

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

[TÍTULO]

Daniel Blando Deluiggi

PROJETO FINAL DE GRADUAÇÃO

CENTRO TÉCNICO CIENTÍFICO - CTC DEPARTAMENTO
DE INFORMÁTICA Curso de Graduação em Ciência
da Computação

Rio de Janeiro, Março de 2013



Daniel Blando Deluiggi

[TITULO]

Relatório de Projeto Final, apresentado ao programa
Ciência da Computação da PUC-Rio como requisito
parcial para a obtenção do título de Bacharel em
Ciência da Computação.

Orientador: Rogério Costa

Rio de Janeiro Março de 2013

1. Introdução

O mercado atual para ferramentas de ensino está em alta. Os alunos buscam cada vez mais um aprendizado independente e acelerado como alternativas do ensino tradicional, focado, exclusivamente em salas de aula. Iniciativas como *Coursera*[1] e *Codeschool*[2] têm propostas de ensinar o usuário de maneiras diferentes, mas com o mesmo objetivo. O *Coursera* é um site focado em aulas universitárias e com um tutor para ajudar ao longo do curso. Já o *Codeschool* tem uma iniciativa inovadora que vem fazendo muito sucesso, o site possui inúmeros tutoriais onde o aluno completa etapas que incluem uma explicação teórica e um exercício prático.

Entretanto, essas iniciativas de ensino têm uma grande escassez de ferramentas para aprendizado de banco de dados. O *Coursera*, por exemplo, fornece cursos voltados a banco de dados, mas sem nenhuma ferramenta disponível para um aprendizado prático. Os alunos encontram grande dificuldade ao procurar uma ferramenta que os auxilie de forma mais interativa.

Assim, nosso objetivo é desenvolver uma ferramenta para ajudar no ensino em banco de dados, de uma forma simples, porém eficaz e interativa. Sua função será auxiliar os alunos no aprendizado de SQL relacionando-o à linguagem natural. Essa ferramenta será usada como um programa a ser instalado no computador e que se conecta ao banco de dados fornecido pelo aluno. Terá como funcionalidade receber uma pergunta feita em linguagem natural dentro de uma sintaxe pré-definida. A partir dessa pergunta e das informações extraídas do banco de dados, ela irá gerar o comando SQL, que terá como resultado as informações do próprio banco para tal pergunta. Vale ressaltar que esse fluxo deve ocorrer apenas com a conexão ao banco de dados, sem a necessidade de nenhuma informação semântica.

Na ferramenta conhecida como *PrimeQue* foram encontradas as mesmas características desse projeto, onde o sistema consegue guiar o usuário a elaborar perguntas para receber em forma gráfica a resposta no *dashboard*, porém o usuário inicialmente deve informar conceitos semânticos sobre os dados que estão no banco de dados. Outro diferencial é que o *PrimeQue* tem como foco o mercado, o que não se aplica ao intuito da ferramenta proposta.

2. Estado da Arte

As propostas existentes que auxiliam o ensino de banco de dados seguem caminhos diversos. Algumas das abordagens encontradas são ferramentas de gerenciamento de banco de dados, cursos online, sites com bancos disponíveis para manipulação online e tutorias sobre SQL. Todos esses métodos podem ser utilizados para aprendizagem, mas eles não se encaixam na proposta de projeto de ensinar SQL de uma forma mais intuitiva, onde o aluno pode relacionar a linguagem natural com o SQL.

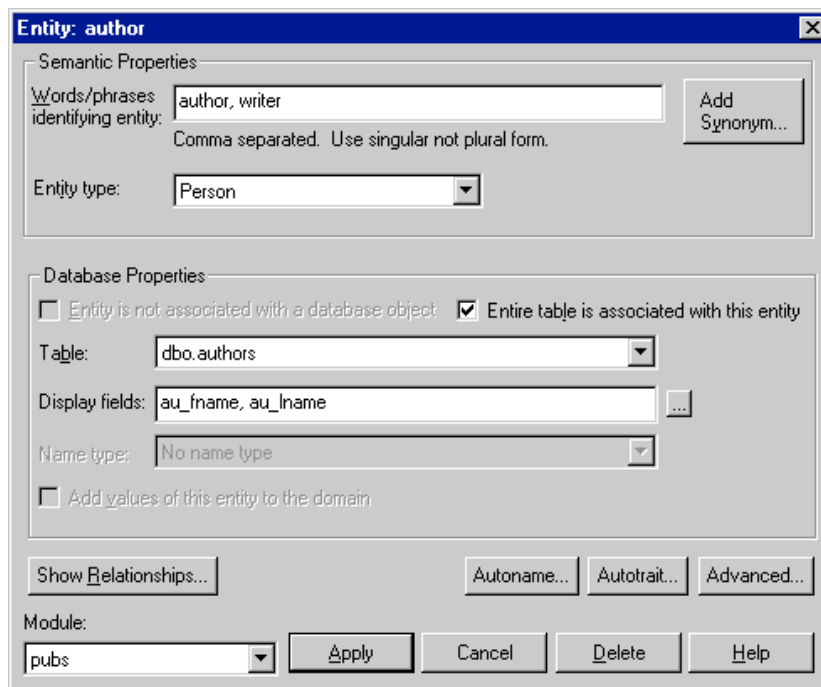
Seguindo o método proposto não foi encontrada nenhuma ferramenta em funcionamento com o intuito de auxiliar o ensino de banco de dados aos alunos. Apresentaremos, abaixo, algumas ferramentas semelhantes, mas que não tem a mesma proposta:

- *Delver* [3]

Esta ferramenta tem uma grande semelhança ao proposto nesse projeto, pois traduz perguntas em linguagem natural para SQL, e pode ser usada a partir de API ou SDK, entretanto, ela ainda não foi lançada e suas únicas informações são teóricas e encontradas em seu site, assim não temos como comprovar de que forma e como será seu funcionamento.

- Microsoft English Query [4]

Ferramenta utilizada para tradução de linguagem natural em SQL, mas por meio de mapeamento prévio da linguagem natural. O usuário deve inicialmente definir palavras chaves e regras de sintaxe no programa para que o mesmo possa traduzir a pergunta e responder em SQL. As palavras chaves são entidades que podem se relacionar, e as regras são verbos que relacionam as entidades. A figura 1 apresenta como é criado uma palavra chave no sistema, na imagem pode-se notar a criação da ligação entre tabelas e colunas do banco com palavras chaves. Já na figura 2 existe um exemplo de ligação de duas palavras chaves por meio de um verbo. Esse método de tradução cria uma dificuldade enorme para o usuário, pois necessita que o mesmo introduza regras para cada sintaxe diferente da realizada no banco.



Entity: author

Semantic Properties

Words/phrases identifying entity: Add Synonym...

Entity type:

Database Properties

☐ Entity is not associated with a database object ☒ Entire table is associated with this entity

Table:

Display fields: ...

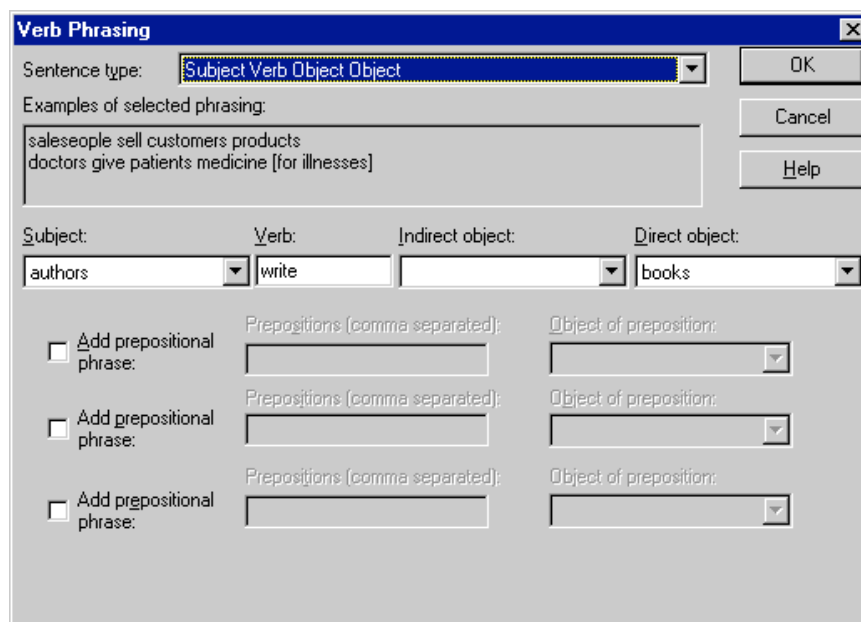
Name type:

☐ Add values of this entity to the domain

Show Relationships... Autoname... Autotrait... Advanced...

Module: Apply Cancel Delete Help

Figura 1: Tela para criar entidades. O usuário cria palavras chaves que se relacionam com tabelas e colunas existentes no banco



Verb Phrasing

Sentence type: OK

Examples of selected phrasing:

salespeople sell customers products
doctors give patients medicine [for illnesses]

Cancel Help

Subject:	Verb:	Indirect object:	Direct object:
<input type="text" value="authors"/>	<input type="text" value="write"/>	<input type="text"/>	<input type="text" value="books"/>

☐ Add prepositional phrase: Prepositions (comma separated): Object of preposition:

☐ Add prepositional phrase: Prepositions (comma separated): Object of preposition:

☐ Add prepositional phrase: Prepositions (comma separated): Object of preposition:

Figura 2: Tela para criação de regras. Com as regras o usuário pode relacionar entidades criadas previamente.

- *PrimeQue* [5]

É a ferramenta que mais se aproxima da proposta desse projeto, pois ajuda o usuário na formulação de perguntas para um sistema específico com uma interface limpa e simples. A figura 3 exibe um exemplo das opções fornecidas pelo sistema para a próxima palavra que pode ser usada pelo usuário.

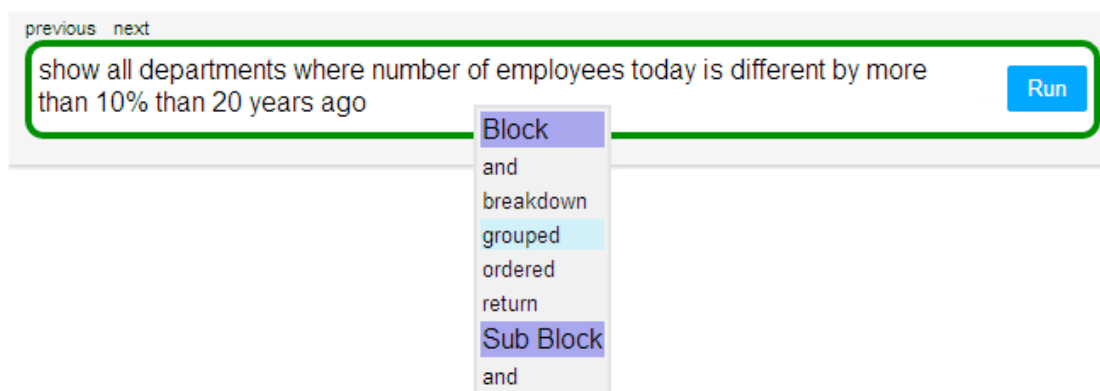


Figura 3: Formulação da pergunta pelo usuário na ferramenta *PrimeQue*

O problema é que ela também necessita que o usuário insira informação semântica sobre a base de dados. O usuário pode criar sua própria linguagem definindo um mapa semântico com nomes para campos, conexões entre os campos, entre outros. A figura 4 mostra como pode ser criado um mapa semântico para a tabela *employees*.

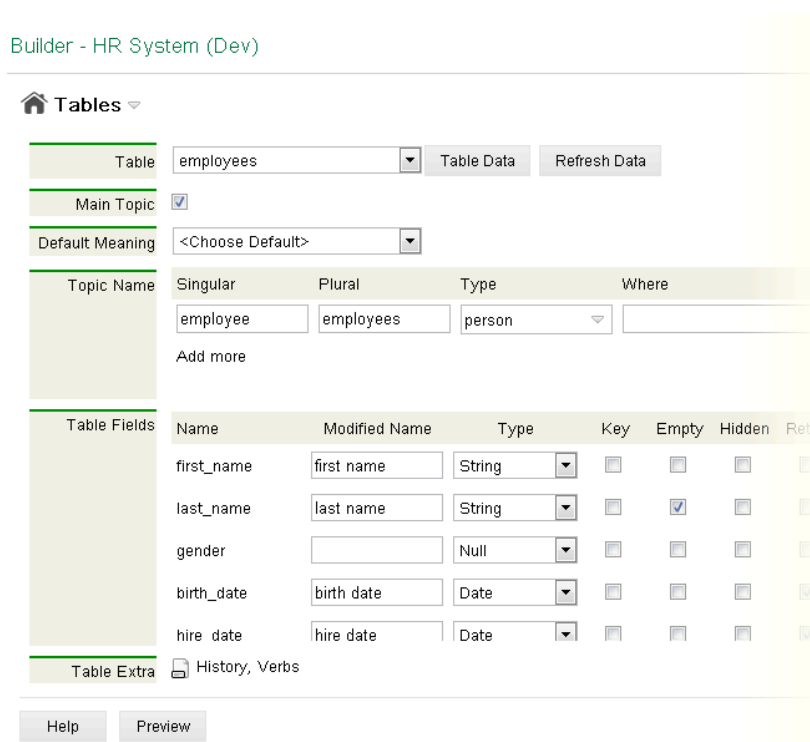


Figura 4: Construção da semântica na ferramenta *PrimeQue*

A *PrimeQue* também possui diferentes funcionalidades como *dashboard*, histórico de perguntas e diferentes níveis de usuários. O *dashboard* pode ser personalizado com gráficos e estatísticas sobre o banco, bem como um histórico de perguntas para análise posterior e reutilização. Os usuários podem ser: *PrimeQue Admin*, *DB Admin* e *User*. O usuário *PrimeQue Admin* tem privilégio de criar conexão com novos bancos e novos usuários, o usuário *DB Admin* é o responsável por fornecer as informações semânticas sobre o banco, e *User* é o usuário padrão de um banco de dados que pode consultar os dados e ter seu próprio *dashboard*.

Entretanto, o maior problema é que a *PrimeQue*, infelizmente, é voltada para o mercado e não possui um âmbito educativo.

Finalizando, concluímos que não foi possível encontrar uma ferramenta de auxílio ao aprendizado de banco de dados que consiga a partir de uma pergunta em linguagem natural e informações extraídas do banco responder com uma consulta SQL. Todas as ferramentas existentes atualmente necessitam de uma predefinição sintática sobre o banco.

3.Proposta e Objetivos do trabalho

O projeto se baseia no desenvolvimento de uma ferramenta gratuita e de código-aberto para auxiliar no ensino de banco de dados, trazendo uma nova solução para os alunos aprenderem de forma mais simples e intuitiva a linguagem SQL. Essa ferramenta tem o propósito de apoiar alunos com pouco ou nenhum conhecimento de SQL.

Seu principal objetivo será receber uma pergunta em linguagem natural e retornar o equivalente em SQL para que o aluno possa avaliar e executar o comando, com o intuito de que ele entenda a ligação entre sua pergunta e as funções da linguagem SQL. Seu requisito mínimo será um banco de dados já existente e conectado a ferramenta para extração de informação, o banco de dados pode estar conectado local ou remotamente, vide figura 5.

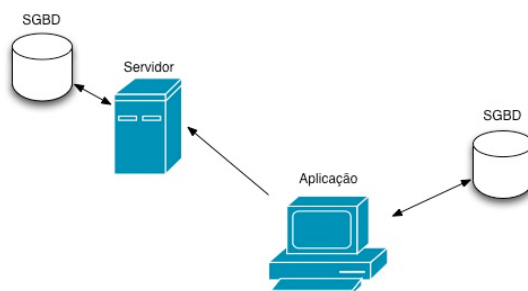


Figura 5: Diagrama da arquitetura

Ao iniciar, o programa recolherá dados do banco como nome de tabelas, colunas e restrições para que possa identificar na pergunta palavras chaves usadas pelo aluno. Como entrada, o programa irá receber uma pergunta feita pelo aluno com base no banco de dados fornecido e também irá guiar o usuário na criação dessa pergunta. O guia, vide figura 6, funcionará de forma a restringir as opções de palavras que o aluno pode usar com relação às funcionalidades existentes no programa. A cada comando de SQL implementado a lista de palavras e opções de frases aumentará.

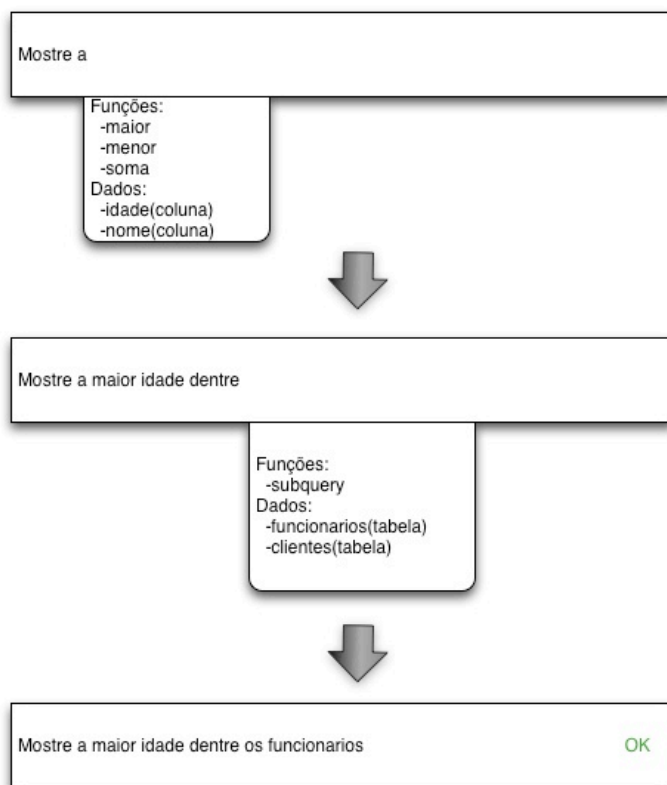


Figura 6: Exemplo do guia para formulação de perguntas pela ferramenta

Ainda está sendo estudado uma forma de outros desenvolvedores implementarem novos comandos a partir de suas necessidades. De saída o aluno irá receber o SQL referente à sua pergunta e o resultado que o mesmo obteve no banco, vide figura 7.

SELECT max(t1.idade) FROM funcionarios t1	
	idade
1	57

Figura 7: Exemplo de um resultado a partir de um pergunta no programa

4.Plano de Ação

O projeto final 1, figura 8, é focado no estudo sobre os componentes que farão parte do sistema. Esse cronograma foi dividido em quatro fases: estudo sobre SQL, estudo sobre compiladores, estudo sobre *natural language processing* (NPL), e definição do sistema. Segue uma explicação para fase do cronograma.

Primeiramente, serão aprofundados os estudos dos comandos SQL para se criar um escopo do que será implementado pela ferramenta. A princípio todos os comandos de DML (*data manipulation language*) estão previstos no escopo. Vale lembrar que para cada comando implementada existirá uma ou mais palavras no guia que o representem.

Na segunda fase do estudo com um tempo mais prolongado será avaliada a forma de implementação da ferramenta, que é um compilador de uma linguagem natural pré-definida para SQL. Nessa mesma fase será dado foco ao estudo de como pode ser garantido um funcionamento correto do sistema.

Para finalizar os estudos, foi adicionado o conhecimento sobre *natural language processing* (NLP). O intuito dessa fase é definir a liberdade que o aluno terá para elaborar suas perguntas. O programa terá um grande desafio para compreender a semântica das palavras, sinônimos entre elas e até suas ambiguidades. Esse estudo será muito importante para definir uma sintaxe que seja compatível e capaz de ser implementada na ferramenta.

Com os estudos anteriores concluídos já pode ser iniciada a etapa de definição do sistema, onde o objetivo é concretizar a arquitetura do mesmo, definir os seus componentes e especificar os requisitos.

TAREFA	Março	Abril	Maio	Junho
Elaboração da Proposta				
Estudo sobre SQL				
Estudo sobre Compiladores				
Estudo sobre NLP				
Definição do Sistema				
Elaboracao do Relatório Final				

Figura 8: Plano de ação para o projeto final 1

O projeto final 2, figura 9, será dividido por duas etapas principais, a fase de implementação teórica e prática sobre o sistema.

A primeira, consiste na especificação da ferramenta. Essencialmente, serão produzidos casos de uso, casos de teste, e diagramas de sequência.

Na próxima fase será iniciada a implementação da ferramenta, que terá início pelo compilador, com as seguintes etapas: léxica, sintática, semântica e geração de código. A ideia inicial será implementar o léxica e sintática de forma bem definida, com frases simples e poucas possibilidades de palavras. A partir desse ponto será iniciada uma evolução lenta das alternativas de escrita que a ferramenta estará apta a compreender. Essa evolução tem por base o conhecimento adquirido sobre NLP no projeto final 1.

A interface de entrada e saída será feita após o funcionamento do compilador, pois se necessita dele para que a interface funcione como desejado.

As funcionalidades, como guia, tabela do resultado e expansão da ferramenta por desenvolvedores, serão desenvolvidas no final e com baixa prioridade. O escopo de funcionalidades do sistema pode aumentar dependendo do tempo restante.

Ao final de todas as etapas, serão iniciados testes para medir o nível de acerto da ferramenta, sendo importante e esperado que ela tenha um alto nível de acerto por se tratar de uma ferramenta voltada para o ensino.

Para concluir será realizada uma apresentação de todo o projeto consolidado e uma demonstração prática da ferramenta.

TAREFAS	Julho	Agosto	Setembro	Outubro	Novembro
Especificação do Sistema					
Implementação do Compilador					
Implementação do Interface					
Implementação das Funcionalidades					
Realização de testes					
Elaboracao do Relatório Final					

Figura 9: Plano de ação para o projeto final 2

5. Referências bibliográficas

1. Coursera, INC; Mountain View, USA; <http://www.coursera.org>
2. Code School LLC; Orlando, USA; <http://www.codeschool.com/>
3. Hotwoofy Ltd; Sheffield, UK; <http://delver.io/>
4. Microsoft; Redmond, USA; <http://technet.microsoft.com/en-us/library/cc966482.aspx>
5. PrimeQue Ltd; Tel-Aviv Israel; <http://primeque.com/>