FGV EMAp - Escola de Matemática Aplicada

Disciplina: Sistemas de Bancos de Dados

FGV EMAp
ESCOLA DE
MATEMÁTICA
APLICADA

Introdução aos Bancos de dados

Renato Rocha Souza

rsouza.fgv@gmail.com

Programa:

Linguagem SQL – Parte II

- Projetando Bancos e Tabelas
- Criando e Populando Tabelas
- Alterando e Apagando Registros
- Carregando um Banco de Dados
- Consultas (queries)
- Consultas com Funções
- Subqueries
- Condições de Pertencimento
- Registros Únicos
- Views
- Joins / Multiple Joins /
- Group By / Having
- Backup de Bancos
- Exercícios

Design de tabelas
Queries avançadas

Slides inspirados no livro:
http://shop.oreilly.com/product/9780596007270.do

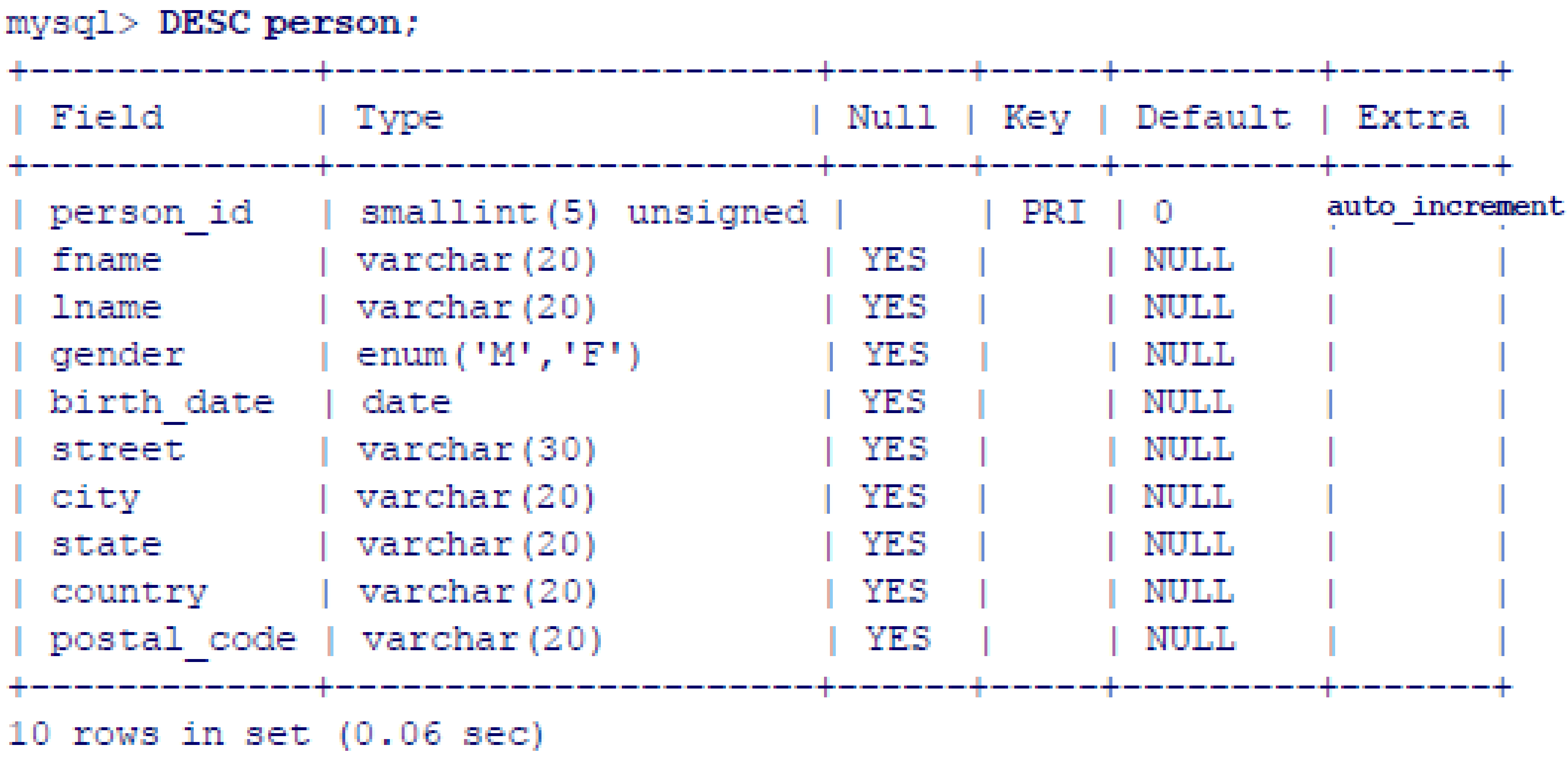
Projetando uma tabela:

- Vamos projetar um banco simples, com duas tabelas:
- Uma tabela com informação de pessoas
- Uma tabela de comidas favoritas.
- As tabelas serão relacionadas entre si, através de uma chave estrangeira.

Criando a primeira tabela:

```
mysql> CREATE TABLE person
-> (person_id SMALLINT UNSIGNED AUTO_INCREMENT,
-> fname VARCHAR(20),
-> lname VARCHAR(20),
-> gender ENUM('M','F'),
-> birth_date DATE,
-> street VARCHAR(30),
-> city VARCHAR(20),
-> state VARCHAR(20),
-> country VARCHAR(20),
-> postal_code VARCHAR(20),
-> CONSTRAINT pk_person PRIMARY KEY (person_id)
-> );
Query OK, 0 rows affected (0.27 sec)
```

Criando a primeira tabela:

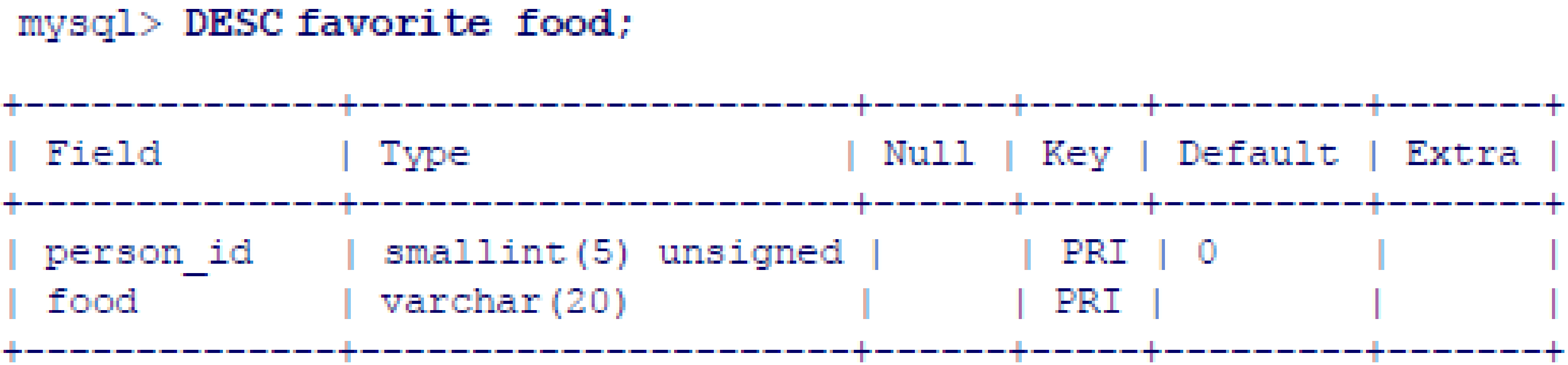


```
mysql> DESC person;
+-------------+----------------------+------+-----+---------+-------+
| Field       | Type                 | Null | Key | Default | Extra |
+-------------+----------------------+------+-----+---------+-------+
| person_id   | smallint(5) unsigned |      | PRI | 0       auto_increment
| fname       | varchar(20)          | YES  |     | NULL    |       |
| lname       | varchar(20)          | YES  |     | NULL    |       |
| gender      | enum('M','F')        | YES  |     | NULL    |       |
| birth_date  | date                 | YES  |     | NULL    |       |
| street      | varchar(30)          | YES  |     | NULL    |       |
| city        | varchar(20)          | YES  |     | NULL    |       |
| state       | varchar(20)          | YES  |     | NULL    |       |
| country     | varchar(20)          | YES  |     | NULL    |       |
| postal_code | varchar(20)          | YES  |     | NULL    |       |
+-------------+----------------------+------+-----+---------+-------+
10 rows in set (0.06 sec)
```

Criando a segunda tabela:

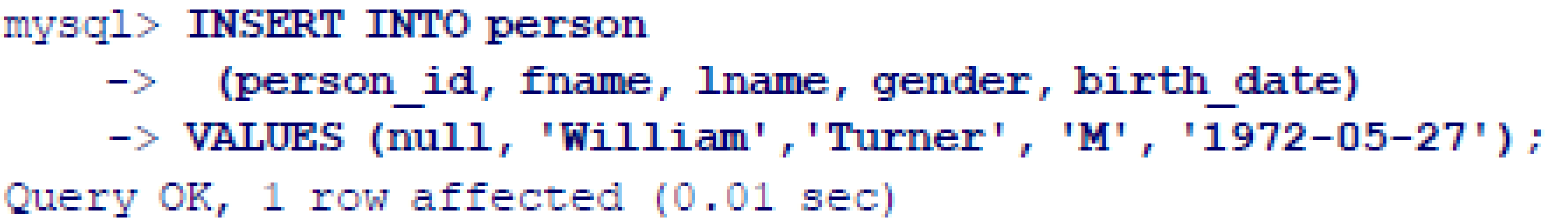
```
mysql> CREATE TABLE favorite_food
-> (person_id SMALLINT UNSIGNED,
-> food VARCHAR(20),
-> CONSTRAINT pk_favorite_food PRIMARY KEY (person_id, food),
-> CONSTRAINT fk_fav_food_person_id FOREIGN KEY (person_id)
-> REFERENCES person (person_id)
-> );
Query OK, 0 rows affected (0.10 sec)
```

Criando a segunda tabela:

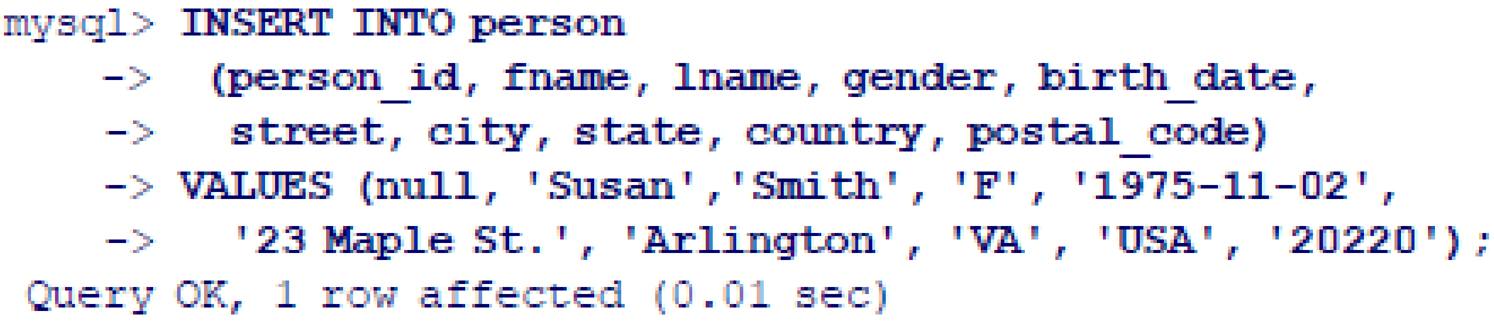


```
mysql> DESC favorite food;

+-------------+----------------------+------+-----+---------+-------+
| Field       | Type                 | Null | Key | Default | Extra |
+-------------+----------------------+------+-----+---------+-------+
| person_id   | smallint(5) unsigned |      | PRI | 0       |       |
| food        | varchar(20)          |      | PRI |         |       |
+-------------+----------------------+------+-----+---------+-------+
```

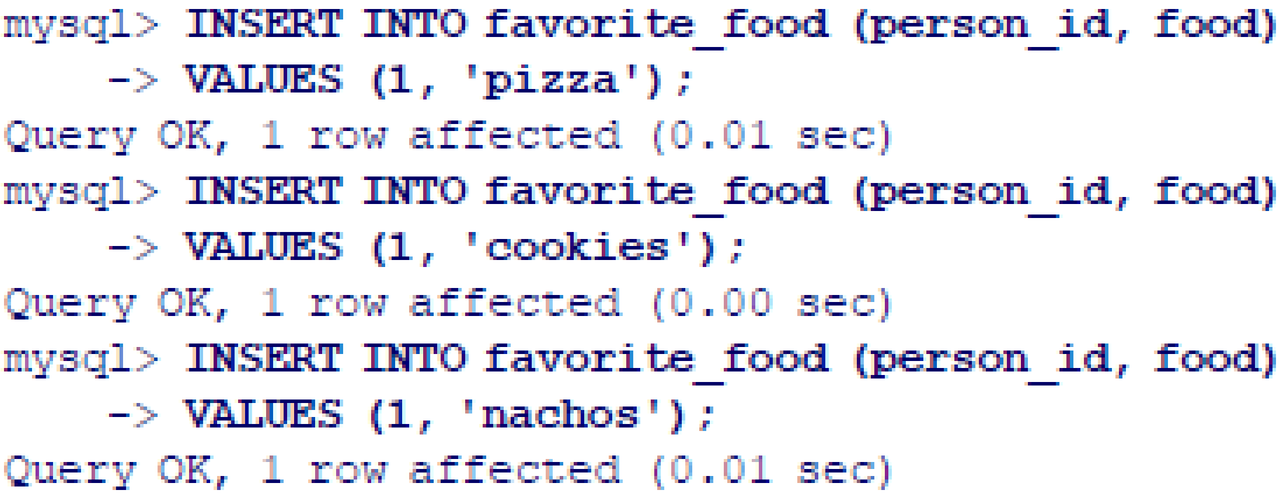
Populando as tabelas…



```
mysql> INSERT INTO person
    ->  (person_id, fname, lname, gender, birth_date)
    -> VALUES (null, 'William','Turner', 'M', '1972-05-27');
Query OK, 1 row affected (0.01 sec)
```

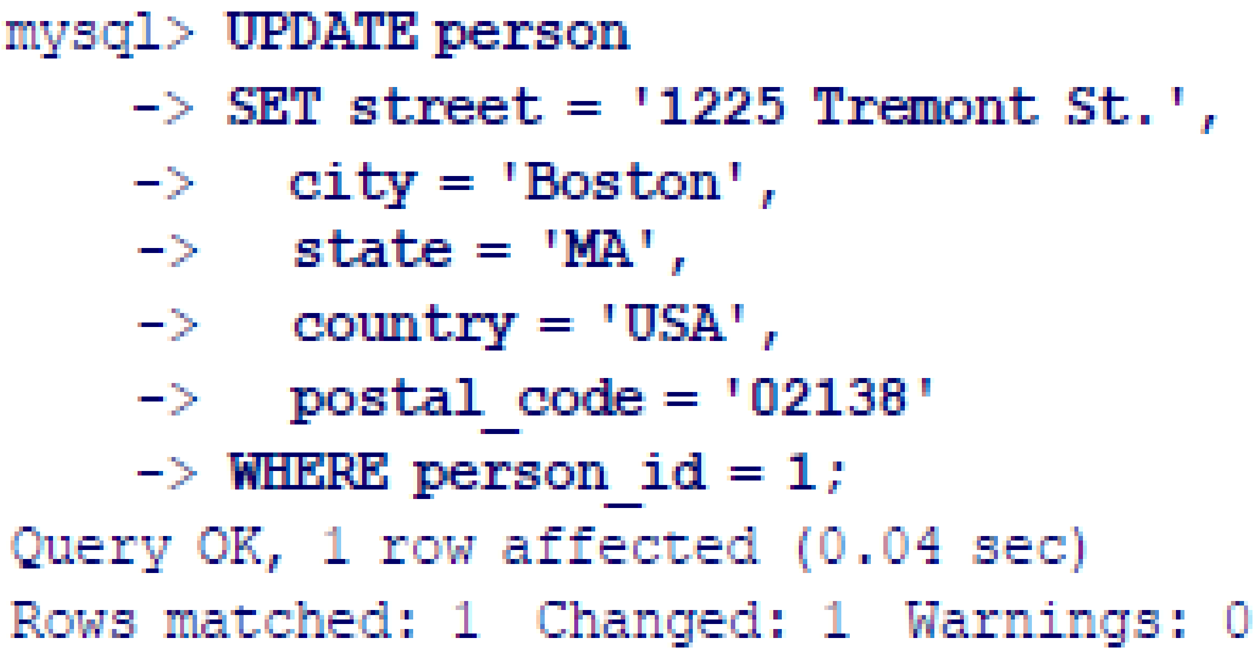


```
mysql> INSERT INTO person
    ->  (person_id, fname, lname, gender, birth_date,
    ->   street, city, state, country, postal_code)
    -> VALUES (null, 'Susan','Smith', 'F', '1975-11-02',
    ->   '23 Maple St.', 'Arlington', 'VA', 'USA', '20220');
 Query OK, 1 row affected (0.01 sec)
```



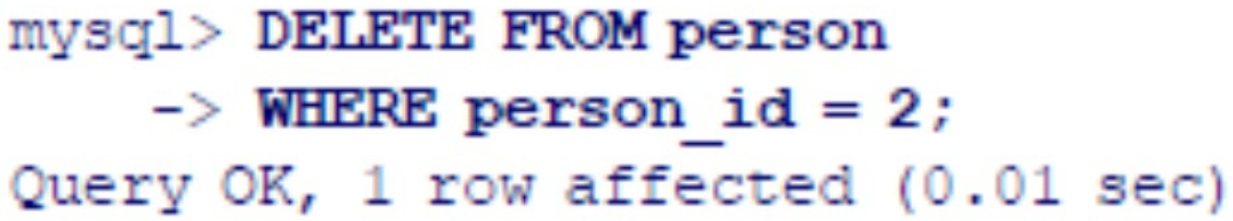
```
mysql> INSERT INTO favorite_food (person_id, food)
    -> VALUES (1, 'pizza');
Query OK, 1 row affected (0.01 sec)
mysql> INSERT INTO favorite_food (person_id, food)
    -> VALUES (1, 'cookies');
Query OK, 1 row affected (0.00 sec)
mysql> INSERT INTO favorite_food (person_id, food)
    -> VALUES (1, 'nachos');
Query OK, 1 row affected (0.01 sec)
```

Alterando registros.



```
mysql> UPDATE person
    -> SET street = '1225 Tremont St.',
    ->   city = 'Boston',
    ->   state = 'MA',
    ->   country = 'USA',
    ->   postal_code = '02138'
    -> WHERE person_id = 1;
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Apagando registros.

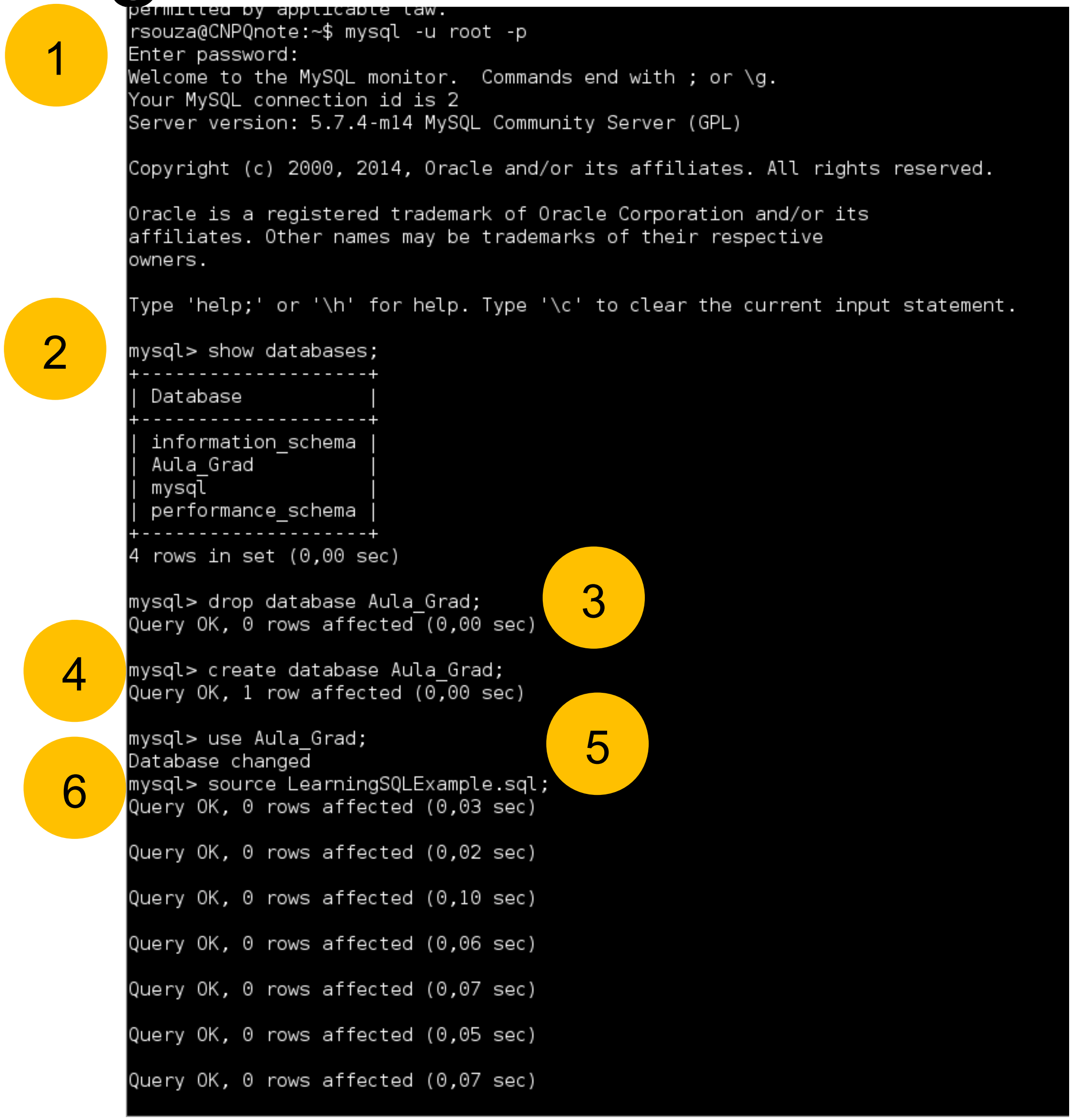


```
mysql> DELETE FROM person
    -> WHERE person_id = 2;
Query OK, 1 row affected (0.01 sec)
```

Carregando um banco de dados

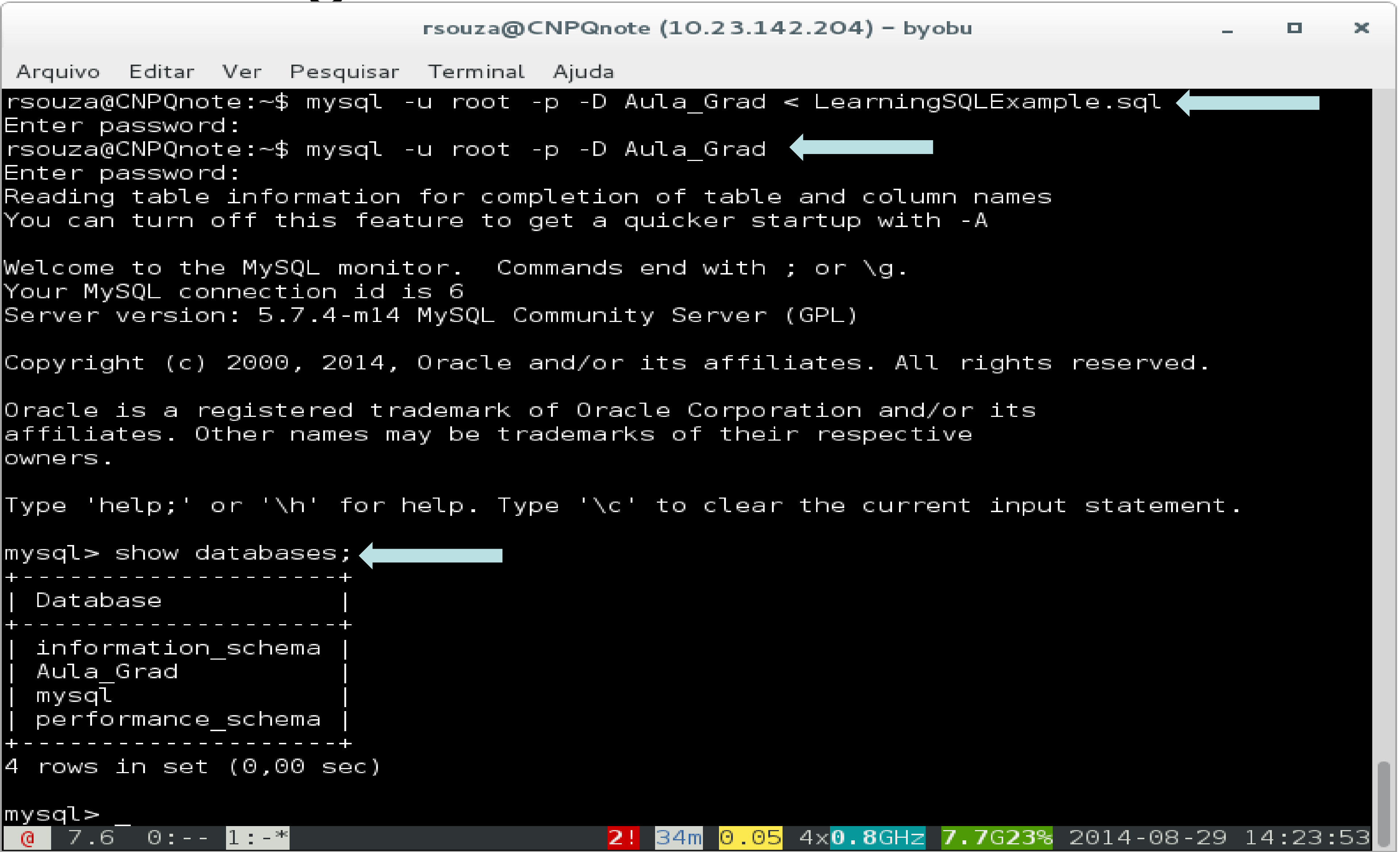
Table name	**Definition**
`Account`	A particular product opened for a particular customer
`Branch`	A location at which banking transactions are conducted
`Business`	A corporate customer (subtype of the `Customer` table)
`Customer`	A person or corporation known to the bank
`Department`	A group of bank employees implementing a particular banking function
`Employee`	A person working for the bank
`Individual`	A noncorporate customer (subtype of the `Customer` table)
`Officer`	A person allowed to transact business for a corporate customer
`Product`	A banking service offered to customers
`Product_type`	A group of products having a similar function
`transaction`	A change made to an account balance

Carregando um banco de dados



```
permitted by applicable law.
rsouza@CNPQnote:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.4-m14 MySQL Community Server (GPL)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| Aula_Grad          |
| mysql              |
| performance_schema |
+--------------------+
4 rows in set (0,00 sec)

mysql> drop database Aula_Grad;
Query OK, 0 rows affected (0,00 sec)

mysql> create database Aula_Grad;
Query OK, 1 row affected (0,00 sec)

mysql> use Aula_Grad;
Database changed
mysql> source LearningSQLExample.sql;
Query OK, 0 rows affected (0,03 sec)

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

Query OK, 0 rows affected (0,10 sec)

Query OK, 0 rows affected (0,06 sec)

Query OK, 0 rows affected (0,07 sec)

Query OK, 0 rows affected (0,05 sec)

Query OK, 0 rows affected (0,07 sec)
```

Carregando um banco de dados

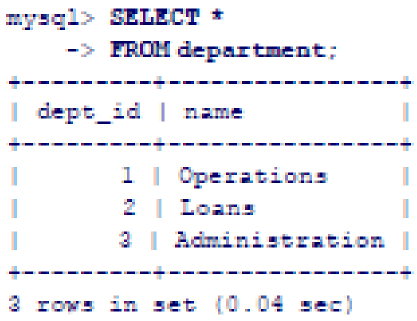


rsouza@CNPQnote (10.23.142.204) - byobu
Arquivo Editar Ver Pesquisar Terminal Ajuda

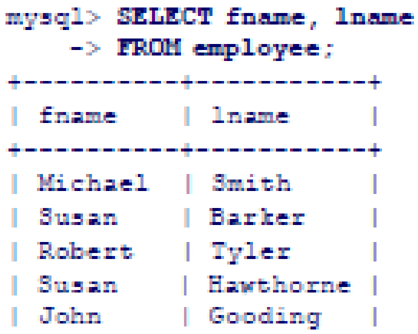
```
rsouza@CNPQnote:~$ mysql -u root -p -D Aula_Grad < LearningSQLExample.sql
Enter password:
rsouza@CNPQnote:~$ mysql -u root -p -D Aula_Grad
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.7.4-m14 MySQL Community Server (GPL)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| Aula_Grad          |
| mysql              |
| performance_schema |
+--------------------+
4 rows in set (0,00 sec)

mysql> _
```

@ 7.6 0:-- 1:-* 2! 34m 0.05 4x0.8GHz 7.7G23% 2014-08-29 14:23:53

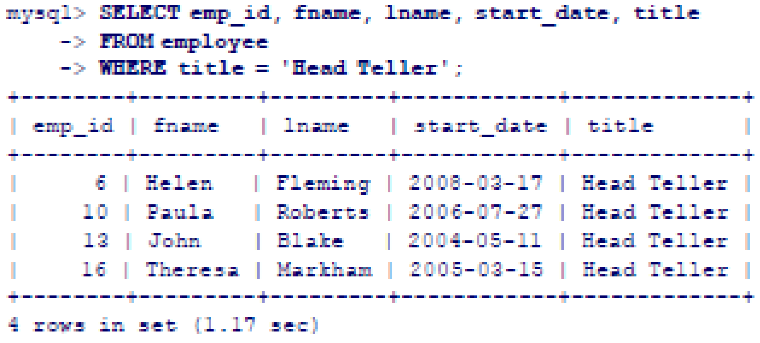
Recordando: consultas



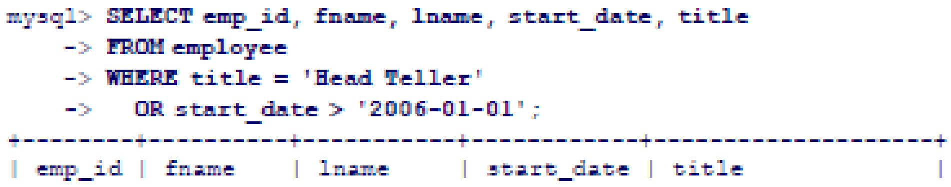
```
mysql> SELECT *
    -> FROM department;
+---------+----------------+
| dept_id | name           |
+---------+----------------+
|       1 | Operations     |
|       2 | Loans          |
|       3 | Administration |
+---------+----------------+
3 rows in set (0.04 sec)
```



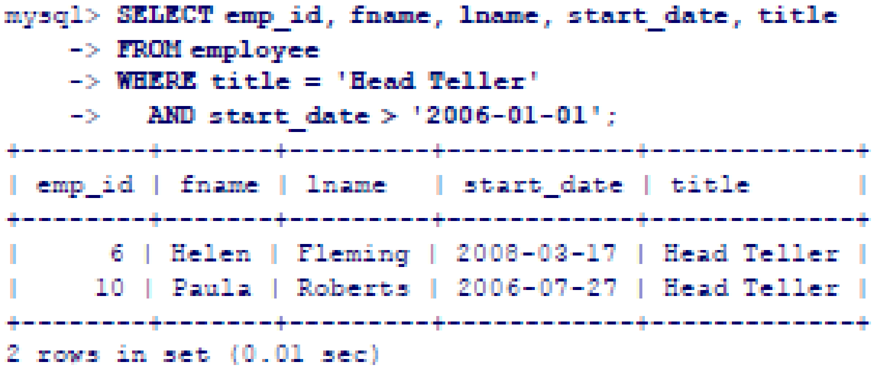
```
mysql> SELECT fname, lname
    -> FROM employee;
+----------+-----------+
| fname    | lname     |
+----------+-----------+
| Michael  | Smith     |
| Susan    | Barker    |
| Robert   | Tyler     |
| Susan    | Hawthorne |
| John     | Gooding   |
```



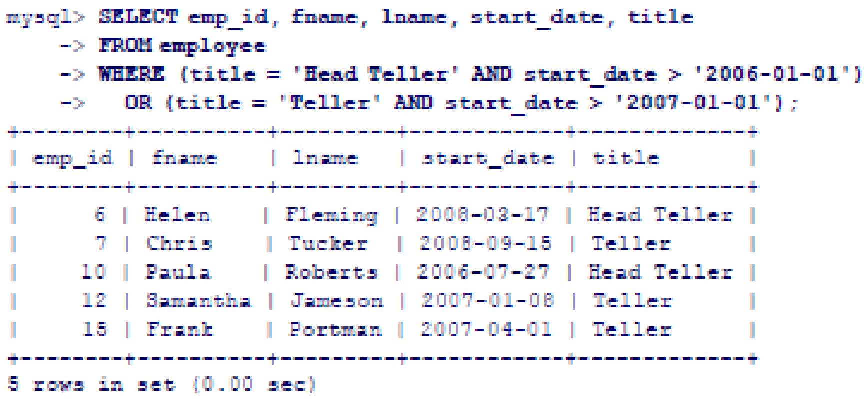
```
mysql> SELECT emp_id, fname, lname, start_date, title
    -> FROM employee
    -> WHERE title = 'Head Teller';
+--------+---------+---------+------------+-------------+
| emp_id | fname   | lname   | start_date | title       |
+--------+---------+---------+------------+-------------+
|      6 | Helen   | Fleming | 2008-03-17 | Head Teller |
|     10 | Paula   | Roberts | 2006-07-27 | Head Teller |
|     13 | John    | Blake   | 2004-05-11 | Head Teller |
|     16 | Theresa | Markham | 2005-03-15 | Head Teller |
+--------+---------+---------+------------+-------------+
4 rows in set (1.17 sec)
```



```
mysql> SELECT emp_id, fname, lname, start_date, title
    -> FROM employee
    -> WHERE title = 'Head Teller'
    ->   OR start_date > '2006-01-01';
+--------+----------+----------+------------+--------------------+
| emp_id | fname    | lname    | start_date | title              |
```

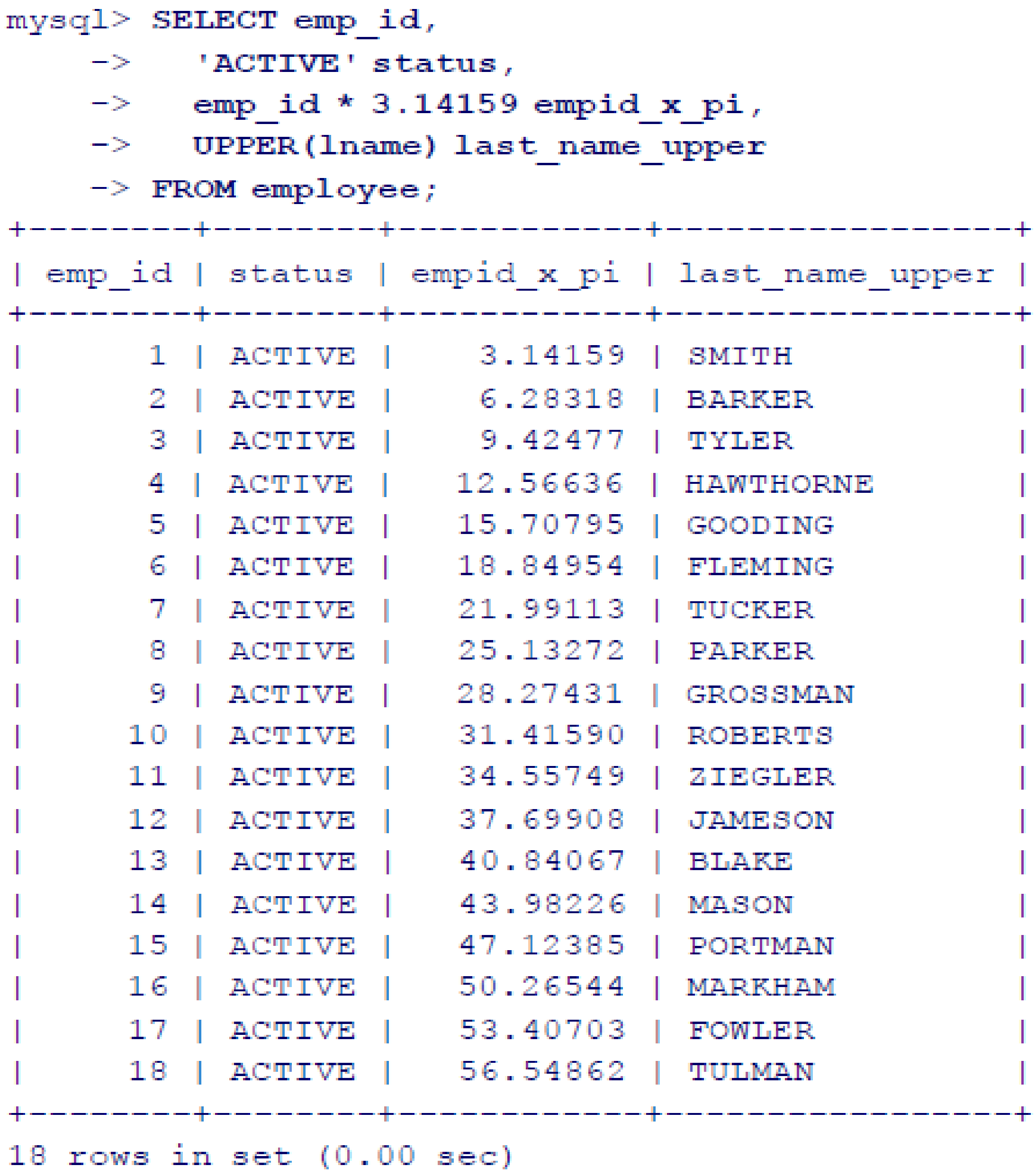


```
mysql> SELECT emp_id, fname, lname, start_date, title
    -> FROM employee
    -> WHERE title = 'Head Teller'
    ->   AND start_date > '2006-01-01';
+--------+-------+---------+------------+-------------+
| emp_id | fname | lname   | start_date | title       |
+--------+-------+---------+------------+-------------+
|      6 | Helen | Fleming | 2008-03-17 | Head Teller |
|     10 | Paula | Roberts | 2006-07-27 | Head Teller |
+--------+-------+---------+------------+-------------+
2 rows in set (0.01 sec)
```



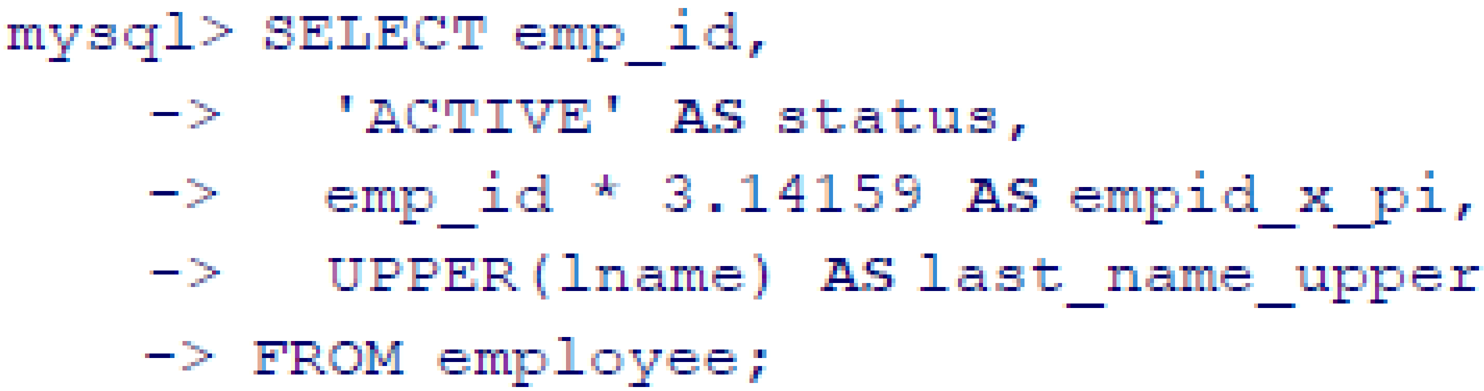
```
mysql> SELECT emp_id, fname, lname, start_date, title
    -> FROM employee
    -> WHERE (title = 'Head Teller' AND start_date > '2006-01-01')
    ->   OR (title = 'Teller' AND start_date > '2007-01-01');
+--------+----------+---------+------------+-------------+
| emp_id | fname    | lname   | start_date | title       |
+--------+----------+---------+------------+-------------+
|      6 | Helen    | Fleming | 2008-03-17 | Head Teller |
|      7 | Chris    | Tucker  | 2008-09-15 | Teller      |
|     10 | Paula    | Roberts | 2006-07-27 | Head Teller |
|     12 | Samantha | Jameson | 2007-01-08 | Teller      |
|     15 | Frank    | Portman | 2007-04-01 | Teller      |
+--------+----------+---------+------------+-------------+
5 rows in set (0.00 sec)
```

Queries com funções



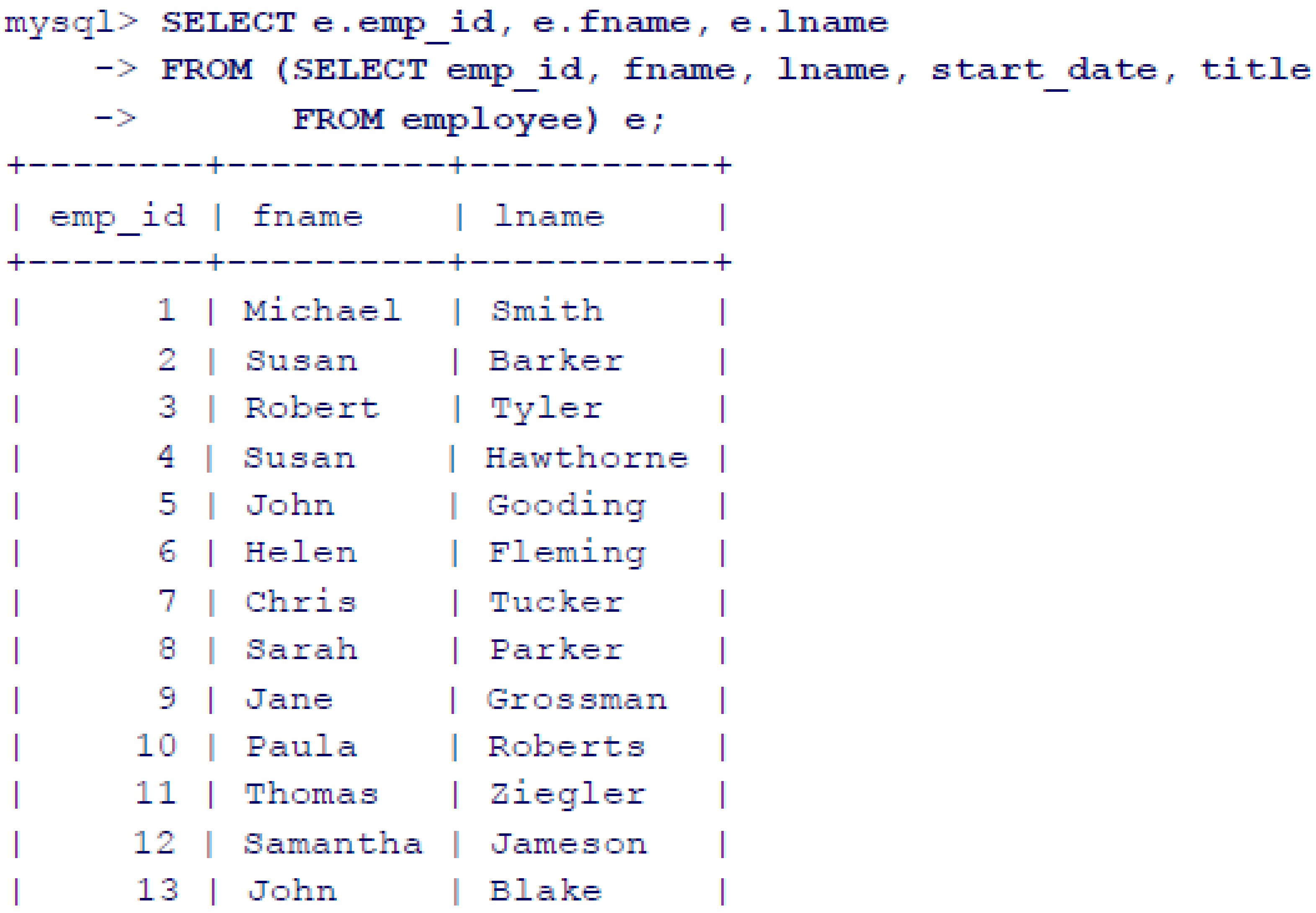
```
mysql> SELECT emp_id,
    ->   'ACTIVE' status,
    ->   emp_id * 3.14159 empid_x_pi,
    ->   UPPER(lname) last_name_upper
    -> FROM employee;
+--------+--------+------------+-----------------+
| emp_id | status | empid_x_pi | last_name_upper |
+--------+--------+------------+-----------------+
|      1 | ACTIVE |    3.14159 | SMITH           |
|      2 | ACTIVE |    6.28318 | BARKER          |
|      3 | ACTIVE |    9.42477 | TYLER           |
|      4 | ACTIVE |   12.56636 | HAWTHORNE       |
|      5 | ACTIVE |   15.70795 | GOODING         |
|      6 | ACTIVE |   18.84954 | FLEMING         |
|      7 | ACTIVE |   21.99113 | TUCKER          |
|      8 | ACTIVE |   25.13272 | PARKER          |
|      9 | ACTIVE |   28.27431 | GROSSMAN        |
|     10 | ACTIVE |   31.41590 | ROBERTS         |
|     11 | ACTIVE |   34.55749 | ZIEGLER         |
|     12 | ACTIVE |   37.69908 | JAMESON         |
|     13 | ACTIVE |   40.84067 | BLAKE           |
|     14 | ACTIVE |   43.98226 | MASON           |
|     15 | ACTIVE |   47.12385 | PORTMAN         |
|     16 | ACTIVE |   50.26544 | MARKHAM         |
|     17 | ACTIVE |   53.40703 | FOWLER          |
|     18 | ACTIVE |   56.54862 | TULMAN          |
+--------+--------+------------+-----------------+
18 rows in set (0.00 sec)
```

Queries com funções (usando AS)



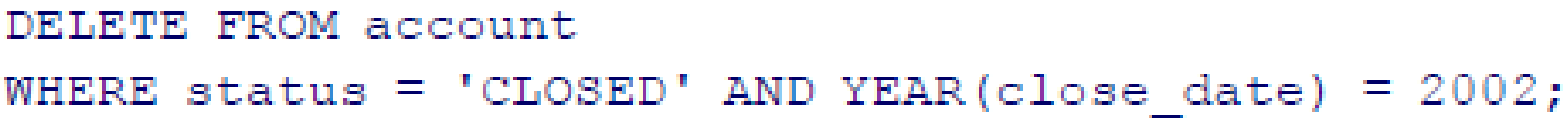
```
mysql> SELECT emp_id,
    ->   'ACTIVE' AS status,
    ->   emp_id * 3.14159 AS empid_x_pi,
    ->   UPPER(lname) AS last_name_upper
    -> FROM employee;
```

Subqueries



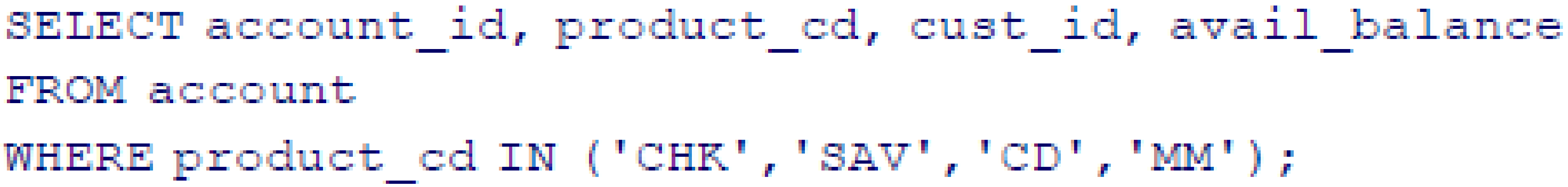
```
mysql> SELECT e.emp_id, e.fname, e.lname
    -> FROM (SELECT emp_id, fname, lname, start_date, title
    ->       FROM employee) e;
+--------+----------+-----------+
| emp_id | fname    | lname     |
+--------+----------+-----------+
|      1 | Michael  | Smith     |
|      2 | Susan    | Barker    |
|      3 | Robert   | Tyler     |
|      4 | Susan    | Hawthorne |
|      5 | John     | Gooding   |
|      6 | Helen    | Fleming   |
|      7 | Chris    | Tucker    |
|      8 | Sarah    | Parker    |
|      9 | Jane     | Grossman  |
|     10 | Paula    | Roberts   |
|     11 | Thomas   | Ziegler   |
|     12 | Samantha | Jameson   |
|     13 | John     | Blake     |
```

Apagando registros



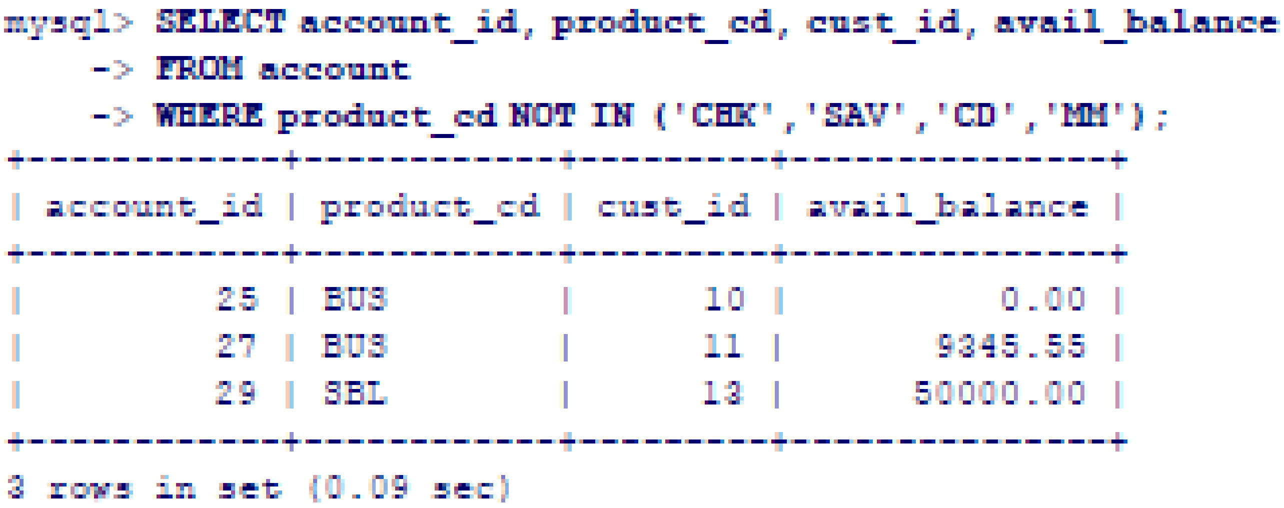
```
DELETE FROM account
WHERE status = 'CLOSED' AND YEAR(close_date) = 2002;
```

Condições de pertencimento



```
SELECT account_id, product_cd, cust_id, avail_balance
FROM account
WHERE product_cd IN ('CHK','SAV','CD','MM');
```

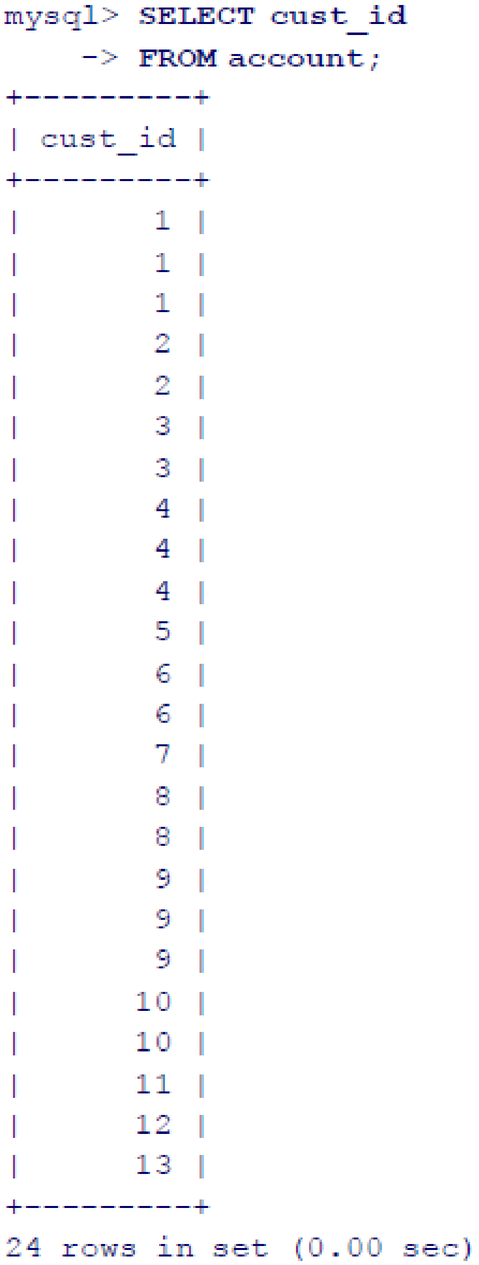
… ou não pertencimento



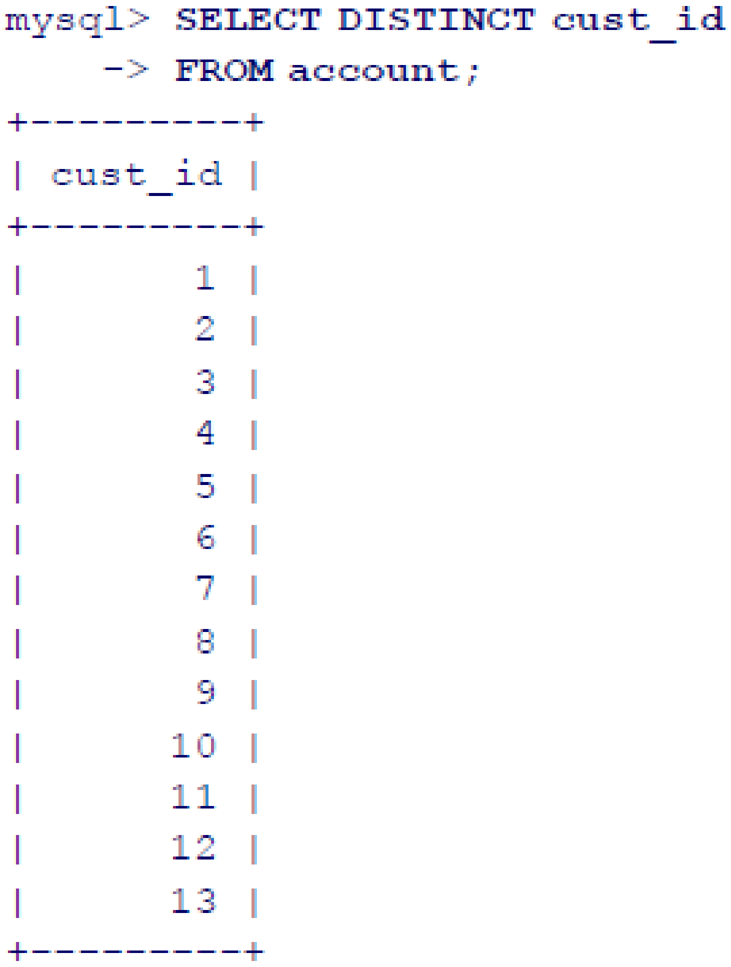
```
mysql> SELECT account_id, product_cd, cust_id, avail_balance
    -> FROM account
    -> WHERE product_cd NOT IN ('CHK','SAV','CD','MM');
+------------+------------+---------+---------------+
| account_id | product_cd | cust_id | avail_balance |
+------------+------------+---------+---------------+
|         25 | BUS        |      10 |          0.00 |
|         27 | BUS        |      11 |       9345.55 |
|         29 | SBL        |      13 |      50000.00 |
+------------+------------+---------+---------------+
3 rows in set (0.09 sec)
```

De maneira geral, o NOT pode ser usado para negar condições

Registros únicos

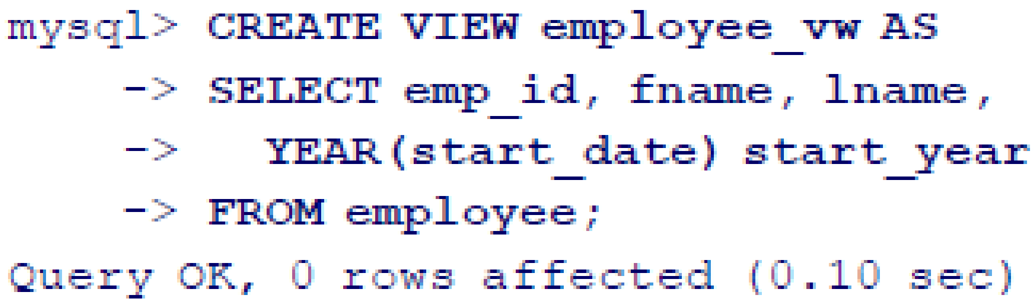


```
mysql> SELECT cust_id
    -> FROM account;
+---------+
| cust_id |
+---------+
|       1 |
|       1 |
|       1 |
|       2 |
|       2 |
|       3 |
|       3 |
|       4 |
|       4 |
|       4 |
|       5 |
|       6 |
|       6 |
|       7 |
|       8 |
|       8 |
|       9 |
|       9 |
|       9 |
|      10 |
|      10 |
|      11 |
|      12 |
|      13 |
+---------+
24 rows in set (0.00 sec)
```

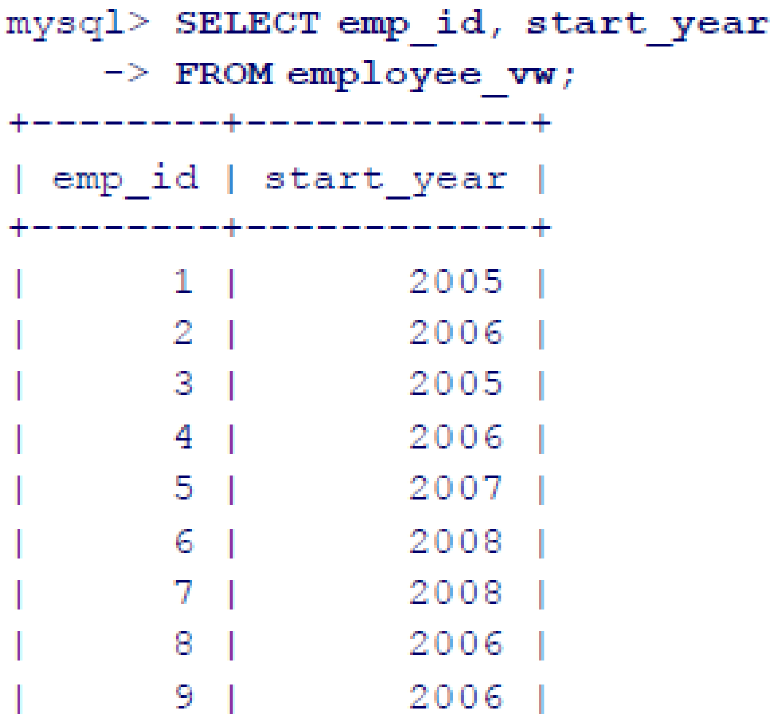


```
mysql> SELECT DISTINCT cust_id
    -> FROM account;
+---------+
| cust_id |
+---------+
|       1 |
|       2 |
|       3 |
|       4 |
|       5 |
|       6 |
|       7 |
|       8 |
|       9 |
|      10 |
|      11 |
|      12 |
|      13 |
+---------+
```

Views

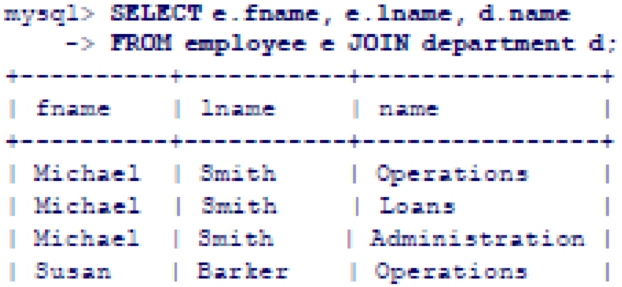


```
mysql> CREATE VIEW employee_vw AS
    -> SELECT emp_id, fname, lname,
    ->   YEAR(start_date) start_year
    -> FROM employee;
Query OK, 0 rows affected (0.10 sec)
```



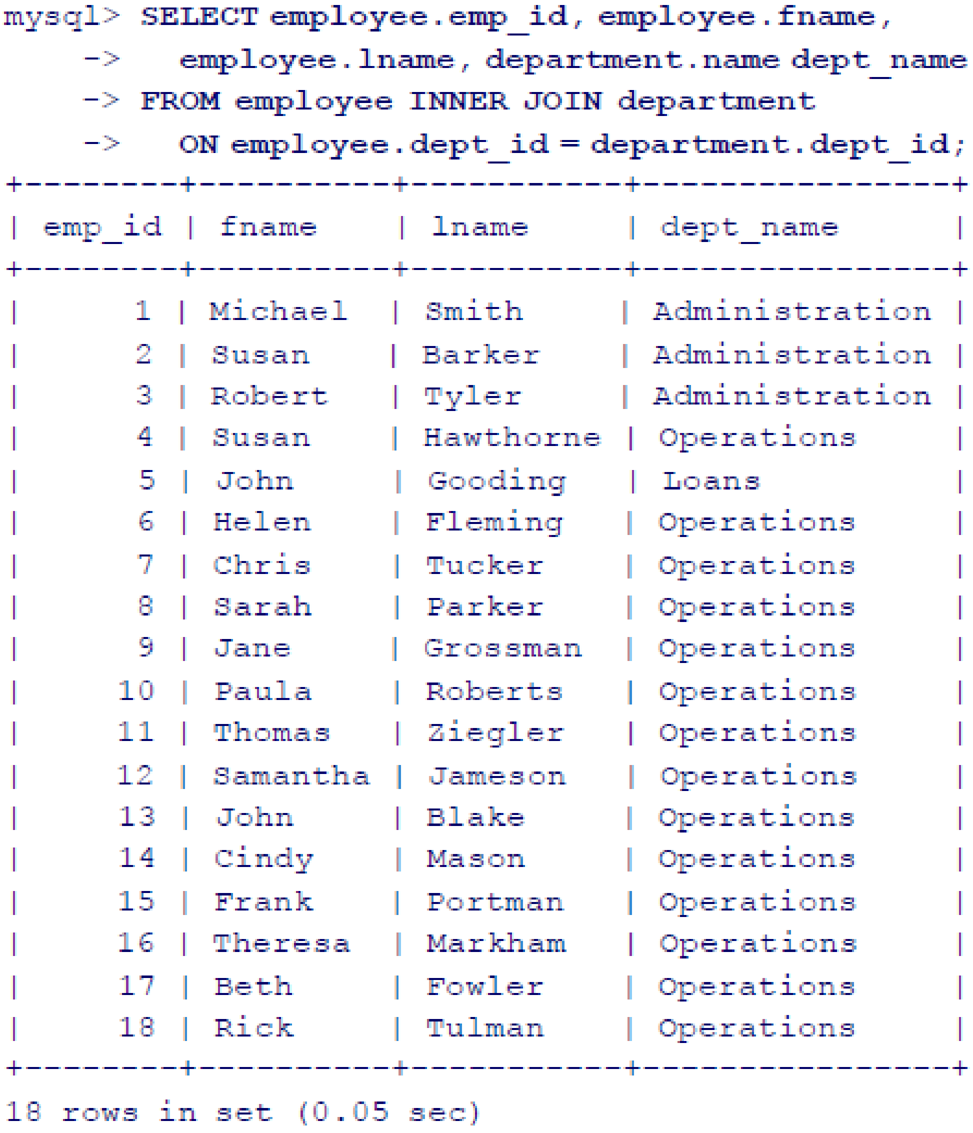
```
mysql> SELECT emp_id, start_year
    -> FROM employee_vw;
+--------+------------+
| emp_id | start_year |
+--------+------------+
|      1 |       2005 |
|      2 |       2006 |
|      3 |       2005 |
|      4 |       2006 |
|      5 |       2007 |
|      6 |       2008 |
|      7 |       2008 |
|      8 |       2006 |
|      9 |       2006 |
```

Joins



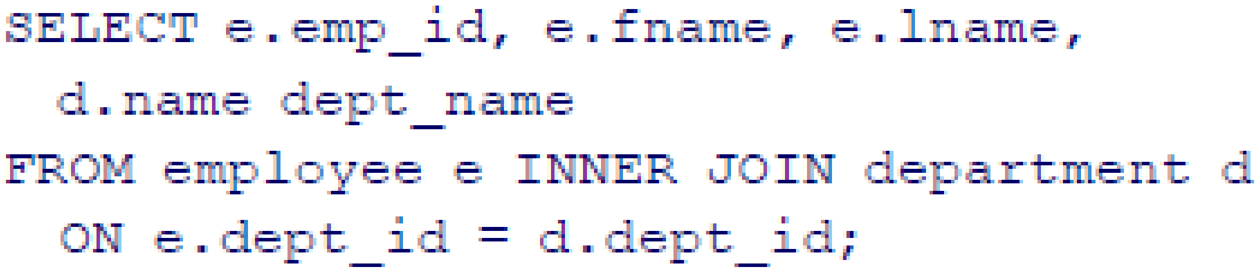
```
mysql> SELECT e.fname, e.lname, d.name
    -> FROM employee e JOIN department d;
+----------+----------+----------------+
| fname    | lname    | name           |
+----------+----------+----------------+
| Michael  | Smith    | Operations     |
| Michael  | Smith    | Loans          |
| Michael  | Smith    | Administration |
| Susan    | Barker   | Operations     |
```

???



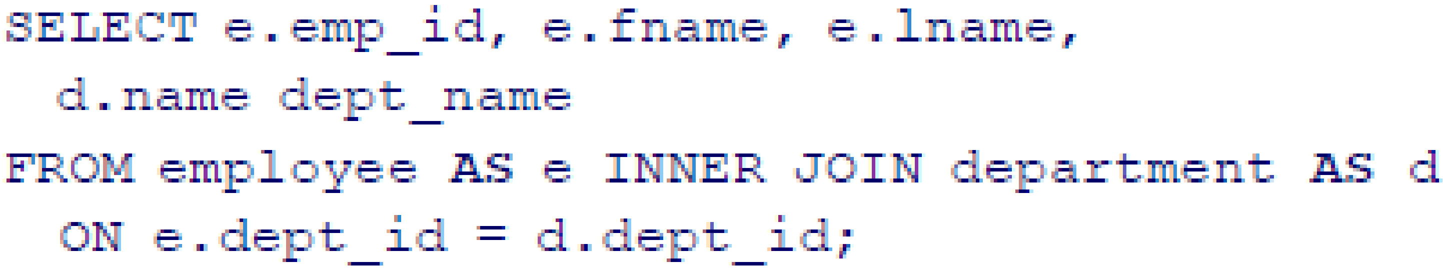
```
mysql> SELECT employee.emp_id, employee.fname,
    ->   employee.lname, department.name dept_name
    -> FROM employee INNER JOIN department
    ->   ON employee.dept_id = department.dept_id;
+--------+----------+-----------+----------------+
| emp_id | fname    | lname     | dept_name      |
+--------+----------+-----------+----------------+
|      1 | Michael  | Smith     | Administration |
|      2 | Susan    | Barker    | Administration |
|      3 | Robert   | Tyler     | Administration |
|      4 | Susan    | Hawthorne | Operations     |
|      5 | John     | Gooding   | Loans          |
|      6 | Helen    | Fleming   | Operations     |
|      7 | Chris    | Tucker    | Operations     |
|      8 | Sarah    | Parker    | Operations     |
|      9 | Jane     | Grossman  | Operations     |
|     10 | Paula    | Roberts   | Operations     |
|     11 | Thomas   | Ziegler   | Operations     |
|     12 | Samantha | Jameson   | Operations     |
|     13 | John     | Blake     | Operations     |
|     14 | Cindy    | Mason     | Operations     |
|     15 | Frank    | Portman   | Operations     |
|     16 | Theresa  | Markham   | Operations     |
|     17 | Beth     | Fowler    | Operations     |
|     18 | Rick     | Tulman    | Operations     |
+--------+----------+-----------+----------------+
18 rows in set (0.05 sec)
```

Join: aliasing



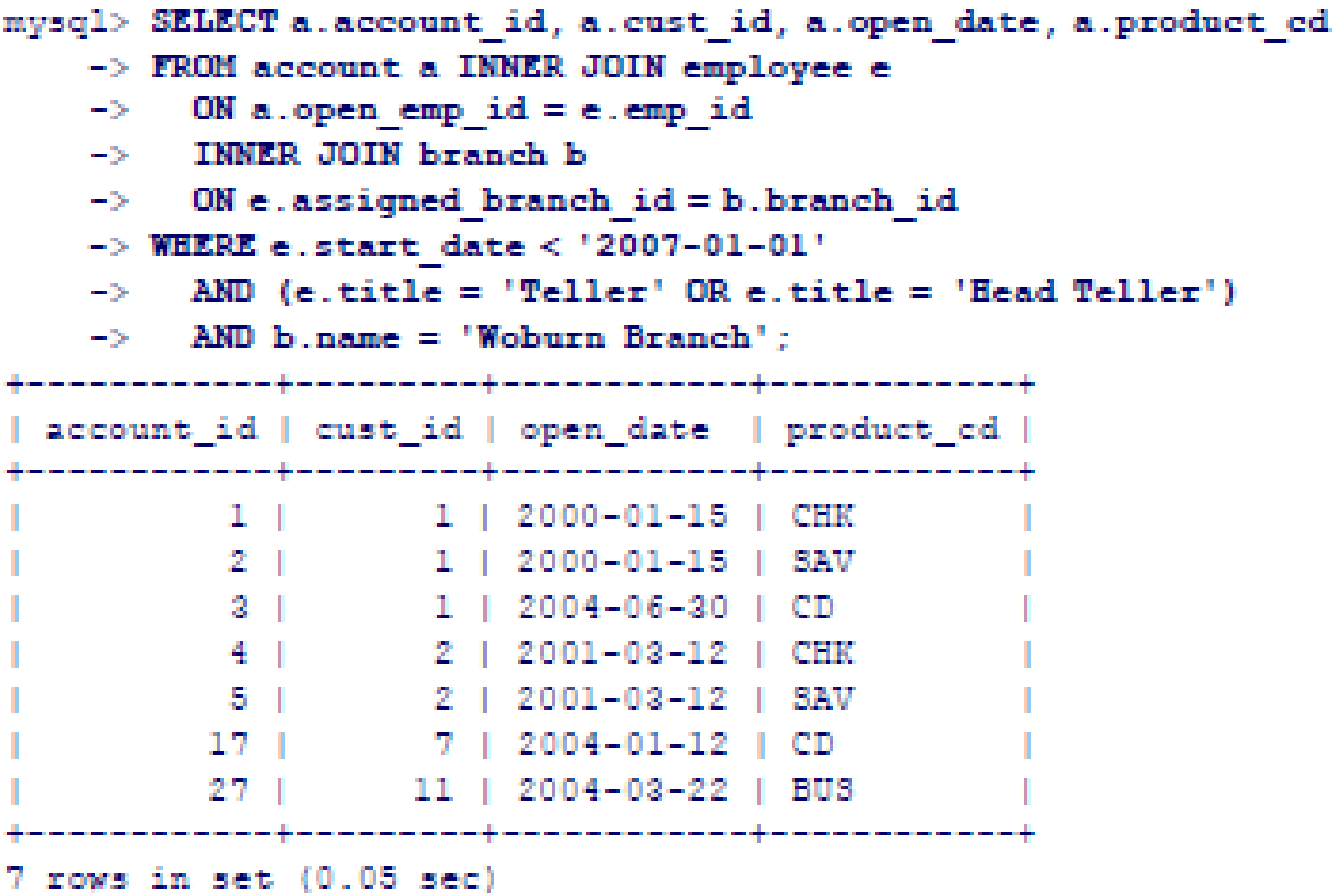
```
SELECT e.emp_id, e.fname, e.lname,
  d.name dept_name
FROM employee e INNER JOIN department d
  ON e.dept_id = d.dept_id;
```

ou



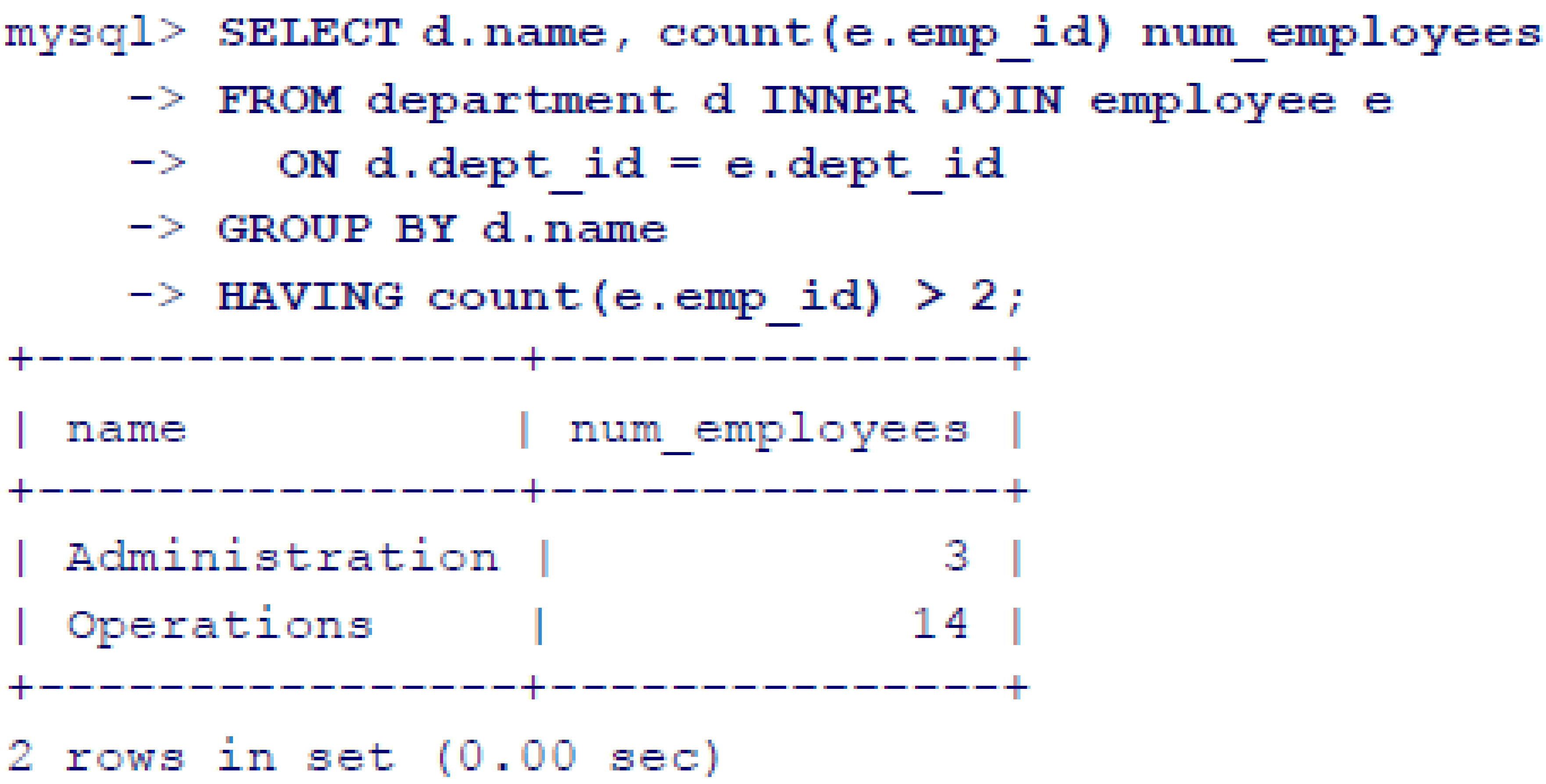
```
SELECT e.emp_id, e.fname, e.lname,
  d.name dept_name
FROM employee AS e INNER JOIN department AS d
  ON e.dept_id = d.dept_id;
```

Multiple joins



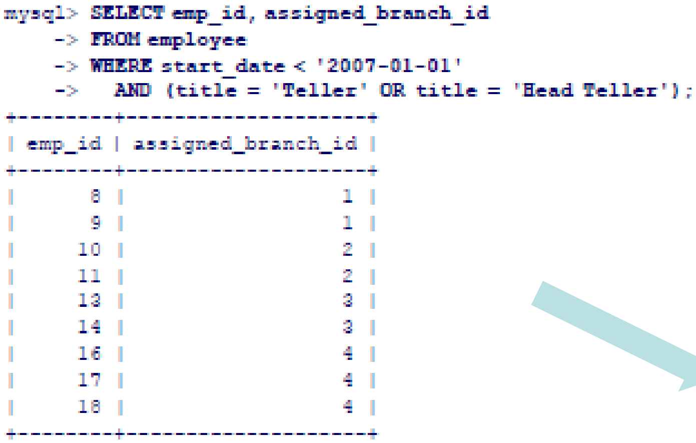
```
mysql> SELECT a.account_id, a.cust_id, a.open_date, a.product_cd
    -> FROM account a INNER JOIN employee e
    ->   ON a.open_emp_id = e.emp_id
    ->   INNER JOIN branch b
    ->   ON e.assigned_branch_id = b.branch_id
    -> WHERE e.start_date < '2007-01-01'
    ->   AND (e.title = 'Teller' OR e.title = 'Head Teller')
    ->   AND b.name = 'Woburn Branch';
+------------+---------+------------+------------+
| account_id | cust_id | open_date  | product_cd |
+------------+---------+------------+------------+
|          1 |       1 | 2000-01-15 | CHK        |
|          2 |       1 | 2000-01-15 | SAV        |
|          3 |       1 | 2004-06-30 | CD         |
|          4 |       2 | 2001-03-12 | CHK        |
|          5 |       2 | 2001-03-12 | SAV        |
|         17 |       7 | 2004-01-12 | CD         |
|         27 |      11 | 2004-03-22 | BUS        |
+------------+---------+------------+------------+
7 rows in set (0.05 sec)
```

Group by & Having

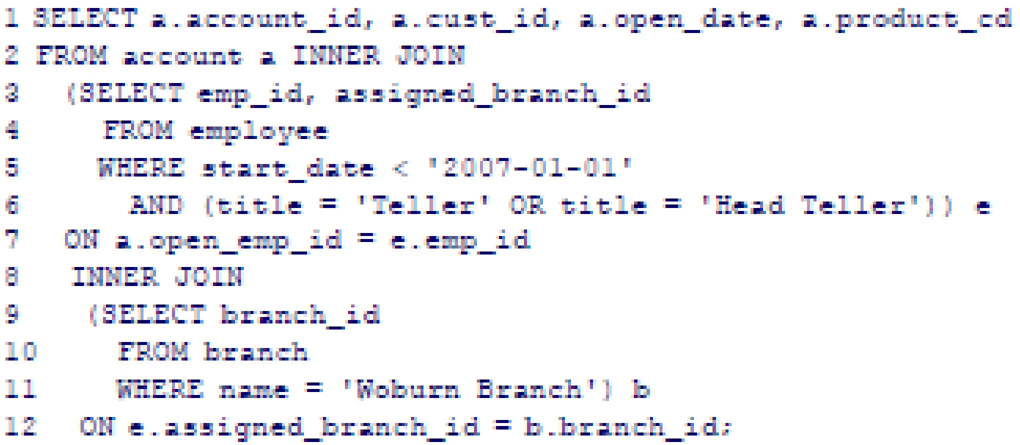


```
mysql> SELECT d.name, count(e.emp_id) num_employees
    -> FROM department d INNER JOIN employee e
    ->   ON d.dept_id = e.dept_id
    -> GROUP BY d.name
    -> HAVING count(e.emp_id) > 2;
+----------------+---------------+
| name           | num_employees |
+----------------+---------------+
| Administration |             3 |
| Operations     |            14 |
+----------------+---------------+
2 rows in set (0.00 sec)
```

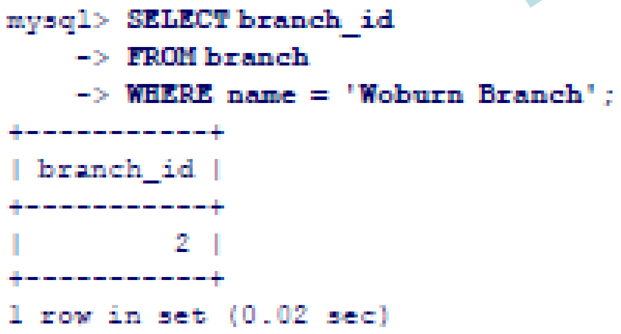
Subqueries II



```
mysql> SELECT emp_id, assigned_branch_id
    -> FROM employee
    -> WHERE start_date < '2007-01-01'
    ->   AND (title = 'Teller' OR title = 'Head Teller');
+--------+--------------------+
| emp_id | assigned_branch_id |
+--------+--------------------+
|      8 |                  1 |
|      9 |                  1 |
|     10 |                  2 |
|     11 |                  2 |
|     13 |                  3 |
|     14 |                  3 |
|     16 |                  4 |
|     17 |                  4 |
|     18 |                  4 |
+--------+--------------------+
```

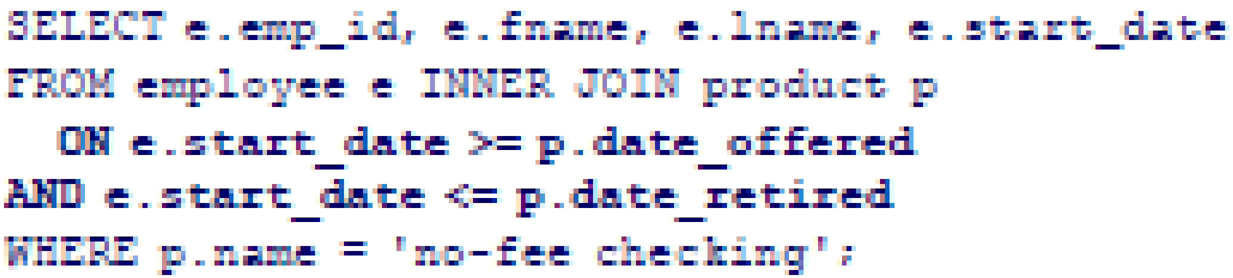


```
SELECT a.account_id, a.cust_id, a.open_date, a.product_cd
FROM account a INNER JOIN
  (SELECT emp_id, assigned_branch_id
    FROM employee
    WHERE start_date < '2007-01-01'
      AND (title = 'Teller' OR title = 'Head Teller')) e
  ON a.open_emp_id = e.emp_id
  INNER JOIN
   (SELECT branch_id
     FROM branch
     WHERE name = 'Woburn Branch') b
   ON e.assigned_branch_id = b.branch_id;
```

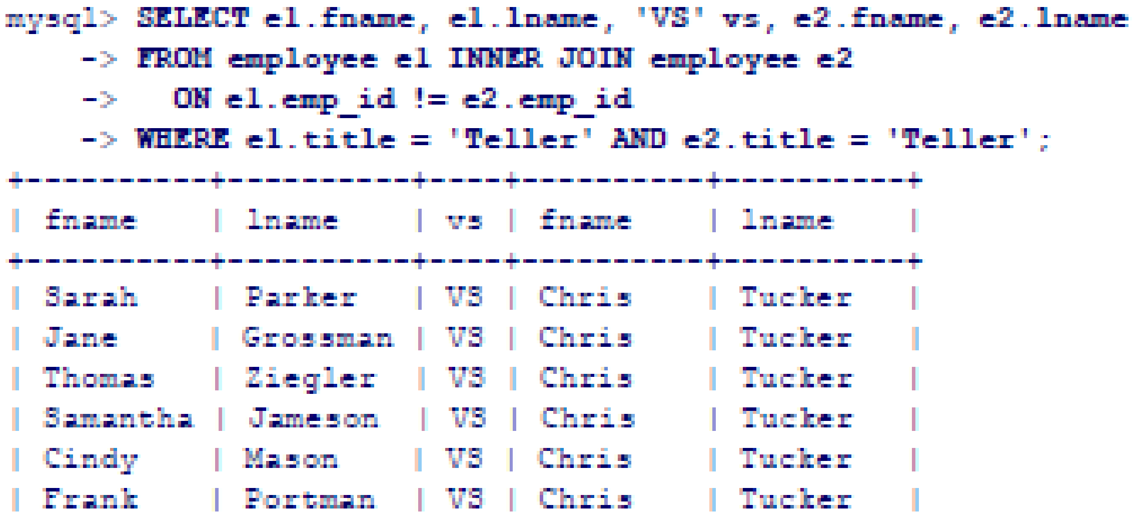


```
mysql> SELECT branch_id
    -> FROM branch
    -> WHERE name = 'Woburn Branch';
+-----------+
| branch_id |
+-----------+
|         2 |
+-----------+
1 row in set (0.02 sec)
```

Non-equi-joins

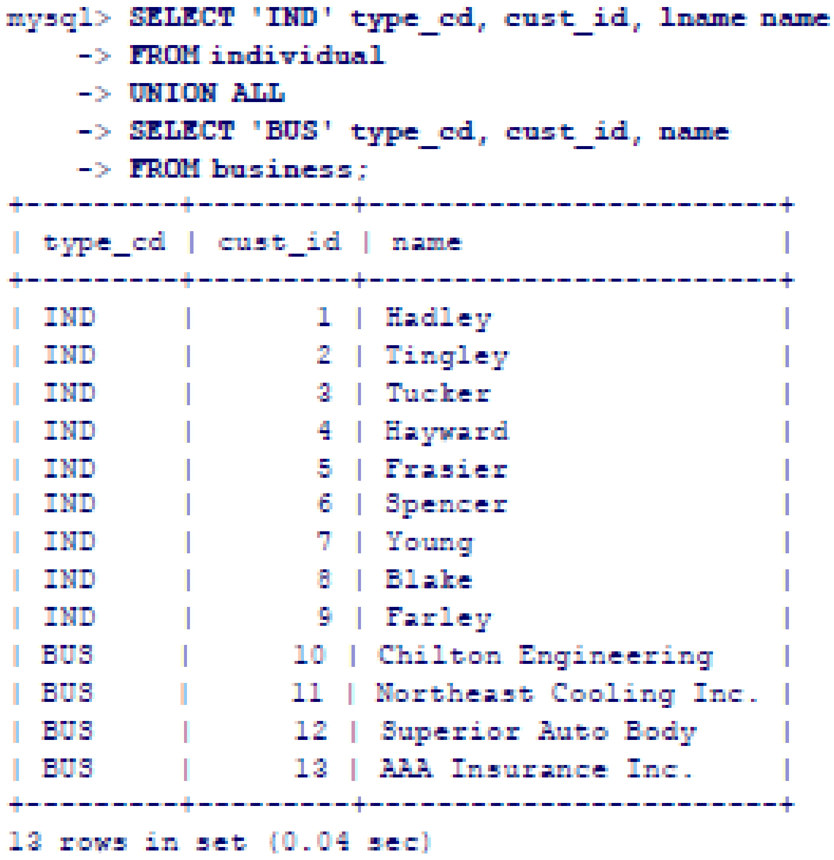


```
SELECT e.emp_id, e.fname, e.lname, e.start_date
FROM employee e INNER JOIN product p
  ON e.start_date >= p.date_offered
AND e.start_date <= p.date_retired
WHERE p.name = 'no-fee checking';
```



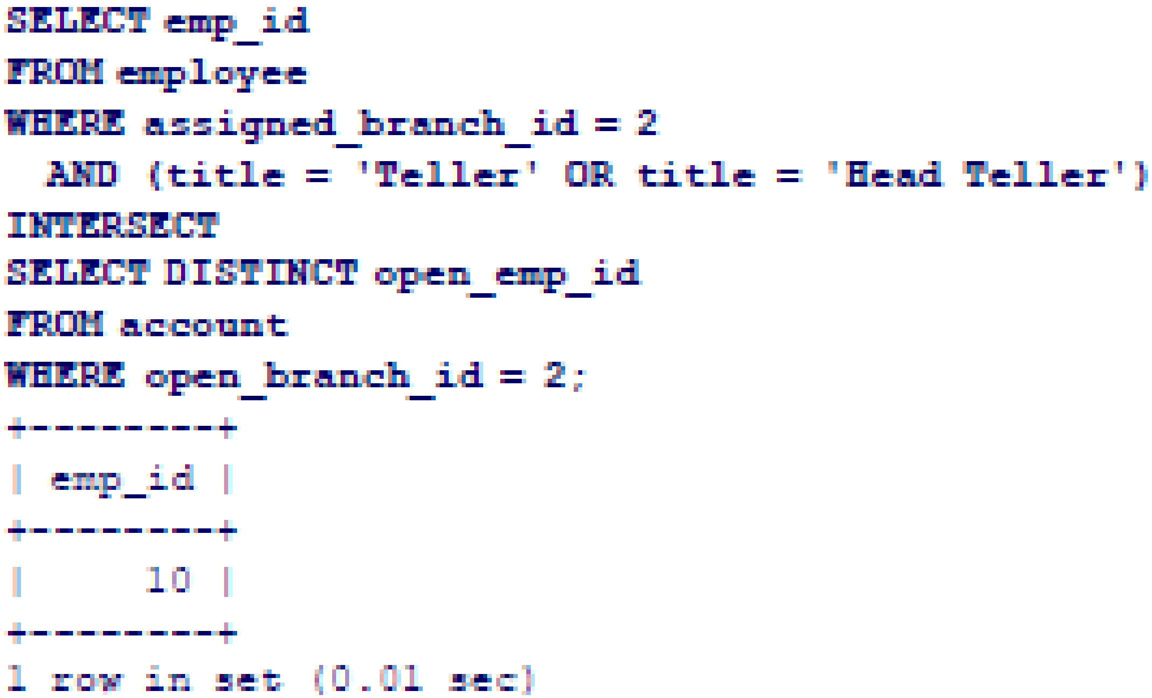
```
mysql> SELECT e1.fname, e1.lname, 'VS' vs, e2.fname, e2.lname
    -> FROM employee e1 INNER JOIN employee e2
    ->   ON e1.emp_id != e2.emp_id
    -> WHERE e1.title = 'Teller' AND e2.title = 'Teller';
+----------+----------+----+----------+----------+
| fname    | lname    | vs | fname    | lname    |
+----------+----------+----+----------+----------+
| Sarah    | Parker   | VS | Chris    | Tucker   |
| Jane     | Grossman | VS | Chris    | Tucker   |
| Thomas   | Ziegler  | VS | Chris    | Tucker   |
| Samantha | Jameson  | VS | Chris    | Tucker   |
| Cindy    | Mason    | VS | Chris    | Tucker   |
| Frank    | Portman  | VS | Chris    | Tucker   |
```

Set operations: União



```
mysql> SELECT 'IND' type_cd, cust_id, lname name
    -> FROM individual
    -> UNION ALL
    -> SELECT 'BUS' type_cd, cust_id, name
    -> FROM business;
+---------+---------+------------------------+
| type_cd | cust_id | name                   |
+---------+---------+------------------------+
| IND     |       1 | Hadley                 |
| IND     |       2 | Tingley                |
| IND     |       3 | Tucker                 |
| IND     |       4 | Hayward                |
| IND     |       5 | Frasier                |
| IND     |       6 | Spencer                |
| IND     |       7 | Young                  |
| IND     |       8 | Blake                  |
| IND     |       9 | Farley                 |
| BUS     |      10 | Chilton Engineering    |
| BUS     |      11 | Northeast Cooling Inc. |
| BUS     |      12 | Superior Auto Body     |
| BUS     |      13 | AAA Insurance Inc.     |
+---------+---------+------------------------+
13 rows in set (0.04 sec)
```

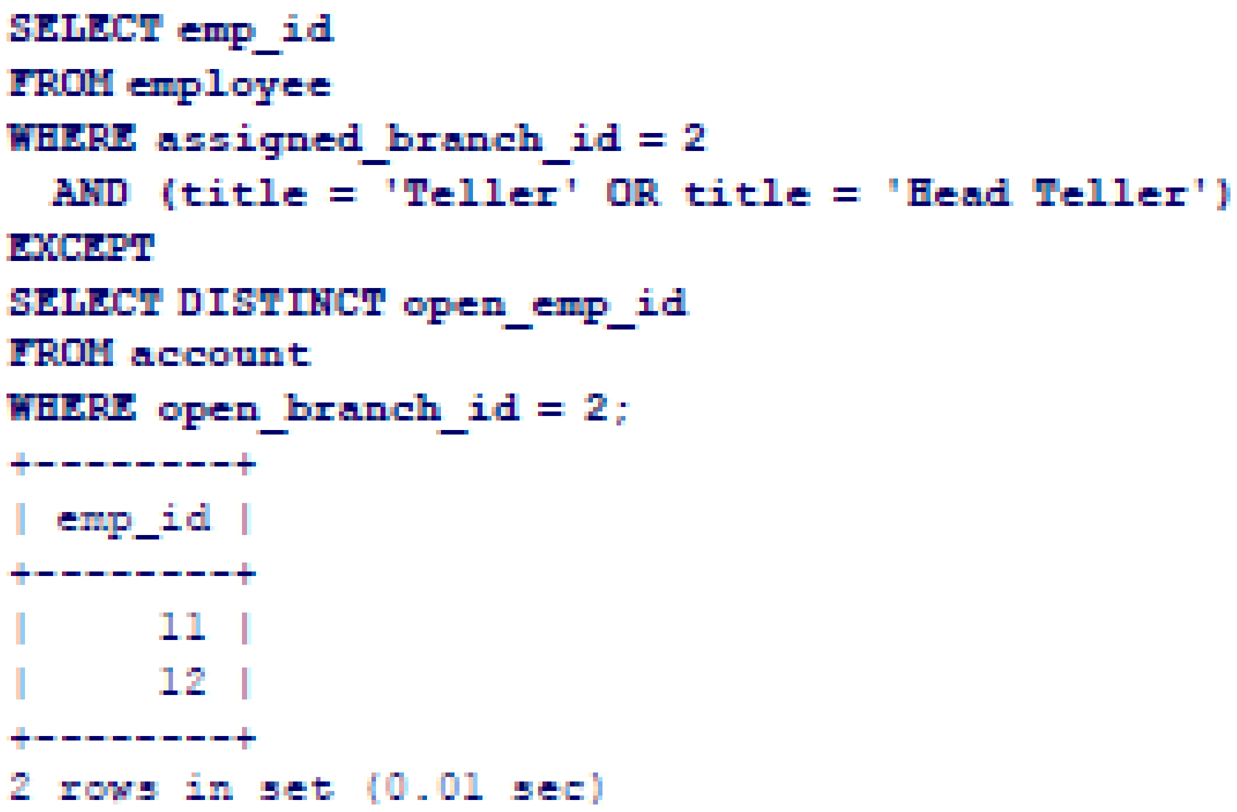
Set operations: Intersecção



```
SELECT emp_id
FROM employee
WHERE assigned_branch_id = 2
  AND (title = 'Teller' OR title = 'Head Teller')
INTERSECT
SELECT DISTINCT open_emp_id
FROM account
WHERE open_branch_id = 2;
+---------+
| emp_id |
+---------+
|      10 |
+---------+
1 row in set (0.01 sec)
```

(não implementada no MySQL)

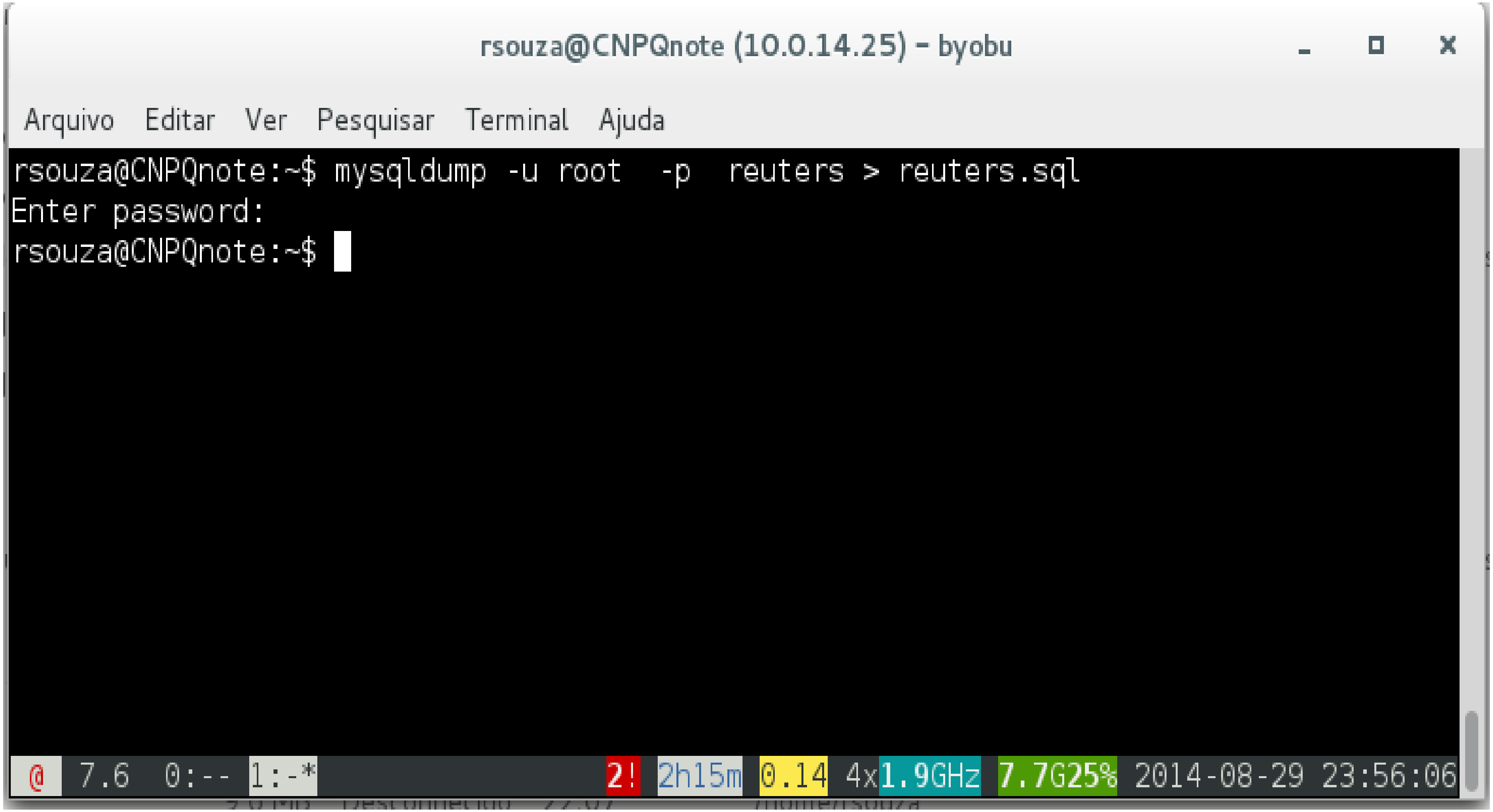
Set operations: complemento



```
SELECT emp_id
FROM employee
WHERE assigned_branch_id = 2
  AND (title = 'Teller' OR title = 'Head Teller')
EXCEPT
SELECT DISTINCT open_emp_id
FROM account
WHERE open_branch_id = 2;
+--------+
| emp_id |
+--------+
|     11 |
|     12 |
+--------+
2 rows in set (0.01 sec)
```

(não implementada no MySQL)

Fazendo backup de um banco:



rsouza@CNPQnote (10.0.14.25) - byobu
Arquivo Editar Ver Pesquisar Terminal Ajuda

```
rsouza@CNPQnote:~$ mysqldump -u root  -p  reuters > reuters.sql
Enter password:
rsouza@CNPQnote:~$
```

@ 7.6 0:-- 1:-* 2! 2h15m 0.14 4x1.9GHz 7.7G25% 2014-08-29 23:56:06

Exercícios (Banco Reuters):

Importe o Banco de Dados reuters.sql e responda:

1. Quantos registros contém $\sigma_{\text{docid=10398\_txt\_earn}}$ (frequência)
2. Quantos registros “term” possuem estas condições $\pi_{\text{term}}(\sigma_{\text{docid=10398\_txt\_earn and count=1}}$(frequência))?
3. Quantos registros retorna esta consulta $\pi_{\text{term}}(\sigma_{\text{docid=10398\_txt\_earn and count=1}}$(frequência)) U $\pi_{\text{term}}(\sigma_{\text{docid=925\_txt\_trade and count=1}}$(frequência)?
4. Quantos documentos contém a palavra “parliament”?
5. Quantos documentos possuem mais de 300 termos, incluindo duplicatas?
6. Quais documentos contém ao mesmo tempo as palavras 'transactions' e 'world‘?

Respostas (Banco Reuters):

```
1. select count(*) from Frequency where docid =
   '10398_txt_earn';

2. select count(*) from Frequency where docid =
   '10398_txt_earn' and count = 1;

3. select count(*) from
(select term from Frequency where docid =
   '10398_txt_earn' and count = 1
union all
select term from Frequency where docid =
   '925_txt_trade' and count = 1);
```

Respostas (Banco Reuters):

```
4. select count(*) from Frequency where term =
   "parliament";

5. select count(*) from
   (select docid, sum(count) from Frequency
   group by docid
   having sum(count) > 300);

6. select count(*) from Frequency a
   inner join Frequency b
   on a.docid = b.docid
   where a.term = 'transactions' and b.term =
   'world';
```

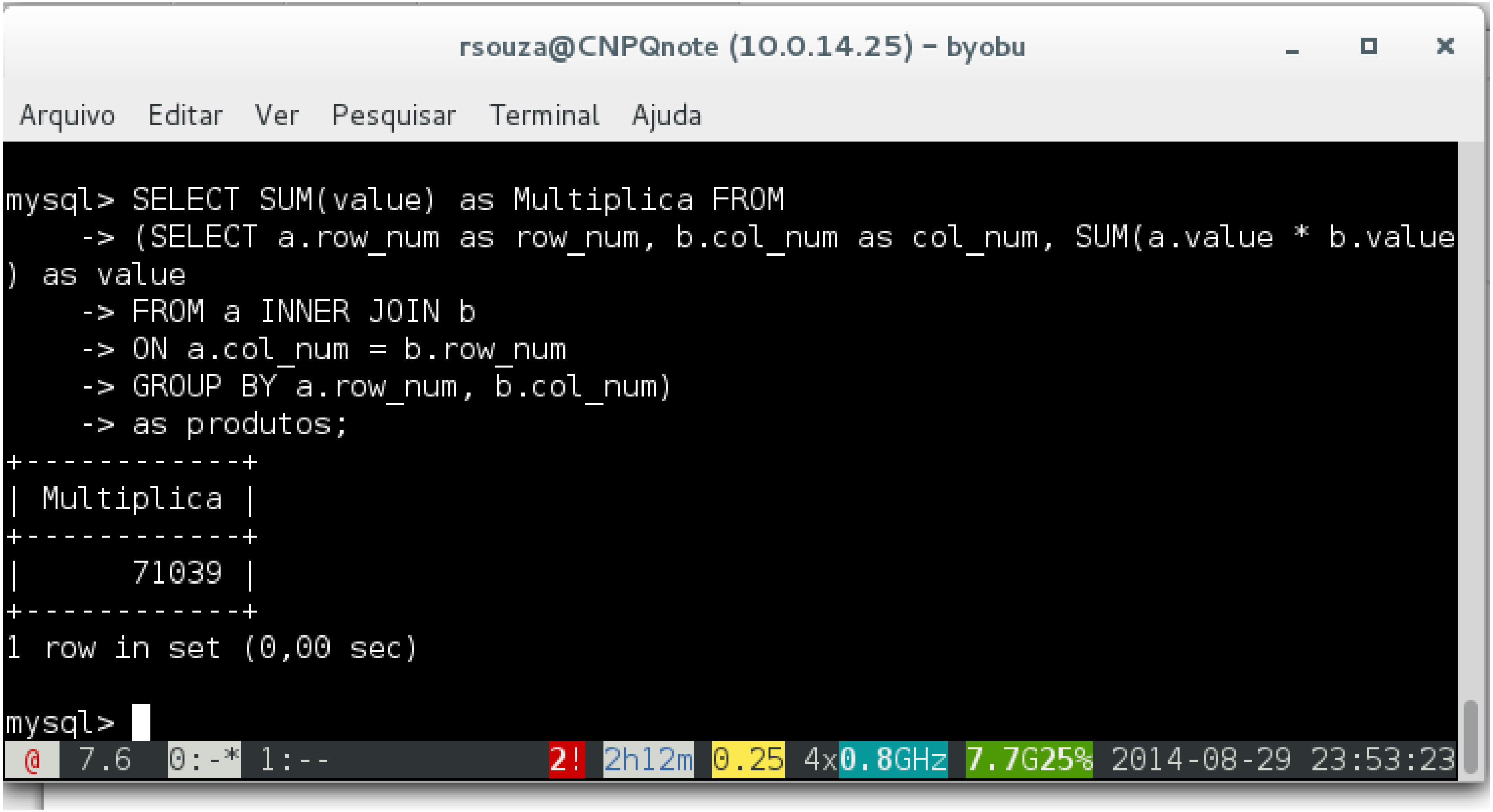
Exercício (Banco Matriz):

Importe o Banco matrix.sql e responda:

- Como multiplicar a matriz A pela matriz B em uma query?

$$(\mathbf{AB})_{ij} = \sum_{k=1}^{m} A_{ik}B_{kj}\,.$$

Resposta (Banco Matriz):


rsouza@CNPQnote (10.0.14.25) - byobu
Arquivo Editar Ver Pesquisar Terminal Ajuda

```
mysql> SELECT SUM(value) as Multiplica FROM
    -> (SELECT a.row_num as row_num, b.col_num as col_num, SUM(a.value * b.value
) as value
    -> FROM a INNER JOIN b
    -> ON a.col_num = b.row_num
    -> GROUP BY a.row_num, b.col_num)
    -> as produtos;
+------------+
| Multiplica |
+------------+
|      71039 |
+------------+
1 row in set (0,00 sec)

mysql> 
```

@ 7.6 0:-* 1:-- 2! 2h12m 0.25 4x0.8GHz 7.7G25% 2014-08-29 23:53:23