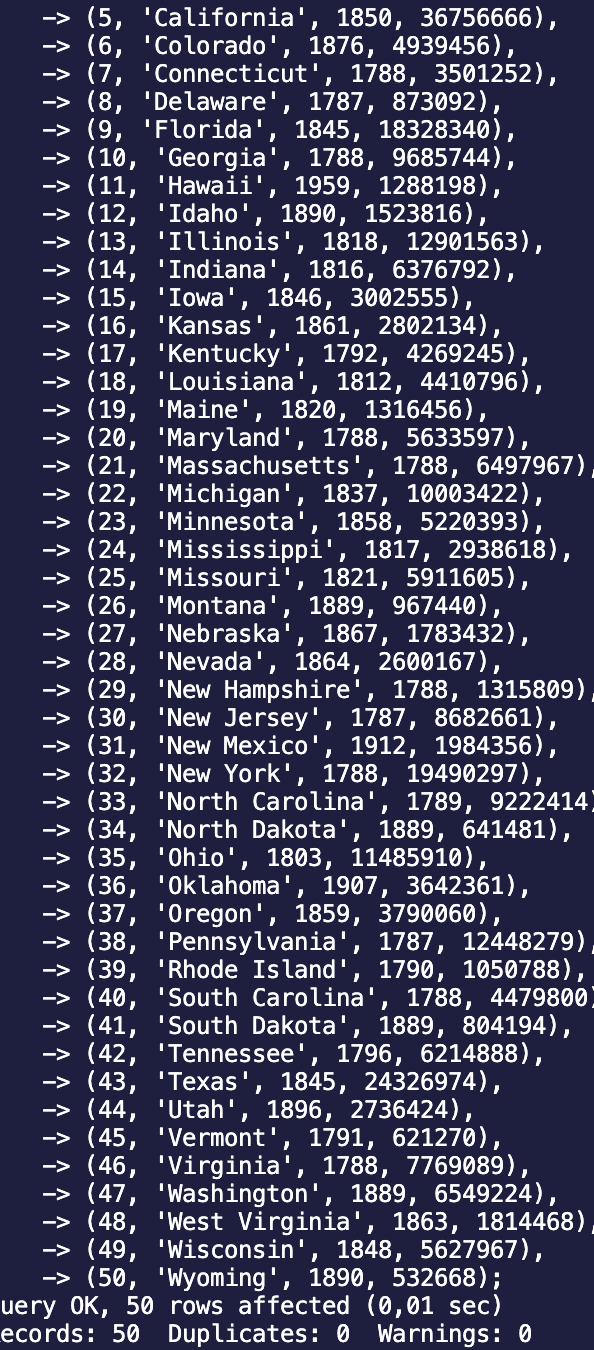
-> (46, 'Virginia', 1788, 7769089),

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

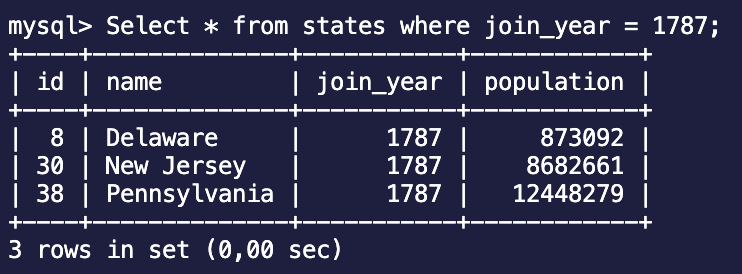
-> (48, 'West Virginia', 1863, 1814468),

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

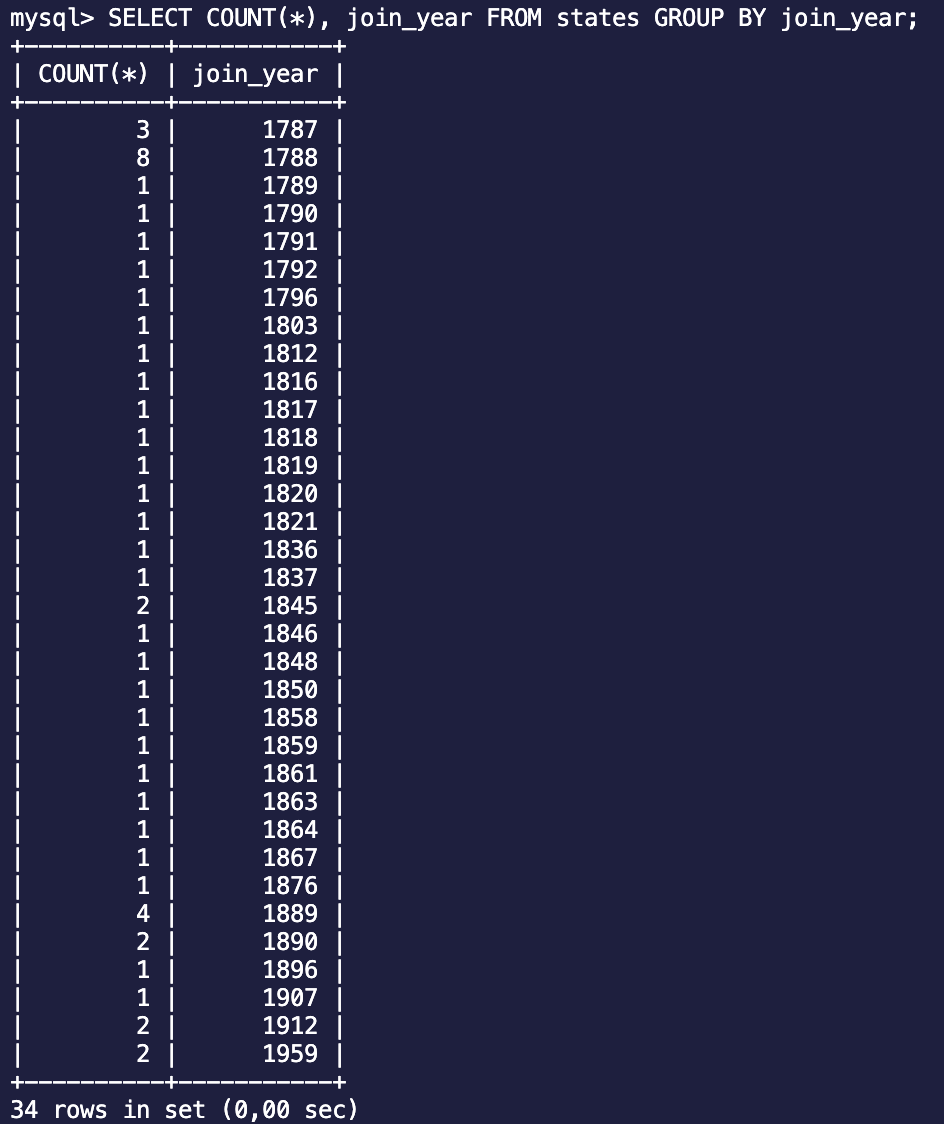
-> (50, 'Wyoming', 1890, 532668);



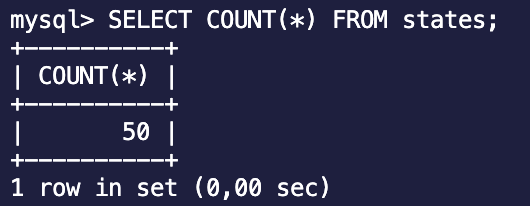
mysql> Select \* from states where join\_year = 1787;



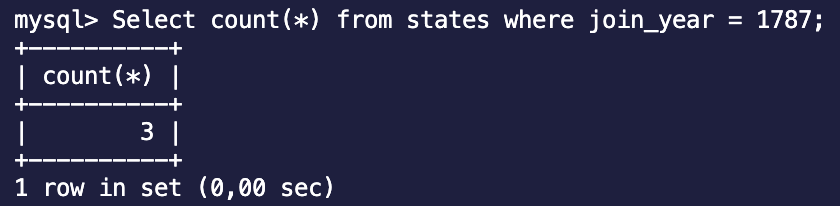
mysql> SELECT COUNT(\*), join\_year FROM states GROUP BY join\_year;



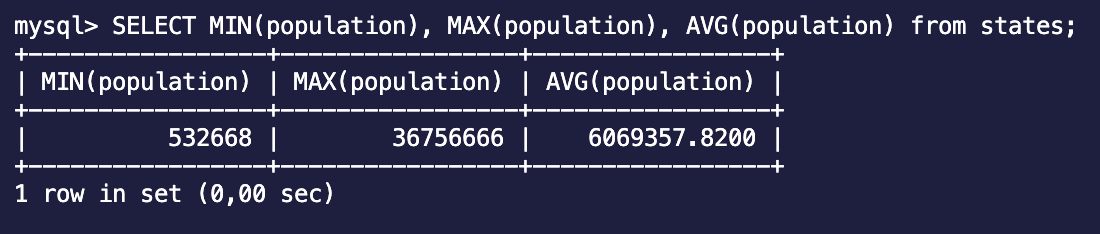
mysql> SELECT COUNT(\*) FROM states;



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

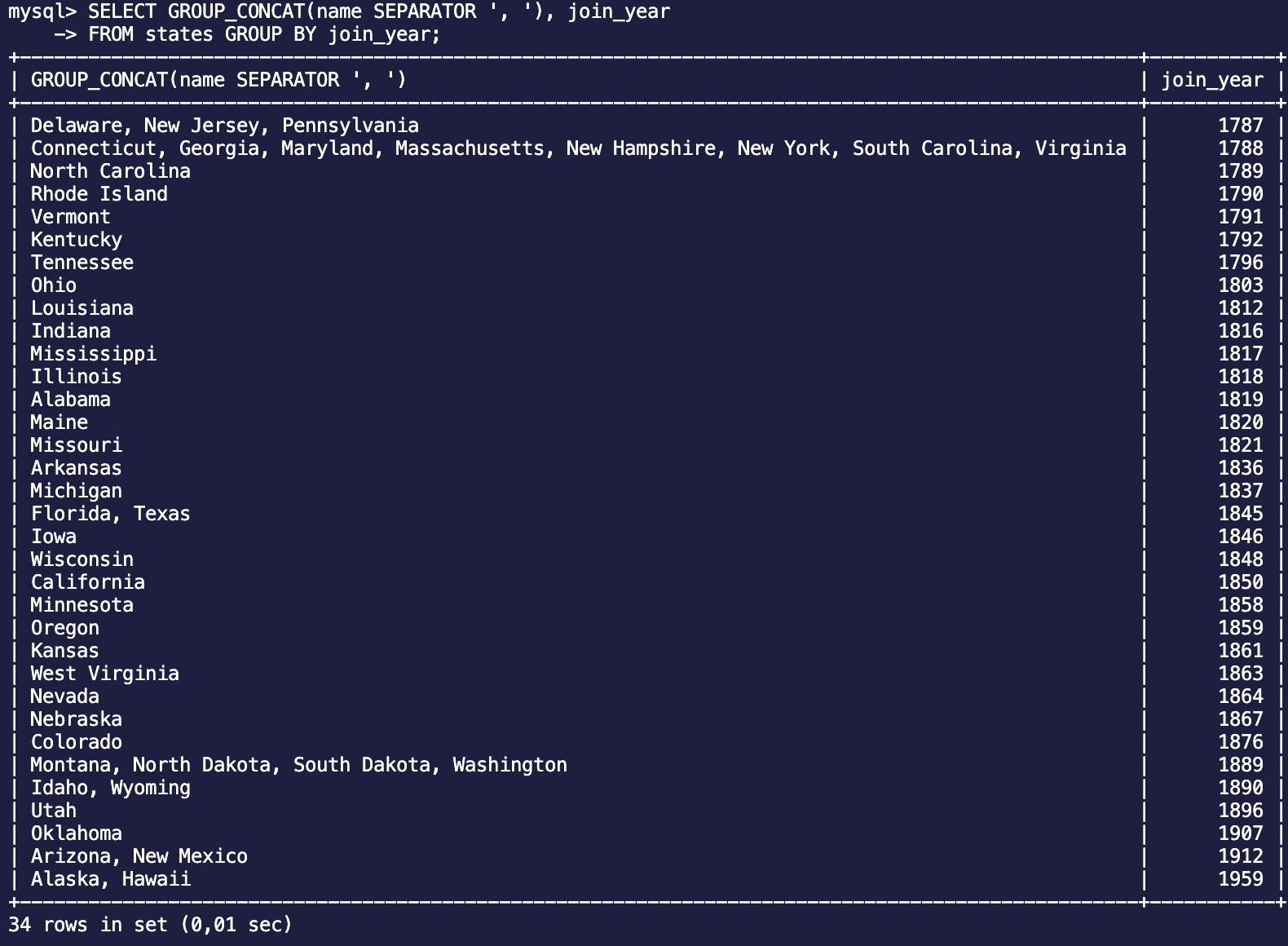


mysql> SELECT MIN(population), MAX(population), AVG(population) from states;

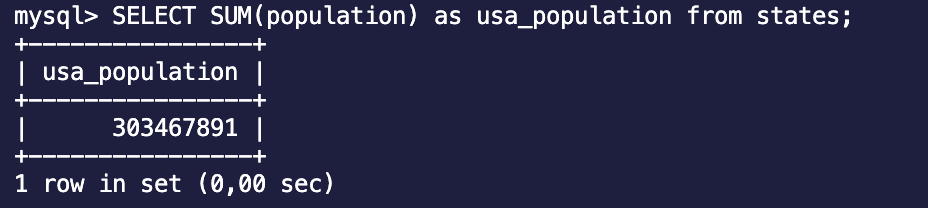


mysql> SELECT GROUP\_CONCAT(name SEPARATOR ', '), join\_year

-> FROM states GROUP BY join\_year;



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



mysql> SELECT

->

-> SUM(

-> IF(population > 5000000, 1, 0)

-> ) AS big\_states,

->

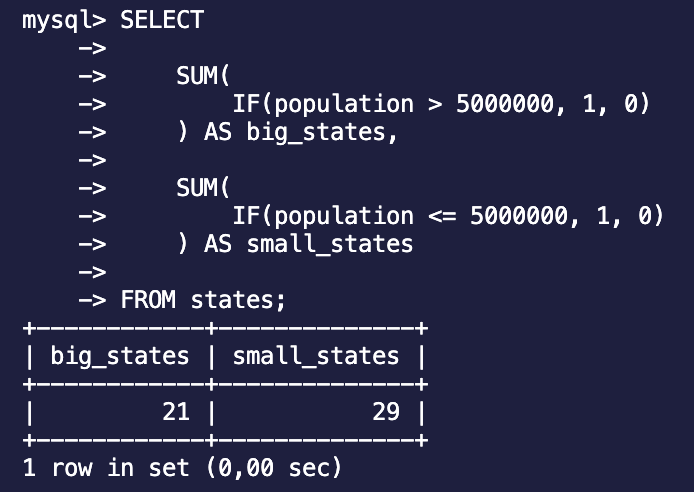
-> SUM(

-> IF(population <= 5000000, 1, 0)

-> ) AS small\_states

->

-> FROM states;



mysql> SELECT

-> COUNT(\*),

-> CASE

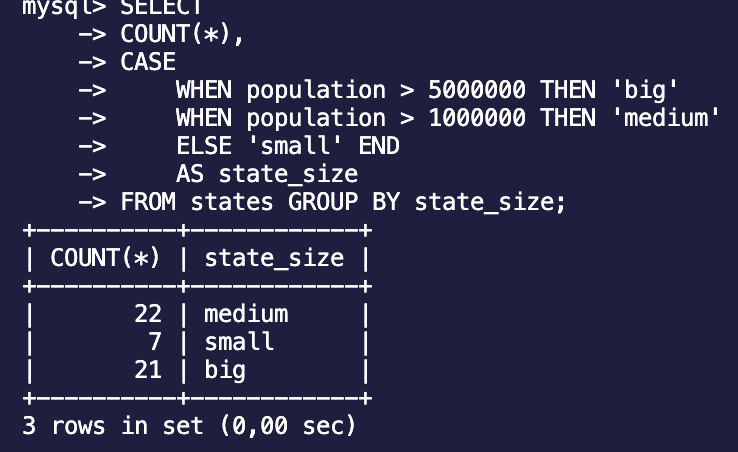
-> WHEN population > 5000000 THEN 'big'

-> WHEN population > 1000000 THEN 'medium'

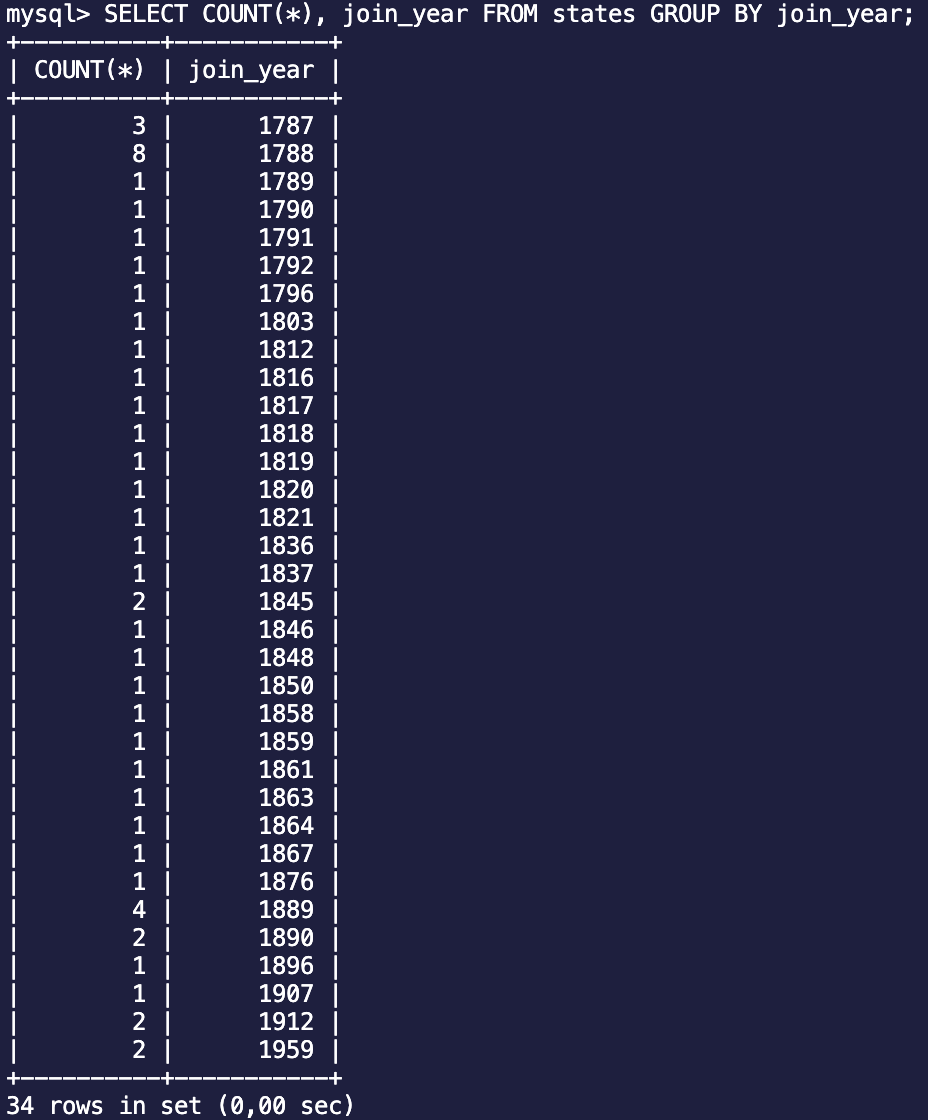
-> ELSE 'small' END

-> AS state\_size

-> FROM states GROUP BY state\_size;



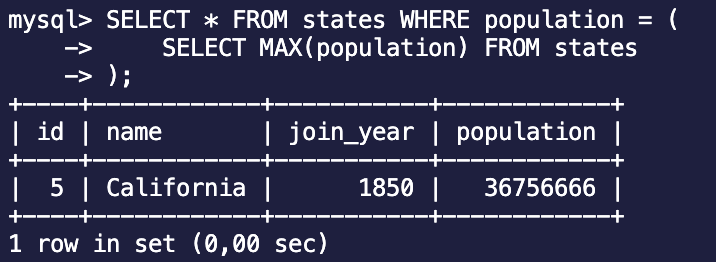
mysql> SELECT COUNT(\*), join\_year FROM states GROUP BY join\_year;



mysql> SELECT \* FROM states WHERE population = (

-> SELECT MAX(population) FROM states

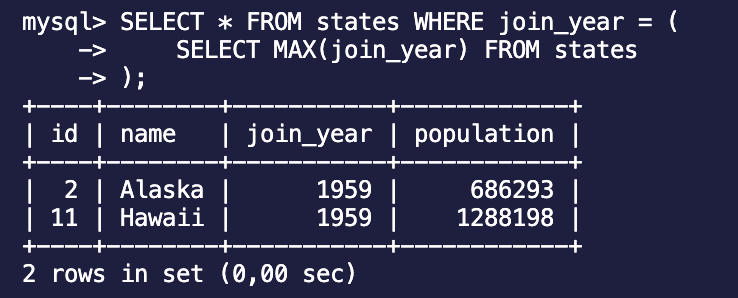
-> );



mysql> SELECT \* FROM states WHERE join\_year = (

-> SELECT MAX(join\_year) FROM states

-> );



mysql> SELECT \* FROM states WHERE join\_year IN (

-> SELECT join\_year FROM states

-> GROUP BY join\_year

-> HAVING COUNT(\*) > 1

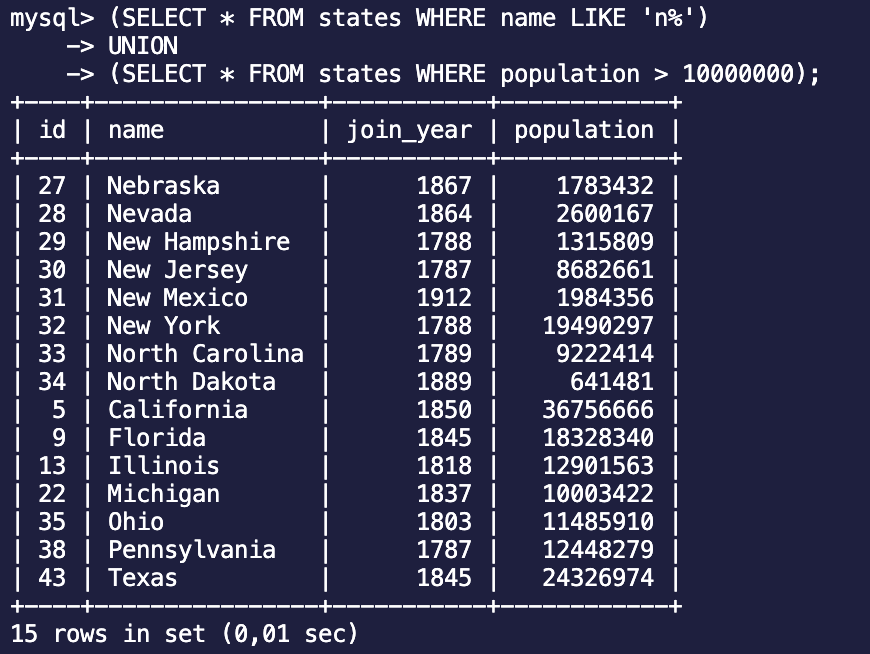
-> ) ORDER BY join\_year;



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

-> UNION

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

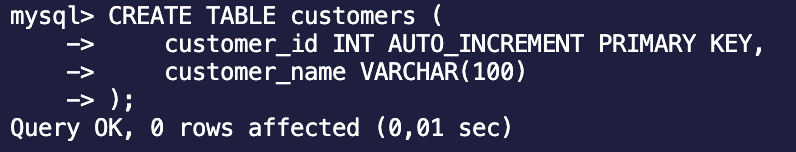


mysql> CREATE TABLE customers (

-> customer\_id INT AUTO\_INCREMENT PRIMARY KEY,

-> customer\_name VARCHAR(100)

-> );



mysql> CREATE TABLE orders (

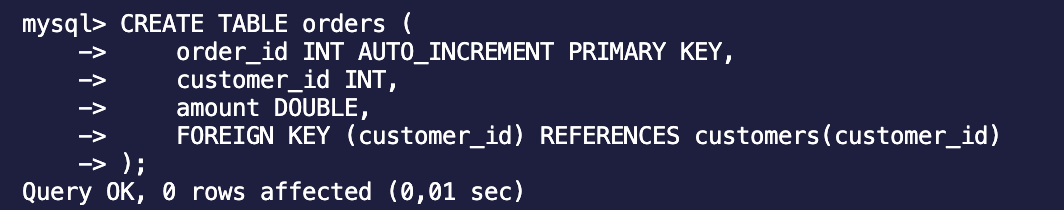
-> order\_id INT AUTO\_INCREMENT PRIMARY KEY,

-> customer\_id INT,

-> amount DOUBLE,

-> FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id)

-> );



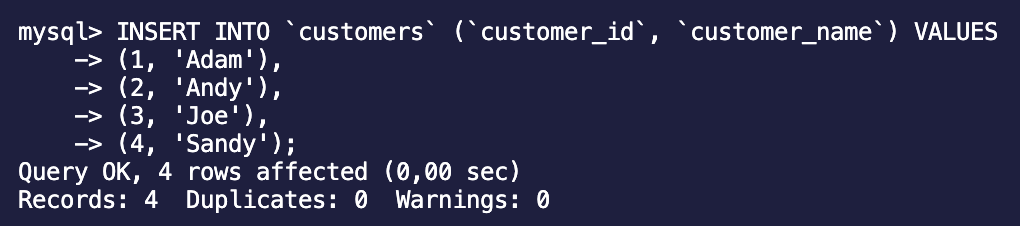
mysql> INSERT INTO `customers` (`customer\_id`, `customer\_name`) VALUES

-> (1, 'Adam'),

-> (2, 'Andy'),

-> (3, 'Joe'),

-> (4, 'Sandy');



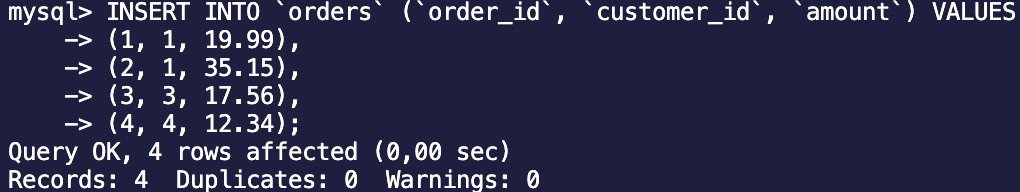
mysql> INSERT INTO `orders` (`order\_id`, `customer\_id`, `amount`) VALUES

-> (1, 1, 19.99),

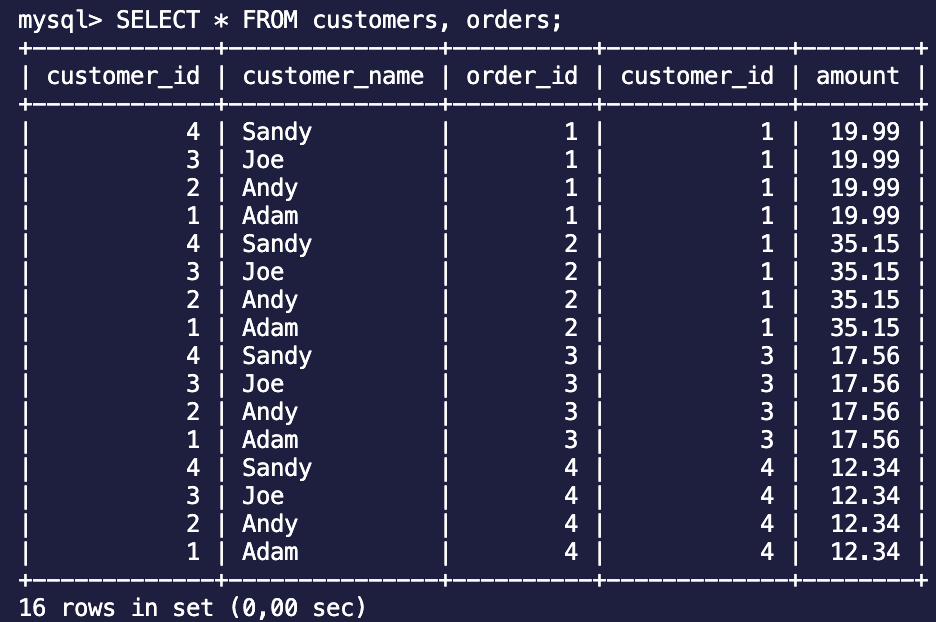
-> (2, 1, 35.15),

-> (3, 3, 17.56),

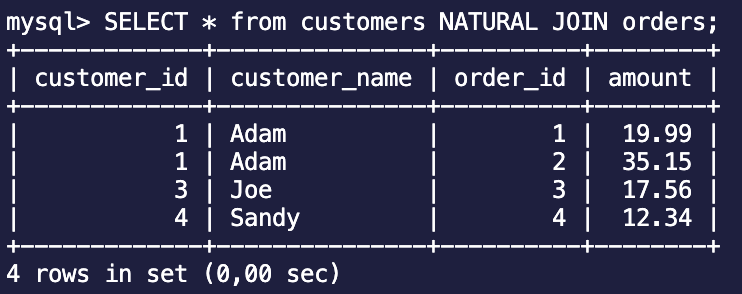
-> (4, 4, 12.34);



mysql> SELECT \* FROM customers, orders;

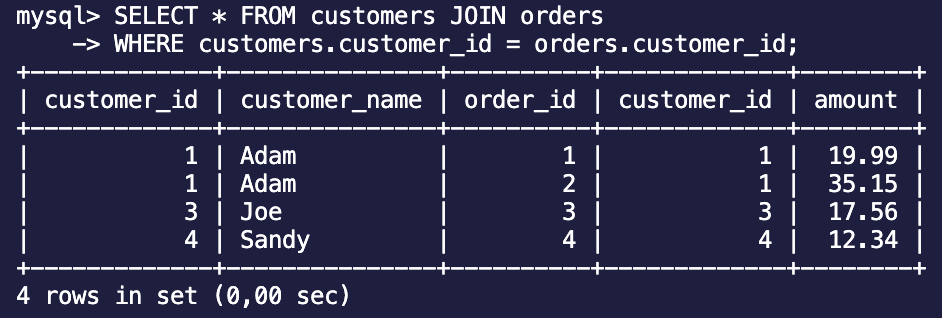


mysql> SELECT \* from customers NATURAL JOIN orders;



mysql> SELECT \* FROM customers JOIN orders

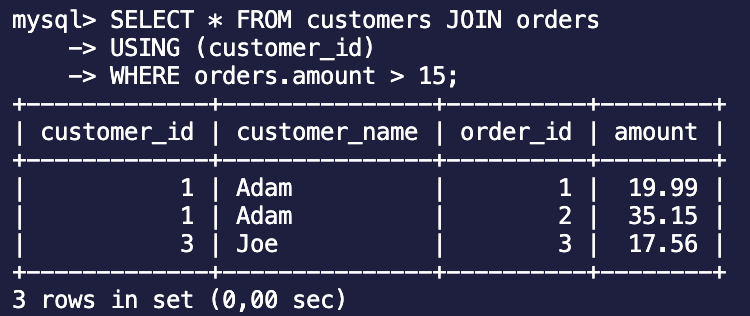
-> WHERE customers.customer\_id = orders.customer\_id;



mysql> SELECT \* FROM customers JOIN orders

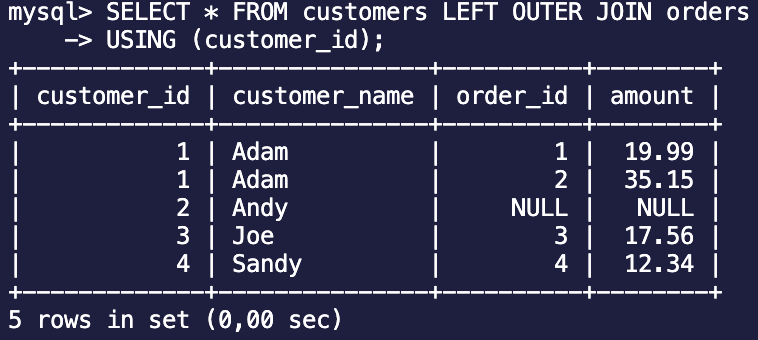
-> USING (customer\_id)

-> WHERE orders.amount > 15;



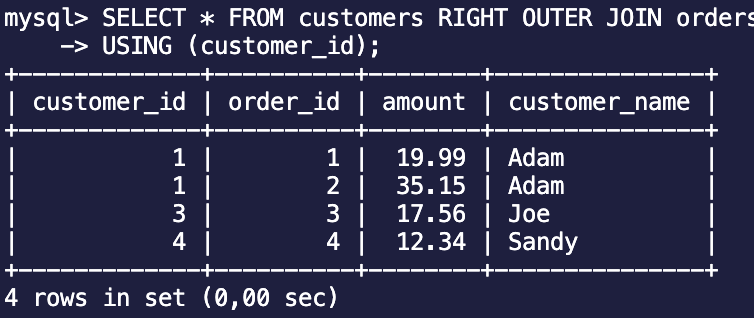
mysql> SELECT \* FROM customers LEFT OUTER JOIN orders

-> USING (customer\_id);



mysql> SELECT \* FROM customers RIGHT OUTER JOIN orders

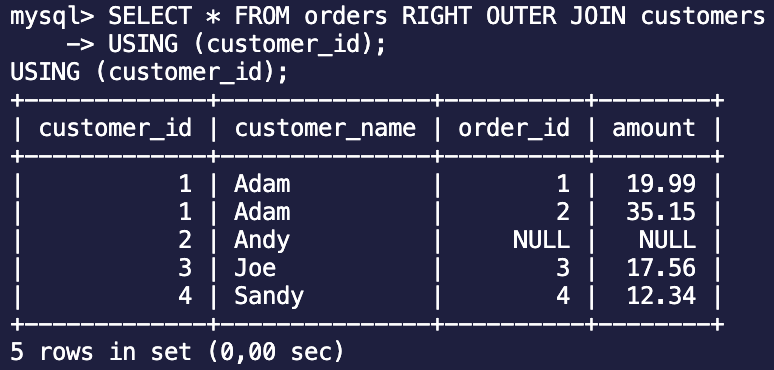
-> USING (customer\_id);



mysql> SELECT \* FROM orders RIGHT OUTER JOIN customers

-> USING (customer\_id);

USING (customer\_id);



# 3. Povestea modificata

– Dormi, Floarea, vorbim altă dată.

        – Zana Primaverii! Insistă Floarea. Nu vreau să dorm! Vreau să vorbim!

        – Bine, Floarea,! îşi deschide un ochi Zana Primaverii,. Despre ce să vorbim?

        – Despre cât de frumos e Padurea Fermecata…

        – Vai, Floarea,, ţi-am povestit de atâtea ori.

        – Mai vreau, Floarea,!.

        – Atunci ascultă… şi Zana Primaverii, îşi începe povestea. Acolo Padurea Fermecata, şopteşte Zana Primaverii,, adică deasupra Padurea Fermecata – e o altă lume. Nu zic, şi la noi e bine şi e frumos, căci aici ne este casa, dar acolo, cum sa-ţi spun, e altfel… În primul rând e multă lumină. Şi sunt multe, multe culori. Totul e colorat. Da… În sfârşit am înţeles… Culorile… Iată de ce sus e atât de frumos.

        – Zana Primaverii,, când ieşim Padurea Fermecata?

        – Vai Floarea,. De câte ori să-ţi zic? Ai niţică răbdare. Acum e Vara, Ne-am ascuns în căsuţa noastră ca sa nu îngheţăm. Multe gâze şi vietăţi mici când e cald mişună pe afară, când e frig se imbraca in Basma . Chiar şi culorile pare-se dorm pe vreme de Vara. Aşa că… Închide ochii, îţi voi cânta un cântecel, iar tu vei face nani.

        – Nu vreau nani! Nu-mi pasă de frig! Nu-mi place aici!

Oricât de Linistita ar fi Zana Primaverii,, până la urmă îşi iese din fire:

– Încetează! strigă Zana Primaverii,. Vom merge sus cu Avion doar când ni se va da de ştire că e posibil! Gata! Nici o vorbă mai mult! Dormi!

– Zana Primaverii,, începe iarăşi Floarea,. Spune-mi… Cine ne va da de ştire?

– Înţelegi Floarea,, deasupra pământului e cerul cu nori. Norii sunt diferiţi… Adică de diferite culori, de diferite mărimi. Când e primăvară şi cerul se acoperă cu nori negri, tuciurii – norul cel mai important îşi ia toba şi începe să bată: Bah! Bah! Bah! Bah! Toba lui se aude până la noi. Deci vorbim după ce auzim bubuitul. Acum dormi!

– Zana Primaverii,! Auzi!? sare Floarea, din pătucul său. Norul cel mai important bate din tobă!

– Da? nu-i vine să creadă Zana Primaverii,. O fi venit Vara?

– A venit, Zana Primaverii,! A venit! Hai sa mergem afară! Repede! Vreau să văd culorile!

– Nu, Floarea,. Încă nu…

– De ce?! M-ai minţit!? Atunci… plec singur, fără tine!

– Stai, Floarea,! Aşteaptă! Doar un pic. Mai întâi de toate ţărâna trebuie să devină reavănă, puhavă, ca să putem săpa tunelul prin care ne vom strecura. Nu durează mult. Acuş picăturile calde de ploaie se vor scurge în sol… Ah! E atât de plăcut când faci prima scăldătoare de primăvară… Atât de plăcut… Vezi? Solul începe să devină umed. Deci… putem să pornim săpatul.

Cele două râme stau într-o grădină plină de flori, plină de culori. Micuţul Floarea, nu încetează să se minuneze de frumuseţea din jur.

– Zana Primaverii,, zice el, nu mă mai întorc acasă. Îmi place atât de mult aici. Mai ales îmi plac florile… sunt foarte frumoase!

– Eh, oftează Zana Primaverii,. Frumos, frumos… Dar locul unei floare e în pământ. Dacă o floare stă mult timp sub razele soarelui, se usucă. Şi nici măcar nu acest lucru e cel mai important. Prea multe primejdii ne înconjoară… Intunericul, iată răul cel mare. Intunericul,!

– Zana Primaverii,.! Priveşte! Ce fiinţă ciudată aleargă spre noi?! E nespus de nostimă! Iar în urma ei – o mulţime de ghemuleţe aurii! Mă duc să-i salut! strigă Floarea,., apoi porneşte cu îndrăzneală în întâmpinarea cloştii.

– Vai, Floarea, ce faci!? e disperată Zana Primaverii, reuşind totuşi să-şi împingă odrasla înapoi în tunel. Am scăpat, răsuflă ea uşurată. Să nu te fi oprit la timp… Dar… să nu mai vorbim. Principalul e că ai văzut… culorile. Cred că altădată nu mai riscăm. Era cât pe ce să ne înghită o Zana Primaverii,.! …

E mult de lucru în împărăţie. Toată ziulica acestea fărâmă solul ca să-l facă pufos. Floarea şi Zana Primaverii sunt convinşi că şi datorită lor din solul pufos răsar în fiecare primăvară… culorile.

1. **Personaje:** „Zana Primaverii”, „Floarea”, „Lumina”, „Intunericul”, „Fluturasul”, „Lacusta”;
2. **Imprejurari:** „Padurea Fermecata”, „Muntii Inaltimilor”, „Raul Mandrei”;
3. **Anotimp:** „Vara”, „Iarna”, „Primavara”;
4. **Haine:** „Camasa”, „Pantaloni”, „Vesta”, „Caciula”, „Papuci”, „Costum”,;
5. **Caracteristici** „Tanara”, „Vesela”, „Neobisnuita”, „Stralucitor”, „Misterios”, „Optimism”;
6. **Transport:** „Avion”, „Corabie”, „Taxi”;

Concluzie

După finalizarea acestui laborator, am dobândit cunoștințe despre comenzile generale și caracteristicile extinse ale limbajului de manipulare a bazelor de date MySQL. Astfel, am ajuns la o înțelegere mai profundă a faptului că MySQL reprezintă un sistem de gestionare a bazelor de date relaționale, dezvoltat de compania suedeză MySQL AB și distribuit sub Licența Publică Generală GNU. Acesta rămâne una dintre cele mai populare și deschise opțiuni de gestionare a bazelor de date la ora actuală, jucând un rol esențial în stiva LAMP (Linux, Apache, MySQL, PHP).

MySQL se integrează strâns în platformele LAMP sau WAMP (Linux/Windows-Apache-MySQL-PHP/Perl/Python) și se bucură de o popularitate continuă, în special în aplicațiile web care utilizează PHP, adesea fiind denumit "Duo-ul Dinamic". Mulți experți recunosc că MySQL este mai accesibil în ceea ce privește învățarea și utilizarea sa în comparație cu alte sisteme de gestionare a bazelor de date. De exemplu, comanda de ieșire este simplă și evidentă: „exit” sau „quit”.

Pentru administrarea bazelor de date MySQL, există două modalități principale: utilizarea liniei de comandă sau descărcarea unor interfețe grafice precum MySQL Administrator și MySQL Query Browser. De asemenea, o alternativă gratuită și larg utilizată pentru administrarea bazelor de date MySQL este phpMyAdmin, o aplicație scrisă în PHP.