

Relatório das stored procedure do banco Sakila

1) Lista dos filmes, e seus respectivos clientes, que foram devolvidos, mas não foram pagos.

```
Delimiter $$
drop procedure if exists devolvido_naopago $$
create procedure devolvido_naopago()
begin
select f.titulo, c.primeiro_nome from aluguel al, filme f, cliente c, inventario i where
al.data_de_devolucao is not null and al.aluguel_id not in (select p.aluguel_id from
pagamento p) and f.filme_id=i.filme_id and i.inventario_id = al.inventario_id and
c.cliente_id=al.cliente_id;
end $$
delimiter ;0
```

✓ MySQL não retornou nenhum registro. (A consulta demorou 0.0381 segundos.)

```
call devolvido_naopago()
```

[Mostrar Caixa do query](#)

2) Lista de atores relacionados ao filme mais locado desde o início do banco de dados.

```
delimiter $$
drop procedure if exists atores_filme $$
create procedure atores_filme()
begin
declare aux int;
select i.filme_id into aux from aluguel a, inventario i where a.inventario_id = i.inventario_id
group by i.filme_id order by count(i.filme_id) DESC limit 1;

select concat(a.primeiro_nome, " ", a.ultimo_nome) as "Nome de atores" from filme_ator f,
ator a where f.ator_id=a.ator_id and f.filme_id=aux;
```

```
end $$  
delimiter ;
```

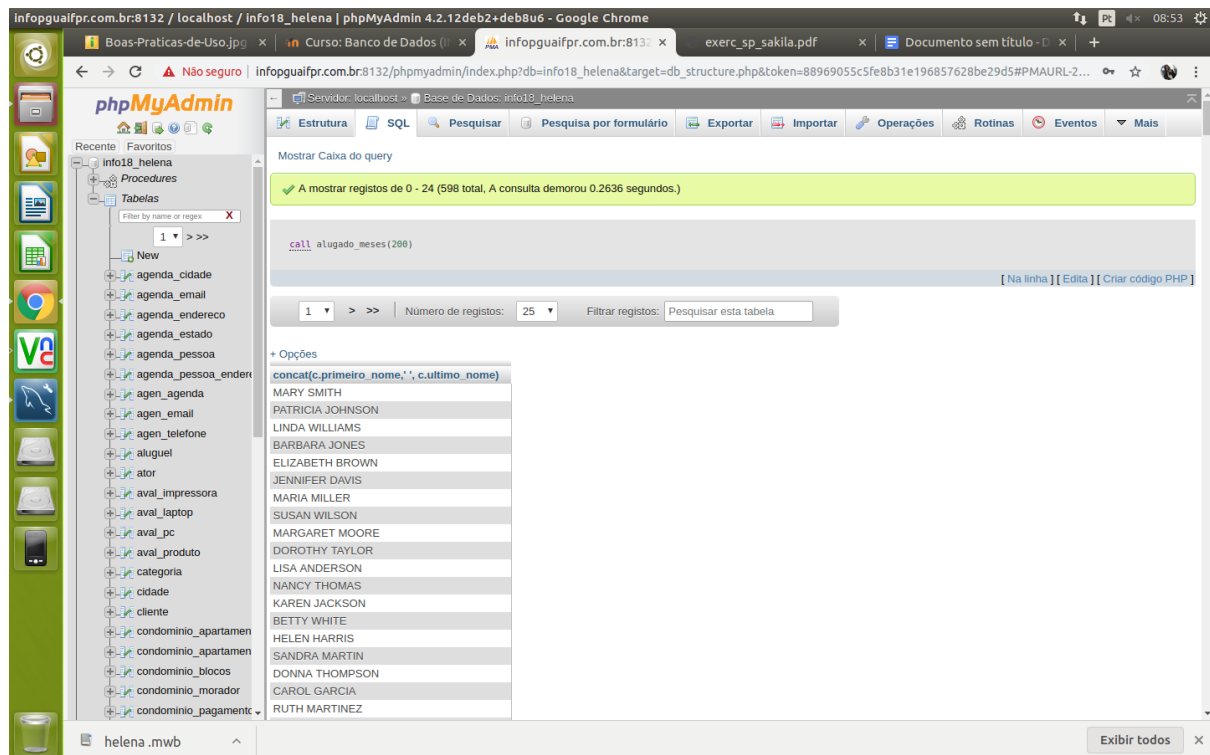


The screenshot shows a database interface. At the top, there is a text input field containing the SQL command `call atores_filme();`. Below this, there is a control bar with two elements: a label 'Número de registros:' followed by a dropdown menu showing the value '25', and a label 'Filtrar registros:' followed by a text input field containing 'Pesquisar esta tabela'. Below the control bar, there is a section titled '+ Opções' which contains a table with actor names.

Nome de atores
RIP CRAWFORD
TIM HACKMAN
GARY PHOENIX
CHARLIZE DENCH
KIRSTEN AKROYD
BURT TEMPLE

3) Lista de clientes que não alugam filmes a mais de X meses.

```
delimiter $$  
drop procedure if exists alugado_meses $$  
create procedure alugado_meses(in aux int)  
begin  
    declare data_referencia datetime;  
    select date_sub(curdate(), interval aux month) into data_referencia;  
  
    select concat(c.primeiro_nome, ' ', c.ultimo_nome) from cliente c, aluguel a where  
    c.cliente_id=a.aluguel_id and data_referencia < a.data_de_aluguel;  
  
end $$  
delimiter ;
```



4) Quanto foi recebido em um determinado mês por categoria de filme.

delimiter \$\$

drop procedure if exists recebido_categoria \$\$

create procedure recebido_categoria(in aux int)

begin

```
select c.categoria_id, c.nome, SUM(p.valor) as total_valor from pagamento p, categoria c,
filme_categoria fc, filme f, inventario i, aluguel al where c.categoria_id=fc.categoria_id and
fc.filme_id=f.filme_id and f.filme_id=i.filme_id and i.inventario_id=al.inventario_id and
al.aluguel_id=p.aluguel_id and MONTH(data_de_pagamento)=aux group by c.categoria_id ;
end $$
```

delimiter ;

✓ A mostrar registros de 0 - 15 (16 total, A consulta demorou 0.0660 segundos.)

```
call recebido_categoria(2)
```

Número de registros:

25 ▾

Filtrar registros:

+ Opções

categoria_id	nome	total_valor
1	Action	55.80
2	Animation	66.75
3	Children	15.94
4	Classics	24.89
5	Comedy	25.90
6	Documentary	7.96
7	Drama	29.92
8	Family	29.90
9	Foreign	24.89
10	Games	39.88
11	Horror	33.88
12	Music	24.91
13	New	30.89
14	Sci-Fi	30.91
15	Sports	40.85
16	Travel	30.91

5) Relação de filmes onde o idioma do filme é igual ao idioma original.

```
delimiter $$
```

```
drop procedure if exists idioma_original $$
```

```
create procedure idioma_original()
```

```
begin
```

```
    select filme_id, titulo from filme where idioma_id=idioma_original_id;
```

```
end $$
```

```
delimiter ;
```

✓ MySQL não retornou nenhum registro. (A consulta demorou 0.0005 segundos.)

```
call idioma_original()
```

Mostrar Caixa do query

6) Qual o nome do funcionário que mais realizou locações em um determinado período.

delimiter \$\$

drop procedure if exists funcionario_periodo \$\$

create procedure funcionario_periodo(in data_inicial date, in data_final date)

begin

select concat(f.primeiro_nome,' ',f.ultimo_nome) as "Nome do funcionario", count(*) as
"Numero de locacoes" from funcionario f , aluguel a where f.funcionario_id = a.funcionario_id
and data_inicial <= a.data_de_aluguel and a.data_de_aluguel <= data_final group by
(f.funcionario_id) order by count(*) desc limit 1;

end \$\$

delimiter ;

✓ A mostrar registos de 0 - 0 (1 total, A consulta demorou 0.0007 segundos.)

```
call funcionario_periodo('2005-05-24', '2005-05-25')
```

Número de registos: 25 ▾ Filtrar registos:

+ Opções

Nome do funcionario	Numero de locacoes
Mike Hillyer	5

7) Retorne o número de filmes disponíveis em uma determinada loja.

```

Delimiter $$
drop procedure if exists loja_disponivel $$
create procedure loja_disponivel(in loja TINYINT)
begin
    select count(*) as 'Numero de filmes' from inventario i where i.loja_id=loja;
end $$
delimiter ;

```

✓ A mostrar registos de 0 - 0 (1 total, A consulta demorou 0.0018 segundos.)

```
call loja_disponivel(1)
```

Número de registos: 25 ▾ Filtrar registos:

+ Opções

filmes_disponiveis
2270

8) Lista de filmes e respectivos clientes que não devolveram os filmes na data prevista.

```

Delimiter $$
drop procedure if exists filmes_atrasados $$
create procedure filmes_atrasados(in max_dias int)
begin
    select f.titulo as Filme, concat(c.primeiro_nome, ' ', c.ultimo_nome) as Cliente from
    filme f, cliente c, aluguel a, inventario i where f.filme_id=i.filme_id and
    a.inventario_id=i.inventario_id and c.cliente_id=a.cliente_id and a.data_de_devolucao is not
    null and datediff(a.data_de_devolucao, a.data_de_aluguel) >= max_dias;
end $$
delimiter ;

```

9) Número de clientes ativos, mês a mês, desde a implantação do banco de dados.

```

Delimiter $$
drop procedure if exists cliente_ativo $$
create procedure cliente_ativo()
begin
    select extract(month from data_criacao) as mes, count(*) from cliente where ativo=1
    group by(extract(month from data_criacao)) order by (extract(month from data_criacao));
end $$
delimiter ;

```

✓ A mostrar registros de 0 - 0 (1 total, A consulta demorou 0.0007 segundos.)

```
call cliente_ativo()
```

Número de registros:

25 ▼

Filtrar registros:

Pesquisar esta tabela

+ Opções

mes	count(*)
2	584

10) Lista de e-mails dos clientes que gostam de uma determinada categoria de filme.

delimiter \$\$

drop procedure if exists email_categoria \$\$

create procedure email_categoria(in categoria varchar(50))

begin

select c.email from cliente c, aluguel a, inventario i, filme_categoria fcat, categoria cat, filme f where c.cliente_id=a.cliente_id and i.inventario_id=a.inventario_id and fcat.categoria_id=cat.categoria_id and fcat.filme_id=f.filme_id and f.filme_id=i.filme_id and cat.nome like categoria;

end \$\$

delimiter ;

✓ A mostrar registros de 0 - 24 (1112 total, A consulta demorou 0.0030 segundos.)

```
call email_categoria('Action')
```

1 > >> | Número de registros: 25 | Filtrar registros:

+ Opções

email	primeiro_nome
JANE.BENNETT@sakilacustomer.org	JANE
DEBRA.NELSON@sakilacustomer.org	DEBRA
REBECCA.SCOTT@sakilacustomer.org	REBECCA
MAXINE.SILVA@sakilacustomer.org	MAXINE
BRANDY.GRAVES@sakilacustomer.org	BRANDY
JESSICA.HALL@sakilacustomer.org	JESSICA
JULIA.FLORES@sakilacustomer.org	JULIA
VERNON.CHAPA@sakilacustomer.org	VERNON
KATHLEEN.ADAMS@sakilacustomer.org	KATHLEEN
HECTOR.POINDEXTER@sakilacustomer.org	HECTOR
SAMANTHA.DUNCAN@sakilacustomer.org	SAMANTHA
NATHANIEL.ADAM@sakilacustomer.org	NATHANIEL
SHERRY.MARSHALL@sakilacustomer.org	SHERRY
BARBARA.JONES@sakilacustomer.org	BARBARA
EDUARDO.HIATT@sakilacustomer.org	EDUARDO