

use sakila;

-- Aula 07

-- UNION -> (vai unir consultas, elimina as colunas repetidas na união, se usar "UNION ALL" ele traz tudo das 2 consultas (repetindo coluna se for o caso)

-- HAVING -> (filtra consulta, não filtra campo (se usa o where nesse caso) e ele filtra a função de agregação(função de agregação(SUM(), COUNT(), MAX(), MIN(), AVG()))

-- Exemplo de consulta sem UNION

```
select count(title) qtde, name as categoria from film join film_category using (film_id) join
category using (category_id) where name in ('Drama', 'Documentary', 'Action') and length >
100
and rental_duration between 2 and 4 and replacement_cost > 20.00 group by name;
```

```
select count(title) qtde, name as categoria from film join film_category using (film_id) join
category using (category_id) where name in ('Comedy', 'Classics', 'Horror')
and length > 100 and rental_duration between 2 and 4 and replacement_cost > 20.00 group
by name;
```

-- Exemplo da mesma consulta com o UNION

```
select count(title) qtde, name as categoria from film join film_category using (film_id) join
category using (category_id) where name in ('Drama', 'Documentary', 'Action') and length >
100
and rental_duration between 2 and 4 and replacement_cost > 20.00 group by name
UNION
select count(title) qtde, name as categoria from film join film_category using (film_id) join
category using (category_id) where name in ('Comedy', 'Classics', 'Horror')
and length > 100 and rental_duration between 2 and 4 and replacement_cost > 20.00 group
by name;
```

```
select distinct year(payment_date) from payment;
```

/*

Crie uma consulta que irá unificar as duas consultas abaixo: (usando UNION)

Consulta 1: Retornar a quantidade dos clientes inativo, que tiveram pagamento em 2005 e que são residentes nos países que começam com a letra A

Consulta 2: Retornar a quantidade dos clientes ativo, que tiveram pagamento em 2005 e são residentes nos países com o critério %B_A%

*/

```
select * from city;
select * from address;
```

```
select * from customer;
select * from payment;
select * from country;
```

```
select ifnull(count(customer_id),0) as qtde, 'Inativo' as tipo_cliente from country join city
using (country_id) join address using (city_id) join customer using (address_id) join payment
using (customer_id)
where year (payment_date) = 2005 and country like "A%" and active = 0 group by
tipo_cliente
UNION ALL
select count(customer_id) as qtde, 'Inativo' as tipo_cliente from country join city using
(country_id) join address using (city_id) join customer using (address_id) join payment using
(customer_id)
where year (payment_date) = 2005 and country like "%B_A%" and active = 1 group by
tipo_cliente;
```

-- Exemplo de HAVING

```
use classicmodels;
```

-- Retorna os vendedores que atenderam mais de 1 cliente

```
select concat(firstname, ' ', lastname) as vendedor, count(customernumber) as qtde from
customers join employees on (salesRepEmployeeNumber = EmployeeNumber) group by
vendedor
having
count(customernumber) > 1;
```

-- Crie uma consulta que retorne a relação de produtos que foram vendidos em 2004, no primeiro trimestre. Selecione aqueles que tiveram faturamento superior a 50.000 dolares

```
select productname, sum(quantityordered * priceeach) as faturamento from products join
orderdetails using (productcode) join orders using (ordernumber) join customers using
(customernumber)
join payments on (customers.customernumber = payments.customernumber) where
year(paymentdate) = 2004 and month(paymentdate) between 1 and 3 group by
productname having faturamento > 50000;
```