



Universidade Federal da Fronteira Sul

Nome dos discentes

Loude Djema Sime

Sebastien Lionel Lubin

Lindy Saphira Elmius

Ashley Saint-Louis

Kerby Lovince

Nome do docente: Denio Duarte

Tarefa: Finding Bugs (uffbdb)

Data 17/11/2024

[Bug 1]

Mensagens de Erro Genéricas no UFFSDB

Um dos principais problemas identificados no UFFSDB é a falta de detalhamento nas mensagens de erro retornadas pelo banco de dados. Quando ocorre uma falha, especialmente em consultas complexas, o sistema retorna apenas "syntax error" ou "Expected object name", sem indicar exatamente o motivo do erro, a linha ou o trecho problemático. Isso dificulta a depuração e a identificação da causa do problema.

```
uffsdb=# SELECT crm, COUNT(*) AS total
FROM Medico
GExpected object name.
ERROR: syntax error.
uffsdb=# ERROR: syntax error.
uffsdb=# ERROR: syntax error.
uffsdb=# GROUP BY crm
HAVERROR: syntax error.
uffsdb=# ERROR: syntax error.
ING uffsdb=# ERROR: syntax error.
uffsdb=# COUNT(*) > 1;
ERROR: syntax error.
uffsdb=# ERROR: syntax error.
uffsdb=# ERROR: syntax error.
uffsdb=# ERROR: syntax error.
uffsdb=# ERROR: syntax error.
uffsdb=# ERROR: syntax error.
uffsdb=# ERROR: syntax error.
uffsdb=# uffsdb=# SELECT rg, COUNT(*) AS total
FRExpected object name.
ERROR: syntax error.
OM Puffsdb=# paciente
ERROR: syntax error.
```

Cenários de Erro:

Exemplo 1: Consulta com 'GROUP BY' Comando: SELECT crm, COUNT(*) AS total FROM Medico GROUP BY crm HAVING COUNT(*) > 1;
Mensagem de erro: ERROR: syntax error.

Possível causa:

- Limitação no suporte a 'GROUP BY' e 'HAVING'.
- Palavras reservadas podem ser interpretadas de forma incorreta.

Exemplo 3: Consulta Simples Comando: SELECT * FROM Paciente; Mensagem de erro: ERROR: Expected object name.

Impacto:

Essas mensagens genéricas dificultam a depuração, obrigando o desenvolvedor a testar incrementalmente o comando ou tentar simplificá-lo até identificar a causa do erro.

[BUG 2]

Mensagens de loop erro comando UPDATE

Ao tentar atualizar um registro na tabela paciente, ocorre uma mensagem de erro de "syntax error" ao executar o comando UPDATE.

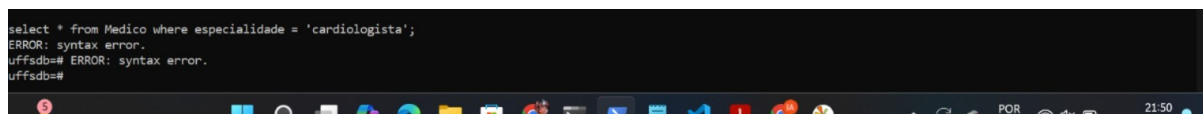
Comando testado:

```
UPDATE paciente SET idade = 18 WHERE rg = 1158750;
```

A atualização do registro na tabela paciente deveria ser realizada com sucesso, alterando o campo idade para 18 onde rg = 1158750, sem exibir mensagens de erro.

[illegible]

[BUG #3]

A screenshot of a terminal window with a dark background. The text in the terminal is white and shows a SQL query followed by an error message. The query is 'select * from Medico where especialidade = 'cardiologista';'. Below it, the error message reads 'ERROR: syntax error.' followed by two lines of hexademical data: 'ufffdb=# ERROR: syntax error.' and 'ufffdb=#'. The terminal window is part of a desktop environment, with a taskbar visible at the bottom showing various application icons and system status indicators like 'POR', signal strength, and the time '21:50'.

Mensagem de error a executar Select:

Comando:

```
select * from Medico where especialidade = 'cardiologista';
```

Ao tentar consultar registros na tabela Medico, ocorre uma mensagem de erro de "syntax error" ao executar o comando SELECT.

A consulta deveria retornar todos os registros da tabela Medico onde a coluna especialidade seja igual a 'cardiologista'. No entanto, o comando gera um erro de sintaxe e impede a execução correta da consulta.

O comportamento esperado é que o comando seja executado sem erros, retornando os registros correspondentes.

[Bug 4]

Erro em Consultas com Subconsultas ('NOT IN')

```
uffsdb=# SELECT * FROM Medico WHERE numeroA NOT IN (SELECT numeroA FROM Ambulatorio);  
Expected object name.  
ERROR: syntax error.  
uffsdb=# SELECT * FROM Consulta WHERE.crm NOT IN (SELECT.crm FROM Medico);  
Expected object name.  
ERROR: syntax error.  
uffsdb=# SELECT * FROM Medico WHERE numeroA NOT IN (SELECT.numeroA FROM Ambulatorio);  
Expected object name.  
ERROR: syntax error.  
uffsdb=# SELECT * FROM Medico WHERE numeroA NOT IN (SELECT.numeroA FROM Ambulatorio);  
Expected object name.  
ERROR: syntax error.  
uffsdb=#
```

Descrição: Consultas que utilizam subconsultas, como 'NOT IN', falham ao verificar relacionamentos inexistentes.

Exemplo: SELECT * FROM Medico WHERE numeroA NOT IN (SELECT numeroA FROM Ambulatorio);

Mensagem de erro: ERROR: syntax error.

Causa provável: Limitações no interpretador SQL para lidar com subconsultas.

Solução:

- Testar com consultas mais simples para isolar o problema: SELECT numeroA FROM Ambulatorio; SELECT * FROM Medico WHERE numeroA = 999;

[BUG 5]

Erro ao fazer uma inserção de múltiplos dados

Durante a inserção de múltiplos registros nas tabelas `Medico` e `Paciente` na database \c banco. Ao tentar inserir novos registros, o sistema apresenta mensagens de erro inesperadas e interrompe o processo de inserção.

```
root@Loude: /mnt/c/Users/Loude/OneDrive/Documentos/Semestre 2024.2/Banco de dados/uffsdB/Fonte#
insert into Medico values (015875135,'DrRicardo','38','Xanxere','dermatologista',317);
insert into Medico values (015875136,'DrRicardo','38','Xanxere','dermatologista',317);
insert into Medico values (015875137,'DrRicardo','38','Xanxere','dermatologista',317);
insert into Medico values (015875138,'DrRicardo','38','Xanxere','dermatologista',317);
insert into Medico values (015875139,'DrRicardo','38','Xanxere','dermatologista',317);
insert into Medico values (015875140,'DrRicardo','38','Xanxere','dermatologista',317);
insert into Medico values (015875141,'DrRicardo','38','Xanxere','dermatologista',317);
insert into Medico values (015875142,'DrRicardo','38','Xanxere','dermatologista',317);
insert into Medico values (015875143,'DrRicardo','38','Xanxere','dermatologista',317);
insert into Medico values (015875144,'DrRicardo','38','Xanxere','dermatologista',317);
insert into Medico values (015875145,'DrRicardo','38','Xanxere','dermatologista',317);
insert into Medico values (015875146,'DrRicardo','38','Xanxere','dermatologista',317);
insert into Medico values (015875147,'DrRicardo','38','Xanxere','dermatologista',317);

insert into Paciente values (1158750,'pigor','18','Palmitos','catarata');
insert into Paciente values (1158751,'pMarcos','15','Chapeco','sarampo');
insert into Paciente values (1158752,'pPedro','6','Palmas','tetano');
insert into Paciente values (1158753,'pPaulo','3','Pato Branco','pneumonia');
insert into Paciente values (1158754,'pRegis','9','Florianopolis','sarampo');
insert into Paciente values (1158755,'pEliton','11','Caibi','tetano');
insert into Paciente values (1158756,'pRicardo','8','Xanxere','catarata');
insert into Paciente values (1158757,'pRicardo','8','Xanxere','catarata');
insert into Paciente values (1158758,'pRicardo','8','Xanxere','catarata');
insert into Paciente values (1158759,'pRicardo','8','Xanxere','catarata');
insert into Paciente values (1158760,'pRicardo','8','Xanxere','catarata');
banco# banco# banco(# banco(# banco(# banco(# ere','catarata');
insert into Paciente values (1158761,'pRicardo','8','Xanxere','catarata');
insert into Paciente values (1158762CREATE TABLE
banco# banco# banco#,'pRicardo','8','Xanxere','catarata');
insert ifree(): invalid pointer
Aborted
root@Loude: /mnt/c/Users/Loude/OneDrive/Documentos/Semestre 2024.2/Banco de dados/uffsdB/Fonte# insert into Ambulatorio values (312,2,27);
-bash: syntax error near unexpected token '('
root@Loude: /mnt/c/Users/Loude/OneDrive/Documentos/Semestre 2024.2/Banco de dados/uffsdB/Fonte# insert into Ambulatorio values (313,4,18);
-bash: syntax error near unexpected token '('
root@Loude: /mnt/c/Users/Loude/OneDrive/Documentos/Semestre 2024.2/Banco de dados/uffsdB/Fonte# insert into Ambulatorio values (314,1,57);
-bash: syntax error near unexpected token '('
atroot@Loude: /mnt/c/Users/Loude/OneDrive/Documentos/Semestre 2024.2/Banco de dados/uffsdB/Fonte# insert into Ambulatorio values (315,5,37);
-bash: syntax error near unexpected token '('
```

Como ele é gerado de forma que seja possível reproduzi-lo.

- Execução de inserção em massa na tabela `Medico` ou `Paciente`.
- Observamos que, após várias inserções, o sistema começa a exibir mensagens de erro repetidas (por exemplo, `CREATE TABLE`, `invalid pointer`) e para de responder.

BUG [6]

Mensagens de erro em português:

Teste com uma coluna que não existe na tabela Medico.

Ao fazer o select: SELECT teste FROM Medico;

Saiu a seguinte mensagem: “A coluna da projecao teste não pertence a tabela.”

```
mydb=# SELECT teste
FROM Medico;mydb-#
A coluna da projecao teste não pertence a tabela.
mydb=#
```

sqlcommands.c

```

598 //select * from t4;
599 int validaProj(Lista *proj,column *colunas,int qtdColunas,char *pert){
600     if(proj->tam == 1 && ((char *)proj->prim->inf)[0] == '*'){
601         free(rmvNodoPtr(proj,proj->prim));
602         proj->prim = proj->ult = NULL;
603         for(int j = 0;j < qtdColunas; j++){
604             pert[j] = 1;
605             char *str = malloc(sizeof(char)*strlen(colunas[j].nomeCampo));
606             strcpy(str,colunas[j].nomeCampo);
607             adcNodo(proj,proj->ult,str);
608         }
609         return 1;
610     }
611     char *validar = malloc(sizeof(char)*proj->tam);
612     for(int j = 0; j < proj->tam; j++) validar[j] = 0;
613     for(int j = 0; j < qtdColunas; j++){
614         int k = 0;
615         for(Nodo *it = proj->prim; it; it = it->prox,k++){
616             pert[j] |= (strcmp((char *)it->inf,colunas[j].nomeCampo) == 0);
617             validar[k] |= pert[j];
618         }
619     }
620     int k = 0;
621     for(Nodo *it = proj->prim; it; it = it->prox,k++){
622         if(!validar[k]){
623             free(validar);
624             printf("A coluna da projecao %s não pertence a tabela.\n",(char *)it->inf);
625             return 0;
626         }
627     }
628     free(validar);
629     return 1;
630 }

```

Teste usando Where para uma coluna que não existe na tabela Medico.

Ao fazer o select: Select * from Medico where teste = 'cardiologista';
 Saiu a seguinte mensagem: "A coluna teste não pertene a tabela."

```

mydb=# Select * from Medico where teste = 'cardiologista';
A coluna teste não pertene a tabela.
mydb=#

```



```

659 int validaColsWhere(Lista *tok,column *colunas,int qtdColunas){
660     if(!tok) return 1;
661     for(Nodo *i = tok->prim; i; i = i->prox){
662         inf_where *iw = (inf_where *)i->inf;
663         if(iw->id == OBJETO){
664             int achou = 0;
665             char *str = (char *)iw->token;
666             for(int j = 0; !achou && j < qtdColunas; j++)
667                 achou = (strcmp(str,colunas[j].nomeCampo) == 0);
668             if(!achou){
669                 printf("A coluna %s não pertence a tabela.\n",str);
670                 return 0;
671             }
672         }
673     }
674     return 1;
675 }

```

Teste usando Where com Tipos diferentes na tabela Medico.

Ao fazer o select: SELECT * FROM Medico where crm = 15875136;
 Saiu a seguinte mensagem: "Operação entre tipos diferentes inválida."

```

mydb=# SELECT *
FROM Medico
where crm = 'pediatra';mydb=# mydb=#
Operação entre tipos diferentes inválida.
mydb=#

```

Expressao.c

```

81 Lista *resArit(Lista *l,Lista *t){
82     if(!validaExp(l,t)){
83         printf("Operação entre tipos diferentes inválida.\n");
84         return NULL;
85     }
86     Lista *op = novalista(NULL);

```

BUG[7]

Carrega lixo do Buffer na criação de uma nova tabela.

Foi testado várias criação de nova Tabela:

Criação da tabela “testebuffer”: O comando foi digitado manualmente no terminal.

Resultado: A tabela foi criada com sucesso e não houve a exibição de lixo no terminal.

Criação da tabela “testebuffer2”: O comando foi colado no terminal usando CTRL + SHIFT + V.

Resultado: A tabela foi criada com sucesso e não houve a exibição de lixo no terminal.

Criação da tabela “testebuffer3”: O comando foi colado no terminal usando CTRL + SHIFT + V, mas foi dividido em múltiplas linhas para melhorar a legibilidade.

Resultado: A tabela foi criada, mas houve a exibição de lixo no terminal.

Esses testes indicam que o problema ocorre quando o comando SQL é colado no terminal e quebra em várias linhas, possivelmente devido a caracteres ocultos ou problemas de formatação que interferem na leitura do buffer.

```

public | visit | tuple | mydb
(7 rows)

mydb=# CREATE TABLE testebuffer ( codigo integer, nome varchar(23), data varchar(30));
CREATE TABLE
mydb=# CREATE TABLE testebuffer2 ( codigo integer, nome varchar(23), data varchar(30));
CREATE TABLE
mydb=# CREATE TABLE testebuffer3 (
        codigo integer,
        nome varchar(23),
        data varchar(30));mydb=# mydb=# mydb=#
CREATE TABLE
mydb=# CREATE TABLE testebuffer3 (
        codigo integer,
        nome varchar(23),
        data varchar(30)
        idade integer);
mydb=# mydb=# mydb=# mydb=# ERROR: syntax error.
mydb=# \d
                List of Relations
Schema | Name | Type | Owner
-----+-----+-----+-----
public | ambulatorio | tuple | mydb
public | medico | tuple | mydb
public | paciente | tuple | mydb
public | consulta | tuple | mydb
public | funcionario | tuple | mydb
public | persons | tuple | mydb
public | visit | tuple | mydb
public | testebuffer | tuple | mydb
public | testebuffer2 | tuple | mydb
public | testebuffer3 | tuple | mydb
(10 rows)

mydb=#

```

Outro exemplo:

```
ashley@ashley-550XDA:~/Documentos/Quinta Fase/BD II/ufffdb/Fonte$ ./ufffdb
ufffdb (16.2).
Type "help" for help.
```

```
You are now connected to database "ufffdb" as user "ufffdb".
```

```
ufffdb=# CREATE DATABASE MyDB;
```

```
ERROR: database already exists
```

```
ufffdb=# \c mydb
```

```
You are now connected to database "mydb" as user "ufffdb".
```

```
mydb=# create table Ambulatorio (
        numeroA integer primary key,
        andar integer,
        capacidade integer);mydb=# mydb=# mydb(
```

```
CREATE TABLE
```

```
mydb=# create table Medico (
        crm integer primary key,
        nome varchar(30),
        idade integer,
        cidade varchar(30),
        especialidade varchar(30),
        numeroA integer references Ambulatorio(numeroA));
```

```
mydb=# mydb=# mydb=# mydb=# mydb=# mydb(
```

```
mydb=# create table Paciente (
        rg integer primary key,
        nome varchar(30),
        idade integer,
        cidade varchar(30),
        doenca varchar(30));mydb=# mydb=# mydb=# mydb(
```

```
CREATE TABLE
```

```
mydb=# create table Consulta (
        crm integer primary key,
        rg integer references Paciente(rg),
        data varchar(40),
        hora varchar(40));mydb=# mydb=# mydb(
```

```
CREATE TABLE
```

```
mydb=# create table Funcionario (
        rg integer primary key,
        nome varchar(30),
        idade integer,
        cidade varchar(30),
        salario double);mydb=# mydb=# mydb(
```

```
CREATE TABLE
```

```
mydb=# █
```

BUG[8]

Inserção valor NULL na Tabela Funcionario

Ao tentar inserir um valor NULL na tabela Funcionario, a inserção não é aceita, mesmo que as colunas da tabela não tenham restrições que impeçam valores NULL.

```
create table Funcionario (  
    rg integer primary key,  
    nome varchar(30),  
    idade integer,  
    cidade varchar(30),  
    salario double);
```

Comandos de teste:

```
insert into Funcionario values (2158750,'fLucas',20);  
insert into Funcionario values (2158778,'fLucas',20,null,null);  
INSERT INTO Funcionario (rg, nome, idade, cidade, salario) VALUES (101, 'Carlos',  
NULL, 'Rio de Janeiro', NULL);
```

```
mydb=# insert into Funcionario values (2158750,'fLucas',20);  
ERROR: INSERT has more expressions than target columns.  
mydb=# insert into Funcionario values (2158778,'fLucas',20,null,null);  
Expected token "VALUES" after object name.  
ERROR: syntax error.  
mydb=# INSERT INTO Funcionario (rg, nome, idade, cidade, salario) VALUES (101, '  
Carlos', NULL, 'Rio de Janeiro', NULL);  
Expected token "VALUES" after object name.  
ERROR: syntax error.  
mydb=#
```