

Widoki

Widok to nazwana kwerenda lub osłona wokół instrukcji *SELECT*

Widoki są podstawowym elementem relacyjnych baz danych, które można porównać do metod w klasach UML

Jesteśmy w stanie dodawać, usuwać wiersze widoku.

Widoki upraszczają zapytania i zwiększają modularność kodu

Korzyści widoków:

- uproszczenie złożonych zapytań
- poprawa wydajności dzięki buforowaniu wyników
- redukcja ilości kodu SQL
- integracja z językami obiektowymi dzięki widokom aktualizowanym
- zastosowanie mechanizmów autoryzacji na poziomie wiersza
- łatwe prowadzenie zmian bez konieczności wdrażania nowego oprogramowania
- implementacja warstwy abstrakcji między bazą danych a aplikacjami wysokopoziomowymi

Różnice od procedur składowanych

ich zależności są utrzymywane w bazie danych. są bardziej wydajne. modyfikacja widoku może być zabroniona z powodu efektu kaskadowych

Tworzenie widoku

```
CREATE [OR REPLACE] [TEMP | TEMPORARY] [RECURSIVE] VIEW nazwa_widoku  
AS
```

Rodzaje widoków

- tymczasowe : automatycznie usuwane na końcu sesji usera;

```
CREATE TEMP VIEW temp_user_view AS  
SELECT  
    id,  
    username,  
    email  
FROM  
    users  
WHERE  
    active=true;
```

- rekurencyjne obsługują rekurencje podobnie jak funkcje w językach wysokiego poziomu. dane hierarchiczne

```
CREATE RECURSIVE VIEW employee_hierarchy AS
-- Poziom bazowy: pracownicy bez przełożonych
SELECT id AS employee_id, name AS employee_name, manager_id
FROM employees
WHERE manager_id IS NULL
UNION ALL
-- Poziom rekurencyjny: pracownicy podlegli przełożonym
SELECT e.id AS employee_id, e.name AS employee_name, e.manager_id
FROM employees e
JOIN employee_hierarchy eh ON e.manager_id = eh.employee_id;
```

- Aktualizowane widoki (materializowane) - przechowują fizycznie dane w tabelach i mogą być okresowo odświeżane. Używane w dużych i często wykonywanych zapytaniach

```
CREATE MATERIALIZED VIEW sales_summary AS
SELECT product_id, SUM(quantity) AS total_sales
FROM sales
GROUP BY product_id;

...

REFRESH MATERIALIZED VIEW nazwa_widoku;
```

Aby usunąć widok z zależnościami

```
DROP VIEW nazwa_widoku_zaleznego;
DROP VIEW nazwa_widoku_pierwotnego;

DROP VIEW nazwa_widoku CASCADE -- usuwa odrazu widok pierwotny
```

widoki materializowane mają możliwość automatycznego aktualizowania, co oznacza że można robić INSERT UPDATE DELETE

warunki dla aktualizowanych widoków:

- oparty na tabeli
- brak: distinct, with, group by, offset, having, limit, union, except, intersect

Widok edytowalny - pozwala edycję danych

```
CREATE VIEW editable_view_customers AS
SELECT id,username,email,balance,is_actibe
FROM accounts
WHERE is_active = TRUE AND balance >0
WITH LOCAL CHECK OPTION;
```

Widok tylko do odczytu

```
CREATE VIEW readonly_view_customers AS
SELECT DISTINCT id,username,email,balance,is_actibe
FROM accounts
WHERE balance =0;
```