

CONSULTAS EM UMA TABELA NO POSTGRESQL

No final do módulo 1, o especialista realizará as consultas apresentadas sobre a tabela ALUNO, usando a Query Tool da ferramenta PG Admin 4, com capturas das telas.

Módulo 1

Estrutura básica de um comando SELECT

```
CREATE TABLE ALUNO (  
    CODIGOALUNO int NOT NULL,  
    NOME varchar(90) NOT NULL,  
    SEXO char(1) NOT NULL,  
    DTNASCIMENTO date NOT NULL,  
    EMAIL varchar(30) NULL,  
    CONSTRAINT ALUNO_pk PRIMARY KEY (CODIGOALUNO));
```

```
INSERT INTO ALUNO (CODIGOALUNO, NOME, SEXO, DTNASCIMENTO,EMAIL) VALUES (1,'JOSÉ FRANCISCO  
TERRA','M','28/10/1989','JFT@GMAIL.COM');  
  
INSERT INTO ALUNO (CODIGOALUNO, NOME, SEXO, DTNASCIMENTO,EMAIL) VALUES (2,'ANDREY COSTA  
FILHO','M','20/10/1999','ANDREYCF@HOTMAIL.COM');  
  
INSERT INTO ALUNO (CODIGOALUNO, NOME, SEXO, DTNASCIMENTO,EMAIL) VALUES (3,'PATRÍCIA TORRES LOUREIRO','F','20/10/1980',  
'PTORRES@GMAIL.COM');  
  
INSERT INTO ALUNO (CODIGOALUNO, NOME, SEXO, DTNASCIMENTO,EMAIL) VALUES (4,'CARLA MARIA MACIEL','F','20/11/1996',NULL);  
INSERT INTO ALUNO (CODIGOALUNO, NOME, SEXO, DTNASCIMENTO,EMAIL) VALUES (5,'LEILA SANTANA COSTA','F','20/11/2001',NULL);
```

Consulta 01

```
SELECT * FROM ALUNO;
```

Consulta 02

```
SELECT CODIGOALUNO, NOME, DTNASCIMENTO  
FROM ALUNO;
```

Consulta 02 pode ser reescrita conforme a seguir:

```
SELECT    CODIGOALUNO AS "Matrícula",  
          NOME AS "Nome do discente",  
          DTNASCIMENTO AS "Data de nascimento"  
FROM ALUNO;
```

Funções de data e hora

```
SELECT CURRENT_DATE AS "Data Atual",  
       CURRENT_TIME AS "Hora Atual",  
       CURRENT_TIMESTAMP "Data e Hora atuais",  
-- DOW 0 - domingo, 1 - segunda, ..., 6 - sábado  
       EXTRACT( DOW FROM CURRENT_DATE) AS "Dia da semana",  
       EXTRACT( DAY FROM CURRENT_DATE) AS "Dia Atual",  
       EXTRACT( DOY FROM CURRENT_DATE) AS "Dia do ano",  
       EXTRACT( MONTH FROM CURRENT_DATE) AS "Mês Atual",  
       EXTRACT( YEAR FROM CURRENT_DATE) AS "Ano Atual",  
       EXTRACT( CENTURY FROM CURRENT_DATE) AS "Século Atual";
```

Comando SELECT sem a cláusula FROM

```
SELECT 5+5;
```

Exibindo o nome do dia da semana

```
SELECT CASE WHEN extract(dow from current_date) = 0 THEN 'domingo'
            WHEN extract(dow from current_date) = 1 THEN 'segunda-feira'
            WHEN extract(dow from current_date) = 2 THEN 'terça-feira'
            WHEN extract(dow from current_date) = 3 THEN 'quarta-feira'
            WHEN extract(dow from current_date) = 4 THEN 'quinta-feira'
            WHEN extract(dow from current_date) = 5 THEN 'sexta-feira'
            WHEN extract(dow from current_date)= 6 THEN 'sábado'
            END AS "Nome do dia da semana";
```

Calculando idade e faixa etária

```
SELECT  NOME,
        DTNASCIMENTO,
        AGE(DTNASCIMENTO) AS "Idade [ano/mês/dia]",
        EXTRACT(YEAR FROM AGE(DTNASCIMENTO)) AS "Idade do Aluno"
FROM ALUNO;
```

Exibir o nome, a idade e a faixa etária dos alunos

```
SELECT  NOME,
        EXTRACT(YEAR FROM AGE(DTNASCIMENTO))      AS "Idade do Aluno",
        CASE WHEN EXTRACT(YEAR FROM AGE(DTNASCIMENTO)) <=20 THEN '1. até 20 anos'
              WHEN EXTRACT(YEAR FROM AGE(DTNASCIMENTO)) BETWEEN 21 AND 30 THEN '2. 21 a 30 anos'
              WHEN EXTRACT(YEAR FROM AGE(DTNASCIMENTO)) BETWEEN 31 AND 40 THEN '3. 31 a 40 anos'
              WHEN EXTRACT(YEAR FROM AGE(DTNASCIMENTO)) BETWEEN 41 AND 50 THEN '4. 41 a 50 anos'
              WHEN EXTRACT(YEAR FROM AGE(DTNASCIMENTO)) BETWEEN 51 AND 60 THEN '5. 51 a 60 anos'
              WHEN EXTRACT(YEAR FROM AGE(DTNASCIMENTO)) > 60 THEN '6. mais de 60 anos'
        END AS "Faixa Etária"
FROM ALUNO;
```

Funções de resumo ou de agregação

```
SELECT
        COUNT(*) AS "Número de alunos",
        MIN(EXTRACT(YEAR FROM AGE(DTNASCIMENTO)))      AS "Menor Idade",
        AVG(EXTRACT(YEAR FROM AGE(DTNASCIMENTO)))      AS "Idade Média",
        MAX(EXTRACT(YEAR FROM AGE(DTNASCIMENTO)))      AS "Maior Idade",
        SUM(EXTRACT(YEAR FROM AGE(DTNASCIMENTO)))/COUNT(*) AS "Idade Média"
FROM ALUNO;
```

Listando resumos em uma linha

```
SELECT COUNT(*) NCURSOS FROM CURSO;
```

```
SELECT COUNT(*) NDISCIPLINAS FROM DISCIPLINA;
```

```
SELECT COUNT(*) NALUNOS FROM ALUNO;
```

ou

```
SELECT
```

```
    (SELECT COUNT(*) NCURSOS FROM CURSO),
```

```
    (SELECT COUNT(*) NALUNOS FROM ALUNO),
```

```
    (SELECT COUNT(*) NDISCIPLINAS FROM DISCIPLINA);
```

Criando tabela a partir de consulta

```
CREATE TABLE TTESTE AS
```

```
SELECT  NOME,
```

```
        EXTRACT(YEAR FROM AGE(DTNASCIMENTO))      AS "Idade do Aluno",
```

```
        CASE WHEN EXTRACT(YEAR FROM AGE(DTNASCIMENTO)) <=20  THEN '1. até 20 anos'
```

```
              WHEN EXTRACT(YEAR FROM AGE(DTNASCIMENTO)) BETWEEN 21 AND 30  THEN '2. 21 a 30 anos'
```

```
              WHEN EXTRACT(YEAR FROM AGE(DTNASCIMENTO)) BETWEEN 31 AND 40  THEN '3. 31 a 40 anos'
```

```
              WHEN EXTRACT(YEAR FROM AGE(DTNASCIMENTO)) BETWEEN 41 AND 50  THEN '4. 41 a 50 anos'
```

```
              WHEN EXTRACT(YEAR FROM AGE(DTNASCIMENTO)) BETWEEN 51 AND 60  THEN '5. 51 a 60 anos'
```

```
              WHEN EXTRACT(YEAR FROM AGE(DTNASCIMENTO)) > 60  THEN '6. mais de 60 anos'
```

```
        END AS "Faixa Etária"
```

```
FROM ALUNO;
```

Criando view a partir de consulta

```
CREATE VIEW VTESTE AS
SELECT  NOME,
        EXTRACT(YEAR FROM AGE(DTNASCIMENTO))      AS "Idade do Aluno",
        CASE WHEN EXTRACT(YEAR FROM AGE(DTNASCIMENTO)) <=20 THEN '1. até 20 anos'
              WHEN EXTRACT(YEAR FROM AGE(DTNASCIMENTO)) BETWEEN 21 AND 30 THEN '2. 21 a 30 anos'
              WHEN EXTRACT(YEAR FROM AGE(DTNASCIMENTO)) BETWEEN 31 AND 40 THEN '3. 31 a 40 anos'
              WHEN EXTRACT(YEAR FROM AGE(DTNASCIMENTO)) BETWEEN 41 AND 50 THEN '4. 41 a 50 anos'
              WHEN EXTRACT(YEAR FROM AGE(DTNASCIMENTO)) BETWEEN 51 AND 60 THEN '5. 51 a 60 anos'
              WHEN EXTRACT(YEAR FROM AGE(DTNASCIMENTO)) > 60 THEN '6. mais de 60 anos'
        END AS "Faixa Etária"
FROM ALUNO;
```

Para executar o código associado à view

```
SELECT * FROM VTESTE;
```

Módulo 2

No final do módulo 2, o especialista realizará as consultas apresentadas sobre a tabela ALUNO, com diferentes condições na cláusula WHERE, usando a Query Tool da ferramenta PG Admin 4, com capturas das telas.

Clausula WHERE e operadores da SQL

```
CREATE TABLE ALUNO (  
    CODIGOALUNO int NOT NULL,  
    NOME varchar(90) NOT NULL,  
    SEXO char(1) NOT NULL,  
    DTNASCIMENTO date NOT NULL,  
    EMAIL varchar(30) NULL,  
    CONSTRAINT ALUNO_pk PRIMARY KEY (CODIGOALUNO));
```

```
INSERT INTO ALUNO (CODIGOALUNO, NOME, SEXO, DTNASCIMENTO,EMAIL) VALUES (1,'JOSÉ FRANCISCO  
TERRA','M','28/10/1989','JFT@GMAIL.COM');  
  
INSERT INTO ALUNO (CODIGOALUNO, NOME, SEXO, DTNASCIMENTO,EMAIL) VALUES (2,'ANDREY COSTA  
FILHO','M','20/10/1999','ANDREYCF@HOTMAIL.COM');  
  
INSERT INTO ALUNO (CODIGOALUNO, NOME, SEXO, DTNASCIMENTO,EMAIL) VALUES (3,'PATRÍCIA TORRES LOUREIRO','F','20/10/1980',  
'PTORRES@GMAIL.COM');  
  
INSERT INTO ALUNO (CODIGOALUNO, NOME, SEXO, DTNASCIMENTO,EMAIL) VALUES (4,'CARLA MARIA MACIEL','F','20/11/1996',NULL);  
INSERT INTO ALUNO (CODIGOALUNO, NOME, SEXO, DTNASCIMENTO,EMAIL) VALUES (5,'LEILA SANTANA COSTA','F','20/11/2001',NULL);
```

Consulta 01

```
SELECT NOME, DTNASCIMENTO  
FROM ALUNO  
WHERE SEXO='F';
```

Consulta 02

```
SELECT NOME, DTNASCIMENTO  
FROM ALUNO  
WHERE SEXO='F' AND EXTRACT (MONTH FROM DTNASCIMENTO)=11;
```

```
SELECT    CURRENT_DATE AS "Data de Emissão do Relatório",  
          CODIGOALUNO AS "Matrícula",  
          NOME AS "Nome do discente",  
          DTNASCIMENTO AS "Data de nascimento",  
          CASE  
              WHEN SEXO='M' THEN 'Masculino'  
              WHEN SEXO='F' THEN 'Feminino'  
          END AS SEXO  
FROM ALUNO;
```

Consulta 03

```
SELECT NOME, DTNASCIMENTO  
FROM ALUNO  
WHERE EXTRACT (MONTH FROM DTNASCIMENTO)IN (7,8,9,10,11,12);
```

Consulta 04

```
SELECT NOME  
FROM ALUNO
```



```
WHERE EXTRACT (YEAR FROM DTNASCIMENTO) BETWEEN 1985 AND 2005;
```

Extrair o mesmo resultado sem o uso do BETWEEN

```
SELECT NOME
```

```
FROM ALUNO
```

```
WHERE EXTRACT (YEAR FROM DTNASCIMENTO) >= 1985 AND EXTRACT (YEAR FROM DTNASCIMENTO) <=2005;
```

Consulta 05

```
SELECT NOME
```

```
FROM ALUNO
```

```
WHERE NOME LIKE '%COSTA%';
```

Consulta 06

```
SELECT NOME
```

```
FROM ALUNO
```

```
WHERE NOME LIKE '_A%';
```

Consulta 07

```
SELECT NOME, DTNASCIMENTO
```

```
FROM ALUNO
```

```
WHERE NOME NOT LIKE '%MARIA%';
```

Consulta 08

```
SELECT COUNT(*) AS QUANTIDADE
```

```
FROM ALUNO
```

```
WHERE EMAIL LIKE '%@GMAIL.%';
```

Consulta 09

```
SELECT NOME, DTNASCIMENTO, EMAIL  
FROM ALUNO  
WHERE EMAIL IS NOT NULL;
```

Consulta 10

```
SELECT NOME  
FROM ALUNO  
WHERE EMAIL IS NULL;
```

Consulta 11

```
SELECT NOME, DTNASCIMENTO  
FROM ALUNO  
ORDER BY NOME;
```

Consulta 12

```
SELECT NOME, DTNASCIMENTO  
FROM ALUNO  
ORDER BY EXTRACT(MONTH FROM DTNASCIMENTO),NOME;
```

Módulo 3

No final do módulo 3, o especialista realizará as consultas apresentadas sobre a tabela FUNCIONARIO, ilustrando o emprego das cláusulas GROUP BY e HAVING, usando a Query Tool da ferramenta PG Admin 4, com capturas das telas.

Consultas com GROUP BY e HAVING

```
CREATE TABLE FUNCIONARIO (  
    CODIGOFUNCIONARIO int NOT NULL,  
    NOME char(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));  
  
INSERT INTO FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CPF, SEXO, DTNASCIMENTO, SALARIO)  
VALUES (1, 'ROBERTA SILVA BRASIL', NULL, 'F', '20/02/1980', 7000);  
INSERT INTO FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CPF, SEXO, DTNASCIMENTO, SALARIO)  
VALUES (2, 'MARIA SILVA BRASIL', NULL, 'F', '20/09/1988', 9500);  
INSERT INTO FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CPF, SEXO, DTNASCIMENTO, SALARIO)  
VALUES (3, 'GABRIELLA PEREIRA LIMA', NULL, 'F', '20/02/1990', 6000);  
INSERT INTO FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CPF, SEXO, DTNASCIMENTO, SALARIO)  
VALUES (4, 'MARCOS PEREIRA BRASIL', NULL, 'M', '20/02/1999', 6000);  
INSERT INTO FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CPF, SEXO, DTNASCIMENTO, SALARIO)  
VALUES (5, 'HEMERSON SILVA BRASIL', NULL, 'M', '20/12/1992', 4000);
```

Exibir todo o conteúdo após a criação da tabela e a inserção dos registros

```
SELECT *  
FROM FUNCIONARIO;
```

Grupo de dados

```
SELECT DISTINCT SEXO  
FROM FUNCIONARIO;
```

Grupo de dados com GROUP BY

```
SELECT SEXO  
FROM FUNCIONARIO  
GROUP BY SEXO;
```

Consulta 01

```
SELECT SEXO, COUNT(*) AS QUANTIDADE  
FROM FUNCIONARIO  
GROUP BY SEXO;
```

Código para exibir a consulta em uma única linha

```
SELECT  
    (SELECT COUNT(*) AS "M" FROM FUNCIONARIO WHERE SEXO='M'),  
    (SELECT COUNT(*) AS "F" FROM FUNCIONARIO WHERE SEXO='F');
```

Consulta 02

```
SELECT SEXO,  
        AVG(SALARIO) AS MEDIASALARIAL  
FROM FUNCIONARIO  
GROUP BY SEXO;
```

Consulta 03

```
SELECT EXTRACT(MONTH FROM DTNASCIMENTO) AS MES,  
        COUNT(*) AS QUANTIDADE,  
        MIN(SALARIO) AS MENORSALARIO,  
        ROUND(AVG(SALARIO)::NUMERIC,0) AS SALARIOMEDIO,  
        MAX(SALARIO) AS MAIORSALARIO  
FROM FUNCIONARIO  
GROUP BY EXTRACT(MONTH FROM DTNASCIMENTO)  
ORDER BY EXTRACT(MONTH FROM DTNASCIMENTO);
```

Consulta 04

```
SELECT EXTRACT(MONTH FROM DTNASCIMENTO) AS MES,  
        SEXO,  
        COUNT(*) AS QUANTIDADE  
FROM FUNCIONARIO  
GROUP BY EXTRACT(MONTH FROM DTNASCIMENTO),SEXO  
ORDER BY EXTRACT(MONTH FROM DTNASCIMENTO);
```

Consulta 05

```
SELECT  EXTRACT(MONTH FROM DTNASCIMENTO) AS MES,  
        COUNT(*) AS QUANTIDADE,  
        SUM(SALARIO*0.05) AS TOTALBONUS  
FROM FUNCIONARIO  
GROUP BY EXTRACT(MONTH FROM DTNASCIMENTO)  
HAVING COUNT(*)>1  
ORDER BY EXTRACT(MONTH FROM DTNASCIMENTO);
```