Criando Bancos e Tabelas no PostgreSQL

16.5

Aula 8

Introdução a linguagem SQL

Introdução ao Where

Where - onde

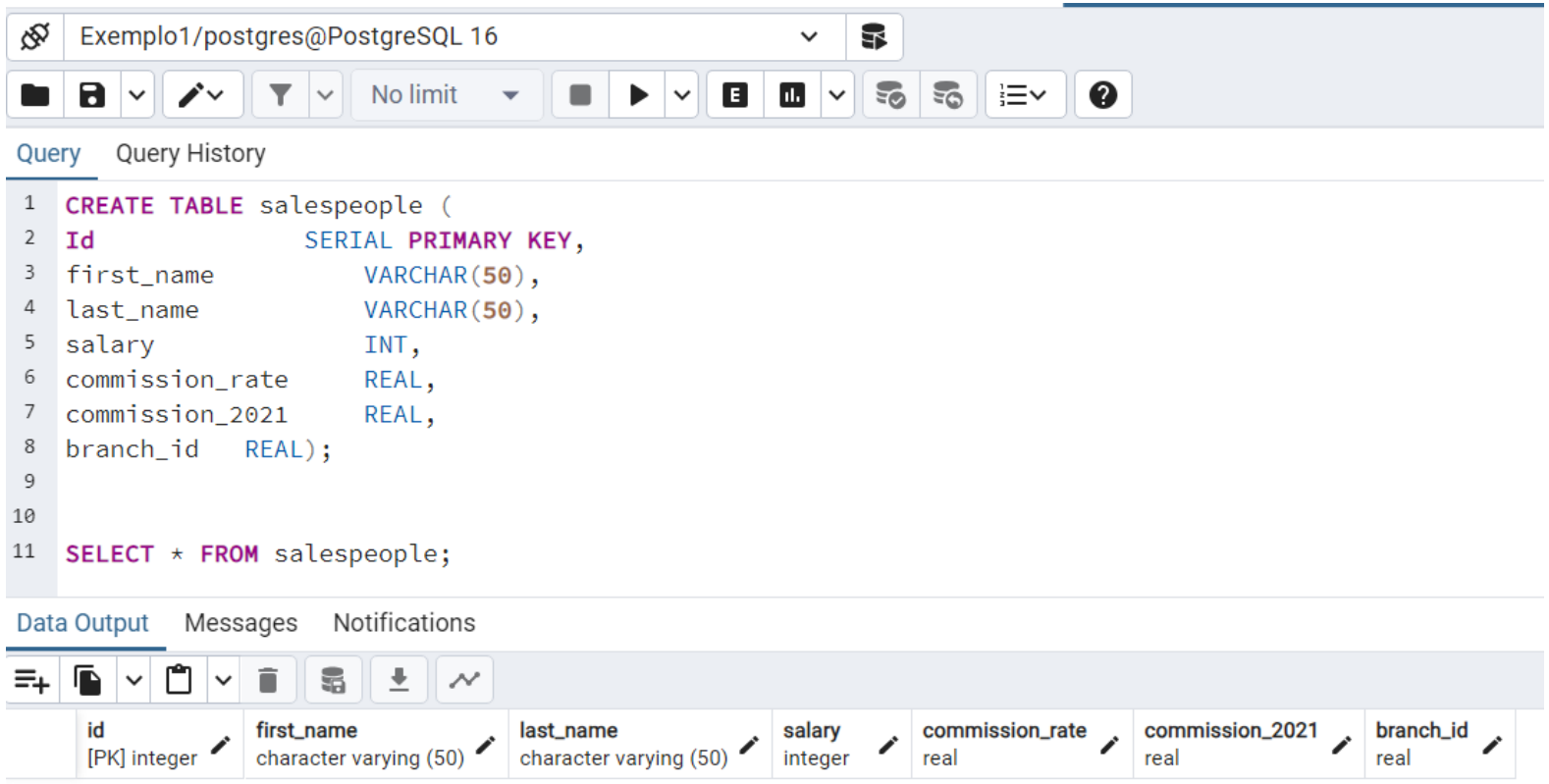
A cláusula WHERE é utilizada para extrair resultados que cumpram determinadas condições, ou seja, funciona como um filtro para que seja retornado apenas resultados específicos em uma consulta.

Comparação

Para valores numéricos, pode-se utilizar estes operadores de comparação:

= Igual a.
!= (ou <>) Não igual a.
< Menos do que.
<= Menor ou igual a.
> Maior do que.
>= Maior ou igual a.

```
CREATE TABLE salespeople (
    Id                  SERIAL PRIMARY KEY,
    first_name          VARCHAR(50),
    last_name           VARCHAR(50),
    salary              INT,
    commission_rate     REAL,
    commission_2021     REAL,
    branch_id           REAL);
```



```
INSERT INTO salespeople (id,
first_name,last_name,salary,commission_rate,commission_2021,branch_id)
VALUES
    (11,'Katarina','Rostova',45000,0.15,47345.60,1),
    (12,'Alina','Park',43000,0.15,45678.90,2),
    (13,'Meera','Malik',50000,0.15,39045.63,2),
    (17,'Samar','Navabi',52000,0.14,23023.45,2),
    (18,'Donald','Ressler',40000,0.14,41345.75,2),
    (20,'Elisabeth','Keen',59000,0.14,45350.00,2),
    (21,'Tom','Keen',41000,0.12,41560.75,1),
    (22,'Dembe','Zuma',40000,0.12,31540.70,5),
    (23,'Aram','Mojtabai',50000,0.12,29050.65,2),
    (30,'Kate','Kaplan',54000,0.10,25760.45,5),
    (32,'Marvin','Gerard',55000,0.10,22500.00,5),
    (34,'Raymond','Reddington',60000,0.10,17570.80,5),
    (35,'Harold','Cooper',57000,0.10,15450.50,2),
    (37,'Ian', 'Garvey',43000,0.08,NULL,1),
    (38,'Ivan','Stepanov',41000,0.08,NULL,1);
```

Query | Query History

```
1  INSERT INTO salespeople (id, first_name,last_name,salary,commission_rate,commission_2021,branch_id)
2  VALUES
3  (11,'Katarina','Rostova',45000,0.15,47345.60,1),
4  (12,'Alina','Park',43000,0.15,45678.90,2),
5  (13,'Meera','Malik',50000,0.15,39045.63,2),
6  (17,'Samar','Navabi',52000,0.14,23023.45,2),
7  (18,'Donald','Ressler',40000,0.14,41345.75,2),
8  (20,'Elisabeth','Keen',59000,0.14,45350.00,2),
9  (21,'Tom','Keen',41000,0.12,41560.75,1),
10 (22,'Dembe','Zuma',40000,0.12,31540.70,5),
11 (23,'Aram','Mojtabai',50000,0.12,29050.65,2),
12 (30,'Kate','Kaplan',54000,0.10,25760.45,5),
13 (32,'Marvin','Gerard',55000,0.10,22500.00,5),
14 (34,'Raymond','Reddington',60000,0.10,17570.80,5),
15 (35,'Harold','Cooper',57000,0.10,15450.50,2),
16 (37,'Ian', 'Garvey',43000,0.08,NULL,1),
17 (38,'Ivan','Stepanov',41000,0.08,NULL,1);
18 SELECT * FROM salespeople
```

Data Output | Messages | Notifications

	id [PK] integer	first_name character varying (50)	last_name character varying (50)	salary integer	commission_rate real	commission_2021 real	branch_id real
1	11	Katarina	Rostova	45000	0.15	47345.6	1
2	12	Alina	Park	43000	0.15	45678.9	2
3	13	Meera	Malik	50000	0.15	39045.63	2
4	17	Samar	Navabi	52000	0.14	23023.45	2
5	18	Donald	Ressler	40000	0.14	41345.75	2
6	20	Elisabeth	Keen	59000	0.14	45350	2

WHERE >=

Como podemos obter os registros de todos os vendedores cujo salário anual é igual ou superior a 50 mil dólares?

```
SELECT *
        FROM salespeople
        WHERE salary >= 50000;
```

Query | Query History

```
1  SELECT *
2      FROM salespeople
3      WHERE salary >= 50000;
4
```

Data Output | Messages | Notifications

	id [PK] integer	first_name character varying (50)	last_name character varying (50)	salary integer	commission_rate real	commission_2021 real	branch_id real
1	13	Meera	Malik	50000	0.15	39045.63	2
2	17	Samar	Navabi	52000	0.14	23023.45	2
3	20	Elisabeth	Keen	59000	0.14	45350	2
4	23	Aram	Mojtabai	50000	0.12	29050.65	2
5	30	Kate	Kaplan	54000	0.1	25760.45	5
6	32	Marvin	Gerard	55000	0.1	22500	5
7	34	Raymond	Reddington	60000	0.1	17570.8	5
8	35	Harold	Cooper	57000	0.1	15450.5	2

WHERE

Como listar todos os vendedores que, graças à sua longa experiência com a empresa, têm uma taxa de comissão acima de 0,12?

```
SELECT *
        FROM salespeople
        WHERE commission_rate > 0.12;
```

Query | Query History

```
1  SELECT *
2      FROM salespeople
3      WHERE commission_rate > 0.12;
4
5
```

Data Output | Messages | Notifications

	id [PK] integer	first_name character varying (50)	last_name character varying (50)	salary integer	commission_rate real	commission_2021 real	branch_id real
1	11	Katarina	Rostova	45000	0.15	47345.6	1
2	12	Alina	Park	43000	0.15	45678.9	2
3	13	Meera	Malik	50000	0.15	39045.63	2
4	17	Samar	Navabi	52000	0.14	23023.45	2
5	18	Donald	Ressler	40000	0.14	41345.75	2
6	20	Elisabeth	Keen	59000	0.14	45350	2

WHERE >

Como enumerar os vendedores cujos ganhos de comissão em 2021 foram maiores do que seu salário anual?

```
SELECT *
        FROM salespeople
        WHERE commission_2021 > salary;
```

Query | Query History

```
1  SELECT *
2  FROM salespeople
3  WHERE commission_2021 > salary;
4
```

Data Output | Messages | Notifications

	id [PK] integer	first_name character varying (50)	last_name character varying (50)	salary integer	commission_rate real	commission_2021 real	branch_id real
1	11	Katarina	Rostova	45000	0.15	47345.6	1
2	12	Alina	Park	43000	0.15	45678.9	2
3	18	Donald	Ressler	40000	0.14	41345.75	2
4	21	Tom	Keen	41000	0.12	41560.75	1

WHERE BETWEEN AND

Como listar todos os vendedores cuja taxa de comissão está entre 0,10 e 0,14, pode-se usar a seguinte consulta?

```
SELECT *
FROM salespeople
WHERE commission_rate BETWEEN 0.10 AND 0.14;
```

Query | Query History

```
1  SELECT *
2      FROM salespeople
3      WHERE commission_rate BETWEEN 0.10 AND 0.14;
4
5
```

Data Output | Messages | Notifications

	id [PK] integer	first_name character varying (50)	last_name character varying (50)	salary integer	commission_rate real	commission_2021 real	branch_id real
1	21	Tom	Keen	41000	0.12	41560.75	1
2	22	Dembe	Zuma	40000	0.12	31540.7	5
3	23	Aram	Mojtabai	50000	0.12	29050.65	2
4	30	Kate	Kaplan	54000	0.1	25760.45	5
5	32	Marvin	Gerard	55000	0.1	22500	5
6	34	Raymond	Reddington	60000	0.1	17570.8	5
7	35	Harold	Cooper	57000	0.1	15450.5	2

Texto

Como recuperar informações sobre todos os vendedores cujo sobrenome (quando ordenado alfabeticamente) esteja antes de "Keen"?

```
SELECT *
FROM salespeople
WHERE last_name < 'Keen';
```

Query | Query History

```
1  SELECT *
2  FROM salespeople
3  WHERE last_name < 'Keen';
4
```

Data Output | Messages | Notifications

	id [PK] integer	first_name character varying (50)	last_name character varying (50)	salary integer	commission_rate real	commission_2021 real	branch_id real
1	30	Kate	Kaplan	54000	0.1	25760.45	5
2	32	Marvin	Gerard	55000	0.1	22500	5
3	35	Harold	Cooper	57000	0.1	15450.5	2
4	37	Ian	Garvey	43000	0.08	[null]	1

LIKE

Como listar todos os vendedores cujo sobrenome comece com K?

```
SELECT *
FROM salespeople
WHERE last_name LIKE 'K%';
```

Coringa

Query | Query History

```
1  SELECT *
2  FROM salespeople
3  WHERE last_name LIKE 'K%';
4
```

Data Output | Messages | Notifications

	id [PK] integer	first_name character varying (50)	last_name character varying (50)	salary integer	commission_rate real	commission_2021 real	branch_id real
1	20	Elisabeth	Keen	59000	0.14	45350	2
2	21	Tom	Keen	41000	0.12	41560.75	1
3	30	Kate	Kaplan	54000	0.1	25760.45	5

Coringa SQL

Os wildcards SQL só funcionam no operador LIKE.

Se você colocar um curinga dentro de uma string comum que não seja um argumento para o operador LIKE, você verá que SQL tratará esse curinga como um caracter literal que aparece na string.

Curingas SQL

%	• Significa qualquer quantidade de caracteres.
_	• Significa um caractere.

Curingas SQL - %

(%)	• é usado para representar um ou mais caracteres.
Para buscar todas as palavras que começam com a letra E.	• select * from NOME_TABELA where nome like 'E%'
Buscar todas as palavras que terminam com a letra ‘a’;	• select * from NOME_TABELA where nomes like '%a';
Buscar todas as palavras que contenham a sílaba ‘li’.	• select * from NOME_TABELA where nomes like '%li%';
Buscar todas palavras que contenham a letra ‘a’ alguma coisa e depois a letra ‘e’ mais alguma coisa escrita.	• select * from NOME_TABELA where nomes like '%a%e%';

Curingas SQL - _

O coringa _ (underline)	• substitui um caractere, ou seja, uma letra.
Para buscar por exemplo o nome Elias, sem considerar a primeira letra do nome.	• select * from NOME_TABELA where nomes like '_lias';
Podemos substituir quantos forem necessários, só colocar mais “underlines”, exemplo:	• select * from NOME_TABELA where nomes like '__ias';
Podemos combinar com o coringa %.	• select * from NOME_TABELA where nomes like '_li%';

LIKE e ILIKE

LIKE

- é um comando que serve para trazer partes de palavras ou números. O like é **case-sensitive,** então ‘maria’ é diferente de ‘MARIA’.

ILIKE

- possui as mesmas características do like, porém não é **case-sensitive,** ou seja, não faz distinção entre maiúsculas e minusculas, então ‘MARIA’ é igual ‘maria’ para o ilike.

NULL

Como recuperar todos os registros que têm NULL na coluna ?

```
SELECT *
        FROM salespeople
        WHERE commission_2021 IS NULL;
```

Query | Query History

```
1  SELECT *
2  FROM salespeople
3  WHERE commission_2021 IS NULL;
4
```

Data Output | Messages | Notifications

	id [PK] integer	first_name character varying (50)	last_name character varying (50)	salary integer	commission_rate real	commission_2021 real	branch_id real
1	37	Ian	Garvey	43000	0.08	[null]	1
2	38	Ivan	Stepanov	41000	0.08	[null]	1

IN

Como recuperar uma lista de funcionários cujos ganhos precisam ser verificados.?

```
SELECT *
FROM salespeople
WHERE last_name IN ('Kaplan', 'Gerard', 'Zuma');
```

Query | Query History

```
1  SELECT *
2  FROM salespeople
3  WHERE last_name IN ('Kaplan', 'Gerard', 'Zuma');
4
```

Data Output | Messages | Notifications

	id [PK] integer	first_name character varying (50)	last_name character varying (50)	salary integer	commission_rate real	commission_2021 real	branch_id real
1	22	Dembe	Zuma	40000	0.12	31540.7	5
2	30	Kate	Kaplan	54000	0.1	25760.45	5
3	32	Marvin	Gerard	55000	0.1	22500	5

Condições de filtragem em WHERE

O operador AND	• exibe um registro se todas as condições forem verdadeiras.
O operador do OR	• exibe um registro se alguma das condições for verdadeira.
O operador do NOT	• exibe um registro se a condição correspondente não for verdadeira.

AND

Como listar todos os vendedores que trabalham no ramo #5 **e** têm salários iguais ou superiores a $50K?

```
SELECT *
        FROM salespeople
        WHERE branch_id = 5 AND salary >= 50000;
```

Query | Query History

```
1  SELECT *
2  FROM salespeople
3  WHERE branch_id = 5 AND salary >= 50000;
4
```

Data Output | Messages | Notifications

	id [PK] integer	first_name character varying (50)	last_name character varying (50)	salary integer	commission_rate real	commission_2021 real	branch_id real
1	30	Kate	Kaplan	54000	0.1	25760.45	5
2	32	Marvin	Gerard	55000	0.1	22500	5
3	34	Raymond	Reddington	60000	0.1	17570.8	5

AND

Como recuperar todos os registros
onde o sobrenome é 'Kaplan' **ou** 'Reddington'?

```
SELECT *
        FROM salespeople
        WHERE branch_id = 5 OR last_name = ‘Reddington’;
```

Query | Query History

```
1  SELECT *
2  FROM salespeople
3  WHERE last_name = 'Kaplan' OR last_name = 'Reddington';
4
```

Data Output | Messages | Notifications

	id [PK] integer	first_name character varying (50)	last_name character varying (50)	salary integer	commission_rate real	commission_2021 real	branch_id real
1	30	Kate	Kaplan	54000	0.1	25760.45	5
2	34	Raymond	Reddington	60000	0.1	17570.8	5

NOT

Como obter informações sobre todos os vendedores , **exceto** os que trabalham no ramo nº 2?

```
SELECT *
        FROM salespeople
        WHERE NOT branch_id = 2;
```

Query | Query History

```
1  SELECT *
2  FROM salespeople
3  WHERE NOT branch_id = 2;
4
```

Data Output | Messages | Notifications

	id [PK] integer	first_name character varying (50)	last_name character varying (50)	salary integer	commission_rate real	commission_2021 real	branch_id real
1	11	Katarina	Rostova	45000	0.15	47345.6	1
2	21	Tom	Keen	41000	0.12	41560.75	1
3	22	Dembe	Zuma	40000	0.12	31540.7	5
4	30	Kate	Kaplan	54000	0.1	25760.45	5
5	32	Marvin	Gerard	55000	0.1	22500	5
6	34	Raymond	Reddington	60000	0.1	17570.8	5
7	37	Ian	Garvey	43000	0.08	[null]	1
8	38	Ivan	Stepanov	41000	0.08	[null]	1

Consula mais elaborada

Como recuperar todos os registros onde o sobrenome de um funcionário é 'Keen' ou 'Park', seus ganhos em comissões foram superiores ao seu salário em 2021, e eles não estão trabalhando no ramo #2?

```
SELECT *
        FROM salespeople
        WHERE (last_name = 'Keen' OR last_name = 'Park')
        AND (commission_2021 > salary)
        AND (NOT branch_id = 2);
```

Query | Query History

```
1  SELECT *
2  FROM salespeople
3  WHERE (last_name = 'Keen' OR last_name = 'Park')
4  AND (commission_2021 > salary)
5  AND (NOT branch_id = 2);
6
```

Data Output | Messages | Notifications

	id [PK] integer	first_name character varying (50)	last_name character varying (50)	salary integer	commission_rate real	commission_2021 real	branch_id real
1	21	Tom	Keen	41000	0.12	41560.75	1

DISTINCT

Como obter todos os salários, não repetidos que existem?

```
SELECT
        DISTINCT salary
        FROM salespeople
```

```
1  SELECT
2      DISTINCT salary
3      FROM salespeople
4
5
```

Data Output | Messages | Notifi

	salary integer
1	52000
2	50000
3	43000
4	59000
5	40000
6	54000
7	57000
8	60000
9	41000
10	55000
11	45000

DISTINCT trata-se de uma cláusula para eliminar repetições em consultas, considerando as colunas informadas na listagem de colunas do comando SELECT que contenham valores iguais como o mesmo registro.

DISTINCT e ORDER BY

Como obter todos os salários, não repetidos que existem em ordem crescente?

```
SELECT
        DISTINCT salary
        FROM salespeople ORDER BY salary
```

Query | Query History

```
1  SELECT
2      DISTINCT salary
3      FROM salespeople ORDER BY salary
4
5
```

Data Output | Messages | Notifications

	salary integer
1	40000
2	41000
3	43000
4	45000
5	50000
6	52000
7	54000
8	55000
9	57000
10	59000
11	60000

DISTINCT e ORDER BY

Como obter todos os salários, não repetidos que existem em ordem decrescente?

```
SELECT
        DISTINCT salary
        FROM salespeople ORDER BY salary DESC
```

Query | Query History

```
1  SELECT
2      DISTINCT salary
3      FROM salespeople ORDER BY salary DESC
4
5
```

Data Output | Messages | Notifications

	salary integer
1	60000
2	59000
3	57000
4	55000
5	54000
6	52000
7	50000
8	45000
9	43000
10	41000
11	40000

DISTINCT e ORDER BY

Preciso de uma informação que chamarei de Salário Alto para todos valores de salário maiores que 50.000?

ALIAS

Um alias de coluna permite que você atribua um nome SELECT temporário a uma coluna ou expressão na lista de seleção de uma instrução.

O alias da coluna existe temporariamente durante a execução da consulta.

```
SELECT
        DISTINCT salary AS Salario_alto
        FROM salespeople
        WHERE salary >= 50000
```

Query | Query History

```
1  SELECT
2      DISTINCT salary AS Salario_alto
3      FROM salespeople
4      WHERE salary >= 50000
5
6
```

Data Output | Messages | Notifications

	salario_alto integer
1	60000
2	52000
3	50000
4	59000
5	55000
6	54000
7	57000

Subquery

SELECT SELECT

DISTINCT e ORDER BY

Qual é o primeiro nome e sobrenome da pessoa que ganha o maior salário da empresa?

```
SELECT
        salespeople.first_name AS PRIMEIRO_NOME, salespeople.last_name AS ULTIMO_NOME, T.MAIOR_SALARIO
        FROM salespeople, (SELECT MAX(salary) AS MAIOR_SALARIO FROM salespeople) T
        WHERE salespeople.salary = T.MAIOR_SALARIO;
```

Query | Query History

```
1  SELECT
2      salespeople.first_name AS PRIMEIRO_NOME, salespeople.last_name AS ULTIMO_NOME, T.MAIOR_SALARIO
3      FROM salespeople, (SELECT MAX(salary) AS MAIOR_SALARIO FROM salespeople) T
4      WHERE salespeople.salary = T.MAIOR_SALARIO
5
6
```

Data Output | Messages | Notifications

	primeiro_nome character varying (50)	ultimo_nome character varying (50)	maior_salario integer
1	Raymond	Reddington	60000

SUBSELECT

Uma subconsulta (mais conhecida como SUBQUERY ou SUBSELECT) é uma instrução do tipo SELECT dentro de outra instrução SQL.

Desta forma, se torna possível efetuar consultas que de outra forma seriam extremamente complicadas ou impossíveis de serem feitas de outra forma.

Sub querys

Suponha que seja necessário listar da tabela produtos, todos os registros que tenham um preço acima da média dos outros produtos.

```
SELECT
   nome,
   preco
FROM
   produto
WHERE
   preco > ( SELECT  AVG(preco) FROM produto )
```

Query | Query History

```
1  SELECT
2      nome,
3      preco
4  FROM
5      produto
6  WHERE
7      preco > ( SELECT
8                    AVG(preco)
9                    FROM produto
10 )
11
```

Data Output | Messages | Notifications

	nome character varying (50)	preco real
1	Notebook	3500

Sub querys

Temos uma tabela com salários (salespeople) e queremos saber qual os funcionários que recebem salário acima da média salarial da empresa..

```
SELECT *
FROM salespeople
WHERE salary > (SELECT AVG(salary) FROM
salespeople);
```

SELECT AVG(salary) FROM salespeople;

	avg numeric
1	48666.666666666667

Query | Query History

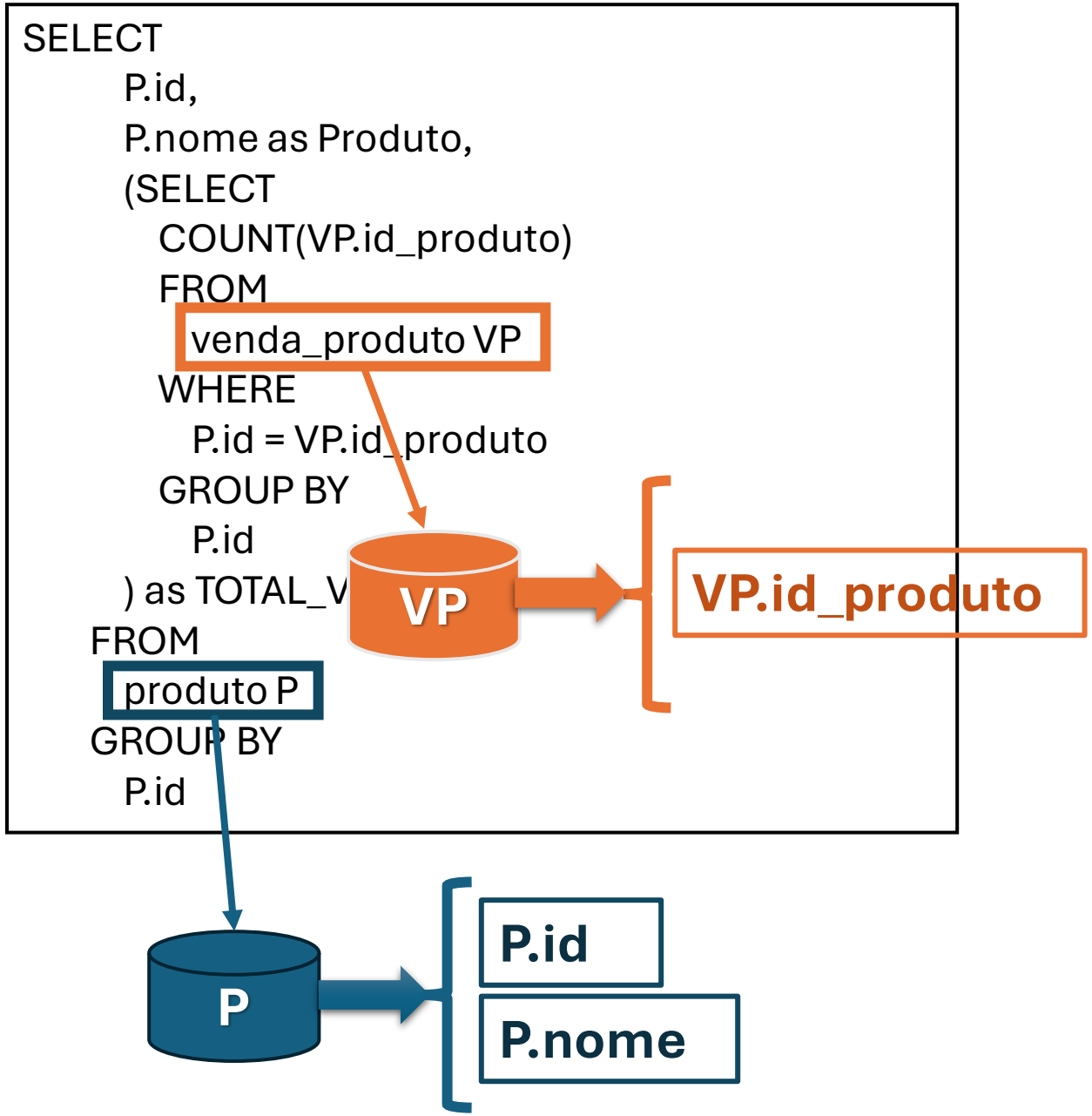
```
1 SELECT *
2 FROM salespeople
3 WHERE salary > (SELECT AVG(salary) FROM salespeople);
```

Data Output | Messages | Notifications

	id [PK] integer	first_name character varying (50)	last_name character varying (50)	salary integer	commission_rate real	commission_2021 real	branch_id real
1	13	Meera	Malik	50000	0.15	39045.63	2
2	17	Samar	Navabi	52000	0.14	23023.45	2
3	20	Elisabeth	Keen	59000	0.14	45350	2
4	23	Aram	Mojtabai	50000	0.12	29050.65	2
5	30	Kate	Kaplan	54000	0.1	25760.45	5
6	32	Marvin	Gerard	55000	0.1	22500	5
7	34	Raymond	Reddington	60000	0.1	17570.8	5
8	35	Harold	Cooper	57000	0.1	15450.5	2

Sub querys

Queremos saber qual a quantidade e quais os produtos que foram vendidos



Query | Query History

```sql
SELECT
            P.id,
            P.nome as Produto,
            (SELECT
                COUNT(VP.id_produto)
                FROM
                    venda_produto VP
                WHERE
                    P.id = VP.id_produto
                GROUP BY
                    P.id
            ) as TOTAL_VENDIDO
        FROM
            produto P
        GROUP BY
            P.id
```

Data Output | Messages | Notifications

	id [PK] integer	produto character varying (50)	total_vendido bigint
1	1	Bola	3
2	2	Patinete	2
3	3	Carrinho,	1
4	4	Skate,	[null]
5	5	Notebook	2
6	6	Monitor LG 19	1
7	7	O Diário de Anne Frank	1
8	8	O dia do Curinga	2
9	9	O mundo de Sofia	1
10	10	Através do Espelho	[null]