
Projeto de Bases de Dados, Parte 4

Nome	Número de Aluno	Percentagem Relativa de Contribuição	Esforço
Pedro Alegre Caldeira	83539	33,33%	12 horas
Francisco Lopes Pereira de Carvalho	84050	33,33%	12 horas
João Miguel Fernandes Pina	85080	33,33%	12 horas

Grupo 10 - turno de quarta-feira às 8:00

Docente do laboratório: André Vasconcelos

Restrições de Integridade

- a) O fornecedor (primario) de um produto não pode existir na relação fornece_sec para o mesmo produto.

```
--TRIGGER

create or replace function checkFornPrimario() returns trigger as $$
begin
    IF (new.nif,new.ean) in (SELECT forn_primario as p_nif, ean as p_ean FROM produto) THEN
        RAISE EXCEPTION 'O fornecedor primario de um produto nao pode existir na relacao fornece_sec para o mesmo produto';
    END IF;
    RETURN NEW;
end;
$$ language plpgsql;

create trigger update_produtoSecIn before insert on fornece_sec for each row execute procedure checkFornPrimario();
```

- b) O instante mais recente de reposição tem de ser sempre anterior ou igual à data atual.

```
create table reposicao(
    ean numeric(13,0) not null,
    nro integer not null,
    lado char(1) not null,
    altura char(1) not null,
    operador varchar(64) not null,
    instante timestamp not null,
    unidades integer not null,

    primary key(ean,nro,lado,altura,operador,instante),
    foreign key(ean,nro,lado,altura)
        references planograma(ean,nro,lado,altura)
        ON DELETE CASCADE,
    foreign key(operador,instante)
        references evento_reposicao(operador,instante),
    check(instante <= CURRENT_TIMESTAMP)
);
```

Índices

- a) Indique, justificando, que tipo de índice(s), sobre que atributo(s) e sobre que tabela(s) faria sentido criar de modo a acelerar a execução destas interrogações. Analise cada uma das consultas separadamente.
- Query 1: Necessitamos de criar dois índices BTREE. O primeiro é um índice primário criado na tabela **fornecedor** com chave de procura **nif**. Este índice terá que ser agrupado e esparso. O facto de estar agrupado ajuda na procura do nif. O segundo índice é um índice secundário composto e é criado na tabela produto com chave de procura (**forn_primario, categoria**), de forma a obrigar a procurar primeiro o nif do fornecedor e só depois a categoria. O índice terá que ser denso e desagrupado. Com estes dois índices conseguimos acelerar bastante a procura já que ocorre um *Merge Join*.
 - Query 2: Necessitamos de criar um índice BTREE primário na tabela **produto** com chave em **ean**. O índice terá que ser agrupado e esparso, de forma a tornar mais eficiente a operação de *group by*.
- b) Cria o(s) índice(s) em SQL, se necessário.

- Query 1:

```
CREATE INDEX forn_nif ON fornecedor(nif);  
CREATE INDEX prod_prim ON produto(forn_primario, categoria);
```

- Query 2:

```
CREATE INDEX prod_ean ON produto(ean);
```

Modelo Multidimensional

```
create table d_produto(
    cean numeric(13,0) not null,
    categoria varchar(64) not null,
    nif_fornecedor_principal numeric(9,0) not null,

    primary key(cean),
    foreign key(cean)
        references produto(ean),
    foreign key(categoria)
        references categoria(nome),
    foreign key(nif_fornecedor_principal)
        references fornecedor(nif)
);

create table d_tempo(
    tempo_id date not null,
    dia numeric(2,0) not null,
    mes numeric(2,0) not null,
    ano numeric(4,0) not null,
    primary key(tempo_id)
);

create table factos(
    prod_id numeric(13,0) not null,
    tempo_id date not null,

    primary key(prod_id, tempo_id),

    foreign key(prod_id)
        references d_produto(cean),
    foreign key(tempo_id)
        references d_tempo(tempo_id)
);

CREATE TABLE digits (
    number INT);
INSERT INTO digits VALUES (0),(1),(2),(3),(4),(5),(6),(7),(8),(9);

CREATE TABLE numbers (
    number INT);
INSERT INTO numbers
    SELECT thousands.number * 1000 + hundreds.number * 100 + tens.number * 10 + ones.number
    FROM digits thousands, digits hundreds, digits tens, digits ones;

insert into d_produto (cean, categoria, nif_fornecedor_principal)
    SELECT ean, categoria, forn_primario
    FROM produto;

insert into d_tempo (tempo_id, dia, mes, ano)
SELECT
    date as tempo_id,
    date_part('day', date) as dia,
    date_part('month', date) as mes,
    date_part('year', date) as ano
FROM (
    SELECT number, date '2017-01-01' + (number * interval '1 day') as date
    FROM digits
    WHERE date '2017-01-01' + (number * interval '1 day') BETWEEN '2017-01-01' AND '2017-12-31'
) AS result
ORDER BY number;

insert into factos(prod_id, tempo_id)
select distinct cean as prod_id, tempo_id
from d_produto natural join d_tempo
where tempo_id in (select distinct instante::date from reposicao where cean = ean);
```

Data Analytics

```
SELECT distinct count(*)
FROM factos NATURAL JOIN d_produto NATURAL JOIN d_tempo
WHERE nif_fornecedor_principal = 123455678
GROUP BY ROLLUP (categoria, ano, mes);
```