Banco de Dados Hospitalar

SQL e PL/SQL

SELECT M1.NOME, COUNT (M1.NOME) AS QTD_SUPERVISIONADO

FROM MEDICO M1 INNER JOIN

MEDICO M2 ON M2.SUPERVISOR = M1.CPF

GROUP BY M1.NOME

SELECT med.nome, COUNT(tr.cpf_paciente) FROM medico med

LEFT OUTER JOIN trata tr ON tr.cpf medico = med.cpf

GROUP BY (med.nome);

SELECT NOME

FROM PACIENTE

WHERE CPF IN (SELECT CPF

FROM ARTE);

SELECT NOME FROM PACIENTE WHERE CPF NOT IN (SELECT CPF_PACIENTE FROM UTILIZA));

```
SELECT med.nome FROM medico med WHERE

(SELECT COUNT(tr.cpf_paciente) FROM trata

tr WHERE med.cpf = tr.cpf medico ) > 1;
```

SELECT pac.nome, med.nome FROM trata tr

WHERE

INNER JOIN paciente pac ON pac.cpf = tr.cpf paciente

INNER JOIN medico med ON med.cpf = tr.cpf medico (pac.cidade, pac.rua) IN

(SELECT cidade, rua FROM medico WHERE cpf = med.cpf);

SELECT NOME FROM PACIENTE

UNION

SELECT NOME FROM MEDICO;

```
CREATE OR REPLACE PROCEDURE Residentes_do_Quarto(quartoNumero INTEGER ) IS
    CURSOR c1 IS
    SELECT nome from paciente where n_quarto=quartoNumero;
    rec1 c1%rowtype;
    begin
        for rec1 in c1 loop
        dbms_output.put_line(rec1.nome);
```

```
create or replace procedure pacientesComCertaQuantidadeDeArtesOuMais(quantidade number)
cursor c1 is
      SELECT p.nome, count(*) as numero de artes
      FROM paciente p
      inner join arte a
      on p.cpf = a.cpf
      group by (p.nome)
  where numero de artes >= quantidade;
rec1 c1%rowtype;
dbms output.put line(rec1.nome||' '||rec1.numero de artes);
```

```
CREATE OR REPLACE TRIGGER QUARTO LOTADO BEFORE INSERT OR UPDATE OF N QUARTO ON
PACIENTE
FOR EACH ROW
DECLARE
   CONTADOR NUMBER;
BEGIN
    SELECT COUNT(*) INTO CONTADOR FROM PACIENTE WHERE N QUARTO = :NEW.N QUARTO;
    IF CONTADOR > 3 THEN
        RAISE APPLICATION ERROR (-20205, 'O quarto está cheio');
END;
```

```
CREATE OR REPLACE TRIGGER novo supervisor BEFORE
INSERT OR UPDATE OF SUPERVISOR ON MEDICO
FOR EACH ROW
BEGIN
     (:NEW.SUPERVISOR<>:OLD.SUPERVISOR) AND (:OLD.SUPERVISOR IS NOT NULL) THEN
      dbms output.put line('Atribuido novo supervisor: ':NEW.SUPERVISOR);
      ELSIF (:NEW.SUPERVISOR IS NULL) AND (:OLD.SUPERVISOR IS NOT NULL)
           dbms output.put line('O médico de CRM' :NEW.CRM 'Não possui mais
supervisor');
          ELSE
               dbms output.put line('Novo supervisor':NEW.CRM);
```

END IF;

END;

```
CREATE OR REPLACE FUNCTION is supervisor(cpf_medi VARCHAR2) RETURN VARCHAR IS
       SELECT *
       FROM MEDICO;
  FOR reg medico in cursor medicos loop
      if( reg_medico.SUPERVISOR = cpf_medi) then return 'TRUE';
      end if;
  when NO DATA FOUND then
      dbms_output.put_line('Dados indisponiveis da tabela);
```

END,

return null;