VIEW, STORED PROCEDURE, FUNCTION AND TRIGGER

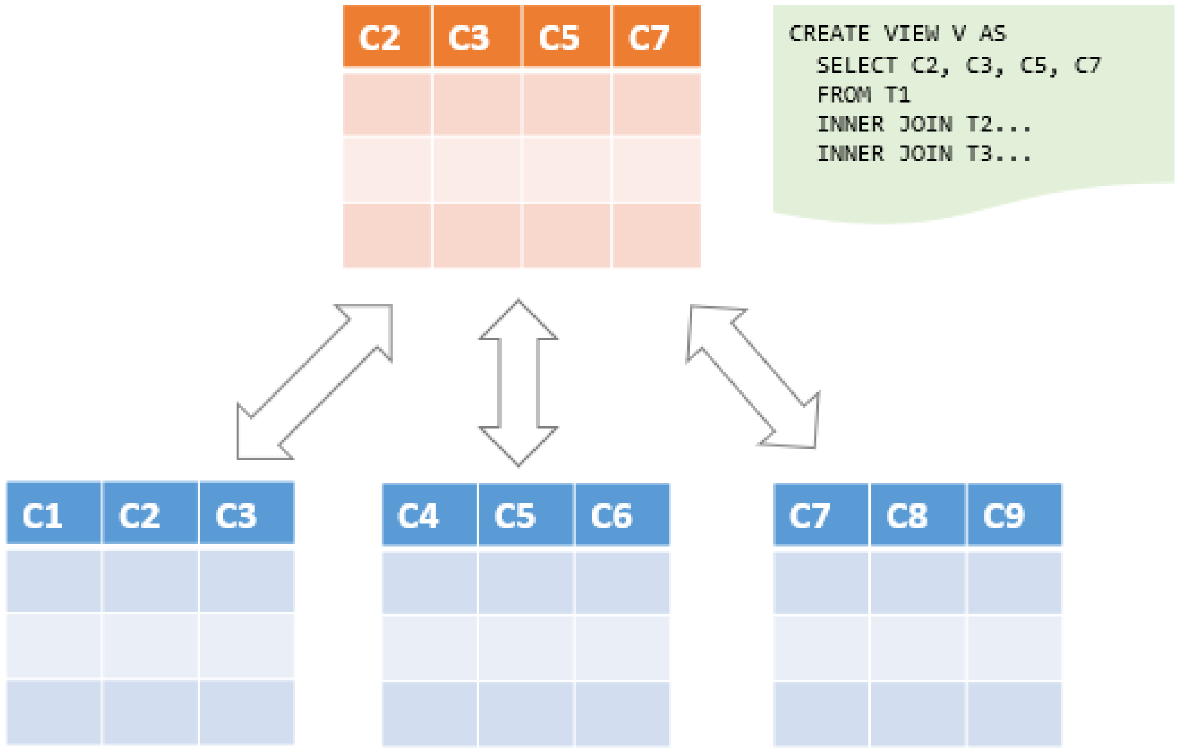
Lê Hồng Hải
UET-VNUH

Today's Overview

1. View
2. Stored Procedure, Function
3. Trigger

View Definition

- A view is a named query stored in the database catalog

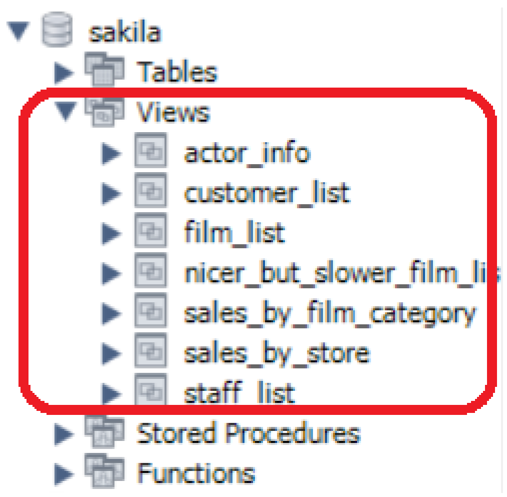


```
CREATE VIEW V AS
  SELECT C2, C3, C5, C7
  FROM T1
  INNER JOIN T2...
  INNER JOIN T3...
```

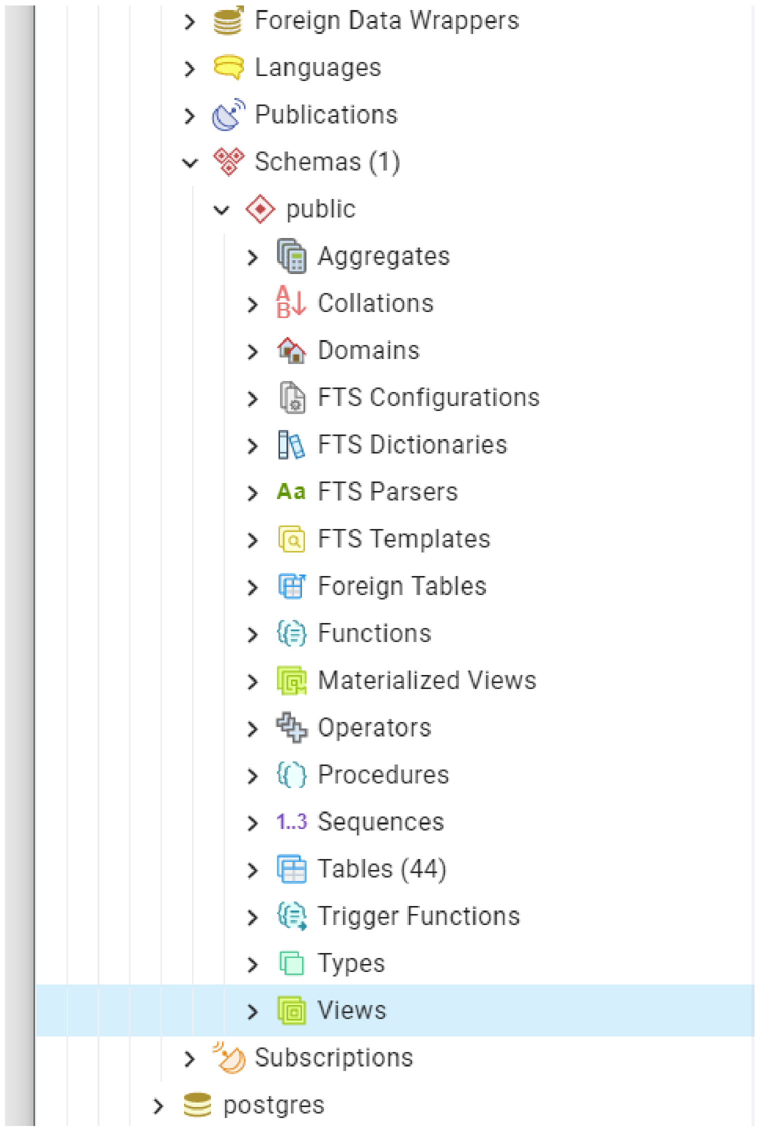
Advantages of Views

- Simplify complex query
  - You can create a view and reference to the view by using a simple SELECT statement instead of typing the query all over again
- Add extra security layers
  - A table may expose a lot of data including sensitive data such as personal and banking information
  - By using views and privileges, you can limit which data users can access by exposing only the necessary data to them

View in MySQL & PostgreSQL



sakila
Tables
Views
actor_info
customer_list
film_list
nicer_but_slower_film_li
sales_by_film_category
sales_by_store
staff_list
Stored Procedures
Functions



Foreign Data Wrappers
Languages
Publications
Schemas (1)
public
Aggregates
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Operators
Procedures
Sequences
Tables (44)
Trigger Functions
Types
Views
Subscriptions
postgres

Create a View

```
CREATE VIEW staff_list
AS
SELECT *
FROM staff AS s JOIN address AS a ON s.address_id =
    a.address_id JOIN city ON a.city_id = city.city_id
    JOIN country ON city.country_id =
    country.country_id;
```

Using View

```
SELECT first_name, last_name, address
FROM staff_list
```

Today's Overview

1
- View

2
- Stored Procedure, Function

3
- Trigger

Stored procedure

- When you use MySQL Workbench or mysql shell to **send the SQL query to MySQL Server**
- If you want to save this query on the database server for execution later, one way to do it is to use a stored procedure

Stored Procedure Example

```
CREATE PROCEDURE film_in_stock(IN p_film_id INT, IN
    p_store_id INT, OUT p_film_count INT)
READS SQL DATA
BEGIN
    SELECT inventory_id
    FROM inventory
    WHERE film_id = p_film_id
    AND store_id = p_store_id
    AND inventory_in_stock(inventory_id);

    SELECT FOUND_ROWS() INTO p_film_count;
END
```

Stored Procedure Example

```
CREATE FUNCTION inventory_in_stock(p_inventory_id INT) RETURNS BOOLEAN
READS SQL DATA
BEGIN
    DECLARE v_rentals INT;
    DECLARE v_out     INT;

    SELECT COUNT(*) INTO v_rentals
    FROM rental
    WHERE inventory_id = p_inventory_id;

    IF v_rentals = 0 THEN
      RETURN TRUE;
    END IF;

    SELECT COUNT(rental_id) INTO v_out
    FROM inventory LEFT JOIN rental USING(inventory_id)
    WHERE inventory.inventory_id = p_inventory_id
    AND rental.return_date IS NULL;

    IF v_out > 0 THEN
      RETURN FALSE;
    ELSE
      RETURN TRUE;
    END IF;
END $$
```

Stored Procedure

- Set value for variable
  - Using *SET or SELECT INTO*.
- Call SP:
  - *Call film_in_stock(1,1, @film_count);*
  - *Select @film_count;*

Stored Procedure

- A stored procedure can have parameters so you can pass values to it and get the result back
- Also, a stored procedure may contain control flow statements such as IF, CASE, and LOOP
- A stored procedure can call other stored procedures or stored functions, which allows you to organize your code more effectively

Stored Procedures Advantages

- Reduce network traffic
  - Applications have to send only the name and parameters of stored procedures.
- Centralize business logic in the database
  - You can use the stored procedures to implement business logic that is reusable by multiple applications
- Make the database more secure
  - The database administrator can grant appropriate privileges to applications that only access specific stored procedures without giving any privileges to the underlying tables.

Stored Procedures Disadvantages

- Lack of Portability
    - SQLServer uses T-SQL
    - Oracle uses PL-SQL
- Developing and maintaining stored procedures often requires a specialized skill

SP Syntax in MySQL

- *BEGIN*
  - *DECLARE* variables;
  - *DECLARE* cursors;
  - *DECLARE* conditions;
  - *DECLARE* handler;
  - other SQL commands;
- *END*;

IF THEN clause

```
IF condition THEN
    commands;
[ELSE IF condition THEN
    commands;]
[ELSE commands;]
END IF;
```

CASE clause

CASE *expression*

WHEN *value1* THEN *commands*;

[WHEN *value2* THEN *commands*;]

[ELSE *commands*;]

END CASE;

REPEAT UNTIL clause

```
[loopname:]
REPEAT commands;
UNTIL condition
END REPEAT [loopname];
```

WHILE clause

[loopname:]
WHILE condition *DO* commands;
 END WHILE [loopname];

Cursor

- The cursor is used to iterate through the rows of results returned by the query and process each row individually

Cursor Syntax

- DECLARE *cursor_name* CURSOR FOR *SELECT_statement*;
- OPEN *cursor_name*;
- Extract each record and move to the next record using the FETCH command

  FETCH *cursor_name* INTO *variable list*;
- CLOSE *cursor_name*;

Today's Overview

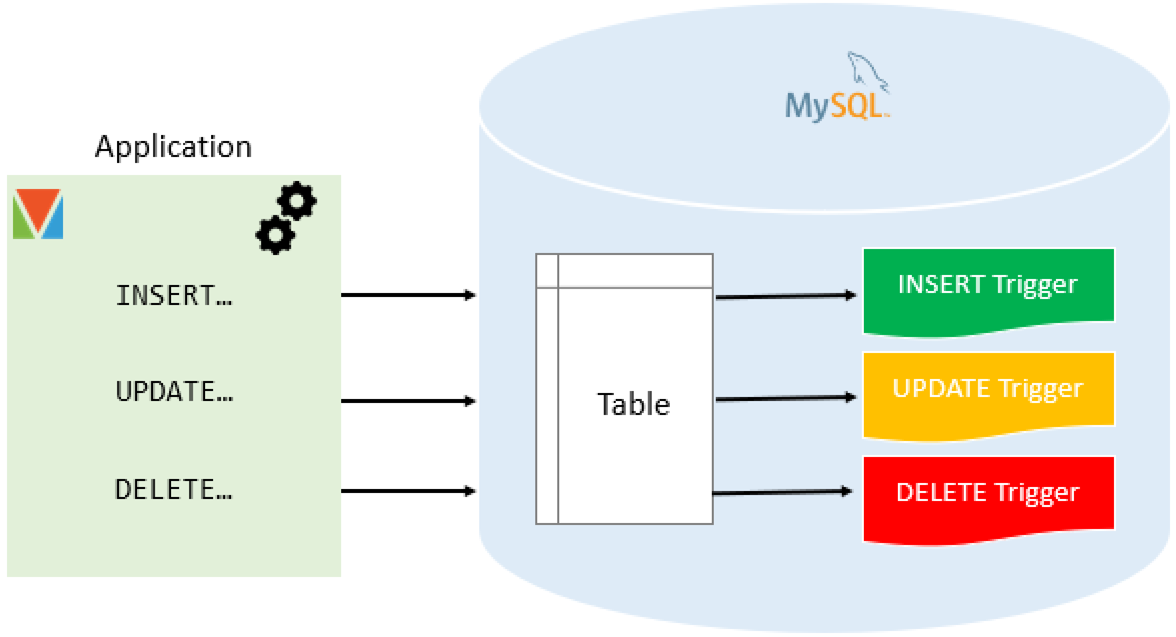
1
- View

2
- Stored Procedure, Function

3
- Trigger

Trigger

- a trigger is a stored program invoked automatically in response to an event such as insert, update, or delete that occurs in the associated table



Trigger Example

```
CREATE TRIGGER `upd_film` AFTER UPDATE ON `film`
  FOR EACH ROW BEGIN
  IF (old.title != new.title) or (old.description !=
  new.description)
  THEN
     UPDATE film_text
        SET title=new.title,
           description=new.description,
           film_id=new.film_id
     WHERE film_id=old.film_id;
  END IF;
 END
```

Trigger

- *OLD* is the row before being updated or deleted
- *NEW* is the row to insert or update into the table

Advantages of Triggers

- Helps us to automate the data alterations
- Helps us to detect errors on the database level
- Allows easy auditing of data

Disadvantages of Triggers

- Triggers can be difficult to troubleshoot because they execute automatically in the database
- Triggers may increase the overhead of the MySQL server

THANKS YOU