

CONSULTAS COM VÁRIAS TABELAS NO POSTGRESQL

No final do **módulo 1**, o especialista realizará as consultas sobre as tabelas NIVEL e CURSO, usando a Query Tool da ferramenta PG Admin 4, demonstrando as operações de CROSS JOIN, INNER JOIN e OUTER JOIN, com capturas das telas.

```
CREATE TABLE NIVEL (  
    CODIGONIVEL int NOT NULL,  
    DESCRICAO varchar(90) NOT NULL,  
    CONSTRAINT CHAVEPNIVEL PRIMARY KEY (CODIGONIVEL));
```

```
CREATE TABLE CURSO (  
    CODIGOCURSO int NOT NULL,  
    NOME varchar(90) NOT NULL UNIQUE,  
    DATACRIACAO date NULL,  
    CODIGONIVEL int NULL,  
    CONSTRAINT CHAVEPCURSO PRIMARY KEY (CODIGOCURSO));
```

```
ALTER TABLE CURSO ADD FOREIGN KEY (CODIGONIVEL) REFERENCES  
NIVEL;
```

```
INSERT INTO NIVEL (CODIGONIVEL, DESCRICAO) VALUES  
(1, 'Graduação');  
INSERT INTO NIVEL (CODIGONIVEL, DESCRICAO) VALUES  
(2, 'Especialização');  
INSERT INTO NIVEL (CODIGONIVEL, DESCRICAO) VALUES  
(3, 'Mestrado');  
INSERT INTO NIVEL (CODIGONIVEL, DESCRICAO) VALUES  
(4, 'Doutorado');  
INSERT INTO CURSO (CODIGOCURSO, NOME, DATACRIACAO, CODIGONIVEL)  
VALUES (1, 'Sistemas de Informação', '19/06/1999', 1);  
INSERT INTO CURSO (CODIGOCURSO, NOME, DATACRIACAO, CODIGONIVEL)  
VALUES (2, 'Medicina', '10/05/1990', 1);  
INSERT INTO CURSO (CODIGOCURSO, NOME, DATACRIACAO, CODIGONIVEL)  
VALUES (3, 'Nutrição', '19/02/2012', NULL);  
INSERT INTO CURSO (CODIGOCURSO, NOME, DATACRIACAO, CODIGONIVEL)  
VALUES (4, 'Pedagogia', '19/06/1999', 1);  
INSERT INTO CURSO (CODIGOCURSO, NOME, DATACRIACAO, CODIGONIVEL)  
VALUES (5, 'Saúde da Família', '10/09/1999', 3);  
INSERT INTO CURSO (CODIGOCURSO, NOME, DATACRIACAO, CODIGONIVEL)  
VALUES (6, 'Computação Aplicada', '10/09/1999', NULL);
```

```
SELECT * FROM NIVEL;  
TABLE CURSO;  
SELECT * FROM CURSO, NIVEL;  
SELECT * FROM CURSO CROSS JOIN NIVEL;
```

```
SELECT *  
FROM CURSO, NIVEL  
WHERE NIVEL.CODIGONIVEL=CURSO.CODIGONIVEL;
```

```
SELECT *  
FROM CURSO INNER JOIN NIVEL ON  
(NIVEL.CODIGONIVEL=CURSO.CODIGONIVEL);
```

```
SELECT *  
FROM CURSO INNER JOIN NIVEL USING(CODIGONIVEL);
```

```
SELECT CURSO.CODIGOCURSO, CURSO.NOME,  
       NIVEL.CODIGONIVEL, NIVEL.DESCRICAO  
FROM CURSO INNER JOIN NIVEL ON  
(NIVEL.CODIGONIVEL=CURSO.CODIGONIVEL);
```

```
SELECT C.CODIGOCURSO, C.NOME,  
       N.CODIGONIVEL, N.DESCRICAO  
FROM CURSO C INNER JOIN NIVEL N ON  
(N.CODIGONIVEL=C.CODIGONIVEL);
```

```
SELECT C.CODIGOCURSO, C.NOME,  
       N.CODIGONIVEL, N.DESCRICAO  
FROM CURSO C LEFT JOIN NIVEL N ON (N.CODIGONIVEL=C.CODIGONIVEL);
```

```
SELECT C.CODIGOCURSO, C.NOME,  
       N.CODIGONIVEL, N.DESCRICAO  
FROM CURSO C RIGHT JOIN NIVEL N ON  
(N.CODIGONIVEL=C.CODIGONIVEL);
```

```
SELECT C.CODIGOCURSO, C.NOME,  
       N.CODIGONIVEL, N.DESCRICAO  
FROM CURSO C LEFT JOIN NIVEL N ON (N.CODIGONIVEL=C.CODIGOCURSO);
```

```
SELECT C.CODIGOCURSO, C.NOME,  
       N.CODIGONIVEL, N.DESCRICAO  
FROM NIVEL N RIGHT JOIN CURSO C ON  
(N.CODIGONIVEL=C.CODIGOCURSO);
```

```
SELECT C.CODIGOCURSO, C.NOME,  
       N.CODIGONIVEL, N.DESCRICAO  
FROM CURSO C FULL OUTER JOIN NIVEL N ON  
(N.CODIGONIVEL=C.CODIGONIVEL);
```

No final do **módulo 2**, o especialista realizará as consultas sobre as tabelas FUNCIONÁRIO e DEPARTAMENTO, usando a Query Tool da ferramenta PG Admin 4, demonstrando diferentes tipos de consultas aninhadas, com capturas das telas.

```
CREATE TABLE DEPARTAMENTO (  
    CODIGODEPARTAMENTO int NOT NULL,  
    NOME varchar(90) NOT NULL,  
    CONSTRAINT DEPARTAMENTO_pk PRIMARY KEY (CODIGODEPARTAMENTO));  
CREATE TABLE FUNCIONARIO (  
    CODIGOFUNCIONARIO int NOT NULL,  
    NOME varchar(90) NOT NULL,  
    CPF char(15) NULL,  
    SEXO char(1) NOT NULL,  
    DTNASCIMENTO date NOT NULL,  
    SALARIO real NULL,  
    CODIGODEPARTAMENTO int NULL,  
    CONSTRAINT FUNCIONARIO_pk PRIMARY KEY (CODIGOFUNCIONARIO));  
  
ALTER TABLE FUNCIONARIO ADD CONSTRAINT FUNCIONARIO_DEPARTAMENTO  
    FOREIGN KEY (CODIGODEPARTAMENTO)  
    REFERENCES DEPARTAMENTO (CODIGODEPARTAMENTO) ;
```

```
INSERT INTO DEPARTAMENTO(CODIGODEPARTAMENTO,NOME) VALUES  
(1,'Tecnologia da Informação');  
INSERT INTO DEPARTAMENTO(CODIGODEPARTAMENTO,NOME) VALUES  
(2,'Contabilidade');  
INSERT INTO DEPARTAMENTO(CODIGODEPARTAMENTO,NOME) VALUES  
(3,'Marketing');  
INSERT INTO FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CPF, SEXO,  
DTNASCIMENTO, SALARIO,CODIGODEPARTAMENTO)  
VALUES (1,'ROBERTA SILVA BRASIL',NULL,'F','20/02/1980',7000,1);  
INSERT INTO FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CPF, SEXO,  
DTNASCIMENTO, SALARIO,CODIGODEPARTAMENTO)  
VALUES (2,'MARIA SILVA BRASIL',NULL,'F','20/09/1988',9500,2);  
INSERT INTO FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CPF, SEXO,  
DTNASCIMENTO, SALARIO,CODIGODEPARTAMENTO)  
VALUES (3,'GABRIELLA PEREIRA  
LIMA',NULL,'F','20/02/1990',6000,1);  
INSERT INTO FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CPF, SEXO,  
DTNASCIMENTO, SALARIO,CODIGODEPARTAMENTO)  
VALUES (4,'MARCOS PEREIRA BRASIL',NULL,'M','20/02/1999',6000,2);  
INSERT INTO FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CPF, SEXO,  
DTNASCIMENTO, SALARIO,CODIGODEPARTAMENTO)  
VALUES (5,'HEMERSON SILVA  
BRASIL',NULL,'M','20/12/1992',4000,NULL);
```

```
SELECT CODIGOFUNCIONARIO, NOME
FROM FUNCIONARIO
WHERE SALARIO=
            (SELECT MAX(SALARIO)
             FROM FUNCIONARIO);
```

```
SELECT CODIGOFUNCIONARIO, NOME, SALARIO
FROM FUNCIONARIO
WHERE SALARIO>
            (SELECT AVG(SALARIO)
             FROM FUNCIONARIO);
```

```
SELECT CODIGOFUNCIONARIO, NOME, SALARIO
FROM FUNCIONARIO
WHERE SALARIO<
            (SELECT AVG(SALARIO)
             FROM FUNCIONARIO
             WHERE CODIGODEPARTAMENTO IN (SELECT
CODIGODEPARTAMENTO
                                         FROM DEPARTAMENTO
                                         WHERE
NOME='Tecnologia da Informação')));
```

```
SELECT CODIGOFUNCIONARIO, NOME, SALARIO
FROM FUNCIONARIO F
WHERE SALARIO<
            (SELECT AVG(SALARIO)
             FROM FUNCIONARIO F JOIN DEPARTAMENTO D ON
F.CODIGODEPARTAMENTO=D.CODIGODEPARTAMENTO
             WHERE D.NOME='Tecnologia da Informação');
```

```
SELECT COUNT(*) AS QUANTIDADE
FROM FUNCIONARIO
WHERE SALARIO<
            (SELECT MAX(SALARIO)
             FROM FUNCIONARIO
             WHERE SEXO='F');
```

```
SELECT CODIGOFUNCIONARIO, NOME, SALARIO
FROM FUNCIONARIO F
WHERE SALARIO>
```

```
                (SELECT AVG(SALARIO)
                 FROM FUNCIONARIO
                 WHERE
CODIGODEPARTAMENTO=F.CODIGODEPARTAMENTO);
```

```
SELECT CODIGOFUNCIONARIO, NOME, SALARIO
FROM FUNCIONARIO F
    JOIN
                (SELECT CODIGODEPARTAMENTO,AVG(SALARIO) AS
MEDIA
                FROM FUNCIONARIO
                GROUP BY CODIGODEPARTAMENTO) TESTE
    ON F.CODIGODEPARTAMENTO=TESTE.CODIGODEPARTAMENTO
WHERE SALARIO>MEDIA;
```

```
UPDATE FUNCIONARIO F
SET SALARIO=
                (SELECT MAX(SALARIO)
                 FROM FUNCIONARIO
                 WHERE
CODIGODEPARTAMENTO=F.CODIGODEPARTAMENTO)
WHERE F.CODIGODEPARTAMENTO IS NOT NULL;
```

```
SELECT D.CODIGODEPARTAMENTO, D.NOME
FROM DEPARTAMENTO D
WHERE EXISTS
    (SELECT F.CODIGODEPARTAMENTO
     FROM FUNCIONARIO F
     WHERE D.CODIGODEPARTAMENTO=F.CODIGODEPARTAMENTO);
```

```
SELECT D.CODIGODEPARTAMENTO, D.NOME
FROM DEPARTAMENTO D
WHERE NOT EXISTS
    (SELECT F.CODIGODEPARTAMENTO
     FROM FUNCIONARIO F
     WHERE D.CODIGODEPARTAMENTO=F.CODIGODEPARTAMENTO);
```

No final do **módulo 3**, o especialista realizará consultas sobre as tabelas FUNCIONÁRIO, ALUNO e CLIENTE, usando a Query Tool da ferramenta PG Admin 4, demonstrando as operações UNION, INTERSECT e EXCEPT, com capturas das telas.

```
CREATE TABLE FUNCIONARIO (  
    CODIGOFUNCIONARIO int NOT NULL,  
    NOME varchar(90) NOT NULL,  
    CPF char(15) NULL,  
    SEXO char(1) NOT NULL,  
    DTNASCIMENTO date NOT NULL,  
    SALARIO real NULL,  
    CONSTRAINT FUNCIONARIO_pk PRIMARY KEY (CODIGOFUNCIONARIO));
```

```
CREATE TABLE ALUNO (  
    CODIGOALUNO int NOT NULL,  
    NOME varchar(90) NOT NULL,  
    CPF char(15) NOT NULL,  
    SEXO char(1) NOT NULL,  
    DTNASCIMENTO date NOT NULL,  
    CONSTRAINT ALUNO_pk PRIMARY KEY (CODIGOALUNO));
```

```
CREATE TABLE CLIENTE (  
    CODIGOCLIENTE int NOT NULL,  
    NOME varchar(90) NOT NULL,  
    CPF char(15) NOT NULL,  
    SEXO char(1) NOT NULL,  
    CONSTRAINT CLIENTE_pk PRIMARY KEY (CODIGOCLIENTE));
```

```
INSERT INTO FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CPF, SEXO,  
DTNASCIMENTO, SALARIO)  
VALUES (1,'ROBERTA SILVA BRASIL','82998','F','20/02/1980',7000);  
INSERT INTO FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CPF, SEXO,  
DTNASCIMENTO, SALARIO)  
VALUES (2,'MARIA SILVA BRASIL','9876','F','20/09/1988',9500);  
INSERT INTO FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CPF, SEXO,  
DTNASCIMENTO, SALARIO)  
VALUES (3,'GABRIELLA PEREIRA  
LIMA','32998','F','20/02/1990',6000);  
INSERT INTO FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CPF, SEXO,  
DTNASCIMENTO, SALARIO)  
VALUES (4,'MARCOS PEREIRA BRASIL','9999','M','20/02/1999',6000);  
INSERT INTO FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CPF, SEXO,  
DTNASCIMENTO, SALARIO)  
VALUES (5,'HEMERSON SILVA BRASIL','9111','M','20/12/1992',4000);
```

```
INSERT INTO ALUNO (CODIGOALUNO, NOME, CPF, SEXO, DTNASCIMENTO)  
VALUES (1,'JOSÉ FRANCISCO TERRA','82988','M','28/10/1989');
```

```
INSERT INTO ALUNO (CODIGOALUNO, NOME, CPF, SEXO, DTNASCIMENTO)
VALUES (2,'ANDREY COSTA FILHO','0024','M','20/10/1999');
INSERT INTO ALUNO (CODIGOALUNO, NOME, CPF, SEXO, DTNASCIMENTO)
VALUES (3,'ROBERTA SILVA BRASIL','82998','F','20/02/1980');
INSERT INTO ALUNO (CODIGOALUNO, NOME, CPF, SEXO, DTNASCIMENTO)
VALUES (4,'CARLA MARIA MACIEL','0044','F','20/11/1996');
INSERT INTO ALUNO (CODIGOALUNO, NOME, CPF, SEXO, DTNASCIMENTO)
VALUES (5,'MARCOS PEREIRA BRASIL','9999','M','20/02/1999');
```

```
INSERT INTO CLIENTE (CODIGOCLIENTE, NOME, CPF, SEXO)
VALUES (1,'ROBERTA SILVA BRASIL','82998','F');
INSERT INTO CLIENTE (CODIGOCLIENTE, NOME, CPF, SEXO)
VALUES (2,'MARCOS PEREIRA BRASIL','9999','M');
INSERT INTO CLIENTE (CODIGOCLIENTE, NOME, CPF, SEXO)
VALUES (3,'HEMERSON SILVA BRASIL','9111','M');
```

```
SELECT * FROM FUNCIONARIO;
TABLE ALUNO;
TABLE CLIENTE;
```

```
SELECT NOME, CPF
FROM FUNCIONARIO
UNION
SELECT NOME, CPF
FROM CLIENTE;
```

```
SELECT NOME, CPF
FROM FUNCIONARIO
UNION ALL
SELECT NOME, CPF
FROM CLIENTE;
```

```
SELECT NOME, CPF, 'Dados da tabela FUNCIONARIO' AS ORIGEM
FROM FUNCIONARIO
UNION ALL
SELECT NOME, CPF, 'Dados da tabela CLIENTE' AS ORIGEM
FROM CLIENTE;
```

```
SELECT NOME, CPF
FROM FUNCIONARIO
INTERSECT
SELECT NOME, CPF
FROM CLIENTE;
```

```
SELECT NOME, CPF
FROM FUNCIONARIO
INTERSECT
SELECT NOME, CPF
```

```
FROM CLIENTE
INTERSECT
SELECT NOME, CPF
FROM ALUNO;
```

```
SELECT NOME, CPF
FROM FUNCIONARIO
UNION
SELECT NOME, CPF
FROM CLIENTE
INTERSECT
SELECT NOME, CPF
FROM ALUNO;
```

```
SELECT NOME, CPF
FROM FUNCIONARIO
EXCEPT
SELECT NOME, CPF
FROM CLIENTE;
```

```
SELECT NOME, CPF
FROM CLIENTE
EXCEPT
SELECT NOME, CPF
FROM FUNCIONARIO;
```

```
SELECT NOME, CPF
FROM FUNCIONARIO
EXCEPT
SELECT NOME, CPF
FROM CLIENTE
EXCEPT
SELECT NOME, CPF
FROM ALUNO;
```

```
SELECT NOME, CPF
FROM ALUNO
EXCEPT
SELECT NOME, CPF
FROM CLIENTE;
```

```
SELECT NOME, CPF
FROM ALUNO
UNION ALL
SELECT NOME, CPF
FROM CLIENTE;
```

```
SELECT NOME, CPF
FROM ALUNO
```



```
UNION  
SELECT NOME, CPF  
FROM CLIENTE;
```

```
SELECT NOME, CPF  
FROM ALUNO  
INTERSECT  
SELECT NOME, CPF  
FROM CLIENTE;
```