**Universidade Federal de Itajubá UNIFEI**



**Trabalho final de COM 231 (Banco de dados II)**

**Prof(a). Vanessa Souza**

**Tema: Sistema de Gerenciamento de Acidentes**

**Equipe:**

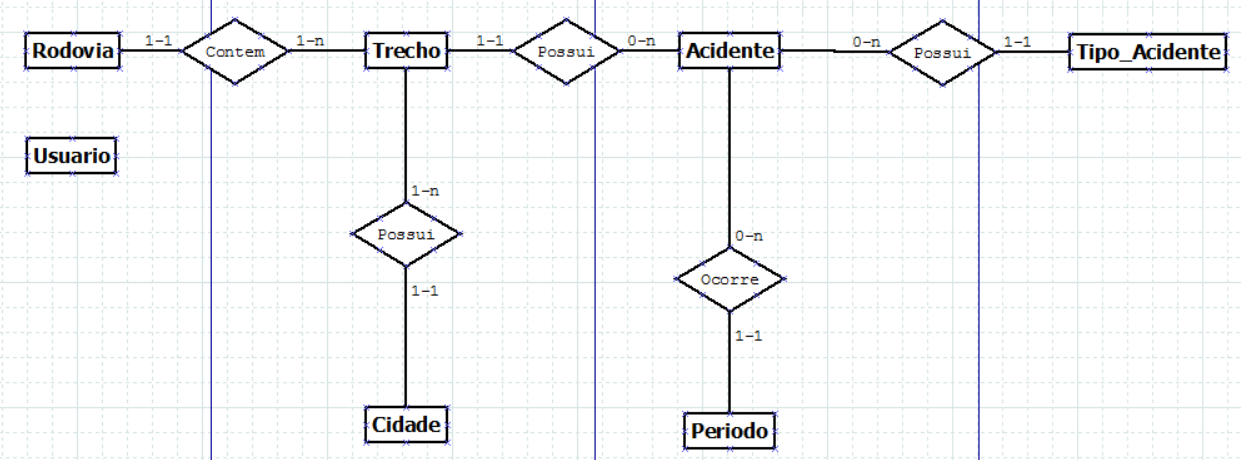
**Alifi Cleiton da Silva - 31412**

**Edmar José Benini - 28971**

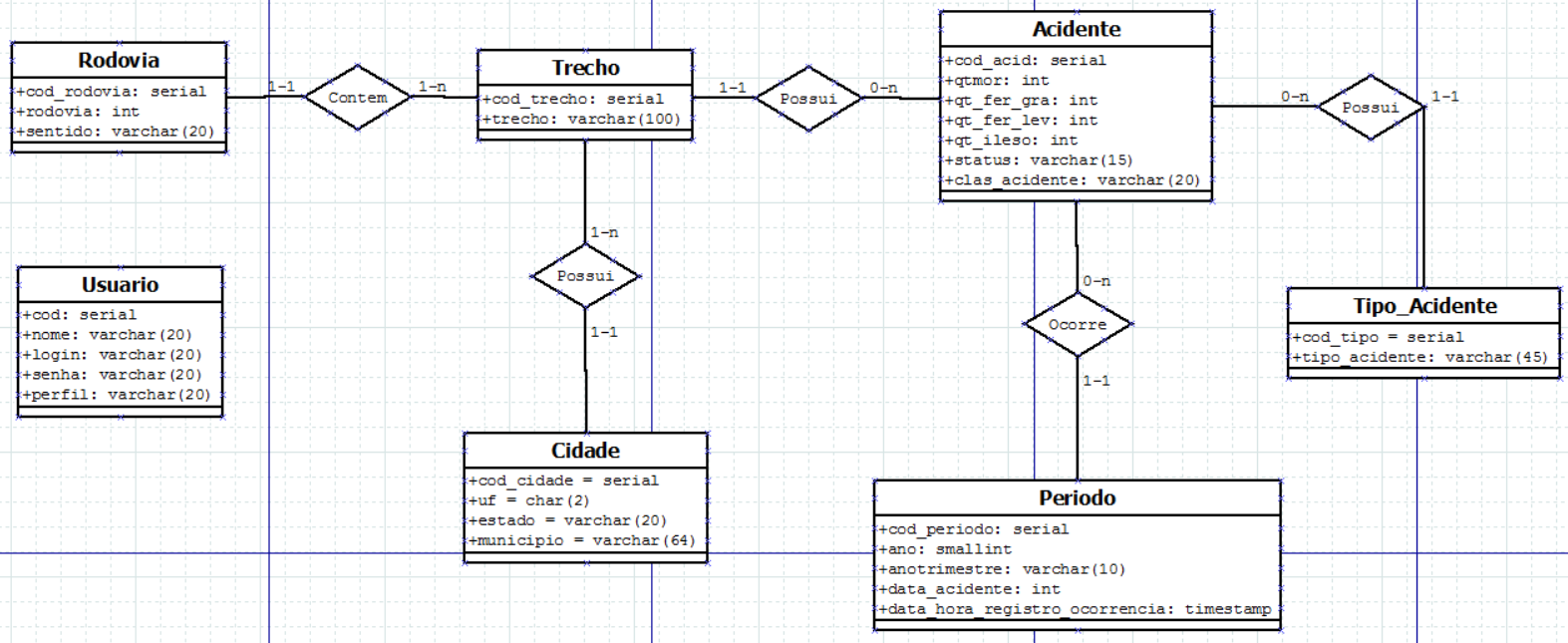
**Natan Barbosa de Morais - 30517**

**Vinicius Eiji Horiguchi - 28836**

1. **Relatório do Banco de Dados**
   1. **Diagrama de Entidade e Relacionamento**



**(Sem atributos)**



**(Com atributos)**

* 1. **Definição de índices e suas justificativas.**

Para fins de teste, foram implementados index para todas as tabelas do banco, com o intuito de quantificar e mensurar a otimização, a partir destes testes, foram escolhidos alguns index que demonstraram diferença satisfatória quando comparados com o SQL sem index. Segue listado abaixo todos os index criados para teste, para resultado de suas performances consultar capítulo 2 deste documento, onde cada teste será justificado, bem como serão apresentados os escolhidos para otimizar a aplicação.

* + 1. **Estado**

Create index cidade\_uf\_municipio on cidade using btree(uf, municipio);

* + 1. **Período**

create index periodo\_data\_hora\_registro\_ocorrencia on periodo(data\_hora\_registro\_ocorrencia);

create index periodo\_ano on periodo(ano);

* + 1. **Rodovia**

create index rodovia\_br on rodovia(br);

* + 1. **Tipo acidente**

create index tipoacidente\_tipo\_acidente on tipoacidente(tipo\_acidente);

* + 1. **Trecho**

create index trecho\_trecho on trecho(trecho);

create index trecho\_id\_cidade\_fk on trecho(id\_cidade\_fk);

* + 1. **Acidente**

create index acidente\_id\_periodo\_fk on acidente(id\_periodo\_fk);

create index acidente\_qtd\_mortos on acidente(qtd\_mortos);

create index acidente\_qtd\_feridos\_graves on acidente(qtd\_feridos\_graves);

create index acidente\_qtd\_feridos\_leves on acidente(qtd\_feridos\_leves);

create index acidente\_qtd\_ilesos on acidente(qtd\_ilesos);

create index acidente\_qtd\_mortos on acidente(qtd\_mortos);

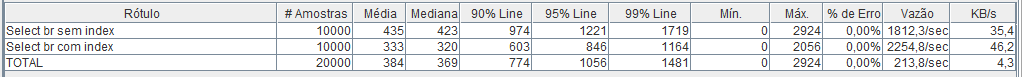
create index acidente\_id\_trecho\_fk on acidente(id\_trecho\_fk);

* 1. **Definição de views, triggers, procedures, funções e suas justificativas.**

Para o desenvolvimento desta aplicação não foram utilizados views, triggers, procedures ou funções, todas as permissões de consultas foram tratadas a nível de aplicação.

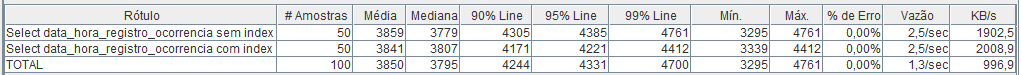
1. **Resultado do teste de performance**
   1. **Rodovia**

Apesar de apresentar melhoria significativa, este índice não foi implementado por não haver nenhum relatório que utiliza tal consulta, desta forma, não se espera que seja uma consulta crítica para o banco, assim foi admitido que manter este índice apenas iria sobrecarregar o banco sem apresentar melhora que justificasse tal sobrecarga, index descartado.



* 1. **Data hora registro ocorrência**

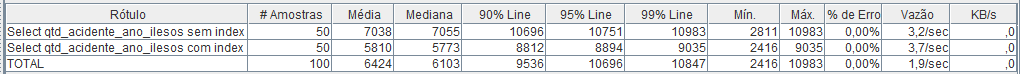
Não apresentou melhora significativa com a criação do índice, index descartado.

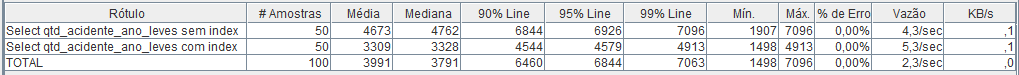


* 1. **Quantidade acidente ano**

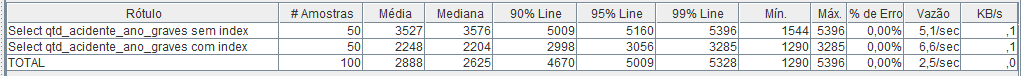
Apresentaram melhora significativa, e como as consultas são utilizadas no [RF12] subentendeu-se que se fazia necessário, para melhor performance da aplicação manter tais índices, index implementado.

* + 1. **Ilesos**

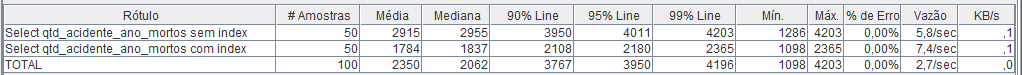
**Feridos Leves**



* + 1. **Feridos Graves**

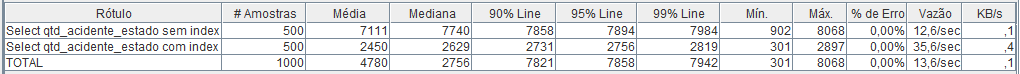
****

* + 1. **Mortos**



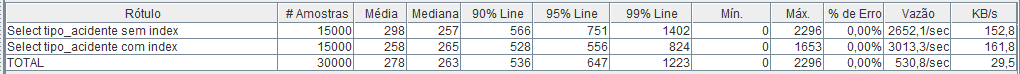
* 1. **Quantidade acidentes por estado**

Apresentou melhora significativa, e como a consulta é utilizada no [RF13] subentendeu-se que se fazia necessário, para melhor performance da aplicação manter tal índice, index implementado.



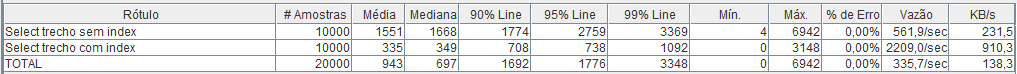
* 1. **Tipo de acidente**

Não apresentou melhora significativa com a criação do índice, index descartado.



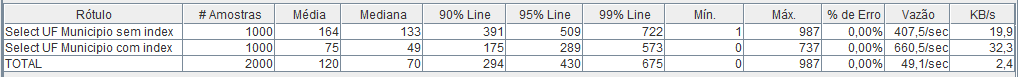
* 1. **Trecho**

Apresentou melhora significativa, apesar de não ser implementada por nenhum relatório, subentende-se que consultas por trecho, bem como por município serão recorrentes no sistema, assim melhorando a performance da aplicação, índex implementado.



* 1. **Município**

Apresentou melhora significativa, apesar de não ser implementada por nenhum relatório, subentende-se que consultas por município, bem como por trecho serão recorrentes no sistema, assim melhorando a performance da aplicação, índex implementado.



1. **Documento de Requisitos**
2. **Backup do Schema.**
   1. **Usuário**

CREATE TABLE usuario (

cod serial NOT NULL, nome character varying(20) NOT NULL,

login character varying(20) NOT NULL,

senha character varying(20) NOT NULL,

perfil character varying(20) NOT NULL,

CONSTRAINT usuario\_pkey PRIMARY KEY (cod) ) WITH ( OIDS=FALSE );

ALTER TABLE usuario OWNER TO postgres;

GRANT ALL ON TABLE usuario TO postgres;

GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE usuario TO aplicacao;

* 1. **Trecho**

CREATE TABLE trecho (

id\_trecho serial NOT NULL,

id\_cidade\_fk integer,

id\_rodovia\_fk integer,

trecho character varying(128),

CONSTRAINT id\_trecho\_pk PRIMARY KEY (id\_trecho),

CONSTRAINT trecho\_id\_cidade\_fk\_fkey FOREIGN KEY (id\_cidade\_fk) REFERENCES cidade (id\_cidade) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION,

CONSTRAINT trecho\_id\_rodovia\_fk\_fkey FOREIGN KEY (id\_rodovia\_fk) REFERENCES rodovia (id\_rodovia) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION ) WITH ( OIDS=FALSE );

ALT,;ER TABLE trecho OWNER TO postgres;

GRANT ALL ON TABLE trecho TO postgres;

GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE trecho TO aplicacao;

-- Index: log\_trecho

-- DROP INDEX log\_trecho;

CREATE INDEX log\_trecho ON trecho USING btree (trecho COLLATE pg\_catalog."default");

* 1. **Tipo Acidente**

CREATE TABLE tipoacidente (

id\_tipoacidente serial NOT NULL,

tipo\_acidente character varying(45),

CONSTRAINT id\_tipoacidente\_pk PRIMARY KEY (id\_tipoacidente) ) WITH ( OIDS=FALSE );

ALTER TABLE tipoacidente OWNER TO postgres;

GRANT ALL ON TABLE tipoacidente TO postgres;

GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE tipoacidente TO aplicacao;

-- Index: log\_tipo\_acidente

-- DROP INDEX log\_tipo\_acidente;

CREATE INDEX log\_tipo\_acidente ON tipoacidente USING btree ( tipo\_acidente COLLATE pg\_catalog."default");

* 1. **Rodovia**

CREATE TABLE rodovia (

id\_rodovia serial NOT NULL,

br smallint,

CONSTRAINT id\_rodovia\_pk PRIMARY KEY (id\_rodovia) ) WITH ( OIDS=FALSE );

ALTER TABLE rodovia OWNER TO postgres;

GRANT ALL ON TABLE rodovia TO postgres;

GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE rodovia TO aplicacao;

-- Index: log\_br

-- DROP INDEX log\_br;

CREATE INDEX log\_br ON rodovia USING btree (br);

* 1. **Período**

CREATE TABLE periodo (

id\_periodo serial NOT NULL,

ano smallint,

ano\_trimestre character varying(10),

data\_acidente integer,

data\_hora\_registro\_ocorrencia timestamp without time zone,

CONSTRAINT id\_periodo\_pk PRIMARY KEY (id\_periodo) ) WITH ( OIDS=FALSE );

ALTER TABLE periodo OWNER TO postgres;

GRANT ALL ON TABLE periodo TO postgres;

GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE periodo TO aplicacao;

-- Index: log\_data\_acidente\_data\_hora\_registro\_ocorrencia

-- DROP INDEX log\_data\_acidente\_data\_hora\_registro\_ocorrencia;

CREATE INDEX log\_data\_acidente\_data\_hora\_registro\_ocorrencia ON periodo USING btree (data\_acidente, data\_hora\_registro\_ocorrencia);

* 1. **Cidade**

CREATE TABLE cidade (

id\_cidade serial NOT NULL,

uf character varying(2),

estado character varying(20),

municipio character varying(64),

CONSTRAINT id\_cidade\_pk PRIMARY KEY (id\_cidade) ) WITH ( OIDS=FALSE );

ALTER TABLE cidade OWNER TO postgres;

GRANT ALL ON TABLE cidade TO postgres;

GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE cidade TO aplicacao;

-- Index: log\_uf\_municipio

-- DROP INDEX log\_uf\_municipio;

CREATE INDEX log\_uf\_municipio ON cidade USING btree (uf COLLATE pg\_catalog."default", municipio COLLATE pg\_catalog."default");

* 1. **Acidente**

CREATE TABLE acidente (

id\_acidente serial NOT NULL,

id\_tipoacidente\_fk integer,

id\_trecho\_fk integer,

id\_periodo\_fk integer,

status character varying(15),

classificacao\_acidente character varying(20),

sentido character varying(20),

qtd\_mortos smallint,

qtd\_feridos\_graves smallint,

qtd\_feridos\_leves smallint,

qtd\_ilesos smallint,

CONSTRAINT id\_acidente\_pk PRIMARY KEY (id\_acidente), CONSTRAINT acidente\_id\_periodo\_fk\_fkey FOREIGN KEY (id\_periodo\_fk) REFERENCES periodo (id\_periodo) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION,

CONSTRAINT acidente\_id\_tipoacidente\_fk\_fkey FOREIGN KEY (id\_tipoacidente\_fk) REFERENCES tipoacidente (id\_tipoacidente) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION,

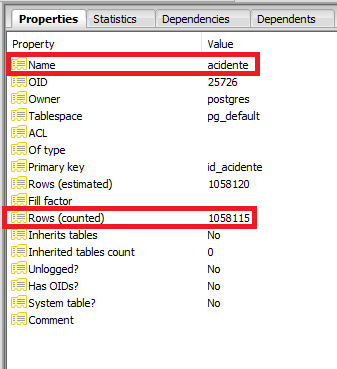
CONSTRAINT acidente\_id\_trecho\_fk\_fkey FOREIGN KEY (id\_trecho\_fk) REFERENCES trecho (id\_trecho) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION ) WITH ( OIDS=FALSE );

ALTER TABLE acidente OWNER TO postgres;

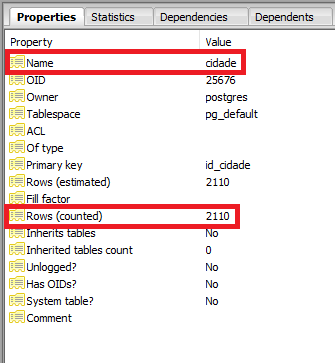
GRANT ALL ON TABLE acidente TO postgres;

GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE acidente TO aplicacao;

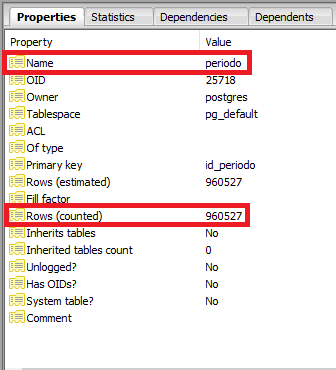
1. **Count das tabelas**
   1. **Tabela Acidente – 1058115**

****

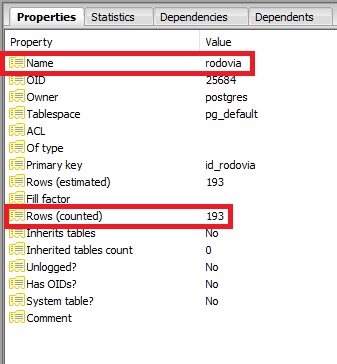
* 1. **Tabela Cidade – 2110**

****

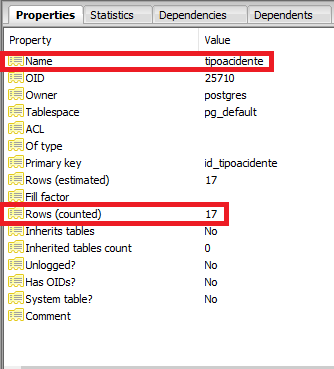
* 1. **Tabela Periodo – 960527**

****

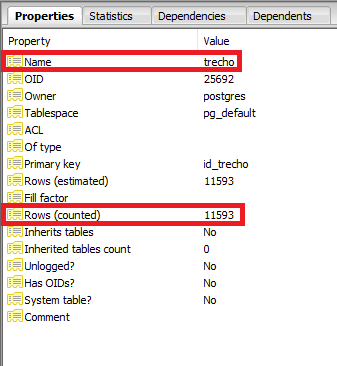
* 1. **Tabela Rodovia - 193**

****

* 1. **Tabela Tipoacidente – 17**

****

* 1. **Tabela Trecho – 11593**

****