

MySQL Views

Sheeri K. Cabral, DBA

The Pythian Group, www.pythian.com

cabral@pythian.com

Boston MySQL User Group – September 8, 2008



The Pythian Group

- www.pythian.com
- Remote Database Management



What is a View?

- Relationally: virtual
- Bank Statement
- SQL Implementation: Define a table in terms of another



MySQL's Implementation

- Closer to relational model
- Dynamic, virtual only
- No “materialized views”



Why is a View Useful?

- Keeps data normalized
- Schema Abstraction
 - Simplicity
 - Security
 - Developer needs no schema knowledge



Simple CREATE VIEW

```
CREATE VIEW staff_name AS  
SELECT first_name, last_name  
FROM sakila.staff;
```



Changing a View

```
CREATE OR REPLACE VIEW staff_name AS  
SELECT CONCAT(first_name, ' ', last_name)  
        AS full_name  
FROM sakila.staff;
```



Dropping a View

```
DROP VIEW IF EXISTS staff_name;
```



Finding Views

- SHOW TABLE STATUS
- SHOW FULL TABLES
- TABLE_TYPE in information_schema.TABLES
- Rows in information_schema.VIEWS



SQL SECURITY

- DEFINER
 - default

- INVOKER

- Syntax:

CREATE

SQL SECURITY {INVOKER|DEFINER}

VIEW ...



View DEFINER

- `DEFINER = CURRENT_USER()`
 - Default
- `DEFINER = user@host`
 - SUPER users only
 - Warning if `user@host` does not exist



Finding View Definitions

- SHOW CREATE VIEW
- VIEW_DEFINITION field in information_schema.VIEWS



Performance Implications

- Views are dynamic
- Abstraction may lead to poor queries
- Not everything is a nail



ALGORITHM

- UNDEFINED
 - Default
- TEMPTABLE
- MERGE



Updatable Views

- Views can be updateable
- Only if:
 - one-to-one row relationships
 - no subqueries in the view definition
 - referenced views are updatable



Updatable View Problems

```
UPDATE staff_list  
  SET address="20 Workhaven Lane"  
 WHERE ID=1;
```



Inserting Into a View

- Same as inserting into a table
- Only if:
 - unique field names
 - non-view fields have default values
 - non-calculated fields



Constraints

- Simulate check constraints
- WITH CHECK OPTION
- Enforces the WHERE clause
- LOCAL **or** CASCADED



Full CREATE VIEW Syntax

CREATE

[OR REPLACE]

[ALGORITHM = {UNDEFINED | MERGE | TEMPTABLE}]

[DEFINER = { user | CURRENT_USER}]

[SQL SECURITY { DEFINER | INVOKER }]

VIEW view_name [(column_list)]

AS select_statement

[WITH [CASCADED | LOCAL] CHECK OPTION]



Changing a View, part 2

- ALTER VIEW
- select_statement always present



ALTER VIEW

ALTER

[ALGORITHM = {UNDEFINED | MERGE | TEMPTABLE}]

[DEFINER = { user | CURRENT_USER}]

[SQL SECURITY { DEFINER | INVOKER }]

VIEW view_name [(column_list)]

AS select_statement

[WITH [CASCADED | LOCAL] CHECK OPTION]



Replication and Views

- CREATE VIEW, ALTER VIEW, DROP VIEW
- Treated like tables
- Be careful!



View Limitations

- Table references
- Variables
- SQL



Questions?

Comments?

Suggestions?

