



Faculdade de Informática e Administração Paulista

Programming
and
Database Management

Contextualização

A hapvida manipula muitos dados sobre o **ODS-3. ODS3 – Saúde e bem-estar**. Para isso foram disponibilizadas as seguintes tabelas que apresentam dados sobre a diabetes, de acordo com a meta 3.4 e indicador 3.4.1:

- diabetes (*)
- discharge_disposition (*)
- admission_type (*)
- admission_source (*)

Essas tabelas são atualizadas constantemente e estão disponíveis em um banco de dados Oracle.

(*) As tabelas pertencem ao meu esquema de banco de dados, portanto para acessá-las vocês devem informar o nome do meu esquema (PF0645)

Exemplos:

```
desc PF0645.diabetes;
select * from PF0645.discharge_disposition;
select * from PF0645.admission_type;
select * from PF0645.admission_source;
```

Portanto vocês devem executar os seguintes comandos a seguir para criar as tabelas no esquema de cada um.

```
create table diabetes
as select * from PF0645.diabetes;

create table discharge_disposition
as select * from PF0645.discharge_disposition;

create table admission_type
as select * from PF0645.admission_type;

create table admission_source
as select * from PF0645.admission_source;

ALTER TABLE diabetes
ADD CONSTRAINT diabetes_pk
PRIMARY KEY (encounter_id);

ALTER TABLE discharge_disposition
ADD CONSTRAINT DISCHARGE_DISPOSITION_PK
PRIMARY KEY (discharge_disposition_id);

ALTER TABLE admission_type
ADD CONSTRAINT ADMISSION_TYPE_PK
PRIMARY KEY (admission_type_id);

ALTER TABLE admission_source
```

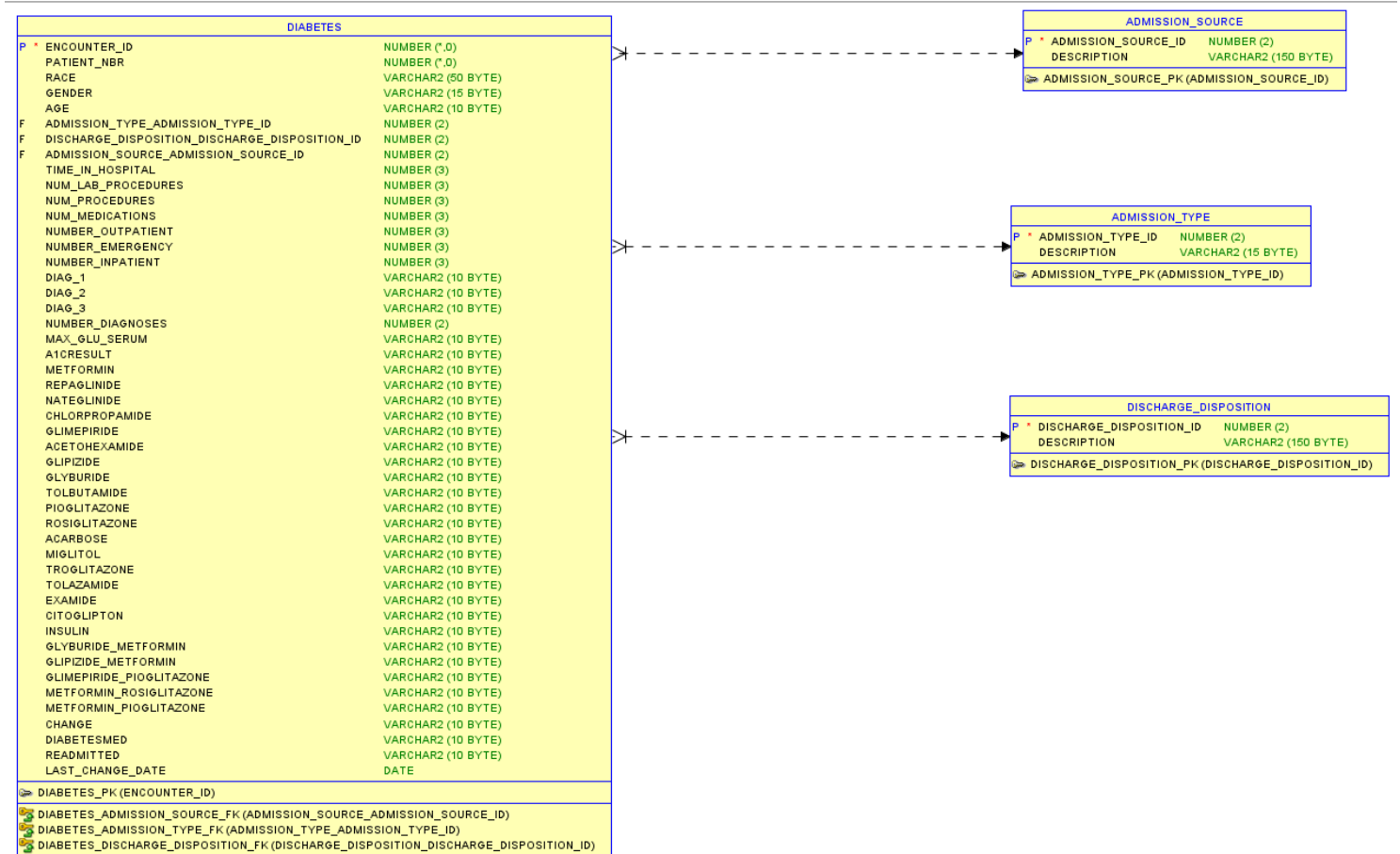
```

ADD CONSTRAINT ADMISSION_SOURCE_PK
PRIMARY KEY (admission_source_id);

ALTER TABLE diabetes
ADD CONSTRAINT diabetes_admission_source_fk
FOREIGN KEY ( admission_source_id )
REFERENCES admission_source ( admission_source_id )
ADD CONSTRAINT diabetes_admission_type_fk
FOREIGN KEY ( admission_type_id )
REFERENCES admission_type ( admission_type_id )
ADD CONSTRAINT diabetes_discharge_disposition_fk
FOREIGN KEY ( discharge_disposition_id )
REFERENCES discharge_disposition (discharge_disposition_id);

```

A seguir é apresentado o modelo relacional referente as tabelas: diabetes, discharge_disposition, admission_type e admission_source



Instruções

A hapvida está enfrentando problemas de lentidão quando executa o procedimento chamado `update_time_in_hospital`. Esse procedimento realiza a adição de um dia na coluna `time_in_hospital` (essa coluna armazena quantos dias um paciente ficou internado).

Cada paciente pode ficar internado de 1 até 14 dias. Quando o procedimento `update_time_in_hospital` é executado ele chama o procedimento `check_eligibility` que irá verificar se poderá ser adicionado 1 dia para todos os pacientes que tem a data de alteração (`last_change_date`) igual ao que foi informado como parâmetro de entrada do procedimento `update_time_in_hospital`.

```
CREATE OR REPLACE PROCEDURE update_time_in_hospital (
    p_last_change_date    IN DATE,
    p_time_in_hospital    IN NUMBER)
IS
    l_eligible            BOOLEAN;
    l_sql_count            INTEGER :=0 ;
BEGIN
    FOR patients_rec IN (SELECT encounter_id
                        FROM    diabetes
                        WHERE    TRUNC(last_change_date) = p_last_change_date)
    LOOP
        DBMS_OUTPUT.PUT_LINE('PACIENTE: '|| patients_rec.encounter_id);
        check_eligibility (patients_rec.encounter_id,
                           p_time_in_hospital,
                           l_eligible);

        IF l_eligible
        THEN
            UPDATE diabetes
            SET    time_in_hospital = time_in_hospital + p_time_in_hospital
            WHERE  encounter_id = patients_rec.encounter_id;
            l_sql_count := l_sql_count+1;
        END IF;
    END LOOP;
    DBMS_OUTPUT.PUT_LINE(l_sql_count || ' rows updated');
    commit;
END update_time_in_hospital;
/
```

```
CREATE OR REPLACE PROCEDURE check_eligibility
    (p_encounter_id      IN  NUMBER
    ,p_time_in_hospital  IN  NUMBER
    ,p_is_eligible       OUT BOOLEAN)
IS
    l_encounter_id NUMBER;
BEGIN
    SELECT encounter_id
    INTO    l_encounter_id
    FROM    diabetes
    WHERE   encounter_id = p_encounter_id
    AND     time_in_hospital + p_time_in_hospital < 14;
    p_is_eligible := TRUE;
EXCEPTION
    WHEN NO_DATA_FOUND then
        p_is_eligible := FALSE;
END;
/
```

O procedimento é executado da seguinte forma:

```
set serverout on  
cl scr  
exec update_time_in_hospital (p_last_change_date => '10-NOV-2023',p_time_in_hospital =>  
08);
```

Nesse exemplo, serão adicionados 8 dias para os pacientes que tem a `last_change_date` igual a 10-NOV-2023 e que estão internados a menos de 14 dias.

A sua tarefa é realizar a otimização dos códigos dos procedimentos `check_eligibility` e `update_time_in_hospital`. Devem ser considerados tanto ajustes nos códigos para evitar a troca de contexto quanto a criação de índices que auxiliam na performance.

Para medir o tempo de execução, utilize o seguinte código:

```
SET SERVEROUT ON  
DECLARE  
    t0 number := dbms_utility.get_time;  
    v_idx number := 1;  
BEGIN  
    update_time_in_hospital (p_last_change_date => '10-NOV-2023',p_time_in_hospital => 01)  
    dbms_output.put_line('Tempo: ' ||  
        ((dbms_utility.get_time - t0) / 100) || ' segundos');  
END;  
/
```

Lembre-se de verificar o plano de execução (Tecla F10 no SQL Developer) após a criação dos índices

Sobre a entrega

Deve ser entregue o arquivo com todos o que foi feito para otimizar a execução dos procedimentos. Coloque os novos procedimentos utilizados e os comandos que criaram os índices, caso tenha sido necessário criar algum índice.

3SI<turma>_GS.SQL

Coloque nas primeiras linhas do arquivo o RM e nome completo dos componentes do grupo

```
/*  
12345 - Xiuderico do Exemplo  
54321 - Clarisvertina de tal  
12121 - Maria das Belranas  
*/  
CREATE OR REPLACE PROCEDURE update_time_in_hospital (  
    p_last_change_date IN DATE,  
    p_time_in_hospital IN NUMBER)  
IS
```

```
...  
  
CREATE OR REPLACE PROCEDURE check_eligibility  
  (p_encounter_id      IN  NUMBER  
  ,p_time_in_hospital IN  NUMBER  
  ,p_is_eligible        OUT BOOLEAN  
  )  
IS  
  l_encounter_id NUMBER;  
...  

```

Postagem

Um e somente um integrante do grupo deverá postar o arquivo (3SI<turma>_GS.SQL) no portal do aluno FIAP na área de entrega de trabalhos.

OS GRUPOS PODEM SER COMPOSTOS POR ATÉ 3 (TRÊS) ALUNOS.

Faça, também, o preenchimento do formulário disponível em:

<https://pt.surveymonkey.com/r/WYWFH5B>