PDP ACADEMY

Viewlar bilan ishlash

Reja:

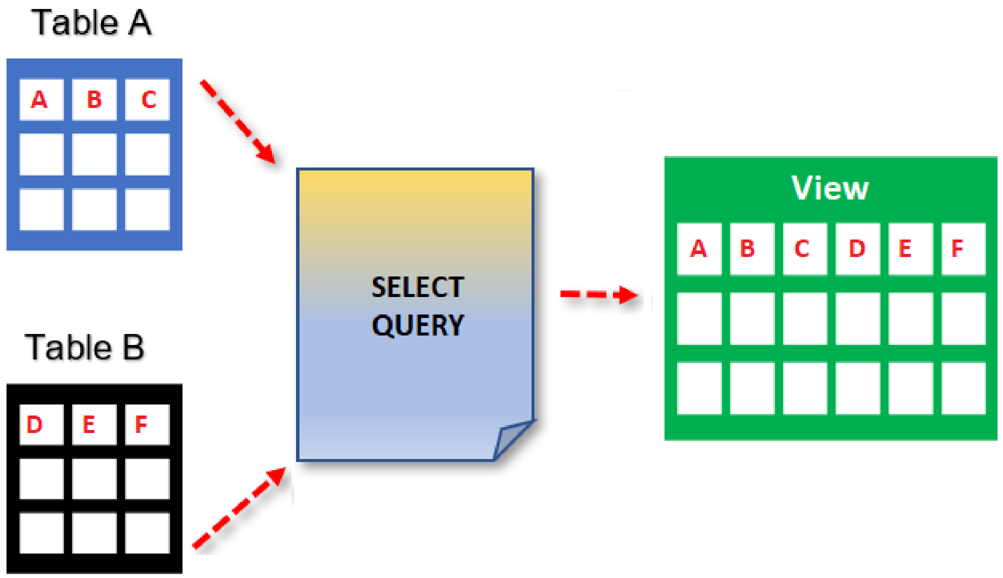
1. Views(Simple, Complex, Inline, Materialized)
2. Managing PostgreSQL Views
   (CREATE, DROP, UPDATABLE)

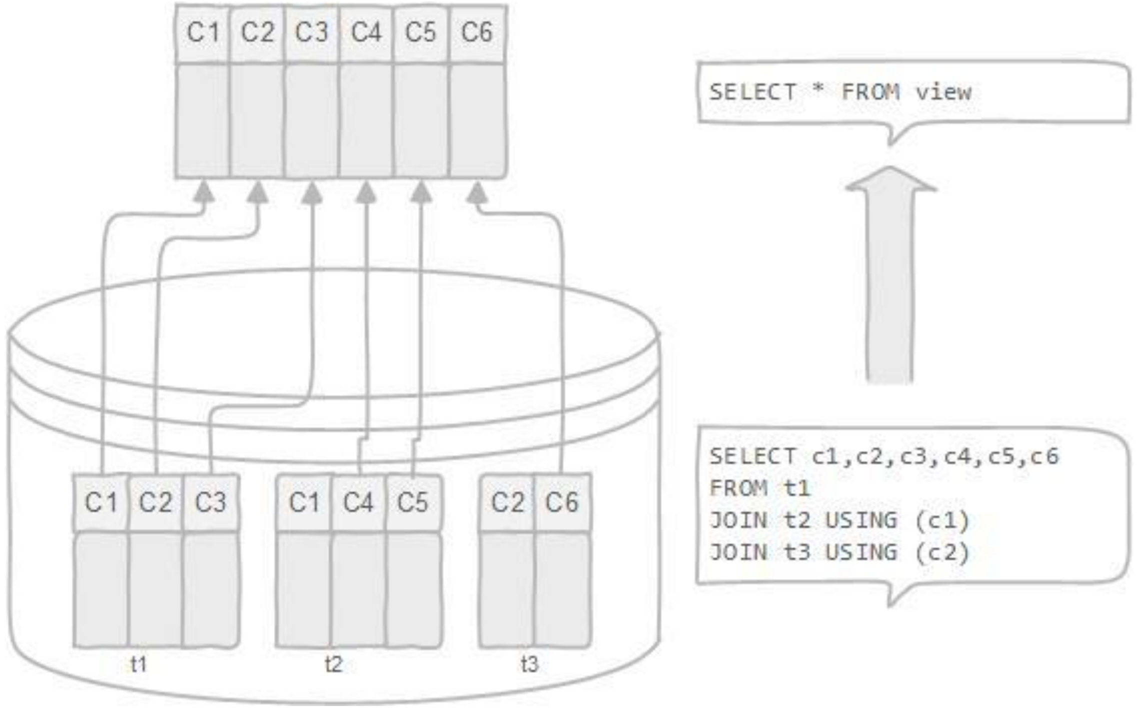
View

View- bu ma`lumotlar ombori jadvalidagi ma`lumotlarni boshqacha tarzda ko`rsatuvchi so`rovdir. View bir yoki bir nechta jadvallarga asoslangan holda yaratiladi. Viewlar odatda murakkab so`rovlarni yaxlit bitta obe`kt sifatida o`rash uchun qo`llaniladi.

- PostgreSQLda **view** psevdo-jadvaldir, ya`ni u haqiqiy jadval emas.
- **View** bir yoki bir nechta jadvallardan yaratilishi mumkin.
- **View** yaratilgan jadvallar asosiy jadvallar deb nomlanadi.
- **View** bir nechta asosiy jadvallarning ayrim ustunlarini o`zida birlashtirgan jadvaldir.

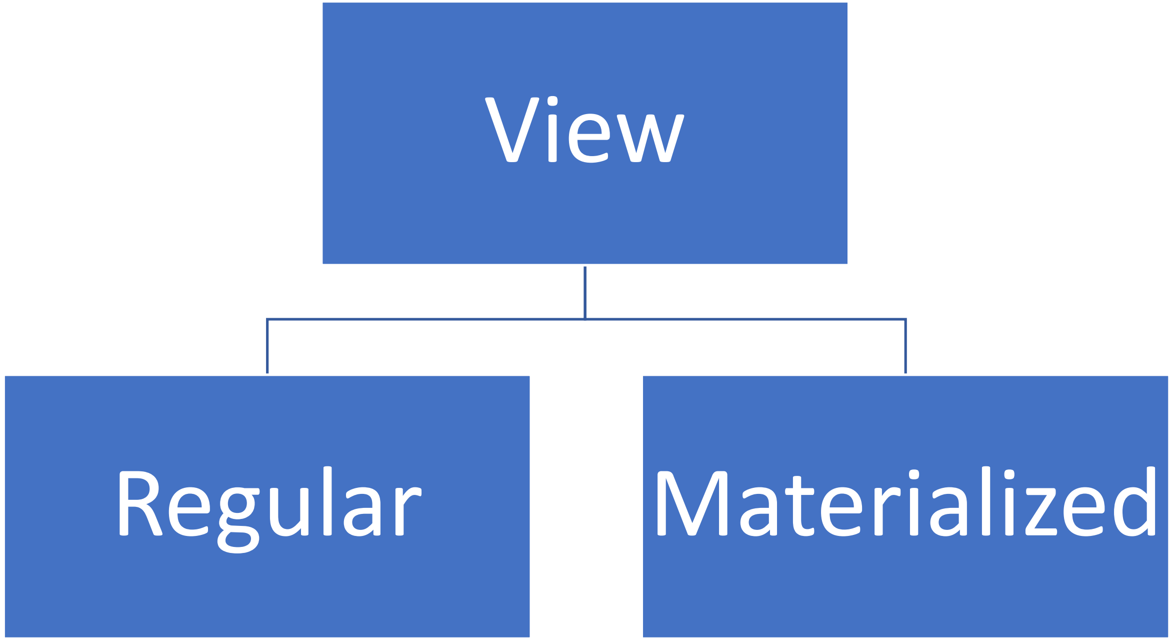
PDP ACADEMY





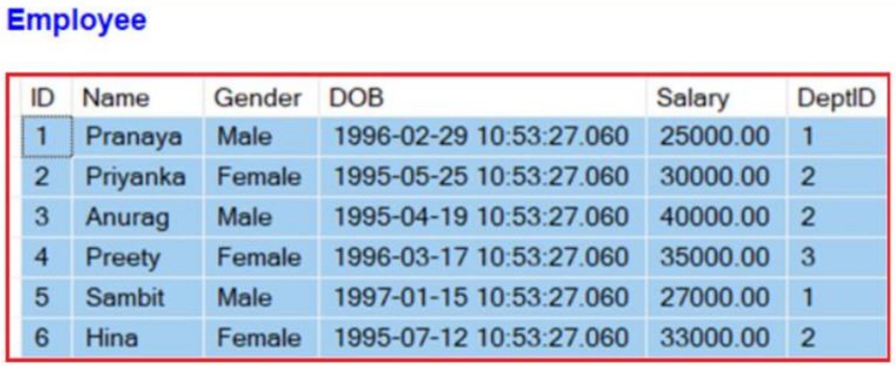
```
SELECT * FROM view
```

```
SELECT c1,c2,c3,c4,c5,c6
FROM t1
JOIN t2 USING (c1)
JOIN t3 USING (c2)
```


View
Regular
Materialized

PostgreSqlda Simple View

View bir bitta jadval asosida yaratilganda, u PostgreSQLda **simple** ***(sodda)*** **view** hisoblanadi. PostgreSQLdagi simple viewlarni tushunish uchun biz quyidagi **Employee** jadvalidan foydalanamiz.

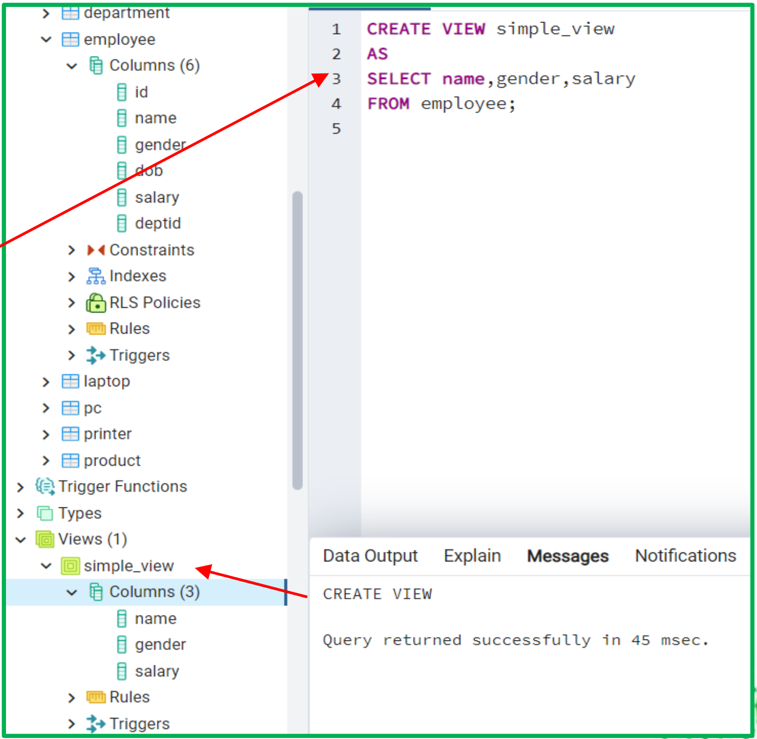


Employee

ID	Name	Gender	DOB	Salary	DeptID
1	Pranaya	Male	1996-02-29 10:53:27.060	25000.00	1
2	Priyanka	Female	1995-05-25 10:53:27.060	30000.00	2
3	Anurag	Male	1995-04-19 10:53:27.060	40000.00	2
4	Preety	Female	1996-03-17 10:53:27.060	35000.00	3
5	Sambit	Male	1997-01-15 10:53:27.060	27000.00	1
6	Hina	Female	1995-07-12 10:53:27.060	33000.00	2

PostgreSqlda Simple View

```
CREATE VIEW simple_view
AS
SELECT name,gender,salary
FROM employee;
```



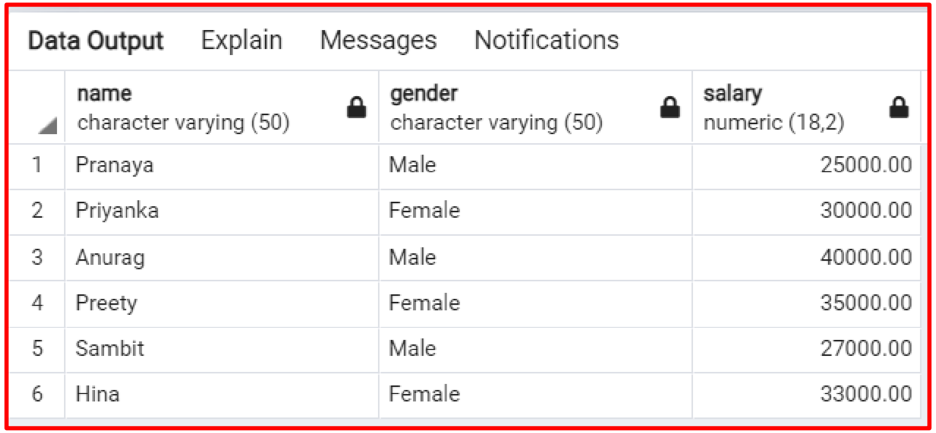
```
CREATE VIEW simple_view
AS
SELECT name,gender,salary
FROM employee;
```

```
CREATE VIEW

Query returned successfully in 45 msec.
```

PDP ACADEMY

PostgreSqlda Simple View

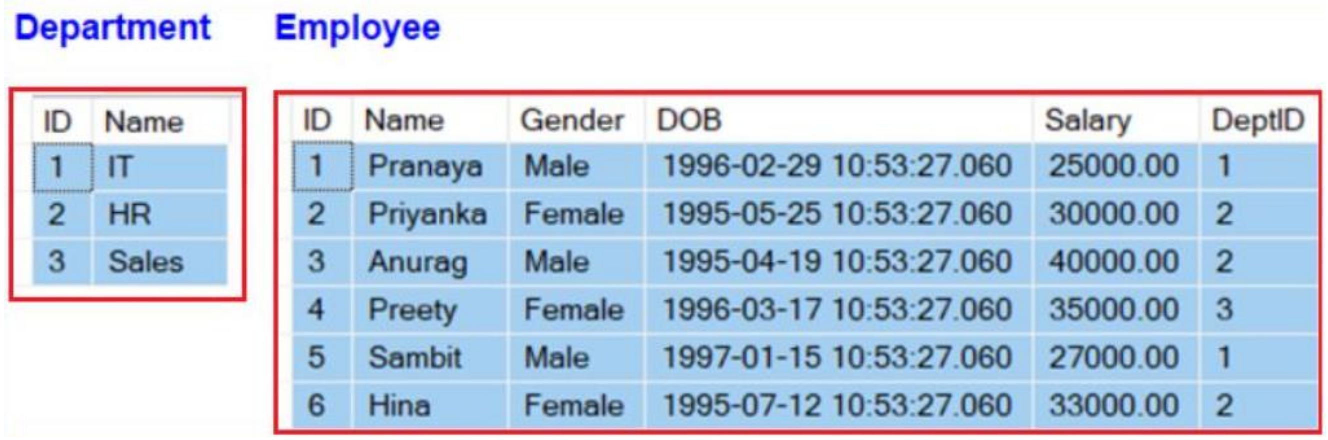
SELECT * FROM simple_view;



	name character varying (50)	gender character varying (50)	salary numeric (18,2)
1	Pranaya	Male	25000.00
2	Priyanka	Female	30000.00
3	Anurag	Male	40000.00
4	Preety	Female	35000.00
5	Sambit	Male	27000.00
6	Hina	Female	33000.00

PostgreSqlda
Complex *(murakkab)* View

View bir nechta jadvallar asosida yaratilganda, u PostgreSQLda **complex** ***(murakkab)*** **view** sifatida yaratiladi. PostgreSQLdagi complex viewlarni tushunish uchun biz quyidagi **Department** va **Employee** jadvallaridan foydalanamiz.



Department

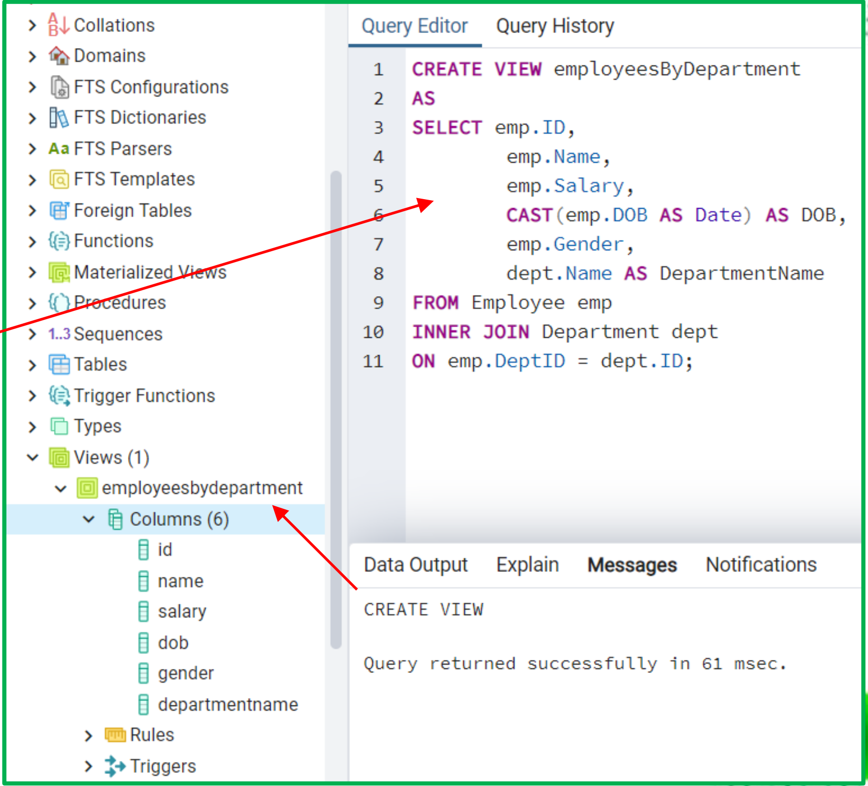
ID	Name
1	IT
2	HR
3	Sales

Employee

ID	Name	Gender	DOB	Salary	DeptID
1	Pranaya	Male	1996-02-29 10:53:27.060	25000.00	1
2	Priyanka	Female	1995-05-25 10:53:27.060	30000.00	2
3	Anurag	Male	1995-04-19 10:53:27.060	40000.00	2
4	Preety	Female	1996-03-17 10:53:27.060	35000.00	3
5	Sambit	Male	1997-01-15 10:53:27.060	27000.00	1
6	Hina	Female	1995-07-12 10:53:27.060	33000.00	2

Yuqoridagi ikkita jadval asosida complex view yaratamiz.

```
CREATE VIEW employeesByDepartment
AS
SELECT emp.ID,
       emp.Name,
       emp.Salary,
       CAST(emp.DOB AS Date) AS DOB,
       emp.Gender,
       dept.Name AS DepartmentName
FROM Employee emp
INNER JOIN Department dept
ON emp.DeptID = dept.ID;
```



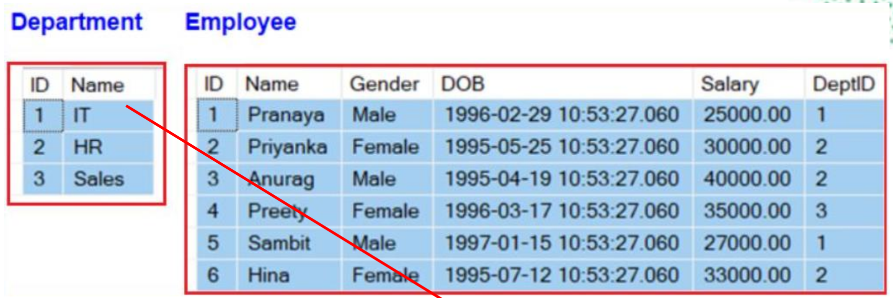
```
CREATE VIEW employeesByDepartment
AS
SELECT emp.ID,
        emp.Name,
        emp.Salary,
        CAST(emp.DOB AS Date) AS DOB,
        emp.Gender,
        dept.Name AS DepartmentName
FROM Employee emp
INNER JOIN Department dept
ON emp.DeptID = dept.ID;
```

```
CREATE VIEW

Query returned successfully in 61 msec.
```

PDP ACADEMY

Complex *(murakkab)* View

```
SELECT * FROM
employeesByDepartment;
```

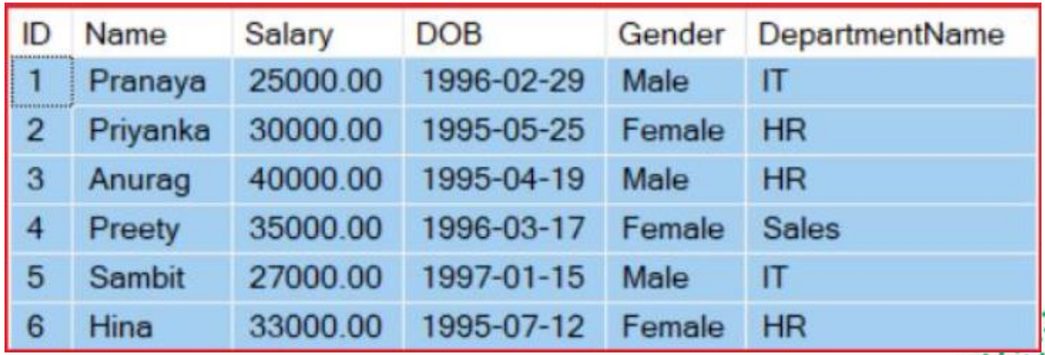


Department

ID	Name
1	IT
2	HR
3	Sales

Employee

ID	Name	Gender	DOB	Salary	DeptID
1	Pranaya	Male	1996-02-29 10:53:27.060	25000.00	1
2	Priyanka	Female	1995-05-25 10:53:27.060	30000.00	2
3	Anurag	Male	1995-04-19 10:53:27.060	40000.00	2
4	Preety	Female	1996-03-17 10:53:27.060	35000.00	3
5	Sambit	Male	1997-01-15 10:53:27.060	27000.00	1
6	Hina	Female	1995-07-12 10:53:27.060	33000.00	2

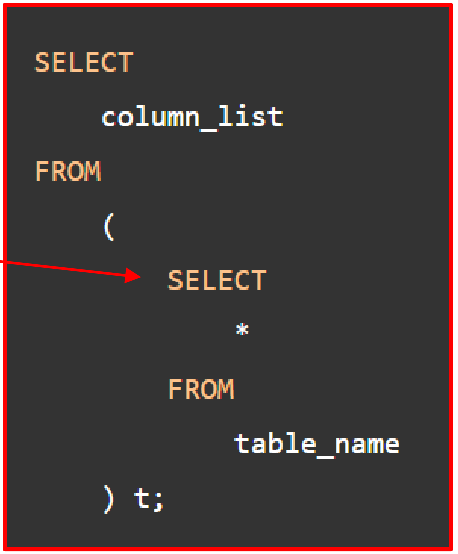


ID	Name	Salary	DOB	Gender	DepartmentName
1	Pranaya	25000.00	1996-02-29	Male	IT
2	Priyanka	30000.00	1995-05-25	Female	HR
3	Anurag	40000.00	1995-04-19	Male	HR
4	Preety	35000.00	1996-03-17	Female	Sales
5	Sambit	27000.00	1997-01-15	Male	IT
6	Hina	33000.00	1995-07-12	Female	HR

Inline View

Viewlarni alohida yaratishdan tashqari, SQL so`rovlarning ichida ham view yaratish mumkin. Bu **inline view** deyiladi.Quyidagi SQL so`rovda FROM dan keyin ko`rsatilgan *(ichki so`rov)* inline view deb ataladi. Inline view so`rovdagi jadval o`rnini bosishi mumkinligi sababli, u hosila jadval deb ham ataladi. Ba`zan siz inline view bilan bir xil ma`noga ega bo`lgan subselect atamasini uchratishingiz mumkin.

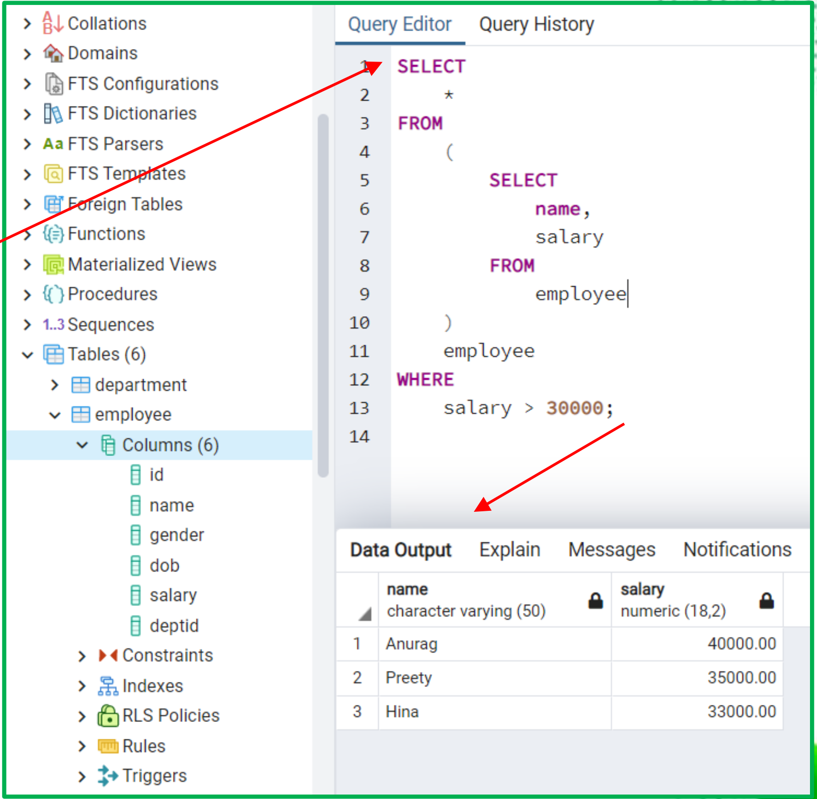
Inline View



```
SELECT
    column_list
FROM
    (
        SELECT
            *
        FROM
            table_name
    ) t;
```

Inline View

```
SELECT
    *
FROM
    (
        SELECT
            name,
            salary
         FROM
            employee
    )
                  employee
WHERE
    salary > 30000;
```

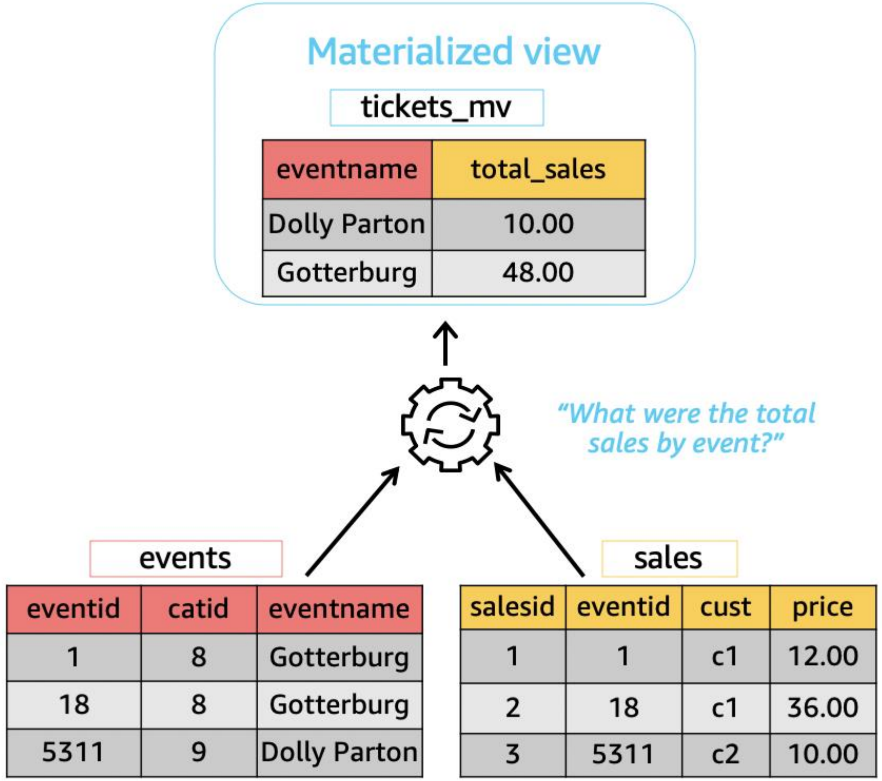


```
SELECT
    *
FROM
    (
        SELECT
            name,
            salary
        FROM
            employee
    )
    employee
WHERE
    salary > 30000;
```

	name character varying (50)	salary numeric (18,2)
1	Anurag	40000.00
2	Preety	35000.00
3	Hina	33000.00

Materialized View

" **Materialized View** " - bu ma`lumotlar bazasi ob`ekti bo`lib, u oldindan hisoblangan ma`lumotlar bazasi so`rovi natijasini saqlaydi va kerak bo`lganda ushbu natijani yangilashni osonlashtiradi. **Materialized Viewlar** deyarli barcha ilg`or ma`lumotlar bazasi tizimlarining ajralmas xususiyati hisoblanadi. Tabiiyki, PostgreSQL ham **Materialized Viewlarni** qo`llab-quvvatlaydi va foydalanuvchiga ko`proq vaqt talab qiladigan so`rovlarni bajarish uchun kuchli vositani taklif qiladi. PostgreSQLda **Materialized View** so`rov natijalarini saqlash va ma`lumotlarni vaqti-vaqti bilan yangilash imkonini beradi. **Materialized View** ma`lumotlarni tez olishni talab qiladigan ko`p hollarda foydalidir.



tickets_mv

eventname	total_sales
Dolly Parton	10.00
Gotterburg	48.00

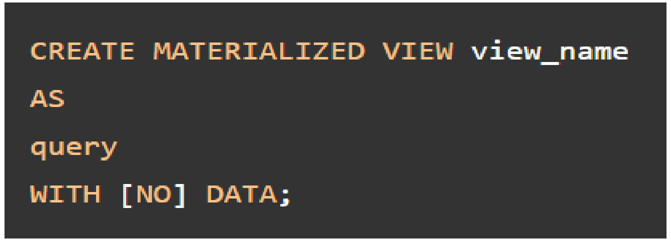
events

eventid	catid	eventname
1	8	Gotterburg
18	8	Gotterburg
5311	9	Dolly Parton

sales

salesid	eventid	cust	price
1	1	c1	12.00
2	18	c1	36.00
3	5311	c2	10.00

Quyida **Materialized Viewni** yaratish sintaksisi keltirilgan.



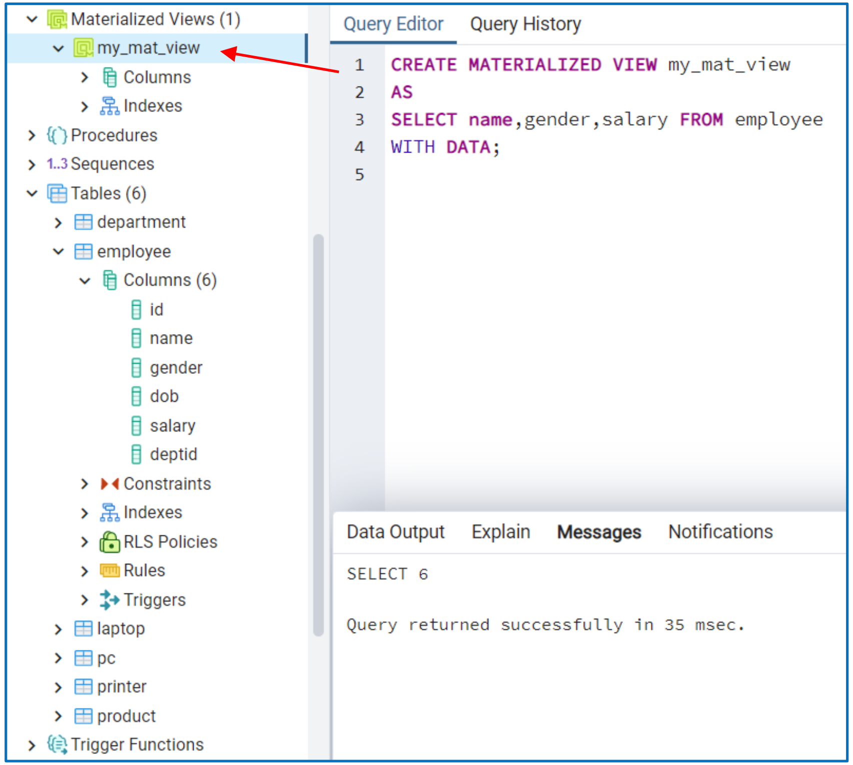
```
CREATE MATERIALIZED VIEW view_name
AS
query
WITH [NO] DATA;
```

•**view_name:** ko`rinish nomini belgilaydi, CREATE MATERIALIZED VIEW bandidan so`ng yoziladi.
•**query:** Bu AS kalit so`zidan keyin ishlatiladi. Bu jadvallardan ma`lumotlarni oladigan so`rovni belgilaydi.
•**With [NO] DATA :** [NO] kalit so`zi ixtiyoriy. Agar u aniqlanmagan bo`lsa, view yaratishda ma`lumotlar viewga yuklangan holda yaratiladi. WITH NO DATA aniqlansa, viewga ma`lumotlar yuklanmagan holda yaratiladi va view o`qilmaydi.

Quyida **Materialized Viewni** yangilash sintaksisi keltirilgan.

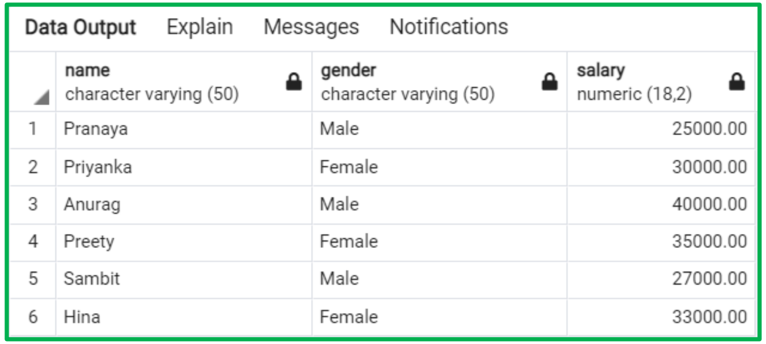
```
REFRESH MATERIALIZED VIEW view_name;
```

Create Materialized View with data



```sql
CREATE MATERIALIZED VIEW my_mat_view
AS
SELECT name,gender,salary FROM employee
WITH DATA;
```

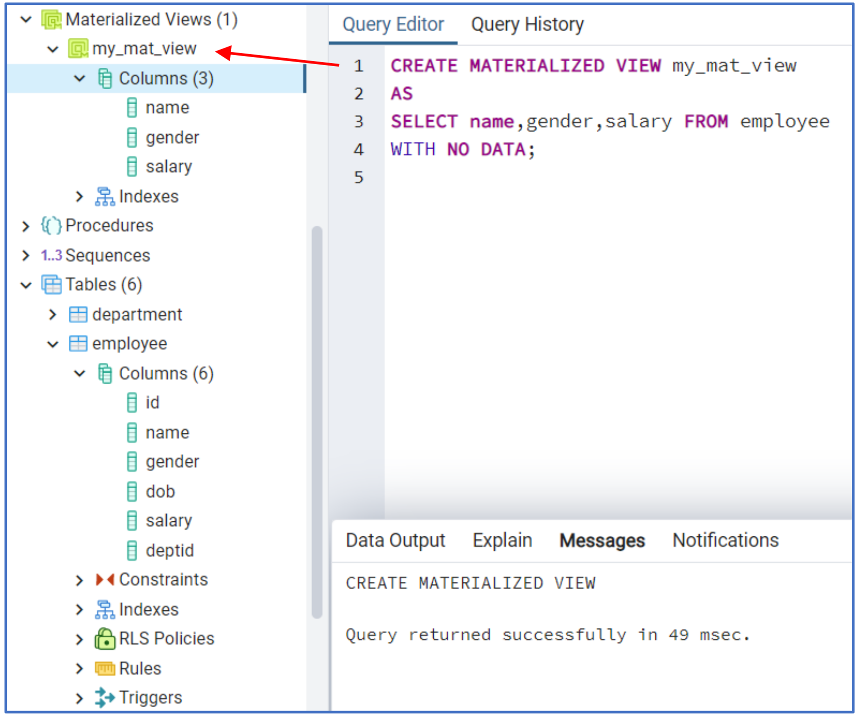
```
SELECT 6

Query returned successfully in 35 msec.
```



	name character varying (50)	gender character varying (50)	salary numeric (18,2)
1	Pranaya	Male	25000.00
2	Priyanka	Female	30000.00
3	Anurag	Male	40000.00
4	Preety	Female	35000.00
5	Sambit	Male	27000.00
6	Hina	Female	33000.00

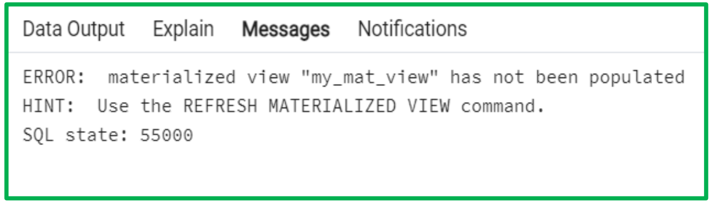
PDP ACADEMY

Create Materialized View with no data



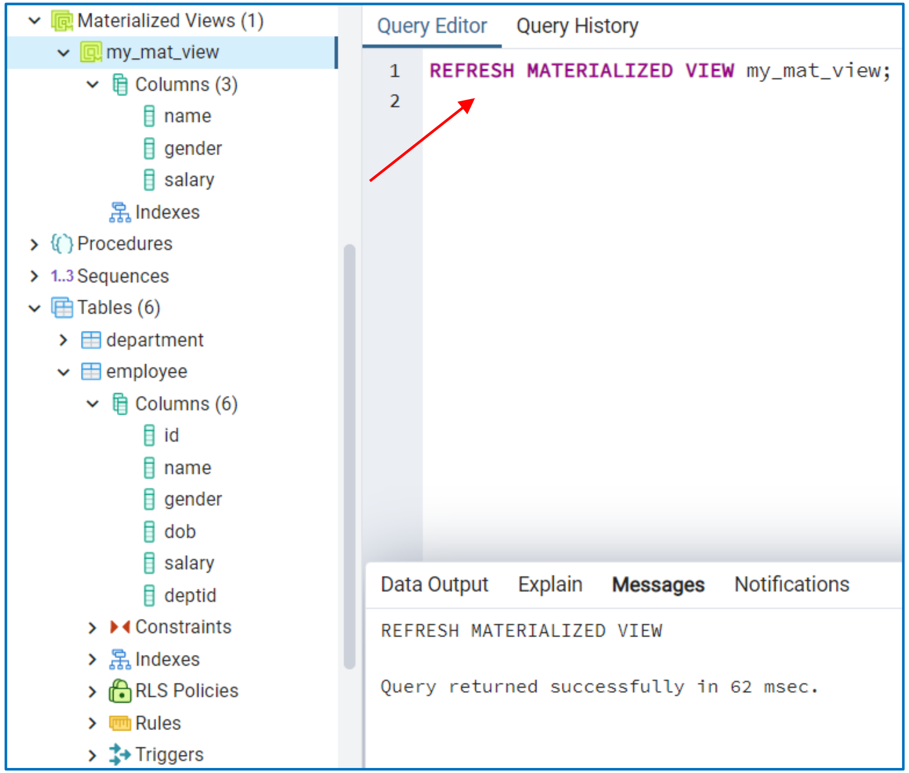
```sql
CREATE MATERIALIZED VIEW my_mat_view
AS
SELECT name,gender,salary FROM employee
WITH NO DATA;
```

```
CREATE MATERIALIZED VIEW

Query returned successfully in 49 msec.
```



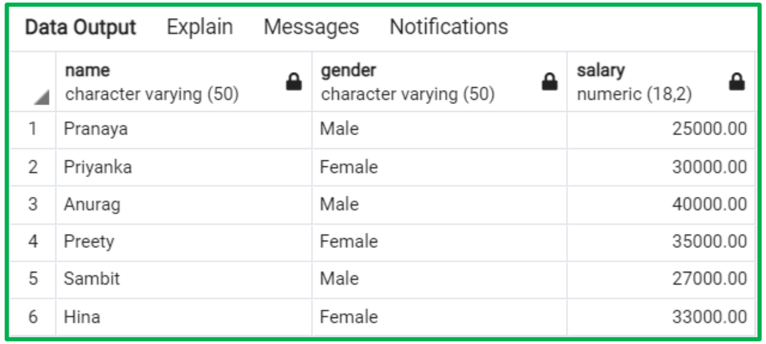
```
ERROR:  materialized view "my_mat_view" has not been populated
HINT:  Use the REFRESH MATERIALIZED VIEW command.
SQL state: 55000
```

Refresh Materialized View



```
REFRESH MATERIALIZED VIEW my_mat_view;
```

```
REFRESH MATERIALIZED VIEW

Query returned successfully in 62 msec.
```



	name character varying (50)	gender character varying (50)	salary numeric (18,2)
1	Pranaya	Male	25000.00
2	Priyanka	Female	30000.00
3	Anurag	Male	40000.00
4	Preety	Female	35000.00
5	Sambit	Male	27000.00
6	Hina	Female	33000.00

Managing PostgreSQL Views

Managing PostgreSQL Views ya`ni PostgreSQLda viewlarni boshqarish. Quyida view qanday yaratilishi, view qanday o`zgartirilishi va view qanday o`chirilishini ko`rib chiqamiz.

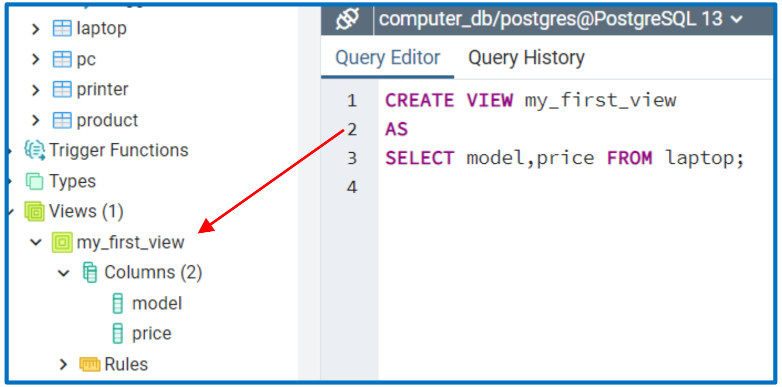
CREATE,
UPDATABLE,
DROP

PDP ACADEMY

Creating PostgreSQL Views

```
CREATE VIEW view_name AS query;
```

Yangi view **CREATE VIEW** buyrug`i orqali yaratiladi. So`ngra **AS** kalit so`zidan keyin query (so`rov) yoziladi. **Masalan:**



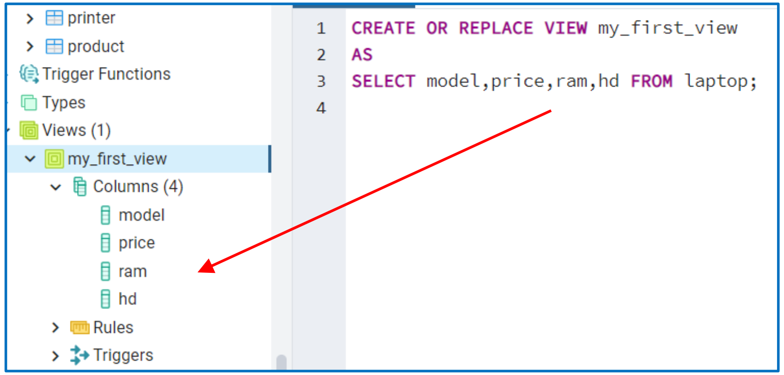
```
CREATE VIEW my_first_view
AS
SELECT model,price FROM laptop;
```

PDP ACADEMY

Updating PostgreSQL Views

```
CREATE OR REPLACE view_name
AS
query
```

Viewni o`zgartirish uchun **CREATE OR REPLACE VIEW** buyrug`i yoziladi. So`ngra **AS** kalit so`zidan keyin query (so`rov) yoziladi. Bunda view mavjud bo`lsa o`zgartiradi, agar yo`q bo`lsa yangi view yaratadi. **Masalan:**

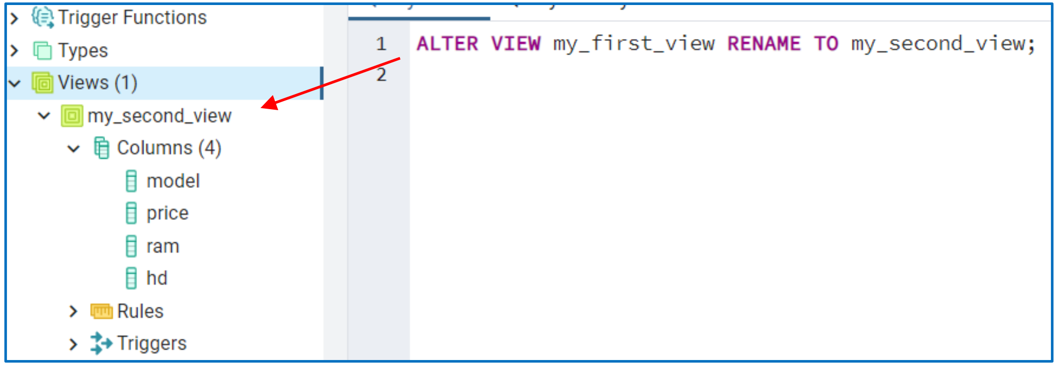


```
CREATE OR REPLACE VIEW my_first_view
AS
SELECT model,price,ram,hd FROM laptop;
```

Updating PostgreSQL Views

```
ALTER VIEW customer_master RENAME TO customer_info;
```

Viewni nomini o`zgartirish uchun **ALTER VIEW** dan keyin eski view nomi va **RENAME TO** dan keyin yangi view nomikiritiladi.
Masalan:



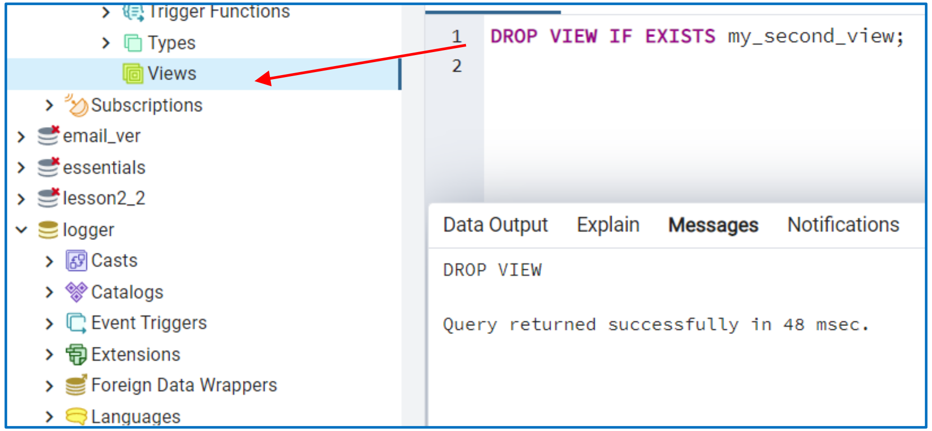
```
ALTER VIEW my_first_view RENAME TO my_second_view;
```

PDP ACADEMY

Removing PostgreSQL Views

```
DROP VIEW [ IF EXISTS ] view_name;
```

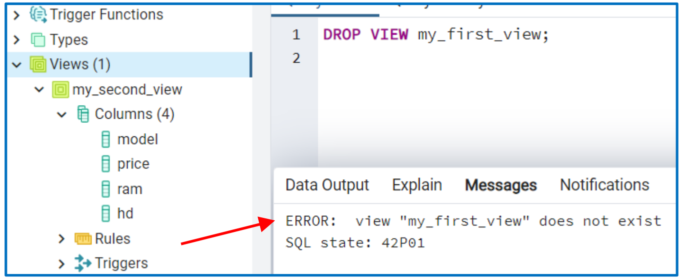
Viewni o`chirish uchun **DROP VIEW** buyrug`idan foydalaniladi, lekin o`chirmoqchi bo`lgan view mavjud bo`lmasa PostgreSQL xatolik beradi. Buning oldini olish uchun **IF EXISTS** dan foydalaniladi. **Masalan:**



```
DROP VIEW IF EXISTS my_second_view;
```

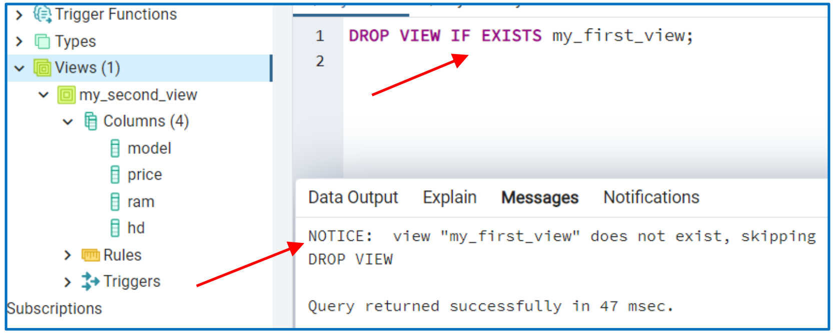
```
DROP VIEW

Query returned successfully in 48 msec.
```

Removing PostgreSQL Views



```
DROP VIEW my_first_view;
```

```
ERROR:  view "my_first_view" does not exist
SQL state: 42P01
```



```
DROP VIEW IF EXISTS my_first_view;
```

```
NOTICE:  view "my_first_view" does not exist, skipping
DROP VIEW

Query returned successfully in 47 msec.
```

PDP ACADEMY

E`TIBORINGIZ UCHUN RAHMAT!