**Skill Track – SQL Fundamentals**

Aprenda as habilidades fundamentais necessárias para interagir com a maioria dos dialetos do SQL – PostgreSQL (usada nesse curso), MySQL, Oracle... As habilidades que você aprende com esses cursos o capacitarão a ingressar em tabelas, resumir dados e responder a suas perguntas sobre análise de dados e ciência de dados.

- 4 cursos, 17h:

**a) Introduction to SQL;**

**b) Joining Data in SQL;**

**c) Intermediate SQL;**

**d) Exploratory Data Analysis in SQL.**

1. **Introduction to SQL**

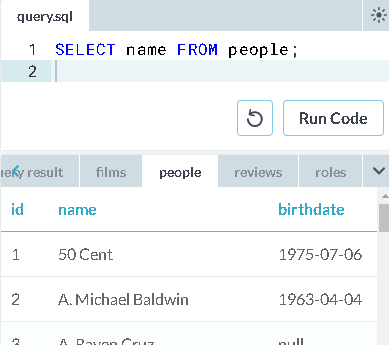
O papel de um cientista de dados é transformar dados brutos em insights. Muitos dos dados brutos do mundo - desde registros médicos eletrônicos até históricos de transações com clientes - vivem em coleções organizadas de tabelas chamadas **bancos de dados relacionais**. Para ser um cientista de dados eficaz, você deve saber como **agrupar e extrair dados** desses bancos de dados usando uma linguagem chamada SQL. Este curso ensina a **sintaxe em SQL** compartilhada por muitos tipos de bancos de dados, como PostgreSQL, MySQL, SQL Server e Oracle. Este curso ensina tudo o que você precisa saber para começar a trabalhar com bancos de dados hoje.

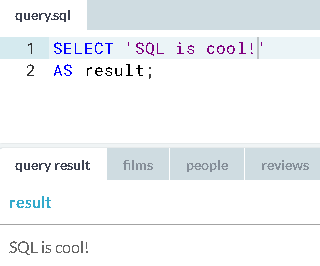
1. Selecting columns: este capítulo fornece uma breve introdução ao trabalho com bancos de dados relacionais. Você aprenderá sobre sua estrutura, como falar sobre eles usando a linguagem do banco de dados e como iniciar uma análise usando comandos SQL simples para selecionar e resumir colunas das tabelas do banco de dados.;
2. Filtering rows: este capítulo se baseia no primeiro, ensinando a filtrar tabelas por linhas que atendem a alguns critérios de interesse. Você aprenderá a usar operadores básicos de comparação, combinar vários critérios, combinar padrões em texto e muito mais.;
3. Aggregate Functions: Este capítulo ensina como usar funções agregadas para resumir dados e obter informações úteis. Você também aprenderá sobre aritmética no SQL e como usar aliases para tornar seus resultados mais legíveis.;
4. Sorting and grouping: este capítulo fornece uma breve introdução à classificação e agrupamento de seus resultados.

SQL (Structured Query Language) ou “Linguagem de Consulta Estruturada” é a linguagem nativa para interagir com dados armazenados em banco de dados relacional.

Para este curso vamos usar um dataset do IMDB contendo infos de quase 5 mil filmes: <https://assets.datacamp.com/production/repositories/641/datasets/d39d51ab6615b8a7f36681ee88490978fb095193/films.sql> .

1. **SELECTing columns**

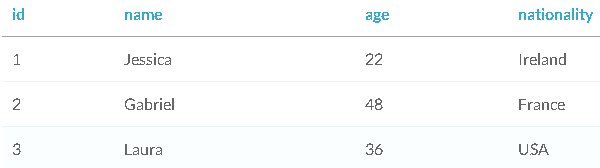




Você pode pensar em um banco de dados relacional como uma **coleção de tabelas**. Uma tabela é apenas um **conjunto de linhas e colunas**, como uma planilha, que representa exatamente um tipo de entidade. Por exemplo, uma tabela pode representar funcionários de uma empresa ou compras feitas, mas não ambas.

**Cada linha ou registro de uma tabela contém informações sobre uma única entidade**. Por exemplo, em uma tabela que representa funcionários, **cada linha representa uma única pessoa**. **Cada coluna ou campo de uma tabela contém um único atributo** para todas as linhas da tabela. Por exemplo, em uma tabela que representa funcionários, podemos ter uma coluna contendo o nome e o sobrenome de todos os funcionários.

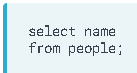
A tabela de funcionários contém 4 campos:



**SELECTing single columns**

Embora o SQL possa ser usado para criar e modificar bancos de dados, o foco deste curso será a consulta a bancos de dados. Uma **consulta é uma solicitação de dados de uma tabela de banco de dados (ou combinação de tabelas)**. A consulta é uma habilidade essencial para um cientista de dados, pois os dados que você precisa para suas análises geralmente ficam em bancos de dados.

No SQL, você pode selecionar dados de uma tabela usando uma instrução SELECT. Por exemplo, a seguinte consulta seleciona a coluna de nome da tabela de pessoas. Veja que as palavras-chave do SQL não são case sensitive, tanto faz usar em letra maiúscula ou minúscula. Por covenção e boas práticas, usamos em letra maiúscula (SELECT , FROM) p/ distingui-las de outras partes da consulta (nome da coluna e nome da tabela) e para melhorar a leitura:

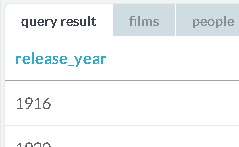
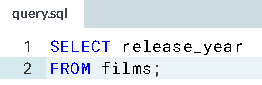


SELECIONE a **coluna** name DA **tabela** people



Também inclua um “ponto e vírgula” ao final da query. Isso diz ao SQL onde termina a sua query.

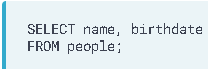
Ex: Selecione a coluna release-year da tabelma films.



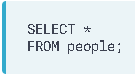
**SELECTing multiple columns**

No mundo real, você frequentemente desejará selecionar várias colunas. O SQL facilita muito isso. Para selecionar várias colunas de uma tabela, basta separar os nomes das colunas com vírgulas.

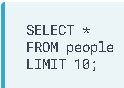
Por exemplo, esta consulta seleciona duas colunas, nome e data de nascimento, na tabela de pessoas:



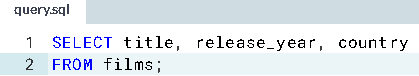
P/ selecionar todas as colunas da tabela, use “\*”, assim vc ñ precisa digitar os nomes de todas as colunas.



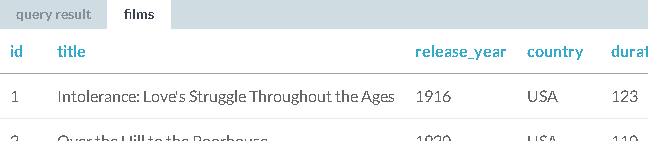
P/ retornar apenas um certo número de resultados, use a palavra-chave LIMIT para limitar o número de linhas retornada:



Ex.: pegue as colunas title, release\_year e country da tabela filmes:



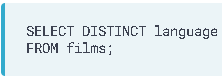




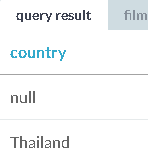
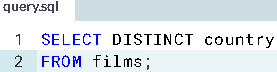
**SELECT DISTINCT**

Frequentemente seus resultados terão valores duplicados. Se você quer selecionar todos os valores únicos de uma coluna, use a palavra-chave DISTINCT.

Isso é útil se, por ex., vc estiver interessado em saber quais línguas estão representadas na tabela films:



Ex.: Pegue todos os países únicos representados na tabela films.

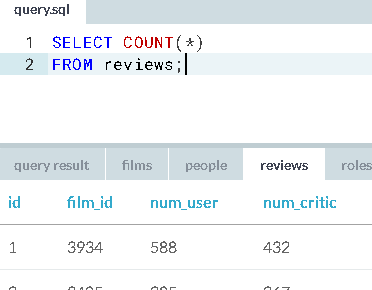


**Learning to COUNT**

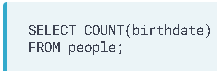
P/ contar o número de funcionários na sua tabela de funcionários use COUNT. COUNT **retorna o número de linhas de uma ou mais colunas**.

Ex. qntas linhas tem na tabela people: selecione e conte todas as linhas da tabela people

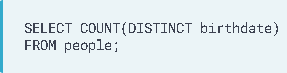




P/ contar o nº valores não ausentes de uma coluna, use COUNT apenas nessa coluna. Ex. p/ contar o nº de birth dates presente na tabela people: selecione e conte o numero de linhas da coluna birthdate da tabela people



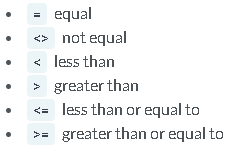
Tbm podemos combinar COUNT com DISTINCT p/ contar o número de valores distintos em uma coluna. Ex. essa consulta conta o nº de birth dates distintas na tabela people:



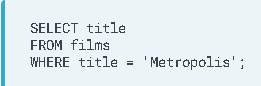
1. **Filtering Results**

WHERE – permite filtrar com base nos valores de texto e numéricos de uma tabela.

Os operadores de comparação q podemos usar:

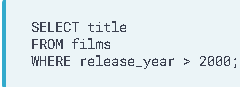


Ex.: filtrar todos os textos de title p/ retornar todos os filmes com o título “Metrópolis”:



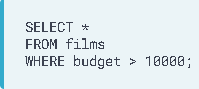
WHERE sempre vem depois do FROM. E vamos usar <> ao invés de !=, pois é o padrão SQL.

Ex. 2: filmes divulgados depois do ano 2000.

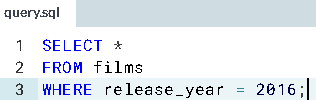


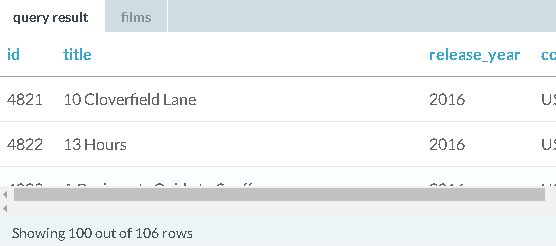
**Filtro simples de valores numéricos**

Ex. query q seleciona todos os detalhes de filmes com orçamento maior q $10k:

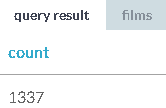
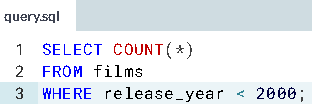


Ex. 2: Pegue todos os detalhes de todos os filmes lançados em 2016:

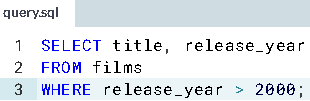


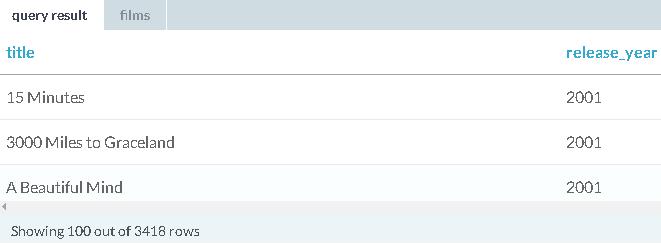


Ex. 3: Pegue o número de filmes lançados antes de 2000:



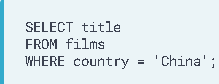
Ex. 4: Pegue o title e release year de filmes lançados depois de 2000:





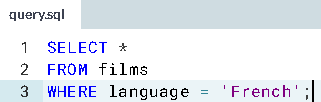
**Filtro simples de texto**

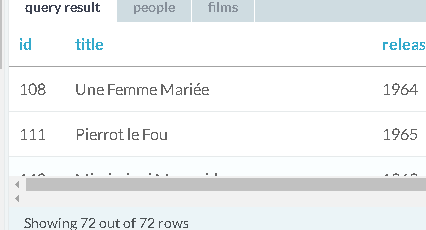
Ex. 1: essa query pega os títulos de todos os filmes q foram filmados na China:



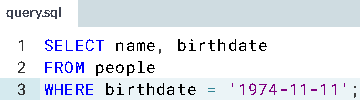
IMP!: Na versão PostgreSQL, q estamos usando, temos q usar aspas simples nas strings com WHERE.

Ex. 2: Pegue todos os detalhes de todos os filmes de língua francesa:



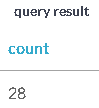
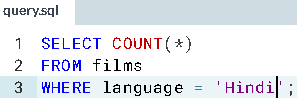


Ex. 3: Pegue o nome e data de nascimento da pessoa nascida em 11 de novembro de 1974. Lembre-se de usar o dateformat ISSO (‘1974-11-11’):

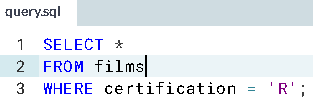


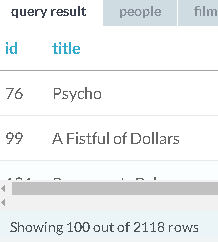


Ex. 4: Pegue o número de filmes de língua Hindu:



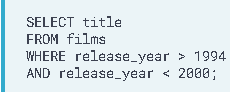
Ex. 5: Pegue todos os detalhes de todos os filmes com R certification:





**WHERE AND**

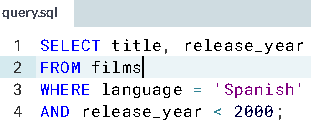
Frequentemente vc pode querer selecionar dados baseado em múltiplas condições. Você pode criar suas consultas WHERE combinando várias condições com a palavra-chave AND. Vc pode adicionar quantos AND quiser. Ex.: pega os títulos de filmes lançados entre 1994 e 2000:

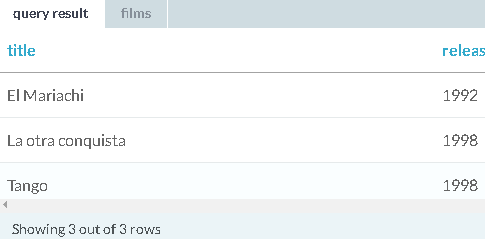


Note que vc precisa especificar o nome da coluna para cada condição. Abaixo está a maneira incorreta de fazer esse filtro:

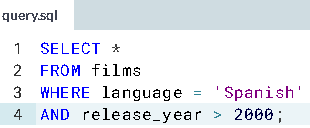


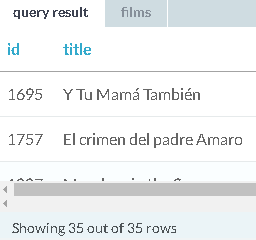
Ex. 1: Pegue o título e release year de todos os filmes de língua espanhola lançados antes de 2000:



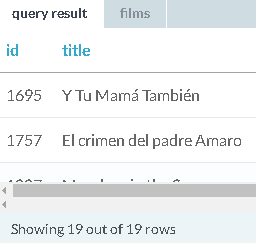
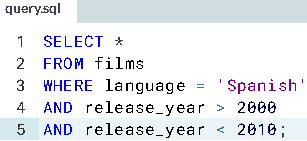


Ex. Pegue tdos detalhes dos filmes de língua espanhola lançados depois de 2000:



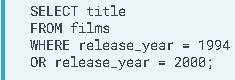


Ex. pegue tdos os detalhes de filmes de língua espanhola lançados depois de 2000 e antes de 2010:

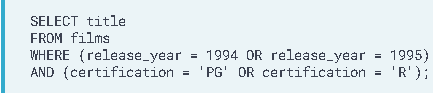


**WHERE AND OR**

P/ selecionar linhas com base em várias condições em que algumas, mas não todas, precisam ser atendidas, use OR. Por exemplo, o seguinte retorna todos os filmes lançados em 1994 ou 2000:

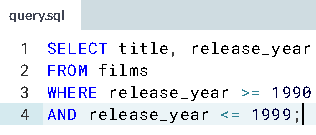


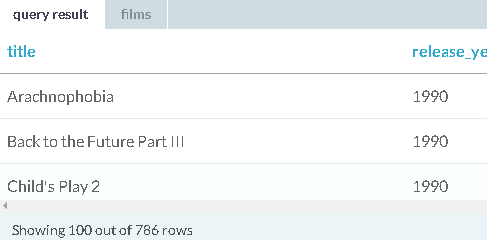
Ao combinar AND e OR, inclua as cláusulas individuais entre parênteses p/ conseguir o resultado desejado, da seguinte maneira:



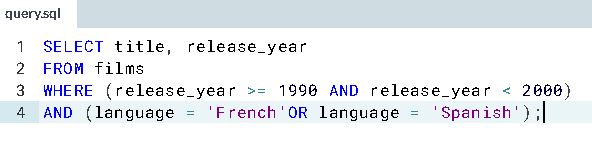
Ex.1: Escreva uma consulta para obter o título e o ano de lançamento dos filmes lançados nos anos 90, em francês ou espanhol e que arrecadaram mais de US $ 2 milhões.

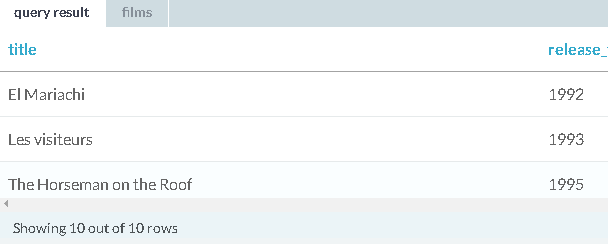
Primeiro, pegue o título e ano de lançamento dos filmes lançados em 90:



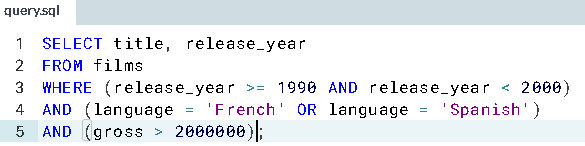


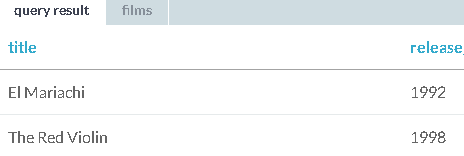
Segundo, faça uma consulta para filtrar os registros e incluir apenas filmes de língua espanhola e francesa:





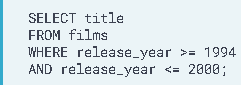
Terceiro, restrinja a consulta a retornar apenas filmes que tenham recebido mais de US $ 2 milhões.



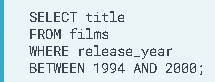


**BETWEEN**

Podemos usar a seguinte consulta para obter títulos de todos os filmes lançados em 1994 e 2000:



A verificação de intervalos como esse é muito comum; portanto, no SQL, a palavra-chave BETWEEN fornece um atalho útil para filtrar valores dentro de um intervalo especificado. Esta consulta é equivalente à acima:

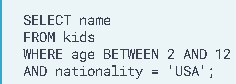


IMP!: BETWEEN é inclusivo, ou seja, os valores inicial e final estão incluídos nos resultados!

**BETWEEN – parte 2**

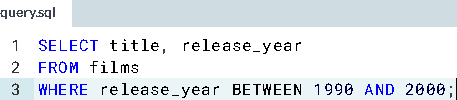
Semelhante à cláusula WHERE, a cláusula BETWEEN pode ser usada com vários operadores AND e OR, para que você possa criar suas consultas e torná-las ainda mais poderosas!

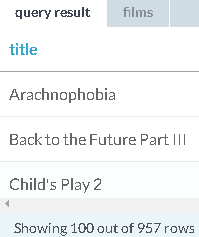
Por exemplo, suponha que tenhamos uma tabela chamada kids. Podemos obter o nome de todas as crianças entre 2 e 12 anos nos Estados Unidos:



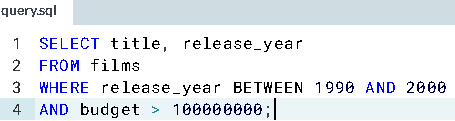
Experimente usar BETWEEN com AND nos dados dos filmes para obter o título e o ano de lançamento de todos os filmes em espanhol lançados entre 1990 e 2000 (inclusive) com orçamentos acima de US $ 100 milhões. Dividimos o problema em etapas menores para que você possa criar a consulta à medida que avança.

Ex.1: pegue o título e ano de lançamento de todos os filmes lançados entre 1990 e 2000 -inclusivo:

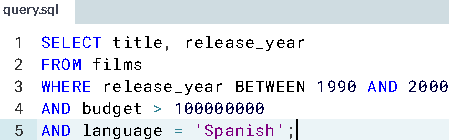


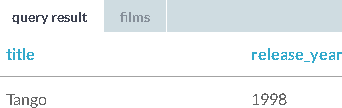


Ex. 2: Agora, aproveite sua consulta anterior para selecionar apenas filmes com orçamentos acima de US $ 100 milhões.

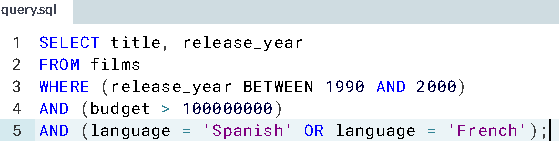


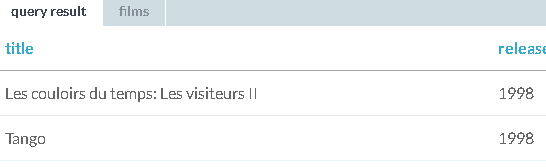
Ex. 3: Agora restrinja a consulta a retornar apenas filmes de língua espanhola:





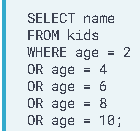
Ex.4: agora modifique a sua consulta anterior p/ incluir todos os filmes de língua espanhola ou francesa com o mesmo critério de antes:



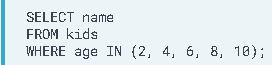


**WHERE IN**

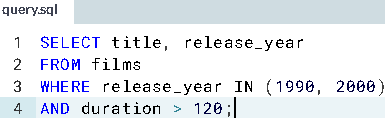
Como você viu, WHERE é muito útil para filtrar resultados. No entanto, se você deseja filtrar com base em várias condições, WHERE pode ficar pesado. Por exemplo:

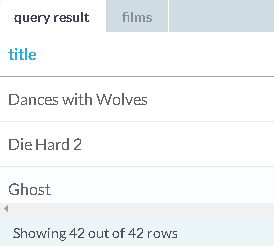


Use o operador IN. O operador IN permite especificar vários valores em uma cláusula WHERE, tornando mais fácil e rápido especificar várias condições OR! O exemplo acima se tornaria simplesmente:

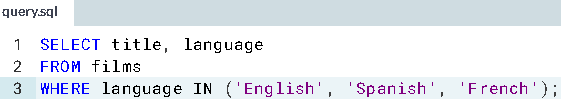


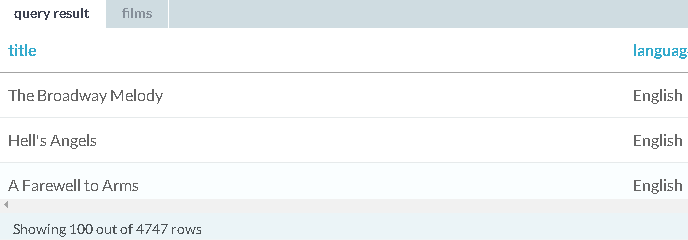
Ex.1: pegue o título e ano de lançamento de todos os filmes lançados em 1990 ou 2000 que tem mais de 2h (duração em minutos):



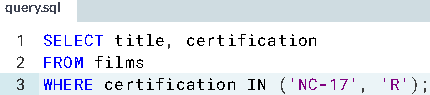


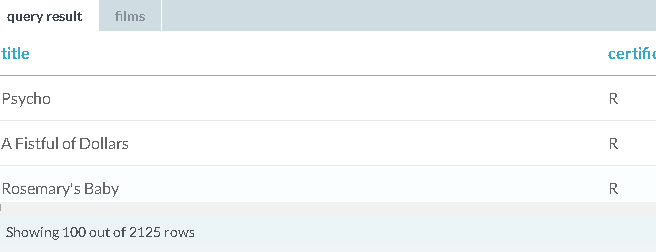
Ex. 2: pegue o título e linguagem de todos os filmes que estão em inglês, espanhol e francês:





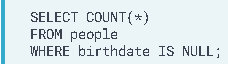
Ex. 3: pegue o título e certificação de todos os filmes com certificado NC-17 e R:





**Introdução ao NULL e IS NULL**

No SQL, NULL representa um valor ausente ou desconhecido. Você pode verificar valores NULL usando a expressão IS NULL. Por exemplo, para contar o número de datas de nascimento ausentes na tabela de pessoas:



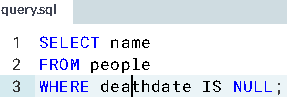
Como você pode ver, IS NULL é útil quando combinado com WHERE para descobrir quais dados estão faltando.

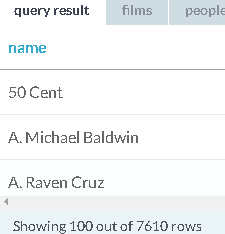
Às vezes, você deseja filtrar os valores ausentes para obter apenas resultados que não sejam NULL. Para fazer isso, você pode usar o operador IS NOT NULL.

Por exemplo, esta consulta fornece os nomes de todas as pessoas cujas datas de nascimento não estão ausentes na tabela de pessoas:

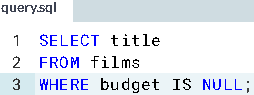


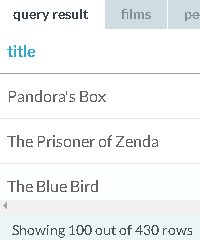
Ex.1: obtenha os nomes das pessoas que ainda estão vivas, ou seja, cujas datas de morte estão vazias:



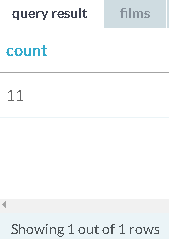
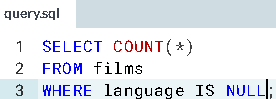


Ex. 2: obtenha o título de todos os filmes que não possuem um budget associado:





Ex. 3: obtenha o número de filmes que não possuem linguagem associada a eles:

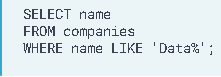


**LIKE and NOT LIKE**

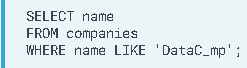
Como você viu, a cláusula WHERE pode ser usada para filtrar dados de texto. No entanto, até agora você só conseguiu filtrar especificando o texto exato em que está interessado. No mundo real, muitas vezes você deseja procurar um padrão em vez de uma sequência de texto específica.

No SQL, o operador LIKE pode ser usado em uma cláusula WHERE para procurar um padrão em uma coluna. Para fazer isso, use algo chamado curinga como espaço reservado para outros valores. Existem dois curingas que você pode usar com o LIKE:

O curinga % corresponderá a zero, um ou muitos caracteres no texto. Por exemplo, a consulta a seguir corresponde a empresas como 'Data', 'DataC' 'DataCamp', 'DataMind' e assim por diante:

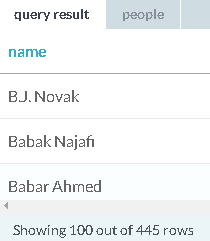
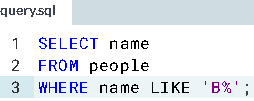


O curinga \_ corresponderá a um único caractere. Por exemplo, a consulta a seguir corresponde a empresas como 'DataCamp', 'DataComp' e assim por diante:

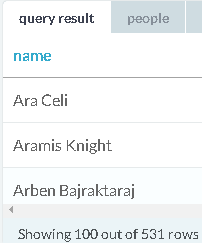
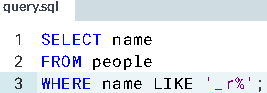


Você também pode usar o operador NOT LIKE para encontrar registros que não correspondem ao padrão especificado.

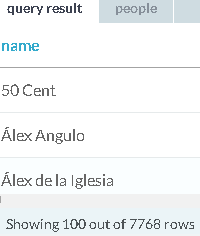
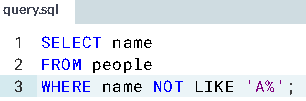
Ex. 1: obtenha os nomes de todas as pessoas cujos nomes começam com a letra “B”.



Ex. 2: obtenha os nomes de pessoas cujos nomes possuem “r” como segunda letra.



Ex. 3: obtenha os nomes de pessoas cujos nomes não comecem com “a”.



1. **Aggregate Functions**

Freqüentemente, você desejará realizar algum cálculo nos dados em um banco de dados. O SQL fornece algumas funções, chamadas funções agregadas, para ajudá-lo. Por exemplo:



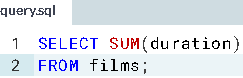
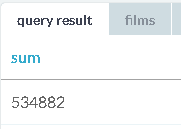
Fornece o valor médio da coluna de orçamento da tabela de filmes. Da mesma forma, a função MAX retorna o orçamento mais alto:



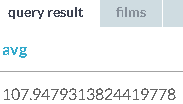
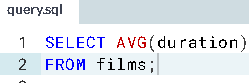
A função SUM retorna o resultado da adição dos valores numéricos em uma coluna:



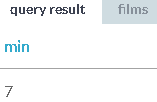
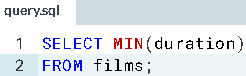
Ex. 1: Use a função SUM p/ obter o total da duração de todos os filmes:

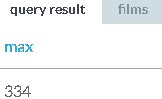
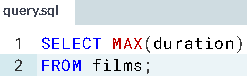
Ex. 2: Obtenha a duração média de todos os filmes:



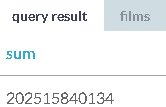
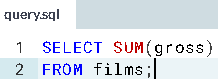
Ex. 3: Obtenha a duração do filme mais curto:



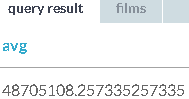
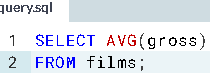
Ex. 4: Obtenha a duração do filme mais longo:



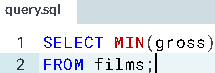
Ex. 5: Use a função SUM p/ obter o valor total gerado por todos os filmes:



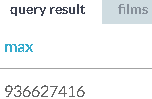
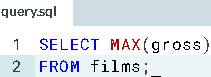
Ex.6: Obtenha o valor médio gerado por todos os filmes:



Ex.7: Obtenha o valor arrecadado pelo filme com pior desempenho.



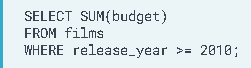
Ex. 8: Obetenha o valor arrecadado pelo filme com melhor desempenho:



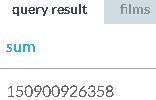
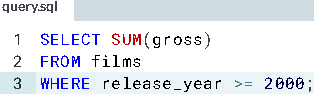
**COMBINANDO FUNÇÕES AGREGADAS COM WHERE**

As funções agregadas podem ser combinadas com a cláusula WHERE para obter informações adicionais de seus dados.

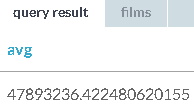
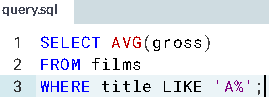
Por exemplo, para obter o orçamento total dos filmes feitos no ano de 2010 ou posterior:



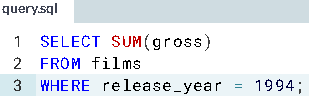
Ex. 1: Use a função SUM para obter o valor total arrecadado por todos os filmes feitos no ano 2000 ou posterior:



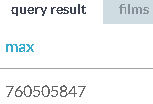
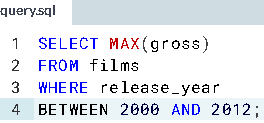
Ex. 2: Obtenha a quantia média gerada por todos os filmes cujos títulos começam com a letra 'A':



Ex.3: Obtenha o valor arrecadado pelo filme com pior desempenho em 1994:



Ex. Obtenha o valor arrecadado pelo filme com melhor desempenho entre 2000 e 2012, inclusive.



Observe que o SQL basicamente fornece vários blocos de construção que você pode combinar de todos os tipos? Daí o nome: Linguagem de Consulta Estruturada.

**Uma nota sobre aritmética**

Além de usar funções agregadas, você pode executar aritmética básica com símbolos como +, -, \* e /. Então, por exemplo, isso resulta em 12:



Aqui o resultado é 1:



O que está acontecendo aqui?

O SQL pressupõe que, se você dividir um número inteiro por um número inteiro, deseja recuperar um número inteiro. Portanto, tenha cuidado ao dividir! Se você deseja obter mais precisão ao dividir, pode adicionar casas decimais aos seus números. Por exemplo, te dá o resultado esperado de :1.333



**É tão simples quanto o aliasing (cognome)**

Você deve ter notado no primeiro exercício deste capítulo que o nome da coluna do seu resultado era apenas o nome da função que você usou. Por exemplo,



fornece um resultado com uma coluna chamada max. Mas e se você usar duas funções como essa, você terá duas colunas de mesmo nome “max”.

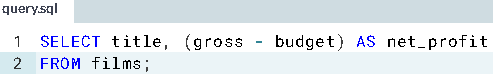


Para evitar situações como essa, o SQL permite fazer algo chamado alias. Alias significa simplesmente que você atribui um nome temporário a alguma coisa. Para o alias, você usa a palavra-chave AS, que você já viu anteriormente neste curso. Por exemplo, no exemplo acima, poderíamos usar aliases para tornar o resultado mais claro:



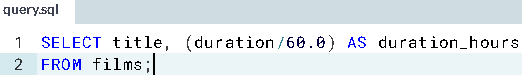
Alias ajuda na readabilidade dos resultados.

Ex.1: Obtenha o título e o lucro líquido (a quantia que um filme gerou, menos o orçamento) de todos os filmes. Aliase o lucro líquido como net\_profit:





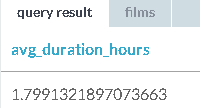
Ex. 2: Obtenha o título e a duração em horas para todos os filmes. A duração é em minutos, portanto, você precisará dividir por 60,0 para obter a duração em horas. Alias a duração em horas como duration\_hours.





Ex. 3: Obtenha a duração média em horas para todos os filmes, cognome como avg\_duration\_hours:





Lembre-se: o SQL assume que, se você dividir um número inteiro por um número inteiro, deseja recuperar um número inteiro. Isso significa que o seguinte resultará erroneamente em 400.0:

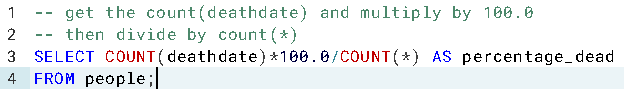


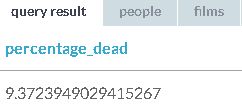
Isso ocorre porque 45/10 é avaliado como um número inteiro (4) e não como um número decimal como seria de esperar. Portanto, **ao dividir, verifique se pelo menos um dos seus números tem uma casa decimal**:



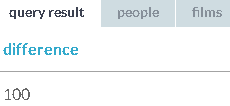
Agora acima teremos a resposta correta: 450.0, já que o numerador da divisão é decimal.

Ex. Obtenha a porcentagem de people que não estão mais vivas. Aliase o resultado como porcentage\_dead.

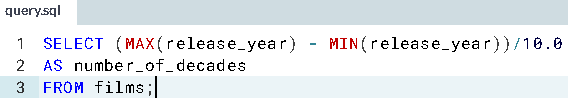


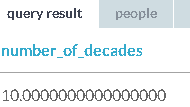


Ex. Obtenha o número de anos entre o filme mais novo e o filme mais antigo. Alias o resultado como diferença.



Ex. Obtenha o número de décadas que a tabela de filmes cobre. Alias o resultado como number\_of\_decades. A metade superior da sua fração deve estar entre parênteses.

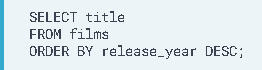




**4. Classificação e agrupamento de resultados**

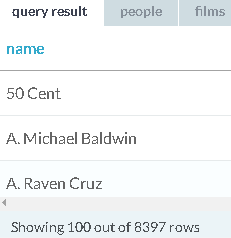
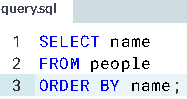
**ORDER BY**

No SQL, a palavra-chave ORDER BY é usada para classificar os resultados em ordem crescente ou decrescente de acordo com os valores de uma ou mais colunas. Por padrão, ORDER BY classificará em ordem crescente. Se você deseja classificar os resultados em ordem decrescente, pode usar a palavra-chave DESC. Por exemplo,

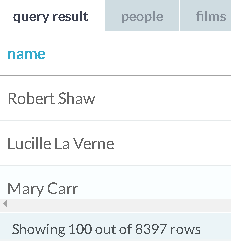
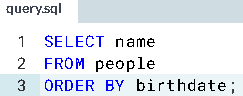


fornece os títulos dos filmes classificados por ano de lançamento, do mais recente ao mais antigo.

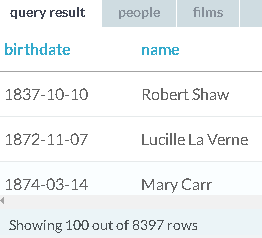
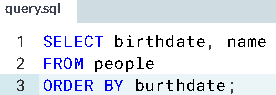
Ex.1: Obtenha os nomes da tabela people, ordenada alfabeticamente:



Ex. 2: Obtenha os nomes de pessoas ordenados pela data de nascimento:

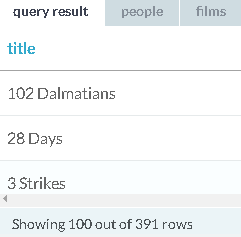
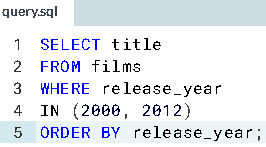


Ex.3: Obtenha a data de nascimento e o nome de cada pessoa, na ordem em que nasceram.

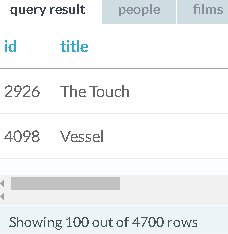
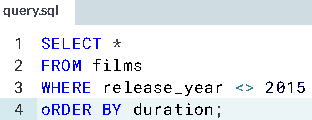


**Ordenando colunas simples**

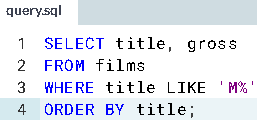
Ex.1: Obtenha o título dos filmes lançados em 2000 ou 2012, na ordem em que foram lançados:

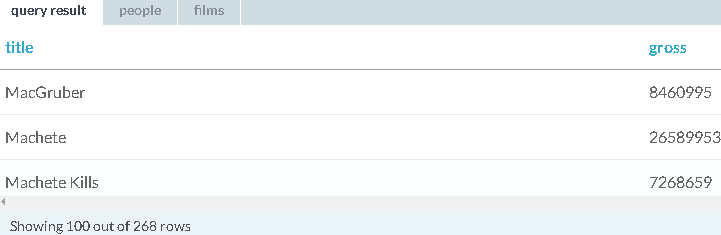


Ex.2: Obtenha todos os detalhes de todos os filmes, exceto os lançados em 2015, e ordene-os por duração:



Ex.3: Obtenha o título e os ganhos brutos dos filmes que começam com a letra 'M' e ordene os resultados em ordem alfabética:



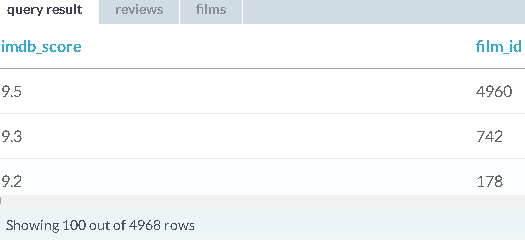
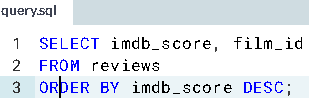


**Ordenando colunas simples (DESC)**

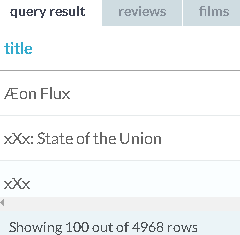
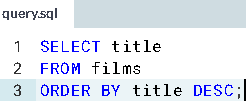
Para ordenar os resultados em ordem decrescente, você pode colocar a palavra-chave DESC depois de ORDER BY. Por exemplo, para obter todos os nomes na tabela de pessoas, em ordem alfabética inversa:



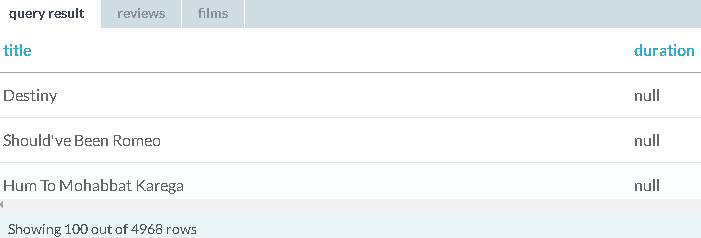
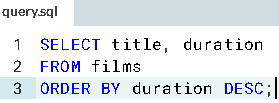
Ex. 1: Obtenha a pontuação do IMDB e o ID do filme para cada filme da tabela de resenhas, classificados da maior para a menor pontuação.



Ex.2: Obtenha o título para todos os filmes na ordem inversa:

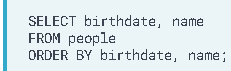


Ex.3: Obtenha o título e a duração de cada filme, do mais longo ao mais curto.

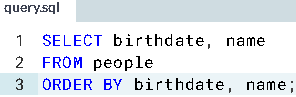
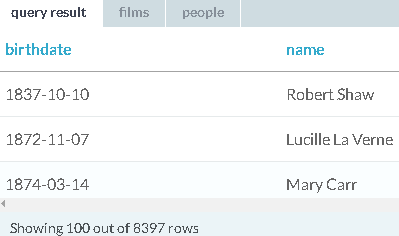


**Ordenando múltiplas colunas**

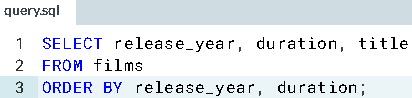
ORDER BY também pode ser usado para classificar em várias colunas. Ele será classificado pela primeira coluna especificada, depois pela próxima, depois pela próxima e assim por diante. Por exemplo, classifica primeiro as datas de nascimento (da mais antiga para a mais recente) e depois classifica os nomes em ordem alfabética. A ordem das colunas é importante!

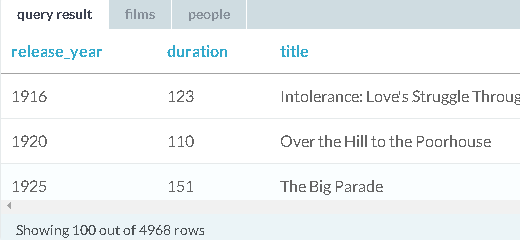


Ex. 1: Obtenha a data de nascimento e o nome das pessoas na tabela de pessoas, na ordem em que nasceram e em ordem alfabética por nome:

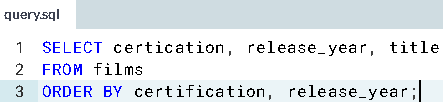
 

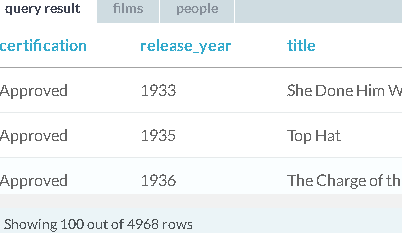
Ex.2: Obtenha o ano de lançamento, a duração e o título dos filmes ordenados por ano de lançamento e duração.



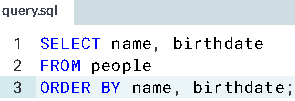
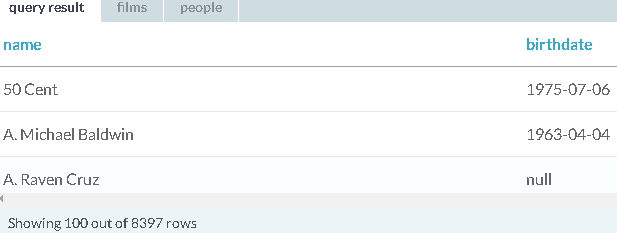


Ex.3: Obtenha certificações, anos de lançamento e títulos de filmes ordenados por certificação (em ordem alfabética) e ano de lançamento.





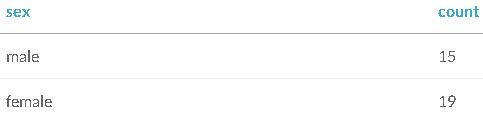
Ex.4: Obtenha os nome e datas de nasicmento da tabela people ordenados por nome e data de nascimento:

Observe como a segunda coluna que você solicita apenas aparece quando a primeira coluna não é decisiva para informar a ordem. A segunda coluna atua como um desempate.

**GROUP BY**

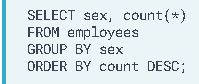
Agora você sabe como classificar os resultados! Muitas vezes, você precisará agregar resultados. Por exemplo, convém contar o número de funcionários do sexo masculino e feminino na sua empresa. Aqui, o que você quer é agrupar todos os homens e contá-los, e agrupar todas as mulheres e contá-las. No SQL, GROUP BY permite agrupar um resultado por uma ou mais colunas, assim:

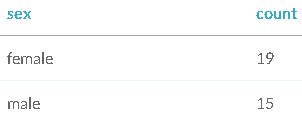
 

Geralmente, GROUP BY é usado com funções agregadas como COUNT () ou MAX (). Note-se que GROUP BY sempre vai depois da cláusula FROM!

O SQL retornará um erro se você tentar SELECIONAR um campo que não esteja na sua cláusula GROUP BY sem usá-lo para calcular algum tipo de valor sobre o grupo inteiro.

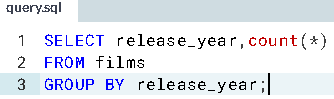
Observe que você pode combinar GROUP BY com ORDER BY para agrupar seus resultados, calcular algo sobre eles e depois ordená-los. Por exemplo,

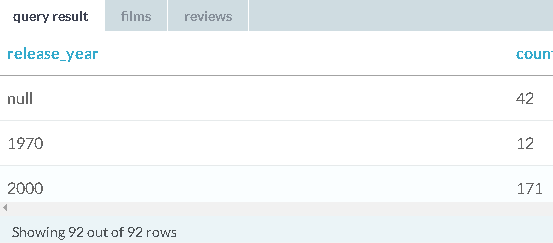




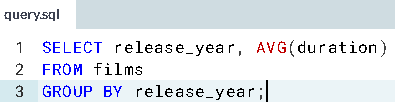
ORDER BY sempre vai depois de GROUP BY. Não dá para ordenar valores que ainda não foram calculados.

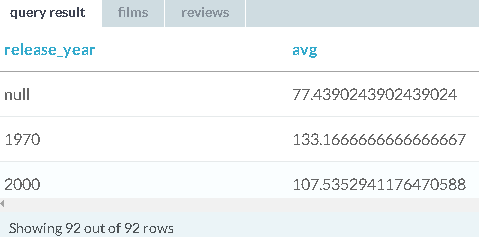
Ex. 1: Obtenha o ano de lançamento e a contagem de filmes lançados em cada ano.

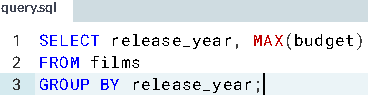




Ex.2: Obtenha o ano de lançamento e a duração média de todos os filmes, agrupados por ano de lançamento.

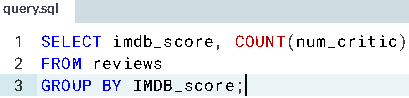


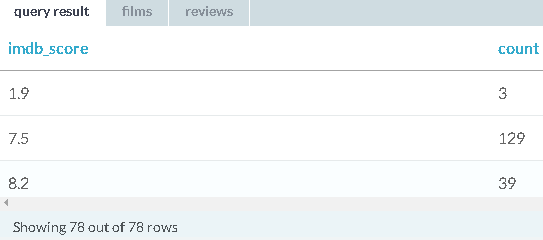


Ex.3: Obtenha o ano de lançamento e o maior orçamento para todos os filmes, agrupados por ano de lançamento.

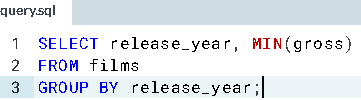


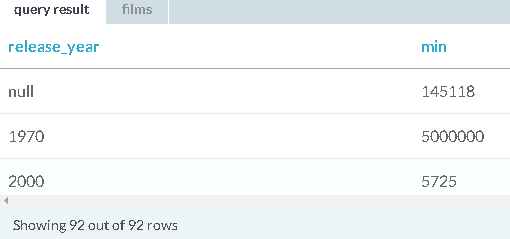
Ex.4: Obtenha a pontuação do IMDB e conte as críticas de filmes agrupadas pela pontuação do IMDB na tabela de críticas.



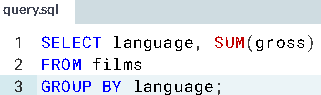


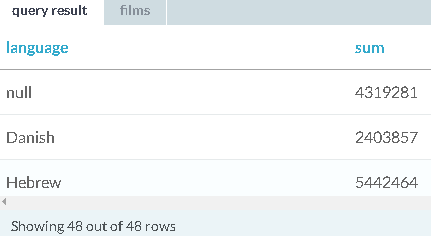
Ex. Obtenha o ano de lançamento e os ganhos brutos mais baixos por ano de lançamento:



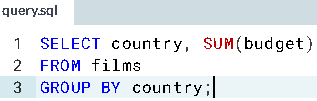


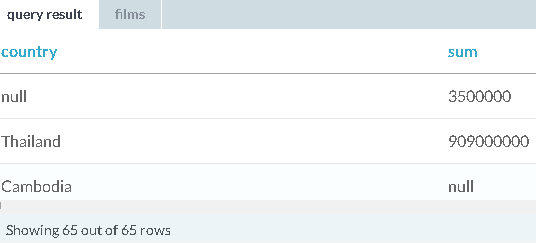
Ex. Obtenha o idioma e a quantidade total de filmes em cada idioma produzido.



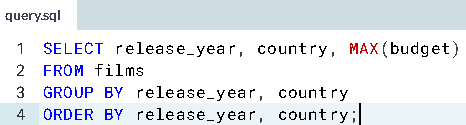


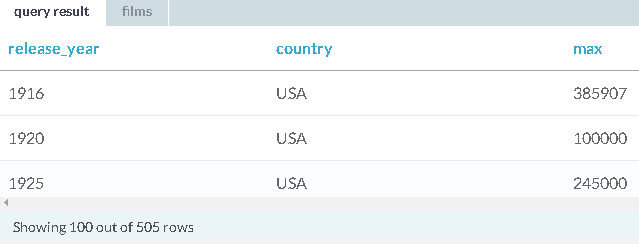
Ex. Obtenha o país e o orçamento total gasto na criação de filmes em cada país.



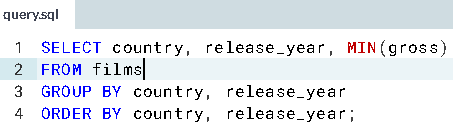


Ex. Obtenha o ano de lançamento, o país e o orçamento mais alto gastos em um filme para cada ano, para cada país. Classifique seus resultados por ano de lançamento e país.



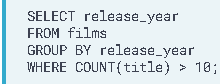


Ex. Obtenha o país, ano de lançamento e a quantia mais baixa gerada por ano de lançamento por país. Ordene seus resultados por país e ano de lançamento.

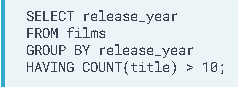


**HAVING**

No SQL, funções agregadas não podem ser usadas nas cláusulas WHERE. Por exemplo, a seguinte consulta é inválida:

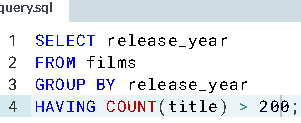


Isso significa que se você deseja filtrar com base no resultado de uma função agregada, precisamos usar outra maneira! É aí que entra a cláusula HAVING. Por exemplo, mostra apenas aqueles anos em que mais de 10 filmes foram lançados.





Ex. Em quantos anos diferentes foram lançados mais de 200 filmes? R: 13

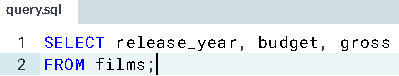


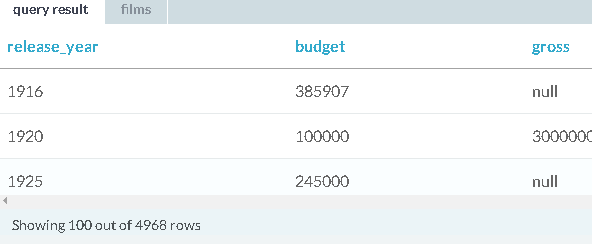


**EXERCÍCIO – ORDER BY, GROUP BY, HAVING**

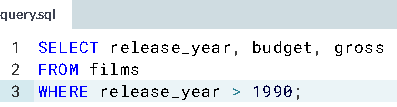
Agora, você escreverá uma consulta que retornará o orçamento médio e o lucro bruto médio dos filmes a cada ano após 1990, se o orçamento médio for superior a US $ 60 milhões.

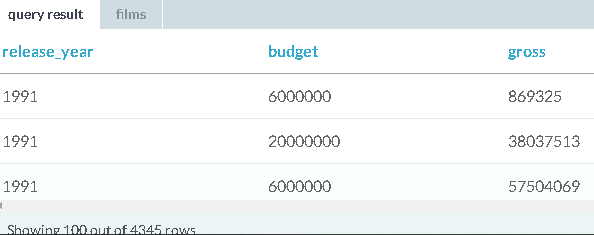
Passo 1: Obtenha o ano de lançamento, o orçamento e os ganhos brutos de cada filme na tabela de filmes:



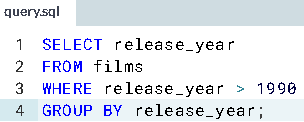


Passo 2: Modifique sua consulta para incluir apenas registros com um release\_ano após 1990.



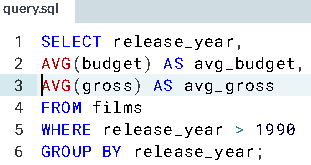
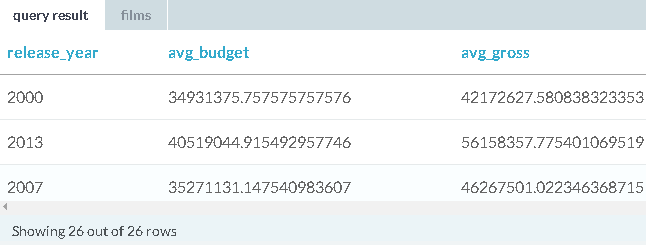


Passo 3: Remova a coluna budget e gross, e agrupe seus resultados por release\_year:

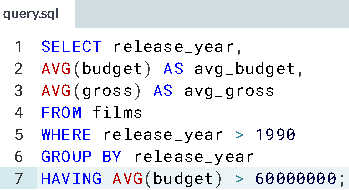


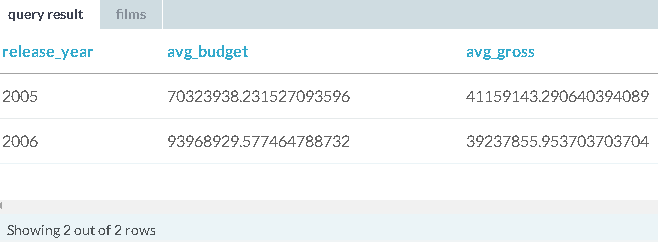


Passo 4: Modifique sua consulta para incluir o orçamento médio e os ganhos brutos médios dos resultados que você tem até o momento. Aliase o orçamento médio como avg\_budget; aliase o lucro bruto médio como avg\_gross.

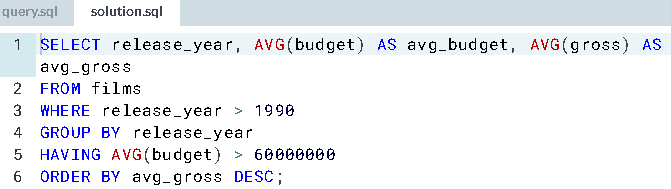
 

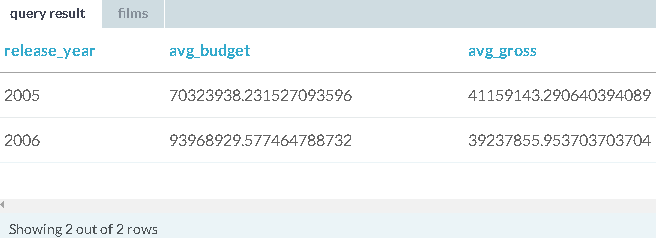
Passo 5: Modifique sua consulta para incluir apenas os anos com um orçamento médio superior a US $ 60 milhões.





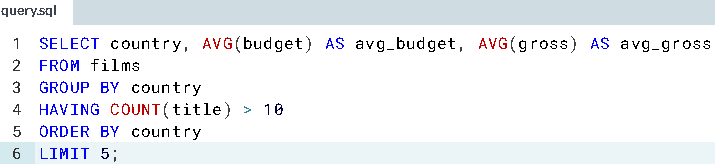
Passo 6: Finalmente, modifique sua query para ordenar os resultados do maior ganho bruto médio para menor.

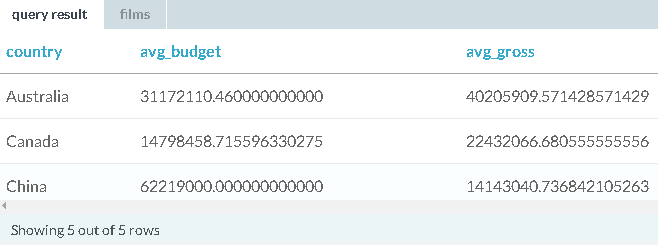




**EXERCÍCIO 2**

Obtenha o país, o orçamento médio e a receita bruta média dos países que fizeram mais de 10 filmes. Ordene o resultado pelo nome do país e limite o número de resultados exibidos para 5. Você deve usar aliase: average as avg\_budget e avg\_gross, respectivamente.



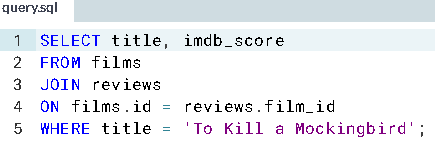


**EXERCÍCIO 3**

Agora temos o conhecimento básico de SQL. Até agora os resultados vêm de apenas uma tabela (films ou people). No mundo real, frequentemente queremos consultar múltiplas tabelas. Por exemplo, e se você quiser ver a pontuação do IMDB para um filme em particular?

Nesse caso, você deseja obter o ID do filme da tabela de filmes e usá-lo para obter informações do IMDB da tabela de comentários. No SQL, esse conceito é conhecido como junção (join), e uma junção básica é mostrada no editor à direita.

A consulta no editor obtém a pontuação do IMDB para o filme To Kill a Mockingbird.



Temos todo um curso de acompanhamento dedicado a eles, chamado Joining Data in PostgreSQL, para você aprimorar ainda mais suas habilidades no banco de dados!

