

Widoki

Możliwe jest zapisanie zapytanie do widoku (view).

Widok jest obiektem bazy danych przechowującym zapytanie.

Na widoku możliwa jest realizacja następujących operacji:

- zapytania na view (SELECT),
- łączenie z normalnymi tabelami,
- używanie widoku do aktualizacji lub wstawiania danych do tabeli na których widok się opiera.

Regularne obiekty view nie przechowują danych.

Widoki

Tworzenie widoku

CREATE OR REPLACE VIEW nazwa AS

SELECT lista_kolumn
FROM nazwa_tabeli;

```
3 CREATE OR REPLACE VIEW view1 AS
4 SELECT company_name, contact_title
5 from suppliers;
6
```

Data Output Explain Messages Notifications

CREATE VIEW

Query returned successfully in 37 msec.

Widoki

Tworzenie widoku

CREATE OR REPLACE VIEW nazwa_widoku AS query

query może być:

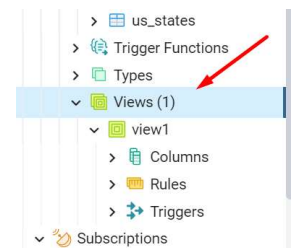
- SELECT
- SELECT z podzapytaniem
- SELECT z łączeniem (JOIN)

```
3 CREATE OR REPLACE VIEW view1 AS
4 SELECT company_name, contact_title
5 from suppliers;
6
```

Data Output Explain Messages Notifications

CREATE VIEW

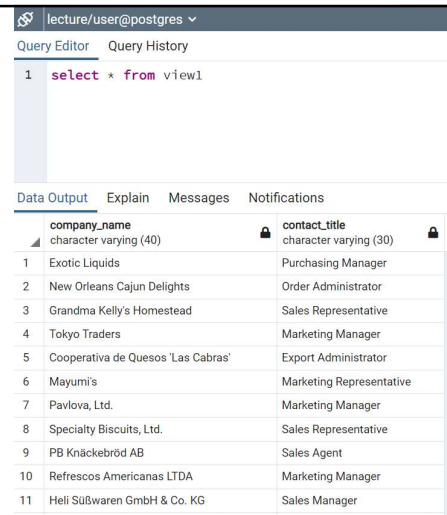
Query returned successfully in 37 msec.



Widoki

Używanie widoku

`SELECT * from nazwa_widoku:`



The screenshot shows a PostgreSQL Query Editor interface. At the top, the user is logged in as 'lecture/user@postgres'. Below the login bar, there are tabs for 'Query Editor' and 'Query History'. The 'Query Editor' tab is active, showing a single line of SQL: `1 select * from view1`. Below the query editor, there are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, displaying a table with two columns: 'company_name' (character varying (40)) and 'contact_title' (character varying (30)). The table contains 11 rows of data.

	company_name character varying (40)	contact_title character varying (30)
1	Exotic Liquids	Purchasing Manager
2	New Orleans Cajun Delights	Order Administrator
3	Grandma Kelly's Homestead	Sales Representative
4	Tokyo Traders	Marketing Manager
5	Cooperativa de Quesos 'Las Cabras'	Export Administrator
6	Mayumi's	Marketing Representative
7	Pavlova, Ltd.	Marketing Manager
8	Specialty Biscuits, Ltd.	Sales Representative
9	PB Knäckebröd AB	Sales Agent
10	Refrescos Americanas LTDA	Marketing Manager
11	Heli Süßwaren GmbH & Co. KG	Sales Manager

Widoki

Widoki z możliwością aktualizacji:

Umożliwiają aktualizację danych w tabelach powiązanych z widokiem. Są możliwe do osiągnięcia, gdy spełnione są następujące zasady:

- zapytanie musi mieć **JEDEN** element **FROM**, który może być tabelą lub innym widokiem z możliwością aktualizacji,

- zapytanie nie może posiadać następujących elementów:

DISTINCT, GROUP BY, WITH, LIMIT, OFFSET, UNION, INTERSECT, EXCEPT, HAVING.

- Nie można używać następujących elementów w liście selekcji:

funkcji widnow, set-returning, agregacji (sup, count, ...)

Widoki

Zmiana nazwy widoku:

`ALTER VIEW nazwa_widoku RENAME TO nowa_nazwa`

Usuwanie widoku:

`DROP VIEW nazwa_widoku`

Widoki

Do aktualizacji danych można używać:

`INSERT,`

`UPDATE,`

`DELETE`

razem z klauzulą `WHERE`

Aby aktualizacja się powiodła użytkownik musi posiadać odpowiednie uprawnienia do widoku **NIE MUSI POSIADAĆ UPRAWNIEN DO TABEL, NA KTÓRYCH WIDOK JEST ZBUDOWANY.**

Widoki

Do aktualizacji danych w tabeli używamy instrukcji **INSERT**, **UPDATE**, **DELETE** wraz z klauzulem **WHERE**.

Aby aktualizacja danych była możliwa, należy nadać odpowiednie uprawnienia na tabeli, na której chcemy działać.

PRZYKŁAD 1: Działanie na tabeli **tabela**.

Query Editor: `select * from tabela`

Data Output:

col1 integer	col2 integer	col3 integer
1	1	1
2	2	3
3	3	2
4	4	4
5	8	7
6	9	9

Query returned successfully in 38 msec.

Widoki

Do aktualizacji danych w widoku używamy instrukcji **INSERT**, **UPDATE**, **DELETE** wraz z klauzulem **WHERE**.

Aby aktualizacja danych była możliwa, należy nadać odpowiednie uprawnienia na widoku, na którym chcemy działać.

PRZYKŁAD 2: Działanie na widoku **view3**.

Query Editor: `SELECT * from view3`

Data Output:

col3 integer
1
2
3
4
5
6

Query returned successfully in 34 msec.

Widoki

Do aktualizacji danych w tabeli używamy instrukcji **INSERT**, **UPDATE**, **DELETE** wraz z klauzulem **WHERE**.

Aby aktualizacja danych była możliwa, należy nadać odpowiednie uprawnienia na tabeli, na której chcemy działać.

PRZYKŁAD 3: Działanie na tabeli **tabela**.

Query Editor: `CREATE OR REPLACE VIEW view3 AS select col3 from tabela`

Messages: `CREATE VIEW`

Query returned successfully in 38 msec.

Widoki

Do aktualizacji danych w widoku używamy instrukcji **INSERT**, **UPDATE**, **DELETE** wraz z klauzulem **WHERE**.

Aby aktualizacja danych była możliwa, należy nadać odpowiednie uprawnienia na widoku, na którym chcemy działać.

PRZYKŁAD 4: Działanie na widoku **view3**.

Query Editor: `INSERT INTO view3(col3) values (100), (200), (300)`

Messages: `INSERT 0 3`

Query returned successfully in 34 msec.

Widoki

Do aktualizacji danych używamy:

INSERT,

UPDATE,

DELETE

razem z klauzulą **WHERE**

Aby aktualizacja się odbyła, użytkownik musi posiadać odpowiednie uprawnienia na tabeli, na której wykonujemy operację. Aby uzyskać uprawnienia, należy użyć następującej instrukcji:

```
GRANT UPDATE ON TABELA TO UZYSKAC UPRAWNIENIA DO TABELA;
```

lecture/user@postgres

Query Editor Query History

```
1 select * from tabela
2
```

Data Output

Explain

Messages

Notifications

	col1 integer	col2 integer	col3 integer	
1	1	1	2	
2	2	3	2	
3	3	2	2	
4	4	4	1	
5	8	7	8	
6	9	9	7	
7	[null]	[null]	100	
8	[null]	[null]	200	
9	[null]	[null]	300	