

DDL

O DDL para criar a tabela alunos no PostgreSQL é o seguinte:

SQL

```
CREATE TABLE alunos (  
    aluno_id SERIAL PRIMARY KEY,  
    p_nome VARCHAR(50) NOT NULL,  
    u_nome VARCHAR(50) NOT NULL,  
    matricula INTEGER NOT NULL,  
    serie VARCHAR(50) NOT NULL,  
    disciplina VARCHAR(50) NOT NULL,  
    nota DECIMAL(2,1) NOT NULL  
);
```

DML

O DML para inserir os dados da tabela alunos no PostgreSQL é o seguinte:

SQL

```
INSERT INTO alunos (p_nome, u_nome, matricula, serie, disciplina,  
nota)  
VALUES  
    ('Vitória', 'Claudino', 5542, '2° ano', 'Matemática', 7.0),  
    ('Luiz', 'Silva', 6215, '1° ano', 'Português', 8.0),  
    ('André', 'Carvalho', 4521, '3° ano', 'Matemática', 9.5),  
    ('Alan', 'Vilela', 3285, '1° ano', 'História', 8.0),  
    ('Figueiredo', 'Santos', 4598, '2° ano', 'Geografia', 9.0);
```

SQL Online IDE

sqliteonline.com

SQLite PostgreSQL PostgreSQL2

```
1 CREATE TABLE alunos (  
2   aluno_id SERIAL PRIMARY KEY,  
3   p_nome VARCHAR(50) NOT NULL,  
4   u_nome VARCHAR(50) NOT NULL,  
5   matricula INTEGER NOT NULL,  
6   serie VARCHAR(50) NOT NULL,  
7   disciplina VARCHAR(50) NOT NULL,  
8   nota DECIMAL(2,1) NOT NULL  
9 );  
10 INSERT INTO alunos (p_nome, u_nome, matricula, serie, disciplina, nota)  
11 VALUES  
12 ('Vitória', 'Claudino', 5542, '2º ano', 'Matemática', 7.0),  
13 ('Luiz', 'Silva', 6215, '1º ano', 'Português', 8.0),  
14 ('André', 'Carvalho', 4521, '3º ano', 'Matemática', 9.5),  
15 ('Alan', 'Vilela', 3285, '1º ano', 'História', 8.0),  
16 ('Figueiredo', 'Santos', 4598, '2º ano', 'Geografia', 9.0);  
17 SELECT * FROM alunos;
```

aluno_id	p_nome	u_nome	matricula	serie	disciplina	nota
1	Vitória	Claudino	5542	2º ano	Matemática	7.0
2	Luiz	Silva	6215	1º ano	Português	8.0
3	André	Carvalho	4521	3º ano	Matemática	9.5
4	Alan	Vilela	3285	1º ano	História	8.0
5	Figueiredo	Santos	4598	2º ano	Geografia	9.0

- Mostre as informações apenas dos alunos aprovados. A aprovação é acima de 7,0;

SQL Online IDE

sqliteonline.com

SQLite PostgreSQL PostgreSQL2

```
8   nota DECIMAL(2,1) NOT NULL  
9 );  
10  
11 INSERT INTO alunos (p_nome, u_nome, matricula, serie, disciplina, nota)  
12 VALUES  
13 ('Vitória', 'Claudino', 5542, '2º ano', 'Matemática', 7.0),  
14 ('Luiz', 'Silva', 6215, '1º ano', 'Português', 8.0),  
15 ('André', 'Carvalho', 4521, '3º ano', 'Matemática', 9.5),  
16 ('Alan', 'Vilela', 3285, '1º ano', 'História', 8.0),  
17 ('Figueiredo', 'Santos', 4598, '2º ano', 'Geografia', 9.0);  
18  
19 SELECT * FROM alunos;  
20  
21 SELECT * FROM alunos WHERE nota > 7;  
22  
23  
24  
25
```

aluno_id	p_nome	u_nome	matricula	serie	disciplina	nota
2	Luiz	Silva	6215	1º ano	Português	8.0
3	André	Carvalho	4521	3º ano	Matemática	9.5
4	Alan	Vilela	3285	1º ano	História	8.0
5	Figueiredo	Santos	4598	2º ano	Geografia	9.0

Exiba as informações dos alunos do primeiro ano com nota maior ou igual a 8,0;

The screenshot shows the SQL Online IDE interface. The left sidebar displays a database structure with a PostgreSQL database containing a table named 'alunos'. The main editor area shows a SQL query that inserts data into the 'alunos' table and then selects all records. The results pane at the bottom shows the data after the query execution.

```
10
11 INSERT INTO alunos (p_nome, u_nome, matricula, serie, disciplina, nota)
12 VALUES
13 ('Vitória', 'Claudino', 5542, '2º ano', 'Matemática', 7.0),
14 ('Luiz', 'Silva', 6215, '1º ano', 'Português', 8.0),
15 ('André', 'Carvalho', 4521, '3º ano', 'Matemática', 9.5),
16 ('Alan', 'Vilela', 3285, '1º ano', 'História', 8.0),
17 ('Figueiredo', 'Santos', 4598, '2º ano', 'Geografia', 9.0);
18
19 SELECT * FROM alunos;
20
21 SELECT * FROM alunos WHERE nota > 7;
22
23 SELECT * FROM alunos WHERE serie = '1º ano' AND nota >= 8;
```

aluno_id	p_nome	u_nome	matricula	serie	disciplina	nota
2	Luiz	Silva	6215	1º ano	Português	8.0
4	Alan	Vilela	3285	1º ano	História	8.0

Exiba apenas os nomes e as notas dos alunos;

The screenshot shows the SQL Online IDE interface. The left sidebar displays a database structure with a PostgreSQL database containing a table named 'alunos'. The main editor area shows a SQL query that selects the first and last names of students along with their grades. The results pane at the bottom shows the data after the query execution.

```
12 VALUES
13 ('Vitória', 'Claudino', 5542, '2º ano', 'Matemática', 7.0),
14 ('Luiz', 'Silva', 6215, '1º ano', 'Português', 8.0),
15 ('André', 'Carvalho', 4521, '3º ano', 'Matemática', 9.5),
16 ('Alan', 'Vilela', 3285, '1º ano', 'História', 8.0),
17 ('Figueiredo', 'Santos', 4598, '2º ano', 'Geografia', 9.0);
18
19 SELECT * FROM alunos;
20
21 SELECT * FROM alunos WHERE nota > 7;
22
23 SELECT * FROM alunos WHERE serie = '1º ano' AND nota >= 8;
24
25 SELECT p_nome, u_nome, nota FROM alunos;
```

p_nome	u_nome	nota
Vitória	Claudino	7.0
Luiz	Silva	8.0
André	Carvalho	9.5
Alan	Vilela	8.0
Figueiredo	Santos	9.0

- Crie uma tabela PROFESSOR que apresente apenas o primeiro e o último nome do professor;

- Crie uma tabela ALUNO com o primeiro e o último nome de cada;

The screenshot shows the SQL Online IDE interface. The left sidebar lists databases: SQLite, MariaDB, PostgreSQL (selected), and MS SQL. Under PostgreSQL, there are tables: alunos, demo, and professor. The main editor shows SQL code for creating the 'professor' table and inserting data. The results pane at the bottom displays the data for the 'professor' table.

```
24
25 SELECT p_nome, u_nome, nota FROM alunos;
26
27 CREATE TABLE professor (
28   professor_id SERIAL PRIMARY KEY,
29   p_nome VARCHAR(50) NOT NULL,
30   u_nome VARCHAR(50) NOT NULL
31 );
32
33 INSERT INTO professor (p_nome, u_nome)
34 VALUES
35   ('Gilberto', 'Ferreira'),
36   ('Elon', 'Musk'),
37   ('Bill', 'Gates'),
38   ('Alan', 'Turing'),
39   ('Eliezer', 'Moraes');
40
41 SELECT * FROM professor;
```

professor_id	p_nome	u_nome
1	Gilberto	Ferreira
2	Elon	Musk
3	Bill	Gates
4	Alan	Turing
5	Eliezer	Moraes

- Mostre o resultado da união entre a tabela ALUNO(PNome, UNome) e a tabela PROFESSOR;

The screenshot shows the SQL Online IDE interface. The left sidebar lists databases: SQLite, MariaDB, PostgreSQL (selected), and MS SQL. Under PostgreSQL, there are tables: alunos, demo, and professor. The main editor shows SQL code for selecting data from the 'professor' and 'alunos' tables and combining them using a UNION. The results pane at the bottom displays the combined data.

```
40
41 SELECT * FROM professor;
42
43 SELECT p_nome, u_nome FROM alunos
44 UNION
45 SELECT p_nome, u_nome FROM professor;
46
47
```

p_nome	u_nome
Eliezer	Moraes
Figueiredo	Santos
Elon	Musk
Alan	Vilela
André	Carvalho
Vitória	Claudino
Alan	Turing
Bill	Gates
Gilberto	Ferreira
Luiz	Silva

- Exiba o resultado da intersecção entre a tabela ALUNO(PNome, UNome) e a tabela PROFESSOR;

Para essa resolução tive que alterar dois registros da tabela professor, conforme segue nas queries.

```
47 UPDATE professor SET p_nome = 'Alan', u_nome = 'Vilela' WHERE p_nome = 'Gilberto';  
48 UPDATE professor SET p_nome = 'André', u_nome = 'Carvalho' WHERE p_nome = 'Elon';
```

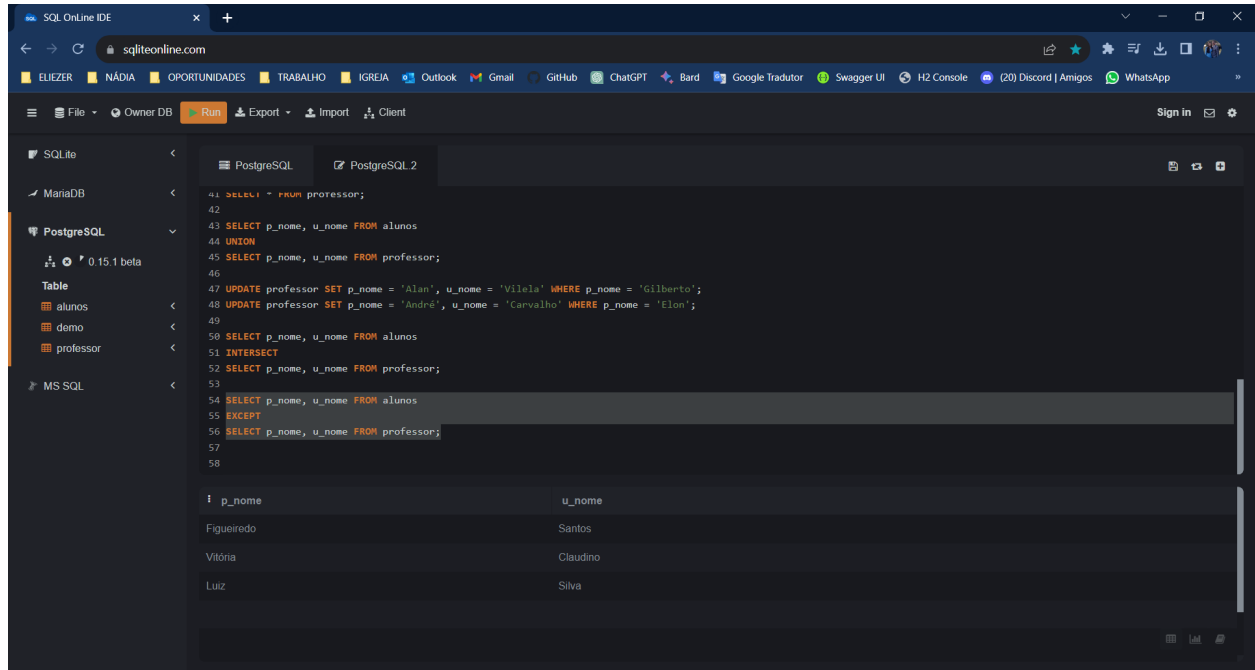
Resultado da intersecção:

The screenshot shows the SQL Online IDE interface. The left sidebar lists databases: SQLite, MariaDB, PostgreSQL (selected), and MS SQL. Under PostgreSQL, there are tables: alunos, demo, and professor. The main editor shows a series of SQL queries. The first query is a SELECT statement. The second query is an UPDATE statement. The third query is an UPDATE statement. The fourth query is a SELECT statement. The fifth query is an INTERSECT statement. The sixth query is a SELECT statement. The results of the queries are displayed in a table with two columns: p_nome and u_nome. The table contains two rows: Alan and Vilela, and André and Carvalho.

```
37 \ DALL, 'WILEY',  
38 ('Alan', 'Turing'),  
39 ('Eliezer', 'Morales');  
40  
41 SELECT * FROM professor;  
42  
43 SELECT p_nome, u_nome FROM alunos  
44 UNION  
45 SELECT p_nome, u_nome FROM professor;  
46  
47 UPDATE professor SET p_nome = 'Alan', u_nome = 'Vilela' WHERE p_nome = 'Gilberto';  
48 UPDATE professor SET p_nome = 'André', u_nome = 'Carvalho' WHERE p_nome = 'Elon';  
49  
50 SELECT p_nome, u_nome FROM alunos  
51 INTERSECT  
52 SELECT p_nome, u_nome FROM professor;  
53  
54
```

p_nome	u_nome
Alan	Vilela
André	Carvalho

- Exiba o resultado da diferença entre a tabela ALUNO(PNome, UNome) e a tabela PROFESSOR.



The screenshot shows the SQL Online IDE interface. The left sidebar displays a database structure for PostgreSQL, including tables 'alunos', 'demo', and 'professor'. The main editor area contains a SQL query with line numbers 41 through 58. The query performs several operations: deleting from 'professor', selecting from 'alunos' and 'professor' with a UNION, updating 'professor' with specific values, and finally selecting the difference between 'alunos' and 'professor' using EXCEPT. The results of the EXCEPT query are displayed in a table below the editor.

```
41 DELETE FROM professor;
42
43 SELECT p_nome, u_nome FROM alunos
44 UNION
45 SELECT p_nome, u_nome FROM professor;
46
47 UPDATE professor SET p_nome = 'Alan', u_nome = 'Vilela' WHERE p_nome = 'Gilberto';
48 UPDATE professor SET p_nome = 'André', u_nome = 'Carvalho' WHERE p_nome = 'Elon';
49
50 SELECT p_nome, u_nome FROM alunos
51 INTERSECT
52 SELECT p_nome, u_nome FROM professor;
53
54 SELECT p_nome, u_nome FROM alunos
55 EXCEPT
56 SELECT p_nome, u_nome FROM professor;
57
58
```

p_nome	u_nome
Figueiredo	Santos
Vitória	Claudino
Luz	Silva