

# Banco de Dados Hospitalar

SQL e PL/SQL

# Consulta com Group by/Having + Junção interna

Mostra o nome do supervisor e a quantidade de supervisionados

```
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
```

## Consultas com junção externa

Mostra a quantidade de pacientes de **todos os** médicos do hospital

```
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);
```

## Semi-Junção

Nome dos pacientes que fizeram alguma arte

```
SELECT NOME  
FROM PACIENTE  
WHERE CPF IN (SELECT CPF  
FROM ARTE) ;
```

## Anti-Junção

Exibe os pacientes que não usam nenhum medicamento

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

# Subconsulta escalar

Vai exibir os medicos que tem mais do que 1 paciente

```
SELECT med.nome FROM medico med WHERE  
  (SELECT COUNT(tr.cpf_paciente) FROM trata  
tr WHERE med.cpf = tr.cpf_medico ) > 1;
```

# Subconsulta do tipo linha

Retorna os médicos que moram na mesma cidade e rua que seu supervisor

```
SELECT med.nome FROM medico med WHERE  
      (med.cidade , med.rua) IN  
      ( SELECT sup.cidade , sup.rua FROM medico sup  
WHERE  sup.cpf = med.supervisor ) ;
```

# Subconsulta do tipo tabela

```
SELECT pac.nome, med.nome FROM trata tr
  INNER JOIN paciente pac ON pac.cpf = tr.cpf_paciente
  INNER JOIN medico med ON med.cpf = tr.cpf_medico
WHERE (pac.cidade, pac.rua) IN
      ( SELECT cidade, rua FROM medico WHERE cpf = med.cpf);
```





# Subconsulta com operação de conjuntos

## Mostra todos os médicos e pacientes do hospital

```
SELECT NOME FROM PACIENTE
```

```
UNION
```

```
SELECT NOME FROM MEDICO;
```

# Procedure com SQL embutido e parâmetro

## Lista todos os residentes do quarto

```
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);
    end loop;
end;
```

# Procedure com SQL embutido e parâmetro

```
create or replace procedure pacientesComCertaQuantidadeDeArtesOuMais(quantidade number)
is
cursor c1 is

    select * from
    (
        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;
begin
for rec1 in c1 loop
dbms_output.put_line(rec1.nome|| ' ' ||rec1.numero_de_artes);
end loop;
end;
```

# Triggers. O gatilho é ativado quando o número de ocupantes do quarto ultrapassa 3

```
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 IF;
END;
```

# Trigger para avisar que o supervisor mudou

```
CREATE OR REPLACE TRIGGER novo_supervisor BEFORE  
INSERT OR UPDATE OF SUPERVISOR ON MEDICO
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    IF (: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) THEN
```

```
        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;
```

# Função para checar se médico é supervisor

```
CREATE OR REPLACE FUNCTION is_supervisor(cpf_medi VARCHAR2) RETURN VARCHAR IS
```

```
    CURSOR cursor_medicos IS
```

```
        SELECT *
```

```
        FROM MEDICO;
```

```
BEGIN
```

```
    FOR reg_medico in cursor_medicos loop
```

```
        if( reg_medico.SUPERVISOR = cpf_medi) then return 'TRUE';
```

```
        end if;
```

```
    end loop;
```

```
    return 'FALSE';
```

```
exception
```

```
    when NO_DATA_FOUND then
```

```
        dbms_output.put_line('Dados indisponiveis da tabela');
```

```
        return null;
```

```
END;
```