ProgramAR - Ferramenta para auxiliar o ensino em Álgebra Relacional

Arley Prates, Daniela Barreiro Claro, Edlane Proencia, João Queiroz

[1]Instituto de Matemática (IM) – Universidade Federal da Bahia (UFBA)
Av. Adhemar de Barros, s/n, Ondina – Salvador – BA – Brasil
Grupo de Pesquisa em Formalismos e Aplicações Semânticas (FORMAS)
Laboratório de Sistemas Distribuídos (LaSiD)

{arprates, edlaneproencia, jpqueiroz}@dcc.ufba.br, dclaro@ufba.br

Abstract. *The difficulty in teaching Algebra and Relational Calculus has been proven through the GPA in Database courses. The abstraction of these languages makes difficult the understanding and culminates with the demotivation in the sequence of the course. However, Algebra and Calculus are the foundations that support the SQL language and especially the foundations to optimize a DBMS. Thus, in order to facilitate and motivate learning, we developed a tool called ProgramAR which has as its main objective the description of expressions in Algebra and Relational Calculus which are converted to the SQL language. These expressions are executed over a DBMS, allowing students to view the results. In order to validate the tool, some tests were conducted with students of the Information Systems and Computer Science courses from the Federal University of Bahia.*

Resumo. *A dificuldade no ensino da Álgebra e do Cálculo Relacional tem sido comprovada através dos rendimentos nas disciplinas de Banco de Dados. A abstração destas linguagens dificulta a compreensão dos discentes e culmina com a desmotivação na sequencia didática. Porém, a Álgebra e o Cálculo são as bases que fundamentam a linguagem SQL e sobretudo os alicerces da otimização em um SGBD. Assim, com o intuito de facilitar e motivar o aprendizado, foi desenvolvida uma ferramenta denominada ProgramAR que tem por principal objetivo descrever expressões em Álgebra e Cálculo Relacional que são convertidas para a linguagem SQL. Estas expressões convertidas são executadas em um SGBD, permitindo que os discentes visualizem os resultados obtidos. Com o propósito de validar a ferramenta, alguns testes foram realizados com os discentes do Bacharelado em Sistemas de Informação e Ciência da Computação da Universidade Federal da Bahia.*

1. Introdução

A Álgebra Relacional é uma linguagem de consulta formal associada ao modelo relacional. Uma expressão em álgebra relacional é recursivamente definida como uma relação [RAMAKRISHNAN and GEHRKE 2008].

A aprendizagem da Álgebra Relacional e do Cálculo Relacional é fundamental para os discentes, pois, ambas as linguagens contribuem para a compreensão na manipulação em um SGBD relacional. O entendimento da Álgebra Relacional e do

Cálculo Relacional, quando ensinadas apenas sob o ponto de vista conceitual, dificulta a aprendizagem devido ao alto grau de abstração exigido. Além disso, a impossibilidade de visualizar os resultados de uma consulta em Álgebra ou Cálculo Relacional, torna menos atrativo o processo de aprendizagem.

A transformação de álgebra e do cálculo relacional em linguagem SQL permite que os resultados das consultas sejam retornados visto que há uma real manipulação de um SGBD. Diversas ferramentas foram desenvolvidas para realizar a conversão da álgebra relacional para a linguagem SQL, tais como EnsinAR[Paes 2004], SIMALG[Lautert 2010] e iDFQL [Appel and Traina Jr. 2004]. Porém, a maioria delas possuem limitações tais como: dificuldades na instalação, interface gráfica pouco intuitiva, falta de tratamento de erros nas expressões descritas e, sintaxe e simbologia não adotada por livros didáticos da área de Banco de Dados.

Assim, o presente trabalho propõe o desenvolvimento de uma ferramenta denominada ProgramAR, onde o discente pode escrever uma expressão em álgebra relacional e visualizar o resultado dessa consulta visto que esta expressão é traduzida para a linguagem SQL. O principal objetivo desta ferramenta é o aprendizado de álgebra relacional e do cálculo relacional com o intuito do discente compreender as suas manipulações e facilitar a compreensão de um SGBD, principalmente a linguagem SQL e a otimização das consultas.

Especificamente, este trabalho destaca as seguintes contribuições:

1. Interface gráfica para manusear as consultas em Álgebra Relacional;
2. Tratamento de erro das expressões descritas;
3. Tradução das expressões em Álgebra Relacional para a linguagem SQL;
4. Utilização dos operadores da Álgebra para realização de consultas em um SGBD;
5. Visualização dos resultados das consultas em Álgebra Relacional.

Esse artigo está organizado da seguinte maneira: a seção 2 apresenta conceitos de Álgebra Relacional, a seção 3 apresenta o funcionamento da ferramenta ProgramAR, a seção 4 demonstra as funcionalidades e a seção 5 descreve os trabalhos relacionados. A seção 6 apresenta o estudo de caso que foi utilizado para validar a ferramenta e por fim, a seção 7 as conclusões e os trabalhos futuros.

2. Álgebra Relacional

Os conhecimentos sobre álgebra relacional permitem entender a execução e otimização de consultas em Sistemas Gerenciadores de Bancos de Dados Relacionais (SGBDR) [Sumathi and Esakkirajan 2007]. As operações da Álgebra Relacional descrevem as consultas que podem ser especificadas para obter os dados de um SGBD. Uma sequência de operações da álgebra relacional forma uma expressão, cujo resultado também será uma relação que representa o resultado de uma consulta de banco de dados [ELMASRI and NAVATHE 2010].

Os operadores da Álgebra Relacional podem ser classificados como dois tipos básicos: os operadores unários e operadores binários. Dentre os operadores unários, destacam-se a Seleção, a Projeção e a Renomeação. Enquanto que os operadores binários, destacam-se as operações baseados na Teoria dos Conjuntos e as Junções.

2.1. Operadores relacionais unários

Uma operação é unária pois atua sobre uma única relação, produzindo um subconjunto apenas com os elementos especificados na condição.

A operação de ***Seleção*** é conhecida também como uma operação que filtra linhas de uma tabela, através da construção de um subconjunto horizontal de uma relação. Seu símbolo é: σ (sigma).

A operação de ***Projeção*** também é unária e produz um subconjunto com apenas as colunas escolhidas na condição especificada, filtrando as colunas destacadas na operação para a criação do novo conjunto. Este tipo de relação é vertical e o seu símbolo é: π (pi).

A operação ***Renomear*** redefine um nome para uma relação e/ou um relacionamento. Esta operação é muito utilizada em casos que contêm expressões relacionais grandes. Dessa forma, pode-se realizar a divisão da expressão em duas ou mais sub-expressões para obter o resultado esperado de forma mais simples. Sua simbologia é:ρ (rho).

2.2. Operações de álgebra relacional com base na teoria dos conjuntos

As principais operações com base na teoria dos conjuntos são a *União*, *Intersecção* e a *Diferença*.

A operação de ***União*** cria uma relação partindo de duas outras relações, estabelecendo assim, a união de todas as tuplas dessa nova relação. Seu símbolo é: $\bigcup$ (união). A união entre duas relações é realizada somente se as relações forem compatíveis, ou seja:

- As relações precisam ter o mesmo número de atributos;
- O domínio dos atributos de cada relação precisam ser os mesmos.

A operação de ***Interseção*** é uma operação binária que cria uma relação partindo de duas outras relações, estabelecendo assim a intersecção das duas, gerando uma nova relação onde estão presentes apenas os dados pertencentes às duas condições pré-estabelecidas. Seu símbolo é: $\bigcap$ (intersecção).

A operação de ***Diferença*** é uma operação que cria uma relação partindo de duas outras relações, o qual resultará em uma nova relação que contenha todas as tuplas que pertençam a primeira relação e não pertença a segunda. Seu símbolo é: $-$ (menos).

2.3. Operações binárias de junção

Além das operações binárias baseadas na teoria dos conjuntos, tem-se as operações de junções.

A operação ***Produto Cartesiano*** permite a combinação de dados de duas relações quaisquer. Assim a relação resultante conterá todas as tuplas da primeira relação relacionadas as da segunda relação. Seu símbolo é: $\times$ (produto cartesiano).

A operação ***Divisão*** resulta em uma tabela que contém apenas os elementos da primeira tabela que se relacionam com todos os elementos da segunda tabela. Seu símbolo é: $\div$ (divisão).

A operação de ***Junção*** tem como objetivo unir duas tabelas, as quais possuem um atributo em comum. Este tipo de operação é muito utilizado quando se trata de relacionamentos com chaves estrangeiras, em que é realizada uma seleção a partir dos atributos de um relacionamento sobre um produto cartesiano das tabelas. Seu símbolo é: $\triangleright\triangleleft$.

A operação ***Junção Natural*** é uma operação binária que tem como objetivo combinar tuplas de duas relações em uma única tupla. Esta operação é muito semelhante à junção, porém na junção natural não é preciso informar a condição de junção, visto que a mesma será formada por atributos que possuem nome e domínio iguais nas duas relações. Seu símbolo é: $*$.

A operação ***Junção Externa a Esquerda*** tem como objetivo combinar tuplas de duas relações em uma única tupla. Esta operação recebe como parâmetro as condições de junção que serão utilizadas para fazer a junção entre as relações. Para cada tupla da primeira relação (Esquerda) é verificado se existe alguma tupla na segunda relação (Direita) que satisfaça as condições de junção. Caso exista, as tuplas são combinadas para a formação de uma única tupla com os valores dos atributos das duas relações, caso contrário é criada uma tupla com os valores dos atributos da primeira relação (Esquerda) e os atributos da segunda relação (Direita) serão preenchidos com valores nulos [ELMASRI and NAVATHE 2010].

A operação ***Junção Externa a Direira*** é uma operação binária que tem como objetivo combinar tuplas de suas relações em uma única tupla. Esta operação recebe como parâmetro as condições de junção que serão utilizadas para fazer a junção entre as relações. Para cada tupla da segunda relação (Direita) é verificado se existe alguma tupla na primeira relação (Esquerda), que satisfaça as condições de junção. Caso exista, as tuplas são combinadas para a formação de uma única tupla com os valores dos atributos das duas relações, caso contrário é criada uma tupla com os valores dos atributos da segunda relação (Direta) e os atributos da primeira relação (Esquerda) serão preenchidos com valores nulos [ELMASRI and NAVATHE 2010].

A operação ***Junção Total*** tem como objetivo combinar tuplas de duas relações em uma única tupla. Esta operação recebe como parâmetro as condições de junção que serão utilizadas para fazer a junção na outra relação. Para cada tupla de uma relação é verificado se existe alguma tupla na outra relação que satisfaça as condições de junção. Caso exista, as tuplas são combinadas para a formação de uma única tupla com os valores dos atributos das relações, caso contrário é criada uma tupla com os valores dos atributos da relação em que a tupla foi selecionada e os valores dos atributos da outra relação serão nulos.

A operação ***Agregação*** é uma operação que tem como objetivo utilizar funções matemáticas de agregação, que recebem uma coleção de valores e retorna um único valor como resultado, como agrupamento de tuplas [ELMASRI and NAVATHE 2010]. Seu símbolo é: $\Im$.

A ***Atribuição*** é uma operação utilizada para atribuir a uma variável um determinado relacionamento. Seu símbolo é: $\leftarrow$.

Com o uso dos operadores da álgebra relacional, é possível transformar as expressões em Álgebra em uma consulta SQL. Além disso, muitos otimizadores de SGBD comerciais utilizam-se da Álgebra Relacional para a geração dos planos de execução das consultas, analisando as similaridades e eficiência entre diversos planos convertidos em Álgebra. Neste sentido, é importante dispor de uma ferramenta que permita aos discentes interagir e consequentemente aprender a Álgebra Relacional e os seus operadores.

3. Funcionamento da ProgramAR

Através da ferramenta ProgramAR, o discente pode escrever as expressões em Álgebra Relacional e verificar o resultado destas consultas. A tradução de uma expressão em Álgebra Relacional para a linguagem SQL requer um conhecimento específico em ambas as linguagens. Esta ferramenta tem como principal objetivo auxiliar o ensino da Álgebra Relacional e facilitar o aprendizado do aluno na manipulação das consultas relacionais. É importante salientar que a ProgramAR utiliza os operadores da Álgebra adotados no livro [ELMASRI and NAVATHE 2010] para facilitar a absorção do conhecimento, visto o mesmo ser um referencial teórico na área de Banco de Dados.

Segundo a Figura 1, a ferramenta ProgramAR executa uma consulta através de uma expressão em Álgebra Relacional seguindo as etapas:

1. Recebe a expressão em Álgebra Relacional escrita pelo discente;
2. Realiza uma *análise sintática* na expressão, para identificar possíveis erros. Caso algum erro seja identificado, é informado ao discente o tipo de erro e a posição do erro na expressão;
3. Caso não tenha sido encontrados erros na análise sintática, a ferramenta faz uma *transformação* da expressão em Álgebra Relacional para SQL;
4. O SQL, gerado na etapa anterior, é executado em um SGBD e o resultado da consulta é enviado ao ProgramAR;
5. A ferramenta disponibiliza o resultado da consulta para o discente.

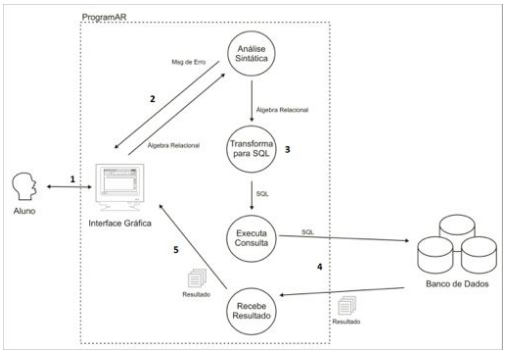

Figura 1. Funcionamento da ferramenta ProgramAR.

3.1. Análise Sintática

A ferramenta ProgramAR possui dois módulos que realizam a análise sintática da expressão em Álgebra Relacional. O primeiro módulo é acionado na medida em que o discente digita a expressão. A cada caractere digitado pelo discente, o módulo é acionado e uma análise sintática da expressão digitada até o momento, é realizada. Este módulo foi criado para facilitar a verificação de erros de sintaxe e para facilitar a visualização da expressão pelo discente.

A cada execução deste módulo são identificadas as partes que compõem a expressão. Estas partes são classificadas como *Norma*, *Operador*, *Sobrescrito*, *Renomear*

Relação ou *Erro*. Cada classificação possui sua fonte, cor e tamanho especifico. Essas características são utilizadas pela ferramenta para fazer uma diferenciação das diversas partes que compõem uma expressão, facilitando, assim, a visualização da expressão.

O segundo módulo é acionado quando o usuário realiza uma consulta a um SGBD utilizando uma expressão em Álgebra Relacional. Este módulo recebe como entrada a expressão digitada pelo discente e faz a verificação da sintaxe da expressão. Caso seja encontrado algum erro de sintaxe, este é informado ao usuário indicando a sua respectiva posição na expressão.

Os dois módulos de análise sintática da ferramenta foram criados tendo como base o conceito de Autômato Finito Determinístico que é um modelo matemático utilizado para reconhecer uma linguagem, que recebe como entrada uma cadeia x e responde *"sim"* se x for uma sentença da linguagem e *"não"* em caso contrário [ULLMAN et al. 1995]. Assim, os módulos recebem a expressão como entrada e realizam uma verificação na expressão, percorrendo todos os caracteres, conforme a Figura 2.

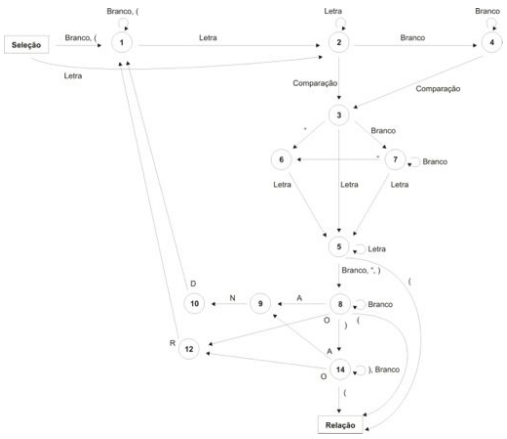

Figura 2. Autômato do Operador Seleção

Para cada operador da Álgebra foi criada uma função que realiza o reconhecimento do respectivo operador. A Figura 2 apresenta o autômato criado para o reconhecimento do operador *Seleção* que possui a seguinte sintaxe:

$$\sigma_{(condição\ de\ seleção)}(Relação)$$

O símbolo σ é utilizado para denotar o operador *seleção* e a *condição de seleção* é denotada por uma expressão booleana que utiliza os atributos da *relação*. A expressão booleana é construída da seguinte forma:

(*atributo*) (*comparação*) (*atributo*)

ou

(*atributo*) (*comparação*) (*valor constante*)

O *atributo* é o nome de um atributo da relação e a *comparação* pode ser um dos operadores $\{>, <, =, \neq, \geq, \leq\}$. Estas cláusulas podem ser conectadas com os operadores booleanos $\{AND, OR\}$ para a formação de novas condições de seleção.

3.2. Tradução da Álgebra Relacional para SQL

Após a realização da análise sintática, todas as partes que compõem a expressão em Álgebra Relacional foram devidamente armazenadas nas variáveis do sistema, para que sejam utilizadas na tradução para o SQL. O armazenamento é feito nas seguintes variáveis: *Nome*, *Predicado*, *Atributos*, *Renomear Atributo*, *Renomear Relação*, *Relação*, *Condições de Join*, *GroupBy* e *Operador Binário*. Após a completa formulação da variável, a ferramenta realiza a tradução da Álgebra para SQL.

Embora as etapas que correspondem à execução de uma consulta em SQL por meio de um SGBD específico sejam importantes, a descrição e o detalhamento das mesmas não fazem parte do escopo deste trabalho.

4. Ferramenta ProgramAR

A ferramenta ProgramAR foi desenvolvida na linguagem Java e utilizada nos seguintes Sistemas Operacionais: Windows (32bits e 64 bits), Linux.

Nesta versão 1.0 da ProgramAR, somente a Álgebra Relacional foi implementada. Especificamente, os operadores implementados foram: *seleção, projeção, união, produto cartesiano, junção, junção externa total, junção externa a direita, junção externa a esquerda, renomear, junção natural e funções de agregação*.

Os operadores *Interseção, Diferença e Divisão* não foram implementados com prioridade devido a sua baixa utilização, mas já estão previstos para a próxima versão.

4.1. Interface Gráfica

A Figura 3 apresenta a interface gráfica da ferramenta ProgramAR com um menu dividido em dois: a) Criar Conexão com SGBD e b) Consulta SGBD.

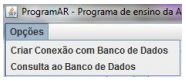

Figura 3. Opções do Menu

Ao *Criar Conexão com Banco de Dados*, o discente pode catalogar suas conexões com o SGBD, conforme pode ser observado na Figura 4. Todas as conexões criadas são armazenadas localmente. Assim, o discente não precisa se conectar ao SGBD toda vez que executar a ferramenta.

Já no menu *Consulta ao Banco de Dados*, detalhado na Figura 5, o discente cria as expressões em Álgebra Relacional e verifica o resultado das consultas. Esta janela é dividida em três áreas principais: Álgebra Relacional, SQL e Resultado.

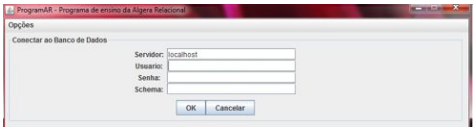

Figura 4. Criar conexão com banco de dados

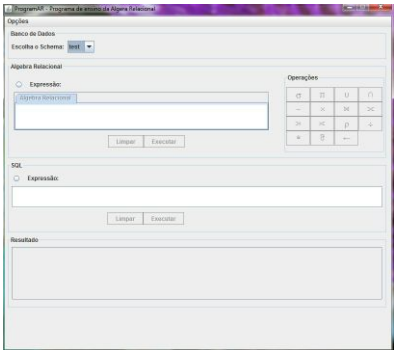

Figura 5. Consulta em Álgebra Relacional

4.2. Álgebra Relacional

Na área *Álgebra Relacional*, o discente deve escrever as expressões em Álgebra Relacional utilizadas na realização das consultas no SGBD específico.

Nesta área, foi desenvolvido um quadro com os operadores da álgebra conforme o livro [ELMASRI and NAVATHE 2010]. À medida que o discente vai escrevendo a expressão em Álgebra, a mesma vai sendo customizada e corrigida, facilitando assim a compreensão. Exemplificando, os símbolos e palavras reservadas ficam na cor azul enquanto que o nome dos atributos, condições de junção e condição de seleção ficam sobrescritos. Caso o discente escreva alguma expressão que possua erro de sintaxe, da posição do erro até o fim da mesma, as letras ficam com a cor vermelha, facilitando assim a identificação do erro.

4.3. SQL

Atualmente, nesta versão do ProgramAR, esta área *SQL* corresponde à expressão em Álgebra traduzida para a linguagem SQL que é utilizada para a consulta ao SGBD designado, conforme pode ser observado na Figura 6.

4.4. Resultado

A área *Resultado* é utilizada para a visualização do resultado de todas as consultas efetuadas no SGBD. As informações são organizadas em uma tabela para facilitar a visualização dos resultados, conforme Figura 6.

Figura 6. Resultado de uma consulta em Algebra Relacional

5. Trabalhos Relacionados

Diversos trabalhos foram analisados, dos quais alguns foram destacados por se tratar de softwares educacionais para a área de Banco de Dados, especificamente os que utilizam algumas técnicas da Álgebra Relacional.

Autores em [Pereira 2012] propuseram uma análise dos ambientes de Ensino de Banco de Dados, os quais destacaram um comparativo entre as principais ferramentas sobre Álgebra relacional, MER e SQL. A partir dessa análise comparativa criteriosa, os autores propuseram um modelo de ambiente capturando as vantagens dos ambientes analisados e corrigindo algumas de suas desvantagens. O ambiente contempla vários requisitos e promete possuir bons resultados, porém ainda não há versões que foram implementadas e disponibilizadas.

Autores em [Appel and Traina Jr. 2004] propuseram desenvolver uma ferramenta de consulta baseado no construcionismo denominada iDFQL (Interactive Data Flow Query Language), cujo objetivo é auxiliar o processo de ensino-aprendizagem da álgebra relacional usando elementos gráficos para representar consultas. A iDFQL permite aos alunos criar e executar consultas de forma interativa, tornando o aprendizado desta linguagem mais estimulante e intuitivo. Diferente da proposta neste trabalho, os autores propõem a construção de árvores de expressões algébricas.

Autores em [Paes 2004] desenvolveram uma ferramenta para o apoio do ensino de álgebra relacional e banco de dados denominada EnsinAR. Esta ferramenta baseia-se na conversão de expressões escritas em álgebra relacional para a SQL (Structured Query Language). A expressão resultante em SQL é então exibida e executada e o resultado é apresentado. A ferramenta é dotada de um compilador desenvolvido: o GALS (Gerador de Analisadores Léxicos e Sintáticos). A execução das consultas e gerenciamento das relações e atributos criados durante a utilização é feita através do SGBD SQL Server Express da Microsoft. Apesar deste trabalho usar expressões algébricas, os autores não evidenciam a utilização de SGBDs diferentes. Além disso, é preciso criar as tabelas na própria aplicação, não contemplando a utilização de uma ferramenta de MER, restringindo assim a sua manipulação.

6. Estudo de Caso

Com o intuito de validar a ferramenta desenvolvida, um estudo de caso foi aplicado aos alunos da disciplina Banco de Dados dos cursos de Bacharelado em Ciência da Computação e Bacharelado em Sistemas de Informação da Universidade Federal da Bahia. Foram desenvolvidos roteiros os quais foram seguidos pelos discentes utilizando a ferramenta ProgramAR.

Primeiramente, foi proposto que os discentes executassem as consultas abaixo e que a escrevessem em Álgebra Relacional na ferramenta ProgramAR, conforme as Figuras 7, 8 e 9:

1. *Recuperar os nomes de todos os empregados do departamento 5 que trabalhem mais de dez horas por semana no projeto 'Product X'.*[1]

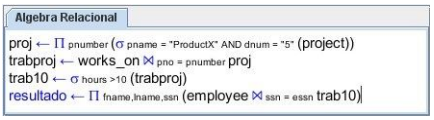

Figura 7. Consulta em Álgebra Relacional na ProgramAR

2. *Para cada projeto, liste o nome do projeto e o total de horas por semana (de todos os empregados) gastas no projeto.*

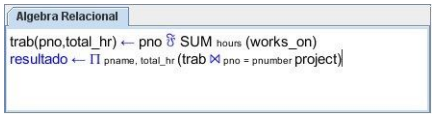

Figura 8. Consulta em Álgebra Relacional na ProgramAR

3. *Recupere o nome do Projeto e os nomes de todos os empregados que trabalhem em pelo menos um projeto.*

Figura 9. Consulta em Álgebra Relacional na ProgramAR

O experimento 1 verifica a utilização dos operadores *seleção, projeção e junção*. Já no experimento 2, observa-se o uso operador ℑ (agregação) na primeira linha, onde recebe uma coleção de valores e retorna um único valor como resultado, realizando a sumarização de uma coleção de valores de um determinado atributo.

O experimento 3 faz uso do operador *Join*, porém, destaca que após o *essn* foi colocado um espaço e então o essn não foi detectado como uma condição de junção, ficando todo o resto da frase em vermelho, sinalizando ao discente que há um erro na expressão gerada, conforme a Figura 10.

Ao final dos experimentos, um questionário sobre a ferramenta foi entregue aos discentes e a maioria informou que não houve dificuldades em executar os comandos pela ferramenta; a formatação (cores, fontes, tamanho) auxiliou na visualização da expressão facilitando a detecção de inconsistências; as funcionalidades dos operadores da álgebra

[1]As consultas foram escritas em inglês para seguir o mesmo exemplo disponibilizado no livro [ELMASRI and NAVATHE 2010].

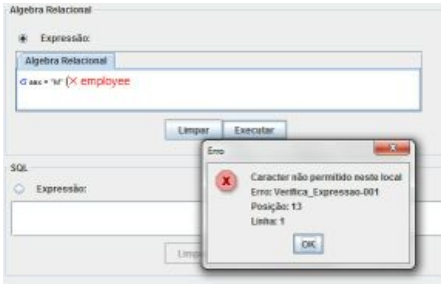

Figura 10. Mensagem de Erro

relacional relatadas foram fáceis de identificar devido a utilização de novas fontes de cores; os operadores utilizados na sala de aula foram os mesmos utilizados na ferramenta facilitando o aprendizado; e por fim a ferramenta tornou-se um ambiente gráfico e interativo que auxiliou o ensino da álgebra relacional.

Os tutoriais de instalação e configuração de um SGBD foram disponibilizados no site da ferramenta, onde há outros exemplos de consultas, além de um arquivo de backup de um SGBD para testes. Alguns discentes notificaram a dificuldade de instalação e configuração do SGBD na plataforma Linux; nenhuma dificuldade na plataforma Windows foi detectada.

7. Conclusão e trabalhos futuros

A ferramenta ProgramAR tem por principal objetivo automatizar o processo de visualização dos resultados das consultas em Álgebra Relacional por meio da transformação em linguagem SQL. Os discentes podem estabelecer a conexão com o SGBD através de uma interface gráfica intuitiva e assim proceder as consultas em Álgebra Relacional.

A visualização dos resultados contribui para o ensino-aprendizagem do conteúdo da disciplina, visto que sem uma ferramenta como a ProgramAR não é possível conferir os resultados das expressões e principalmente avaliar se os resultados estão de acordo com o esperado. Assim, as funcionalidades presentes nessa versão da ferramenta potencializam a aprendizagem dos conceitos de Álgebra Relacional.

Como trabalhos futuros, pretende-se desenvolver uma versão da ProgramAR para plataforma Web e a implementação do Cálculo Relacional, onde as consultas serão realizadas a partir de expressões em Cálculo Relacional de Tupla e de Domínio.

Agradecimentos

Os autores gostariam de agradecer ao Programa Permanecer, Projeto número 2231 com Plano de Atividade número 4946 da Universidade Federal da Bahia.

Referências

Appel, A. P. and Traina Jr., C. (2004). *iDFQL - Uma Ferramenta de Apoio ao Processo de Ensino-Aprendizagem da Álgebra Relacional Baseado no Construcionismo*. PhD thesis, Instituto de Ciências Matemáticas e de Computação - USP. São Carlos, SP.

ELMASRI, R. and NAVATHE, S. B. (2010). *Sistemas de banco de dados*. Pearson Addison Wesley.

Lautert, L. R. (2010). *Implementação de um Simulador de Consultas em Àlgebra Relacional*. PhD thesis, Trabalho de Graduação. Universidade Federal de Santa Maria. Centro de Tecnologia. Curso de Ciência da Computação. Santa Maria, RS.

Paes, E. L. (2004). *EnsinAR: Ferramenta Didática para o Ensino de Álgebra Relacional*. PhD thesis, Universidade Federal de Santa Catarina - Centro Tecnológico. Departamento de Informática e Estatística. Florianópolis.

Pereira, Juliana Alves; Resende, A. M. P. d. (2012). *Uma Análise dos Ambientes de Ensino de Banco de Dados*. PhD thesis, Universidade Federal de Lavras - Departamento de Ciência da Computação - UFLA. Lavras, MG.

RAMAKRISHNAN, R. and GEHRKE, J. (2008). *Sistemas de Gerenciamento de Banco de Dados*. McGraw Hill, 3rd edition.

Sumathi, S. and Esakkirajan, S. (2007). *Fundamentals of Relational Database Management Systems*. Springer.

ULLMAN, J. D., AHO, A. V., and SETHI, R. (1995). *Compiladores - Princípios, Técnicas e Ferramentas*. LTC.