

Construtor Visual de Consultas SQL: SQ-Look

Vitor C. M. Ribeiro

Universidade Federal de Minas Gerais (UFMG) - Belo Horizonte, MG - Brasil

VITORMENEZELLO@DCC.UFMG.BR

Abstract. *The development of the new web application SQ-Learn, website that aims to replace the old Praticando SQL, platform used in the Introduction to Databases course at Universidade Federal de Minas Gerais, allows the implementation of new functionalities that can make the study of database concepts easier. The tool developed in this project is an SQL Visual Query Builder, which is heavily based on user interaction from clicks, without the need of previous SQL knowledge. The query builder, named SQ-Look, will be available on the SQ-Learn platform, after a test phase, and can expect updates to include new functionalities.*

Resumo. *O desenvolvimento da nova aplicação web SQ-Learn, site que visa substituir o antigo Praticando SQL, plataforma utilizada na disciplina de Introdução a Banco de Dados na Universidade Federal de Minas Gerais, possibilita a implementação de novas funcionalidades que podem tornar o estudo dos conceitos de bancos de dados mais fácil. A ferramenta desenvolvida neste trabalho é um Construtor Visual de Consultas SQL, que se baseia fortemente na interação com o usuário a partir de cliques, sem que haja necessidade de conhecimento prévio de SQL. O construtor de consultas, chamado de SQ-Look, será disponibilizado na plataforma SQ-Learn, após uma fase de testes, e poderá contar com atualizações para inclusão de novas funcionalidades.*

1 Introdução

1.1 Motivação

Os bancos de dados se tornaram foco de estudo para diversos níveis de formação: cursos técnicos, graduação, pós-graduação e muitos outros. Como matéria lecionada na Universidade Federal de Minas Gerais, a disciplina de Introdução a Banco de Dados conta com um site para aprendizado chamado Praticando SQL ¹. Nele existem algumas bases de dados e suas definições no modelo relacional para os alunos matriculados praticarem os conhecimentos adquiridos na aula.

O site, no entanto, não é nem um pouco moderno e possui falhas em diversos requisitos funcionais e também não funcionais. Primeiramente, a interface não possui nenhum atrativo, não gerando interesse nos alunos. Além disso, há também problemas relacionados a questões de usabilidade, como um fluxo de navegação pouco intuitivo: ao escolher uma base de dados para se praticar, o usuário é levado a outra página, mas se quiser mudar de base de dados é necessário que volte para a página inicial antes de escolher novamente. O esquema relacional que descreve as bases de dados também pode ser um pouco confuso, dificultando o aprendizado do aluno. Há um link para um tutorial SQL que foi feito em 1998, o que mostra que a ferramenta está um pouco desatualizada. Por último, o site está hospedado em um servidor instável e, com uma certa frequência, cai do ar, deixando os usuários sem ter acesso à ferramenta. A Figura 1 é uma

¹Praticando SQL: <http://www.lbd.dcc.ufmg.br/praticandosql/>

screenshot tirada da página inicial do site, e demonstra alguns dos problemas mencionados acima, evidenciando a necessidade de uma modernização da ferramenta.

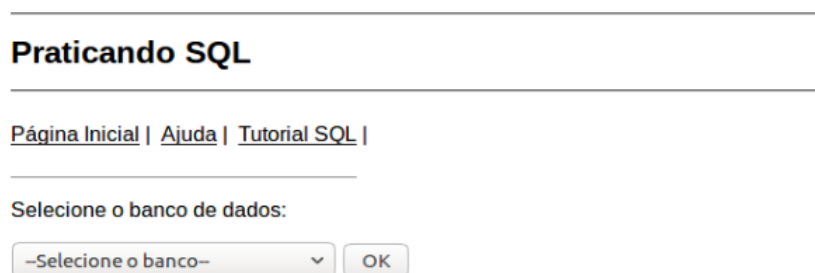


Figura 1: Screenshot da página inicial do site Praticando SQL.

Por esses motivos, a aluna Lívia Almeida Barroso realizou em 2017/2, como seu Projeto Orientado em Computação I, uma modernização do site, dando a ele uma nova interface, trazendo maior interatividade e responsividade, além de cativar mais facilmente a atenção do aluno, e também um novo tutorial SQL integrado na plataforma. Essa nova plataforma foi chamada de SQ-Learn.

1.2 Projeto

O desenvolvimento do novo sistema abre uma porta para a criação de diversas novas ferramentas que podem ser incluídas no site. A ferramenta desenvolvida no trabalho foi um Construtor Visual de Consultas SQL, o qual chamamos de **SQ-Look**, que é um meio de gerar a consulta SQL interativamente, utilizando cliques do *mouse* para gerar o resultado. Isso permite que o usuário observe a construção da consulta passo a passo e compreenda como chegar naquele resultado.

O objetivo principal da criação dessa ferramenta é facilitar o aprendizado dos conceitos de Bancos de Dados para tanto os alunos que estão cursando a disciplina de Introdução a Banco de Dados quanto qualquer outra pessoa que esteja interessada na matéria. O que torna o Construtor Visual de Consultas SQL único é a não necessidade de conhecimento prévio da sintaxe ou da semântica da linguagem SQL para uso. Um usuário leigo deve ser, em teoria, capaz de construir a consulta somente com a visualização do modelo e, com algumas instruções simples, clicar nos elementos e chegar a um resultado correto.

O projeto foi utilizado como trabalho final da disciplina de Bancos de Dados Avançados, realizado em grupo. Os outros integrantes foram as alunas Lívia Almeida Barroso e Marina Monteiro Moreira.

O que foi proposto no início do projeto era uma refatoração da ferramenta antiga, SQ-Learn, de forma a desacoplar as duas partes: interface e requisições às bases de dados. O cliente deveria ser base somente para a interface e funcionalidades principais do construtor, enquanto o servidor deveria receber a *string* com a consulta no formato *JSON*², executá-la no banco de dados e enviar de volta um outro objeto do tipo *JSON* com a resposta, para que ela fosse mostrada em tela para o usuário.

No entanto, algumas dificuldade surgiram e somente o cliente foi realmente desenvolvido. Com

² *JavaScript Object Notation*: um formato comum para transferência de dados em aplicações web que é legível para os humanos e simples para interpretação e geração para as máquinas

a falta do servidor, uma pequena base de dados provisória foi implementada no sistema cliente para que o usuário conseguisse executar algumas consultas e obter seus resultados em tela.

2 Referencial Teórico

Nesta seção discutiremos mais a fundo os conceitos básicos e as ferramentas que já existem para realizar os requisitos do projeto.

2.1 Construtor Visual de Consultas SQL

O Construtor Visual de Consultas SQL é uma ferramenta que já foi desenvolvida para diversas aplicações, e pode ser implementada de diversas maneiras. A maneira planejada de desenvolvimento foi utilizar o modelo relacional das bases de dados, pois elas nos permitem enxergar toda a estrutura do banco com seus relacionamentos, chaves estrangeiras e compostas e entidades complexas com maior facilidade. Ao selecionar campos ou escrever restrições para os atributos, o usuário constrói sua consulta, que é gerada em texto para ser lida e compreendida, antes de ser executada.

A principal referência e inspiração para o projeto foi o SQLeo Visual Query Builder ³, que é um aplicativo para plataformas Desktop, desenvolvida em Java. Nela o usuário tem acesso a um construtor de consultas gráfico que o permite criar consultas SQL complexas facilmente, segundo a própria descrição.

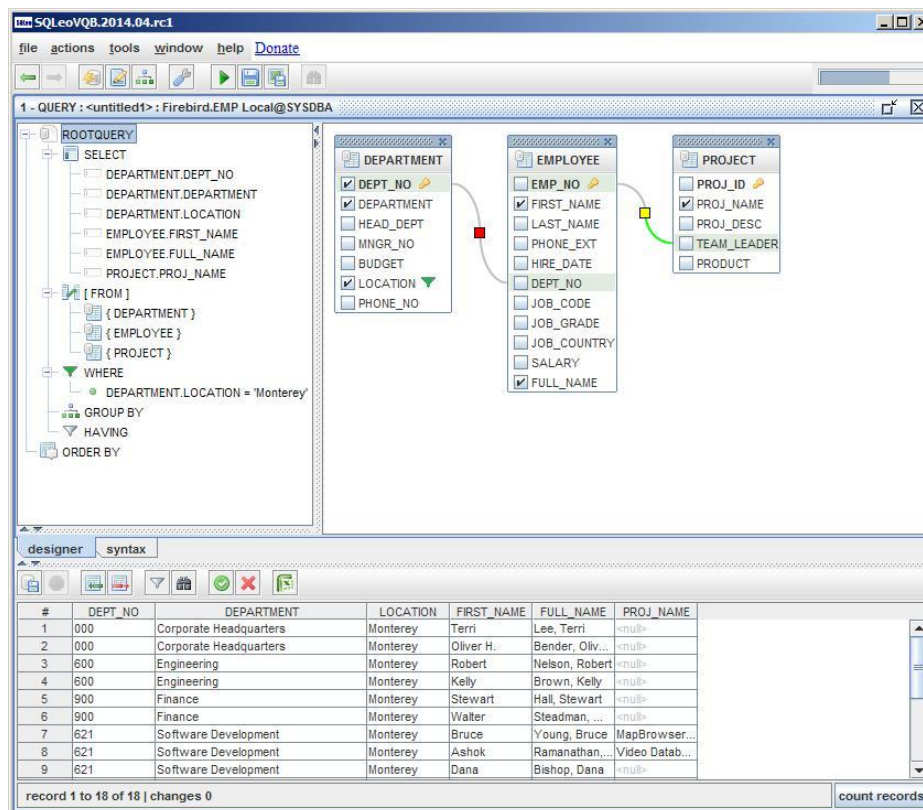


Figura 2: Construtor Visual de Consultas SQL - SQLeo: Screenshot retirada do site oficial.

³SQLeo Visual Query Builder: <http://sqlleo.sourceforge.net/index.html>

A Figura 2 é uma *screenshot* tirada do site da ferramenta que demonstra um exemplo de construção de consulta com a visualização do modelo relacional do banco. No lado esquerdo da imagem podemos ver, dentro de uma caixa, a consulta principal, chamada de ROOTQUERY, e dentro dela há diversos subitens: a seleção dos atributos no subitem SELECT, seguido das tabelas de onde esses atributos estão vindo no subitem FROM, depois uma condição WHERE, e por último os subitens GROUP BY e HAVING que estão vazios. Há também o ORDER BY para definir a ordem de exibição dos resultados, que são exibidos na parte debaixo do construtor, mostrando o nome das colunas selecionadas e quantos resultados foram retornados.

Ainda na imagem, do lado direito podemos ver o esquema relacional do banco, mas somente com algumas das tabelas presentes, representadas por blocos. No topo de cada bloco há o título da tabela, e em seguida uma lista dos seus atributos, com uma *checkbox* ao lado para incluir o elemento na seleção da consulta. Ao lado das chaves primárias de cada tabela há um pequeno ícone de chave, e ao lado dos atributos que estão sendo utilizados no comando WHERE há um ícone que representa um filtro. Além disso há uma linha que conecta as tabelas que possuem chaves estrangeiras, com uma pequena caixa no meio do caminho que indica o tipo de junção que está acontecendo. Não foi encontrada na documentação uma explicação sobre o que cada cor significa exatamente, mas supõe-se, pelos resultados na tabela, que vermelho indica uma junção natural e a amarela com o tracejado verde indica um RIGHT JOIN.

Ao longo do processo de produção do projeto, a ferramenta foi estudada para que meios de reproduzir o seu mecanismo em sistemas web fossem compreendidos. Dificuldades encontradas ao longo da implementação foram resolvidas observando os meios que os desenvolvedores da ferramenta o fizeram, evitando assim perda de tempo desnecessária reinventando a roda.

2.2 Comandos SQL

Para a definição dos comandos principais que deveriam ser incluídos no SQ-Look, foram utilizadas duas referências principais. A primeira delas foi o livro texto utilizado no curso de Introdução a Banco de Dados em que os autores Elmasri e Navathe [1] abordam alguns comandos e conceitos com maior ênfase, evidenciando uma certa importância maior para estes. A segunda referência foi o tutorial SQL presente no site W3Schools⁴, site este que, ao utilizar a ferramenta de busca do Google para pesquisar as palavras chave “SQL tutorial”, é um dos primeiros resultados retornados.

Em ambas referências, há um conjunto de comandos básicos, com conceitos mais complexos sendo usos mais específicos dos mesmos comandos ou comandos similares. Os itens escolhidos estão demonstrados na Figura 3, que é uma imagem editada da lista de lições do tutorial SQL referenciado. É importante também salientar que, observando a ferramenta SQLeo, nota-se que esta se baseia em um conjunto muito similar ao escolhido, o que indica a viabilidade de implementação deste no construtor proposto.

Como SQL é uma linguagem que possui uma quantidade de comandos enorme, a implementação de toda a sua documentação na ferramenta visual é um pouco inviável. A seleção de alguns comandos mais comuns foi um requisito importante para garantir que a ferramenta fosse entregue no prazo.

Em termos de escalabilidade, a ferramenta foi desenvolvida tentando agrupar elementos que possuem o mesmo contexto, como MIN, MAX, COUNT, AVG e SUM, que estão dentro de um contexto de **Funções**, de forma que qualquer nova função que os desenvolvedores decidissem futuramente adicionar ao sistema pudesse simplesmente ser incluída dentro desse grupo. Isso torna o projeto escalável e com possibilidade de lançamento de versões iniciais simples mas já funcionais.

⁴W3Schools - Tutorial SQL: <https://www.w3schools.com/sql/default.asp>

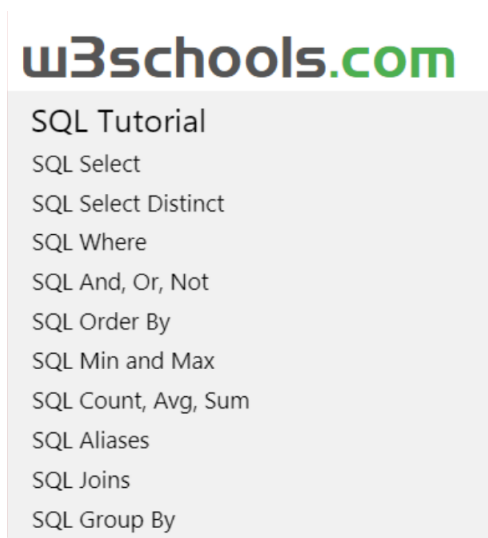


Figura 3: Conjunto de comandos selecionados.

Os desenvolvedores do projeto deixarão links para o repositório no GitHub ⁵ e endereços de e-mail para receberem sugestões dos usuários sobre quais comandos eles gostariam de ter na plataforma, caso eles sintam que há algo que é importante e não está presente, para atender às necessidades do curso e dos alunos.

2.3 Estrutura da Consulta

Com os comandos a serem implementados na ferramenta definidos, um requisito importante é a corretude da consulta gerada. Para isso, foi definida uma estrutura base considerada mínima para a ferramenta na construção de consultas, que está descrita no Código 1. A definição do modelo também foi baseada no livro texto e no tutorial.

Essa estrutura possui o elemento `SELECT` seguido de uma lista de atributos selecionados. Esses atributos fazem parte de uma lista de tabelas, que aparece após o elemento `FROM`. Os próximos elementos são todos opcionais, por isso estão envolvidos por colchetes. São eles: `WHERE` seguido de uma condição para a seleção, `GROUP BY` seguido de uma lista de atributos para agrupamento, `HAVING` seguido de uma condição para o agrupamento e, finalmente, `ORDER BY`, seguido de uma lista de atributos para ordenação.

```
SELECT <lista de atributos>
FROM <lista de tabelas>
[ WHERE <condicao> ]
[ GROUP BY <lista de atributos>
  HAVING <condicao dos grupos> ]
[ ORDER BY <lista de atributos> ]
```

Código 1: Estrutura base das consultas

3 Desenvolvimento

A produção do projeto foi dividida em pequenas etapas descritas a seguir.

⁵GitHub: <https://github.com/>

3.1 Definição de Ferramentas e Implementação

3.1.1 Sistema

O objetivo inicial do projeto era a refatoração do sistema SQ-Learn para posteriormente integrar a ferramenta SQ-Look ao novo sistema. Seriam então desenvolvidas duas partes, um cliente e um servidor, que serão explicados a seguir, em duas possíveis frameworks diferentes. No entanto, o grupo teve dificuldades para chegar em um acordo sobre qual framework utilizar para cada parte.

O sistema cliente deveria ser focado na interface, pois não havia necessidade de um controle de dados robusto sendo que esse tratamento dos dados seria feito pelo servidor. Além disso, o sistema em si é pequeno, com algumas telas apenas. Por isso foi decidido utilizar o micro-framework PHP Lumen ⁶, baseado no framework Laravel.

O desenvolvimento do servidor acabou ficando atrasado e o foco do trabalho era a ferramenta de construção visual, portanto foi adiado. No entanto já está definido e será desenvolvido também utilizando Lumen. As plataformas Lumen e Laravel possuem recursos muito parecidos, mas o Lumen é uma simplificação criada exatamente para micro-aplicações, que é o caso do SQ-Learn. O recurso mais interessante utilizado do framework é o sistema de templates chamado Blade, que será melhor detalhado a seguir.

3.1.2 Interface

Por ser uma ferramenta destinada a uma aplicação web, um importante requisito que guiou o desenvolvimento foi a interação com o usuário. Tendo em mente o requisito de uma interface com boa usabilidade, algumas bibliotecas focadas em UI (*User Interface*) e UX (*User Experience*) foram pesquisadas e dentre algumas possibilidades foi definido que seria utilizada a framework de design da Google, o Material Design ⁷.

Com a framework definida, o primeiro passo foi uma redefinição das cores e do layout de visualização do esquema relacional das bases de dados, de forma a tornar ele o mais legível possível para facilitar a interação do usuário. O novo desenho e esquema de cores foram feitos pela Designer Gráfica formada na UEMG, Amanda Bicalho ⁸. Na Figura 4 podemos ver que a escolha de cores facilita a legibilidade dos elementos. Cada tabela possui uma cor atribuída a ela, de forma que fique clara a distinção entre cada uma. As chaves primárias de cada tabela são indicadas pelo sublinhado colorido. As chaves estrangeiras aparecem como pequenas caixas com a cor da tabela referida e o nome do atributo referido.



Figura 4: Novo esquema relacional.

⁶Micro-Framework PHP Lumen: <https://lumen.laravel.com/>

⁷Material Design: <https://material.io/design/>

⁸Portfólio de Amanda Bicalho. <https://www.behance.net/user/?username=amandabicalho>

Para a implementação do que é definido pelo Material Design havia diversas possibilidades de frameworks com os componentes UI já prontos. A framework escolhida foi AngularJS Material ⁹, que utilizado em conjunto com a biblioteca AngularJS ¹⁰ possui grande capacidade de incluir elementos dinâmicos e eventos interativos na aplicação. Dentre os motivos para essa escolha, a boa documentação das ferramentas e a familiaridade do grupo com o AngularJS se destacam.

Foi necessário também o uso de alguns ícones que representassem visualmente algumas funcionalidades da ferramenta, de forma que os ícones pudessem explicar, sem causar confusão no usuário, o que cada botão significa em termos de funcionalidade. A maioria dos ícones foi retirada do site Material Design Icons ¹¹, com algumas exceções sendo feitas manualmente pela designer Amanda.

Finalmente, uma decisão importante do projeto para a escalabilidade das interfaces foi implementar as telas utilizando componentes. Para isso haviam diversas possibilidades, como o próprio AngularJS, que permite a criação de componentes e diretivas para serem incluídas diretamente no código HTML. Por outro lado, o Lumen possui um sistema de templates chamado Blade, como já mencionado, que permite a criação de componentes de uma forma mais simples, além de outros recursos que foram bem utilizados, como o sistema de extensão de telas de acordo com um arquivo base, similar ao conceito de herança em Orientação a Objetos.

Em resumo, os templates Blade são compilados para geração dos arquivos HTML finais para cada tela. Além das diretivas nativas da linguagem, que foram estilizados e organizados em tela com o uso de CSS, foram utilizados os componentes AngularJS Material. Para os dados dinâmicos foi utilizado o AngularJS.

3.2 Testes

A ferramenta será colocada em fase de testes alfa para que possíveis usuários relatem erros encontrados e façam sugestões de funcionalidades que o SQ-Look não possui ou que não atendeu às expectativas. Para isso será disponibilizado um link para o repositório e os endereços de email dos desenvolvedores na página inicial para que as sugestões e reclamações sejam enviadas.

Após a fase de testes alfa, os erros serão corrigidos e a aplicação terá o lançamento de uma nova versão para testes beta com estudantes de Introdução à Bancos de Dados. Esses testes serão base para garantir que o produto pode ser utilizado na disciplina com a aplicação de trabalhos ou exercícios práticos de acordo com as necessidades da professora Mirella.

3.3 Lançamento

Após lançamento, a ferramenta será acompanhada com atualizações mensais, ou sob necessidade, para correções ou inclusão de novas funcionalidades pelos desenvolvedores do projeto. Ao implementar novas ferramentas, a *branch* de desenvolvimento será utilizado no GitHub, e a versão no ar será a *master*.

O principal objetivo disso é atender às necessidades dos alunos à medida que o curso é modificado, pois, como sabemos, a Ciência da Computação evolui constantemente, e não queremos que o site se torne obsoleto rapidamente e seja olhado de forma negativa como o antigo Praticando SQL.

⁹AngularJS Material: <https://material.angularjs.org/latest/>

¹⁰AngularJS: <https://angularjs.org/>

¹¹Material Design Icons: <https://materialdesignicons.com/>

4 Resultados

A nova interface do SQ-Learn, após a refatoração para adequação ao Material Design da Google, se baseia em diversas ferramentas confiáveis para garantir que o usuário tenha a melhor experiência ao utilizar qualquer funcionalidade da aplicação. O objetivo principal do projeto é trazer maior didática para incentivar os estudos de bancos de dados, e a inclusão do SQ-Look, o construtor visual de consultas SQL, só tem a acrescentar para o site.

O SQ-Look foi implementado como uma página separada, na qual o usuário deve escolher, similar à funcionalidade do antigo Praticando SQL, uma base de dados. Após escolhida a base de dados, o seu modelo relacional é exibido em tela, como mostra a Figura 5. O esquema dessa base de dados em particular possui cinco tabelas, cada uma exibida em uma cor diferente. O título de cada tabela é o texto colorido em destaque e as chaves primárias são os itens sublinhados na lista. Caso a tabela possua chaves estrangeiras, o nome das tabelas referenciadas aparece ao lado do atributo, como é o caso da tabela *movie_info*, que referencia as tabelas *movie* e *movie_genre* com seus atributos *movie_id* e *movie_genre_id*, respectivamente.



Figura 5: Screenshot da ferramenta SQ-Look. Visualização do esquema relacional da base de dados do *imdb250*.

Para construir a consulta, o usuário deve interagir com as *checkboxes* presentes ao lado do nome dos atributos e do nome da tabela. Ao selecionar essas caixas, o atributo é incluído na consulta. Por exemplo, ao clicar em *movie_name*, o resultado seria o exibido no Código 2. Se o usuário seleciona a caixa ao lado do nome da tabela, todos os atributos da tabela são selecionados, resultando no demonstrado no Código 3.

```
SELECT
  movie_name
FROM
  movie
```

Código 2: Exemplo de seleção de um atributo simples


```

SELECT
    *
FROM
    movie

```

Código 3: Exemplo de seleção de todos os atributos da tabela

Quando o usuário seleciona elementos de múltiplas tabelas, o nome da tabela aparece antes do nome do atributo. Ainda com base no esquema relacional da Figura 5, ao selecionar, por exemplo, as colunas *movie_name* de *movie* e *movie_genre_name* de *movie_genre*, o resultado será o mostrado no Código 4. Uma importante ressalva é que a consulta exibida para o usuário é indentada, de forma a deixar claro o que são os elementos dentro de cada cláusula.

```

SELECT
    movie.movie_name ,
    movie_genre.movie_genre_name
FROM
    movie ,
    movie_genre

```

Código 4: Exemplo de seleção em múltiplas tabelas

Ao lado do nome de cada tabela, há um ícone com três pontos verticais que indica um menu de opções. Dentro do menu há uma série de elementos que podem modificar ou inserir novas colunas à consulta. A Figura 6 mostra um exemplo do menu aberto com suas funcionalidades. Dentro do menu, ao lado de cada item, há um ícone que deve ajudar o usuário a entender a função de cada um intuitivamente. As funcionalidades listadas são, em sequência, adicionar uma nova condição à cláusula WHERE, filtrando os resultados, adicionar pseudônimos para as colunas e tabelas com o comando AS, adicionar funções ao resultado como COUNT, realizar junções com outras tabelas e por último criar grupos com a cláusula GROUP BY.



Figura 6: Screenshot da ferramenta SQ-Look. Menu de opções ao lado da tabela *movie*.

Ao clicar em cada um dos itens abre-se uma caixa de diálogo para que o usuário defina melhor como quer que o elemento seja adicionado à consulta. A Figura 7 mostra a caixa que é aberta ao se clicar no elemento *Filtros* da Figura 6. Inicialmente a consulta não possui nenhuma condição, portanto somente o botão para adicionar, indicado pelo ícone com sinal de adição, aparece em tela. Após adicionado um filtro, quatro campos distintos aparecem, exibidos na Figura 8. O primeiro deles indica qual o operador que será colocado antes da condição, sendo as possibilidades combinações dos operadores NOT, AND e OR. No segundo campo o usuário deve selecionar um dos atributos presentes na tabela. O terceiro indica qual tipo de condição será utilizada, que geralmente são elementos simples de comparação, como igualdade. Por ora os elementos

complexos como IN, BETWEEN e LIKE ainda não estão implementados, e serão adicionados futuramente. Por último, o valor a ser comparado com o atributo. O ícone com a letra X no final da linha significa remover a condição. Para tornar uma condição válida, todos os campos devem ser preenchidos corretamente. Ao fechar a caixa de diálogo, todas as condições não corretamente preenchidas são removidas, evitando a adição de uma condição incompleta à consulta.

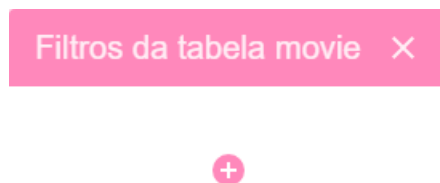


Figura 7: Screenshot da ferramenta SQ-Look. Caixa de diálogo de filtros da tabela *movie* sem nenhuma condição.



Figura 8: Screenshot da ferramenta SQ-Look. Caixa de diálogo de filtros da tabela *movie* com uma condição ainda não preenchida.

Para alguns comandos que não são específicos de cada tabela, foi criado um menu ao lado do nome da base de dados selecionada. Na Figura 9 podemos ver que existem dois itens, o primeiro deles adiciona o comando DISTINCT à consulta e o segundo ordena os resultados com base nos atributos selecionados e uma ordem escolhida entre ascendente ou descendente.

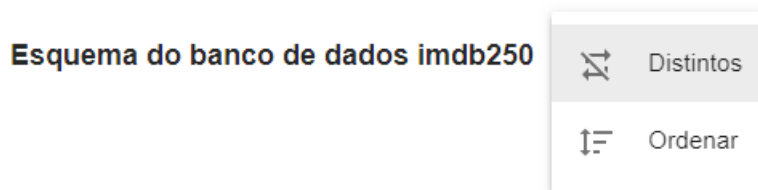


Figura 9: Screenshot da ferramenta SQ-Look. Menu de opções ao lado do título da base de dados.

O uso de menus, como os mostrados acima, facilita a escalabilidade, pois novos elementos podem ser adicionados facilmente dentro dos menus, seja como novo item ou dentro de um dos itens já existentes.

5 Demonstração

Para demonstrar a ferramenta funcionando, foi montado um pequeno exemplo que exhibe o passo a passo da construção de uma consulta utilizando o SQ-Look, incluindo interações simples

porém muito importantes. Nesta demonstração, queremos consultar a base de dados para saber o nome, o gênero e a avaliação de todos os filmes que foram produzidos após 2007. Para isso devemos selecionar tais atributos na visualização do esquema relacional, como demonstrado na Figura 10.



Figura 10: Screenshot da ferramenta SQ-Look. Esquema relacional do *imdb250* com alguns atributos selecionados.

Vamos testar a funcionalidade de adicionar uma label para o nome dos atributos também. Neste exemplo, a coluna *movie_name* será nomeada ‘Nome do Filme’, a coluna *rating* será chamada de ‘Avaliação’ e por último a coluna *movie_genre_name* será chamada de ‘Gênero’. Essas labels estão demonstradas nas Figuras 11.



Figura 11: Screenshot da ferramenta SQ-Look. Caixas de diálogo de labels das tabelas *movie* e *movie_genre*.

Para que a consulta seja correta, devemos fazer a junção das tabelas. A tabela *movie_info* mapeia o relacionamento N para M das tabelas *movie* e *movie_genre*, então utilizaremos ela para mostrar as duas junções. Na Figura 12, podemos ver as possíveis junções que a plataforma permite o usuário incluir com base no modelo relacional. O tipo de junção que selecionamos são as INNER JOINS. Como são chaves que se referenciam, a plataforma realiza automaticamente a condição do INNER JOIN.



Figura 12: Screenshot da ferramenta SQ-Look. Caixa de diálogo das junções da tabela *movie_info*.

Por último, para adicionar a condição da consulta, vamos aos filtros da tabela *movie* e adicionamos a condição, como demonstrado na Figura 13.

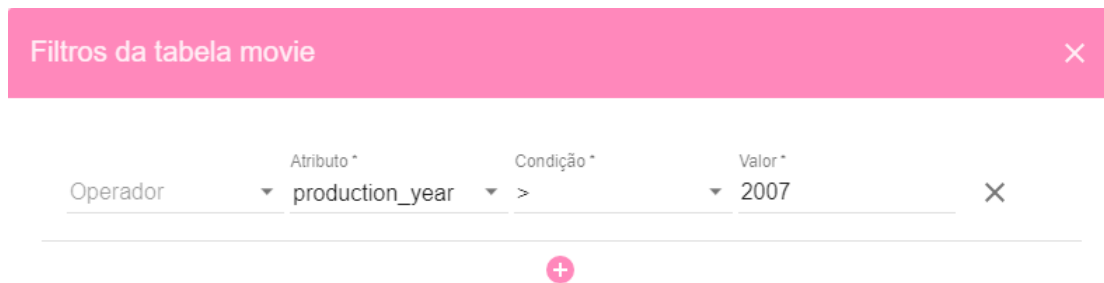


Figura 13: Screenshot da ferramenta SQ-Look. Caixa de diálogo das condições da tabela *movie*.

O resultado final desta consulta será o descrito pelo Código 5.

```

SELECT
  movie.movie_name AS 'Nome do Filme',
  movie.rating AS Avaliação,
  movie_genre.movie_genre_name AS Gênero
FROM
  movie
  INNER JOIN movie_info ON movie.id = movie_info.movie_id
  INNER JOIN movie_genre ON movie_genre.id = movie_info.
  movie_genre_id
WHERE
  movie.production_year > 2007

```

Código 5: Resultado da consulta exemplo gerada no SQ-Look

6 Conclusão e Trabalhos Futuros

Neste artigo descrevemos a motivação para o desenvolvimento da nova ferramenta e os procedimentos necessários para todos os passos de implementação, desde a escolha de bibliotecas e referências até as decisões de projeto. Houveram dificuldades principalmente com a coordenação da refatoração por parte dos desenvolvedores e também na implementação de detalhes do construtor visual de consultas, mas a ferramenta proposta foi desenvolvida e possui um grande lado positivo que é o seu planejamento tendo em mente a escalabilidade e manutenibilidade. Além disso ela

possui objetivo final de ser utilizada como material didático para os estudantes de Introdução a Banco de Dados.

Visto que o sistema possui muito espaço para crescer, podemos citar como possíveis trabalhos futuros a implementação de novas ferramentas, como um tradutor de SQL para Álgebra Relacional, que é um conceito com o qual alunos possuem certa dificuldade e, assim como o SQ-Look, traria apenas benefícios para o ambiente de estudos. Além disso, a manutenção do site com atualizações e inclusão de novos métodos e correções com base na experiência dos usuários do SQ-Look serão de responsabilidade dos desenvolvedores.

No entanto, a construção do tradutor não é uma prioridade, e outros possíveis trabalhos futuros podem ser escolhidos na mesma área como Projeto Orientado em Computação II.

Referências

- [1] ELMASRI, Ramez; NAVATHE, Shamkant. Sistema de Bancos de Dados. 6^a ed. São Paulo: Pearson, 2010.