**Projeto de Bases de Dados**, Parte 4



**Grupo nº 38**

**Turno de Segunda feira 12:30 - Lab14**

**Docente: Taras Lykhenko**

|  |  |  |
| --- | --- | --- |
| **Aluno** | **Esforço (em horas)** | **Percentagem relativa de contribuição** |
| Francisco Nicolau - 86419 | 15 | 33.(3)% |
| Manuel Correia     - 86470 | 15 | 33.(3)% |
| Miguel Valério       - 86483 | 15 | 33.(3)% |

IST, LEIC-A, 2017/2018   Base de Dados

**Restrições de Integridade**

As condições foram satisfeitas recorrendo ao uso de *triggers*.

* Um Coordenador só pode solicitar vídeos de câmaras colocadas num local cujo acionamento de meios esteja a ser (ou tenha sido) auditado por ele próprio.

create or replace function **check\_Solicita**() returns trigger as $body$

begin

if not exists (

select \* from (Vigia natural join EventoEmergencia natural join Audita) T where T.idCoordenador = new.idCoordenador and T.numCamara = new.numCamara )

then

raise exception 'O Coordenador % nao pode solicitar videos da camara %.', new.idCoordenador, new.numCamara

using hint = 'Um Coordenador so pode solicitar vídeos de camaras colocadas num local cujo acionamento de meios esteja a ser (ou tenha sido) auditado por ele proprio.';

end if;

return new;

end;

$body$ language plpgsql;

drop trigger if exists **check\_Solicita\_trigger** on Solicita cascade;

create trigger **check\_Solicita\_trigger** before insert or update on Solicita for each row execute procedure check\_Solicita();

* Um Meio de Apoio só pode ser alocado a Processos de Socorro para os quais tenha sido accionado.

create or replace function **check\_Alocado**() returns trigger as $body$

begin

if not exists (

select \* from Acciona A where A.numMeio = new.numMeio and A.nomeEntidade = new.nomeEntidade and A.numProcessoSocorro = new.numProcessoSocorro )

then

raise exception 'O Meio de Apoio (%, %) nao pode ser alocado pelo Processo de Socorro %.', new.numMeio, new.nomeEntidade, new.numProcessoSocorro

using hint = 'Um Meio de Apoio so pode ser alocado a Processos de Socorro para os quais tenha sido accionado.';

end if;

return new;

end;

$body$ language plpgsql;

drop trigger if exists **check\_Alocado\_trigger** on Alocado cascade;

create trigger **check\_Alocado\_trigger** before insert or update on Alocado for each row execute procedure check\_Alocado();

**Índices**

Em relação à primeira consulta deveriam ser criados dois índices para melhorar a performance. Em primeiro lugar, um índice na tabela Video relativo à coluna numCamara que seja UNIQUE, pois esta coluna contem o número de cada camara e estes são únicos. Em segundo lugar, na tabela Vigia relativo à coluna moradaLocal, de modo a facilmente identificar as linhas de interesse sem ser necessário percorrer toda a tabela, este índice não seria único dado que há várias camaras por morada local.

No que toca à segunda consulta, para ambas as tabelas (Transporta e EventoEmergencia) deveria ser criado um indice relativo à coluna numProcessoSocorro. Para nenhum dos indices usamso UNIQUE dado que em ambas poderá haver multiplas entradas relativas a um dado Processo de Socorro.

**Modelo Multidimensional**

**Data Analytics**