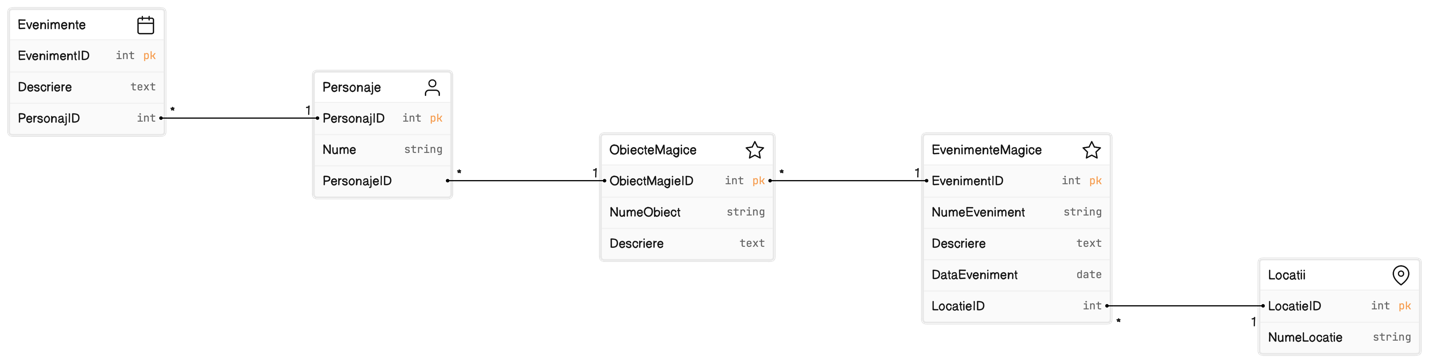
## **Model conceptual**

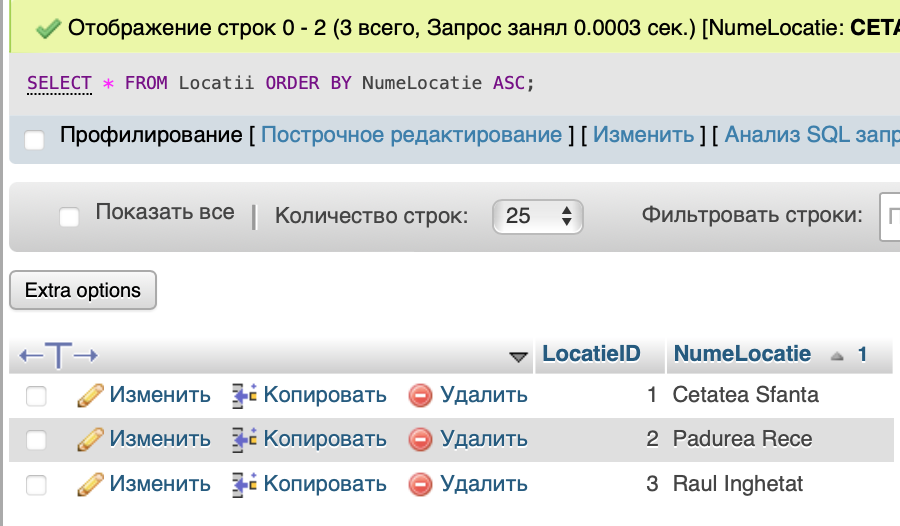
## 

Model Logic



## 

## **Extragere date**

SELECT \* FROM Locatii ORDER BY NumeLocatie ASC;

## SELECT NumeObiect, Descriere FROM ObiecteMagice;

## 

## **Actualizare date**

## 

## **Select IF True**

## SELECT EvenimentID, NumeEveniment, Descriere, DataEveniment, LocatieID, IF(LocatieID = 1, 'Cetatea Sfanta', IF(LocatieID = 2, 'Padurea Rece', IF(LocatieID = 3, 'Raul Inghetat', 'Locatie necunoscuta'))) AS NumeLocatie FROM EvenimenteMagice;

## 

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

CREATE DATABASE my\_first\_db;



USE my\_first\_db;

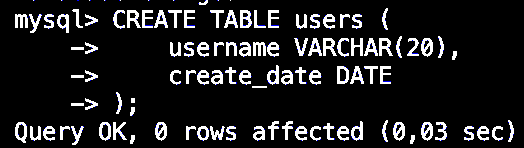


CREATE TABLE users (

    username VARCHAR(20),

    create\_date DATE

);



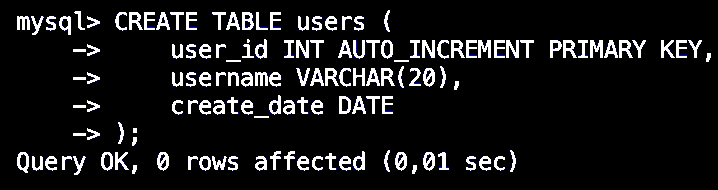
CREATE TABLE users (

user\_id INT AUTO\_INCREMENT PRIMARY KEY,

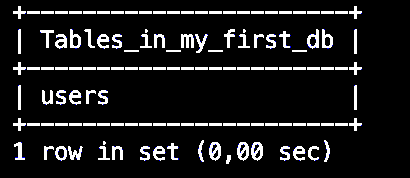
username VARCHAR(20),

create\_date DATE

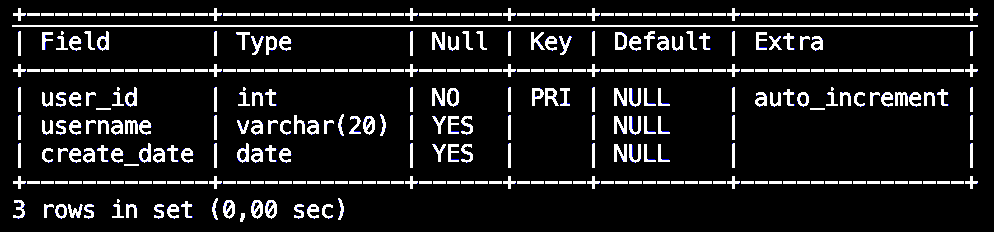
);



Show tables;



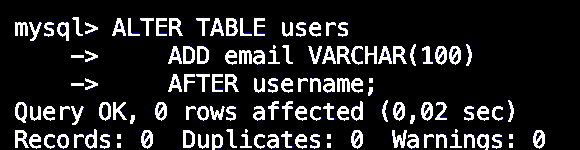
Explain users;



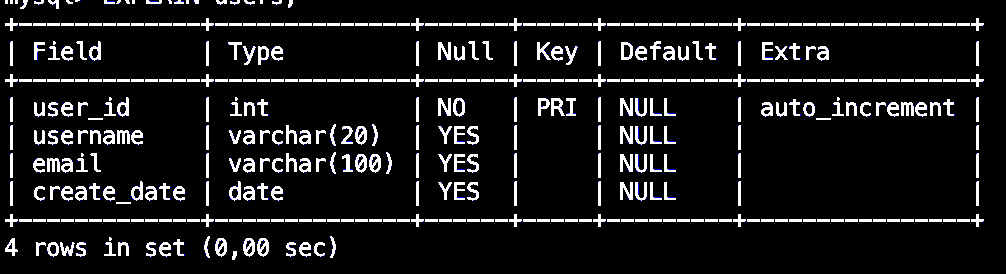
ALTER TABLE users

ADD email VARCHAR(100)

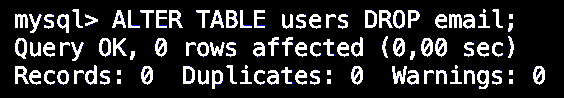
AFTER username;



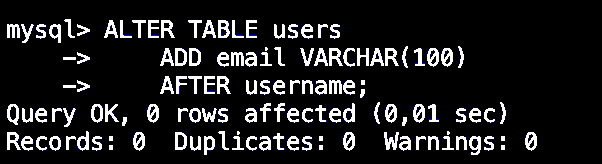
EXPLAIN users;



ALTER TABLE users DROP email;



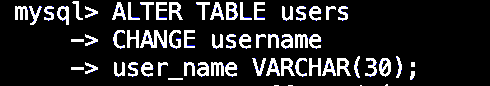
ALTER TABLE users



ALTER TABLE users

CHANGE username

user\_name VARCHAR(30);



INSERT INTO users VALUES (

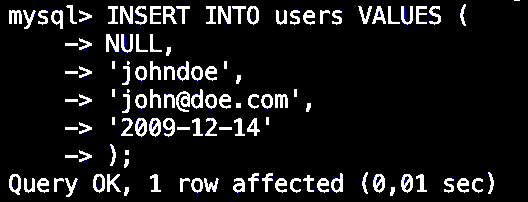
NULL,

'johndoe',

'john@doe.com',

'2009-12-14'

);

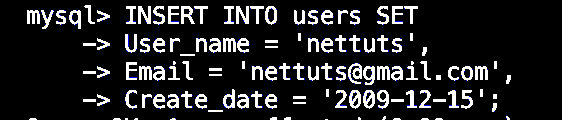


INSERT INTO users SET

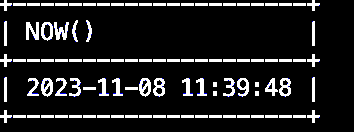
User\_name = 'nettuts',

Email = 'nettuts@gmail.com',

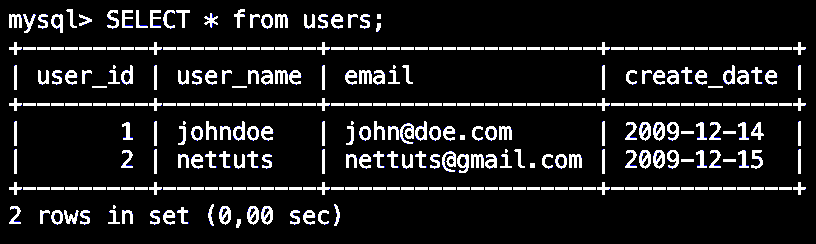
Create\_date = '2009-12-15';



SELECT NOW();



SELECT \* from users;



SELECT \* FROM users

ORDER BY create\_date;

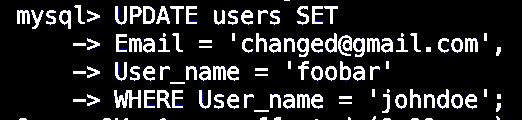


UPDATE users SET

Email = 'changed@gmail.com',

User\_name = 'foobar'

WHERE User\_name = 'johndoe';



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



CREATE TABLE states (

id INT AUTO\_INCREMENT,

name VARCHAR(20),

join\_year INT,

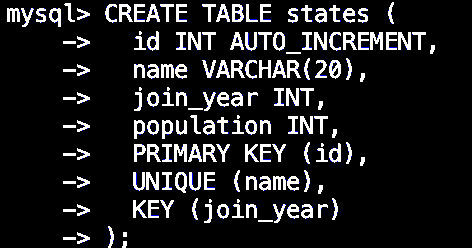
population INT,

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),

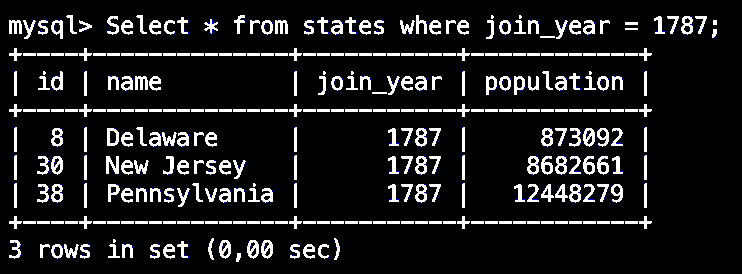
(48, 'West Virginia', 1863, 1814468),

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

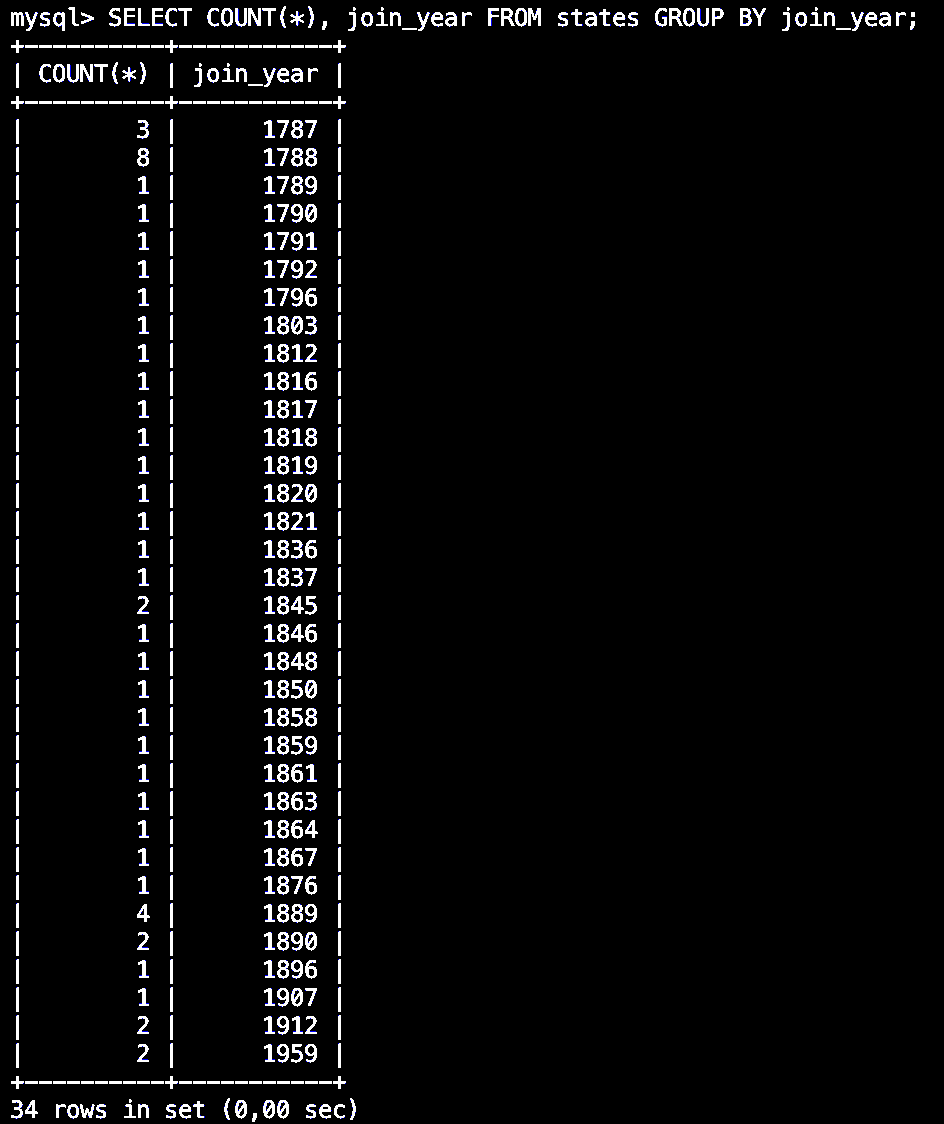
(50, 'Wyoming', 1890, 532668);



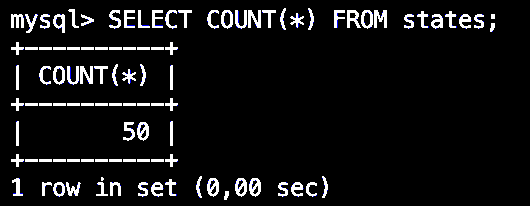
Select \* from states where join\_year = 1787;



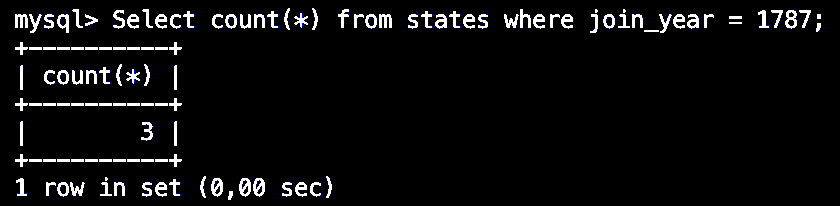
SELECT COUNT(\*), join\_year FROM states GROUP BY join\_year;



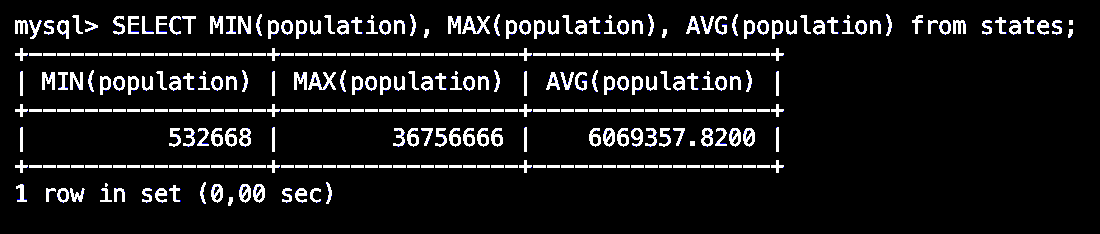
SELECT COUNT(\*) FROM states;



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

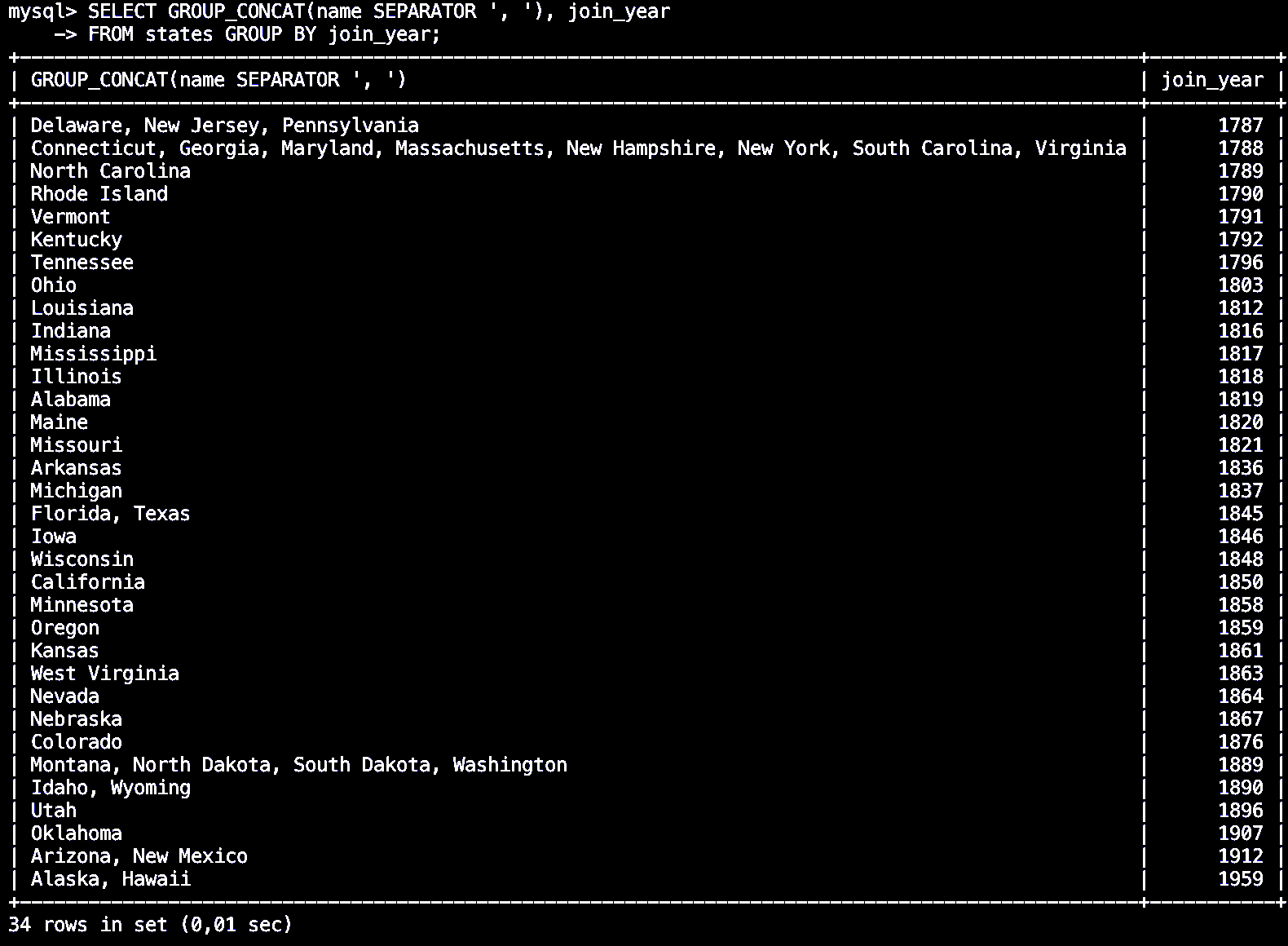


SELECT MIN(population), MAX(population), AVG(population) from states;

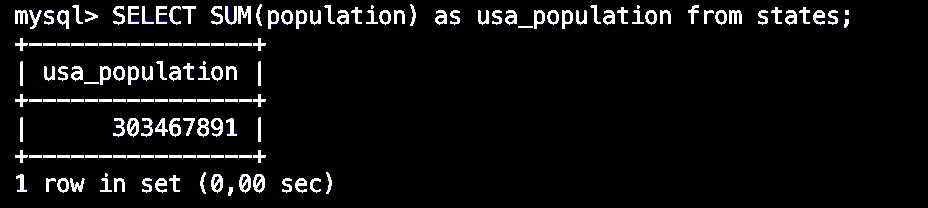


SELECT GROUP\_CONCAT(name SEPARATOR ', '), join\_year

FROM states GROUP BY join\_year;



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



SELECT

SUM(

IF(population > 5000000, 1, 0)

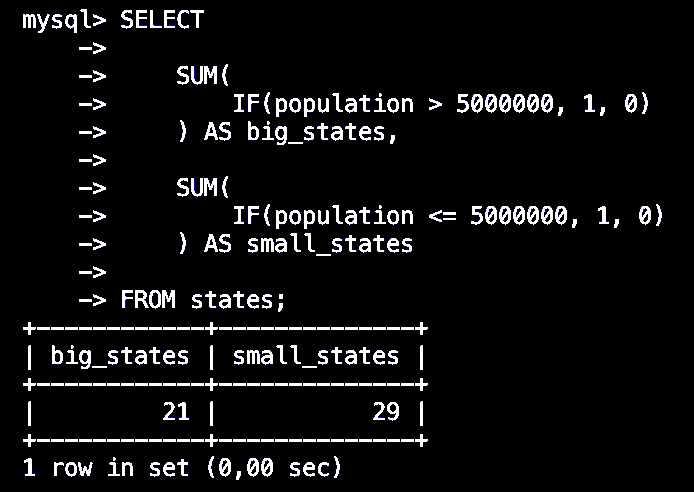
) AS big\_states,

SUM(

IF(population <= 5000000, 1, 0)

) AS small\_states

FROM states;



SELECT

COUNT(\*),

CASE

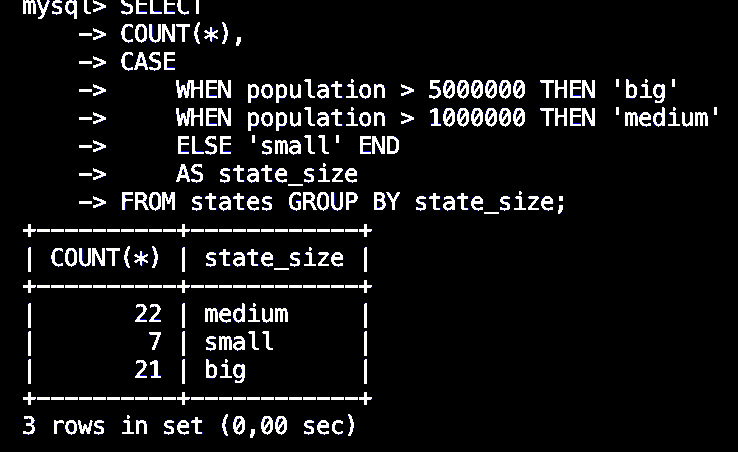
WHEN population > 5000000 THEN 'big'

WHEN population > 1000000 THEN 'medium'

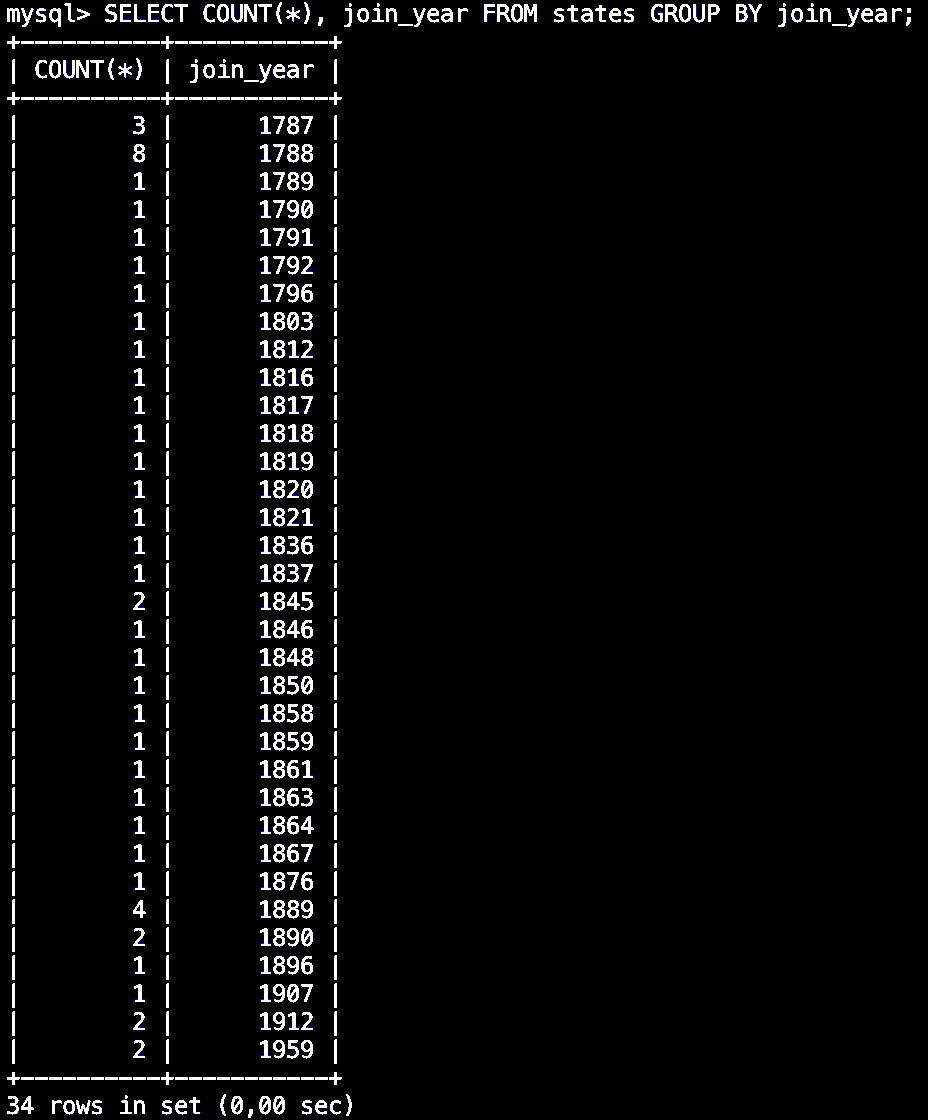
ELSE 'small' END

AS state\_size

FROM states GROUP BY state\_size;



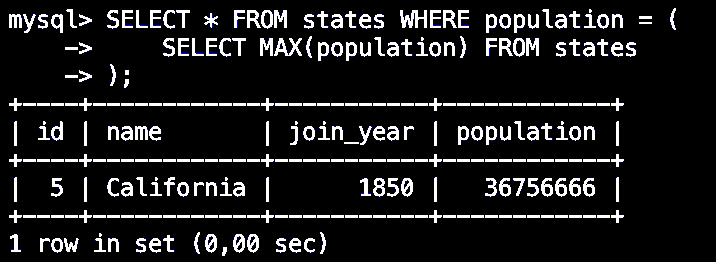
SELECT COUNT(\*), join\_year FROM states GROUP BY join\_year;



SELECT \* FROM states WHERE population = (

SELECT MAX(population) FROM states

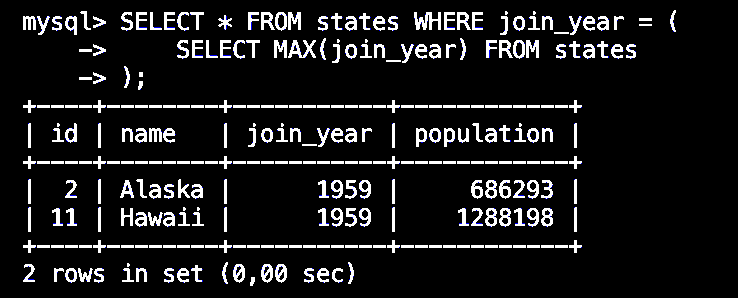
);



SELECT \* FROM states WHERE join\_year = (

SELECT MAX(join\_year) FROM states

);



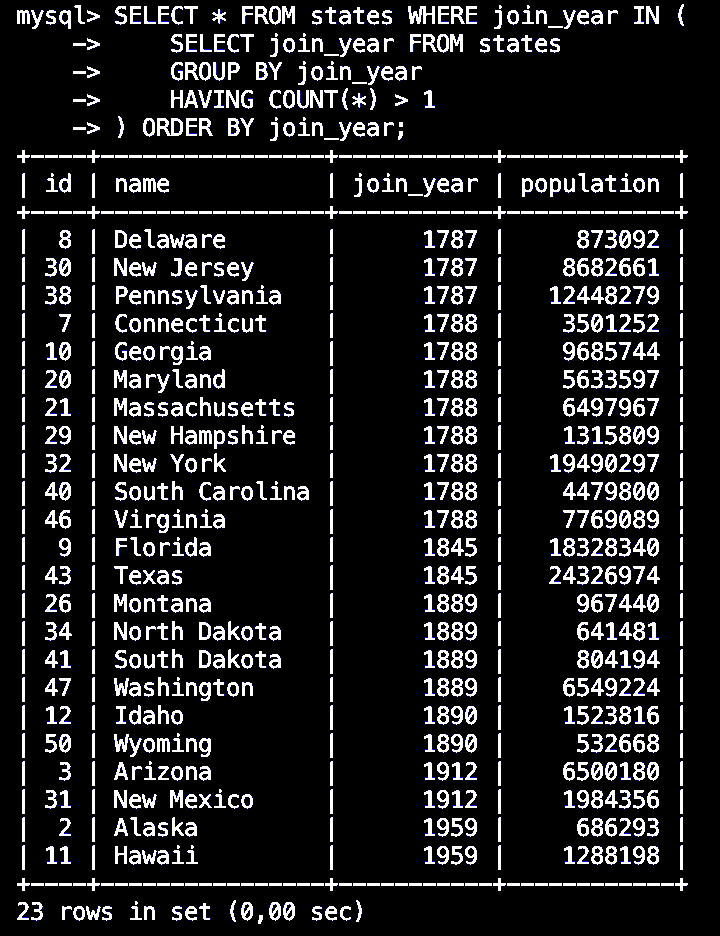
SELECT \* FROM states WHERE join\_year IN (

SELECT join\_year FROM states

GROUP BY join\_year

HAVING COUNT(\*) > 1

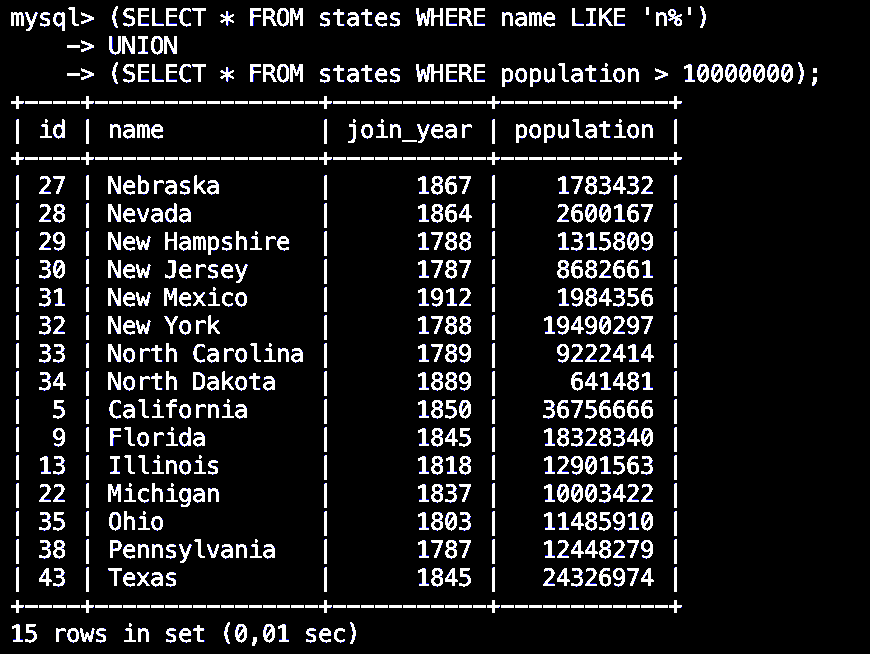
) ORDER BY join\_year;



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

UNION

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

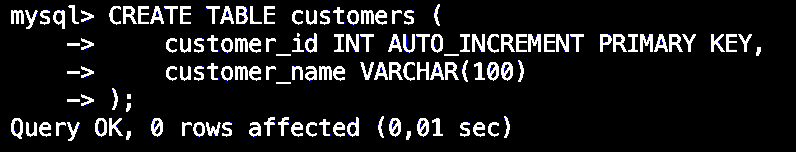


CREATE TABLE customers (

customer\_id INT AUTO\_INCREMENT PRIMARY KEY,

customer\_name VARCHAR(100)

);



CREATE TABLE orders (

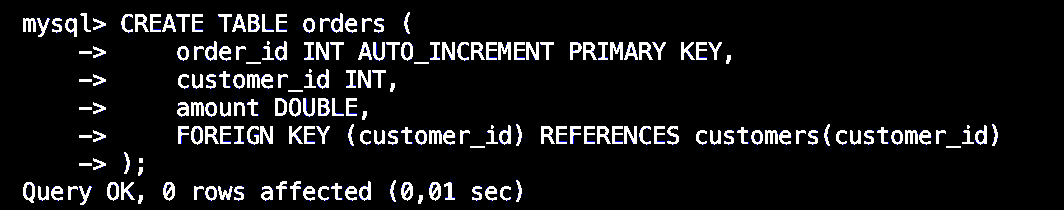
order\_id INT AUTO\_INCREMENT PRIMARY KEY,

customer\_id INT,

amount DOUBLE,

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

);



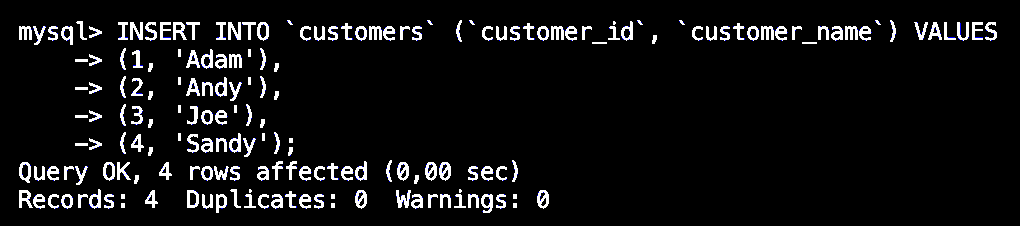
INSERT INTO `customers` (`customer\_id`, `customer\_name`) VALUES

(1, 'Adam'),

(2, 'Andy'),

(3, 'Joe'),

(4, 'Sandy');



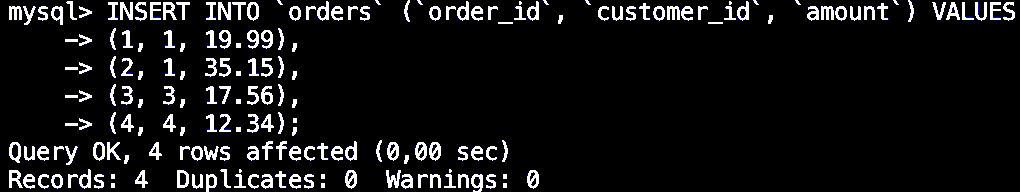
INSERT INTO `orders` (`order\_id`, `customer\_id`, `amount`) VALUES

(1, 1, 19.99),

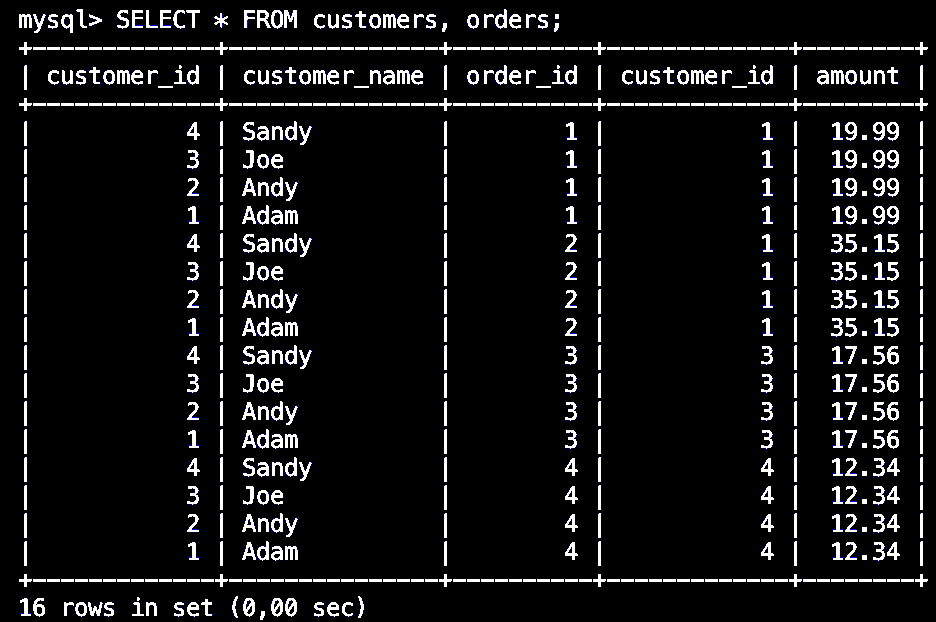
(2, 1, 35.15),

(3, 3, 17.56),

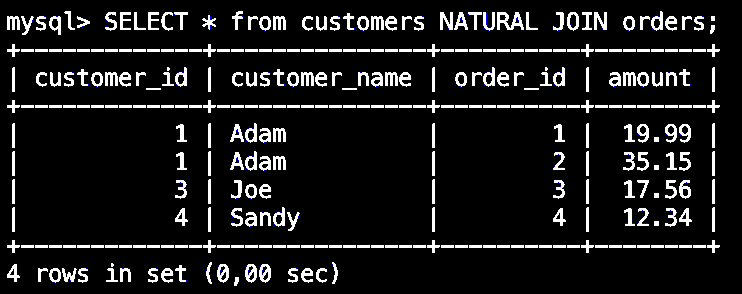
(4, 4, 12.34);



SELECT \* FROM customers, orders;

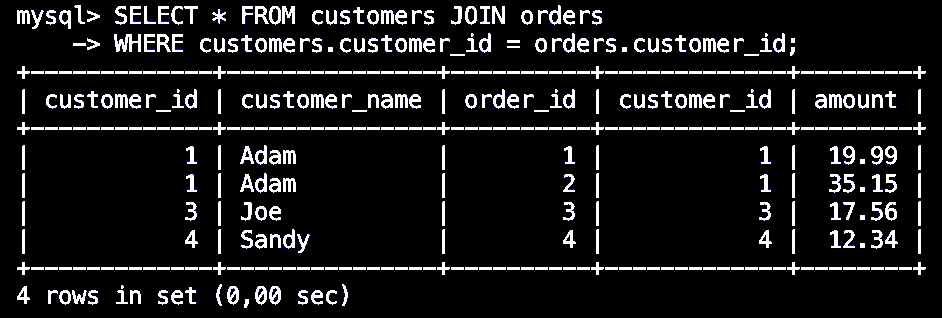


SELECT \* from customers NATURAL JOIN orders;



SELECT \* FROM customers JOIN orders

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



SELECT \* FROM customers JOIN orders

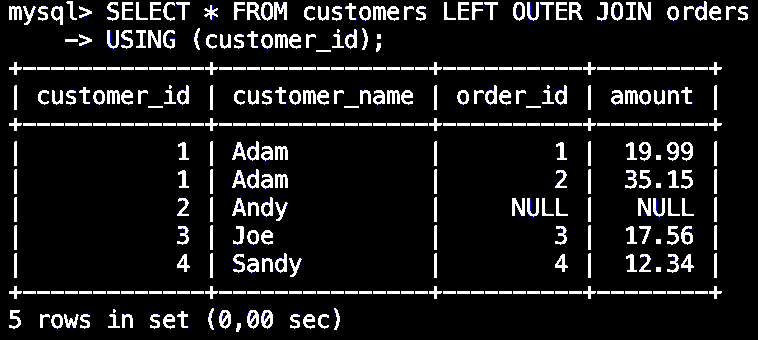
USING (customer\_id)

WHERE orders.amount > 15;



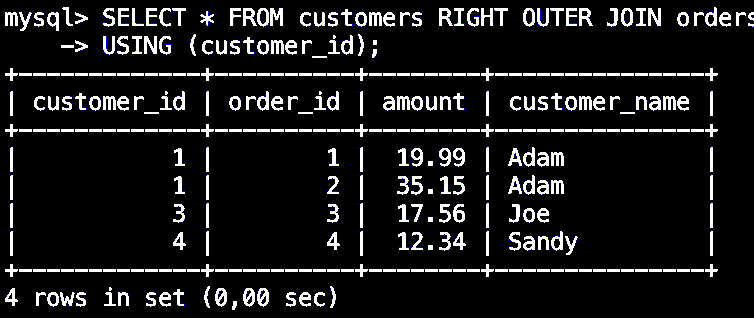
SELECT \* FROM customers LEFT OUTER JOIN orders

USING (customer\_id);



SELECT \* FROM customers RIGHT OUTER JOIN orders

USING (customer\_id);



SELECT \* FROM orders RIGHT OUTER JOIN customers

USING (customer\_id);

USING (customer\_id);

