Enterprise MySQL: Views in MySQL 5.0

Trudy Pelzer, Senior Software Architect & Sanja Byelkin, Software Developer
MySQL AB
trudy@mysql.com / sanja@mysql.com

MySQL Users Conference, 2005
April 18-21, 2005
New DDL Statements

• CREATE VIEW
• ALTER VIEW
• DROP VIEW
What is a View? (I)

• A “virtual” table
• A “view of” a table
• A look at the data in a different way
What is a View? (II)

CREATE VIEW v AS SELECT * FROM t;

SELECT * FROM v;

... same effect as

SELECT * FROM t;
Two Kinds of Tables

Normal tables = BASE TABLES

• Made with CREATE TABLE

Viewed tables = VIEWS

• Made with CREATE VIEW

A view is a named, derived table whose definition is a persistent part of the database.
Two Kinds of Tables

Normal tables = BASE TABLES
• Made with CREATE TABLE

Viewed tables = VIEWS
• Made with CREATE VIEW

A view is a named, derived table whose definition is a persistent part of the database.
Underlying Table (I)

CREATE VIEW v1 AS SELECT * FROM t1;

/* t1 is the "underlying table" for view v1 */
CREATE VIEW view1 AS

SELECT Create_view_priv FROM mysql.user;
Underlying Table (II)

CREATE VIEW v2 AS SELECT 'a';

/* there is no underlying table for view v2 */
CREATE VIEW v3 AS SELECT * FROM v1, v2;

/* there are three underlying tables for view v3 */
CREATE VIEW Syntax

CREATE [OR REPLACE]
[ALGORITHM =
  {MERGE | TEMPTABLE | UNDEFINED}]
VIEW view_name
  [(column_list)]
AS select_statement
  [WITH [CASCADED | LOCAL] CHECK OPTION]
CREATE VIEW Syntax

CREATE  [OR REPLACE]
[ALGORITHM =
    {MERGE | TEMPTABLE | UNDEFINED}]
VIEW view_name
[(column_list)]
AS select_statement
[WITH [CASCADED | LOCAL] CHECK OPTION]

CREATE VIEW view1 ...
Illegal CREATE VIEW: Namespace

mysql> CREATE TABLE tab1 (col1 INT);
Query OK, 0 rows affected (0.30 sec)

mysql> CREATE VIEW tab1 AS SELECT 'a';
ERROR 1050 (42S01): Table 'tab1' already exists
CREATE VIEW Syntax

```
CREATE  [OR REPLACE]
[ALGORITHM =
   {MERGE | TEMPTABLE | UNDEFINED}]
VIEW view_name
[(column_list)]
AS select_statement
[WITH [CASCADED | LOCAL] CHECK OPTION]
```

CREATE OR REPLACE VIEW view1 ...
CREATE VIEW Syntax

CREATE  [OR REPLACE]
[ALGORITHM =
  {MERGE | TEMPTABLE | UNDEFINED}] VIEW view_name
[(column_list)]
AS select_statement
[WITH [CASCADED | LOCAL] CHECK OPTION]
Two Algorithms for Views (I)

CREATE TABLE t (col1 INT);
CREATE VIEW v AS SELECT * FROM t;

**Merge algorithm:**
SELECT * FROM v;
becomes
SELECT * FROM t;
Two Algorithms for Views (II)

CREATE TABLE t (col1 INT);
CREATE VIEW v AS SELECT AVG(col1) FROM t;

Temporary table (temptable) algorithm:
SELECT * FROM v;
becomes
CREATE TEMPORARY TABLE temp_table AS
  SELECT AVG(col1) FROM t;
SELECT * FROM temp_table;
[ALGORITHM = ...]

Clause syntax:
ALGORITHM = {MERGE | TEMPTABLE | UNDEFINED}

Default is UNDEFINED.

- It means “let MySQL choose” the best algorithm.
Letting MySQL Choose (I)

```
CREATE TABLE t (col1 INT);

CREATE VIEW v AS SELECT col1 FROM t;
-- MERGE algorithm will be used
```
Letting MySQL Choose (II)

```
CREATE TABLE t (col1 INT);

CREATE VIEW v AS SELECT AVG(col1) FROM t;
-- TEMPTABLE algorithm will be used
```
Letting MySQL Choose (III)

**TEMPTABLE** algorithm will be used for view definitions with:

- DISTINCT
- Aggregate function
- UNION
- GROUP BY
- HAVING
- Only literal values
Overriding MySQL’s Choice

CREATE ALGORITHM = TEMPTABLE VIEW v AS
    SELECT * FROM t;
CREATE VIEW Syntax

CREATE [OR REPLACE]
[ALGORITHM =
  {MERGE | TEMPTABLE | UNDEFINED}]
VIEW view_name
[(column_list)]
AS select_statement
[WITH [CASCADED | LOCAL] CHECK OPTION]
Omitted [ (column list) ]

By default:
CREATE VIEW v AS SELECT col1, col2 FROM t;

mysql> SELECT * FROM v;
+----------+
| col1    | col2    |
+----------+
| 10 | 20 |
+----------+
Included [ (column list) ]

Different names specified:
CREATE VIEW v (view_col1, view_col2) AS
    SELECT col1, col2 FROM t;

mysql> SELECT * FROM v;
+---------------------+
| view_col1 | view_col2 |
+---------------------+
| 10        | 20        |
+---------------------+
AS Instead of [ (column list) ]

Using AS instead:

CREATE VIEW v AS

    SELECT col1 AS select_list_name FROM t;

mysql> SELECT * FROM v;

+------------------+
<table>
<thead>
<tr>
<th>select_list_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
</tr>
</tbody>
</table>

+------------------+
Illegal: Duplicate Column Names

mysql> CREATE TABLE t (col1 INT);
Query OK, 0 rows affected (0.29 sec)

mysql> SELECT col1, col1 FROM t;
Empty set (0.00 sec)

mysql> CREATE VIEW v AS
    
    -> SELECT col1, col1 FROM t;
ERROR 1060 (42S21): Duplicate column name 'col1'
Columns are Set at

CREATE VIEW Time (I)

CREATE TABLE t (col1 INT);
Columns are Set at

CREATE VIEW Time (I)

CREATE TABLE t (col1 INT);

CREATE VIEW v AS SELECT * FROM t;
Columns are Set at
CREATE VIEW Time (I)

CREATE TABLE t (col1 INT);
CREATE VIEW v AS SELECT * FROM t;

mysql> SELECT * FROM v;
+------+
| col1 |
+------+
| 10   |
+------+
Columns are Set at
CREATE VIEW Time (I)

CREATE TABLE t (col1 INT);

CREATE VIEW v AS SELECT * FROM t;

ALTER TABLE t ADD col2 INT;
Columns are Set at
CREATE VIEW Time (I)

CREATE TABLE t (col1 INT);
CREATE VIEW v AS SELECT * FROM t;
ALTER TABLE t ADD col2 INT;

mysql> SELECT * FROM v;
+------+
| col1 |
+------+
| 10   |
+------+
CREATE VIEW Syntax

```
CREATE [OR REPLACE]
[ALGORITHM =
  {MERGE | TEMPTABLE | UNDEFINED}]
VIEW view_name
[(column_list)]
AS select_statement
[WITH [CASCADED | LOCAL] CHECK OPTION]
```
AS select_statement

The view’s SELECT definition

a.k.a.:

• Oracle – subquery
• SQL Standard – query expression
Illegal : Variable or Parameter (I)

mysql> CREATE VIEW v AS 
   -> SELECT * FROM t 
   -> WHERE col1 = @variable AND col2 = ?;
ERROR 1351 (HY000): View's SELECT contains a variable or parameter
Illegal: Variable or Parameter (II)

```sql
mysql> CREATE PROCEDURE p (param INT)
    -> CREATE VIEW v AS SELECT param;
Query OK, 0 rows affected (0.01 sec)
```

```sql
mysql> CALL p(5);
ERROR 1351 (HY000): View's SELECT contains a variable or parameter
```
Illegal: Temporary Views

CREATE TEMPORARY TABLE tt (col1 INT);

mysql> CREATE VIEW vtt AS SELECT * FROM tt;
ERROR 1352 (HY000): View's SELECT contains a temporary table 'tt'

mysql> CREATE TEMPORARY VIEW tv AS SELECT 'a';
ERROR 1064 (42000): You have an error in your SQL syntax ...
Illegal: Missing Table

mysql> CREATE VIEW v AS
       -> SELECT * FROM no_such_table;
ERROR 1146 (42S02): Table 'db.no_such_table' doesn't exist
Illegal: Subquery in FROM

mysql> CREATE VIEW v AS
   -> SELECT *
   -> FROM (SELECT col1 FROM t) AS x;
ERROR 1349 (HY000): View's SELECT contains a subquery in the FROM clause
Illegal: No Privileges

mysql> CREATE VIEW v AS
    -> SELECT col1 FROM t;
ERROR 1142 (42000): create view command denied to user 'trudy'@'localhost' for table 'v'
CREATE VIEW v AS SELECT
    `column1`,
    column2,
    UPPER(`table1`.`column1`),
    column3 + 10,
    'The rain in Spain' || 'q'
FROM `table1`, table2;
Legal: Niladic Functions

CREATE VIEW v AS SELECT CURRENT_TIME;

mysql> SELECT * FROM v;
+-----------------+
<table>
<thead>
<tr>
<th>current_time</th>
</tr>
</thead>
<tbody>
<tr>
<td>12:00:00</td>
</tr>
</tbody>
</table>
+-----------------+
CREATE VIEW v AS
    SELECT col1 + 1 FROM t
WHERE col1 > 4;
CREATE VIEW v1 AS
    SELECT t1.col1
    FROM t1 JOIN t2 ON t1.col1 = t2.col1;
Legal: View with GROUP BY

CREATE VIEW v AS
    SELECT col1, SUM(col2) AS sum_col2
    FROM t
    GROUP BY col1;
Legal: View with HAVING

CREATE VIEW v AS

    SELECT col1, SUM(col2) AS sum_col2
    FROM t
    GROUP BY col1 HAVING SUM(col2) > 10;
Legal: View with ORDER BY

CREATE VIEW v AS
    SELECT col1, col2
FROM t
ORDER BY col1;
View Queries can Include any SELECT Clause

CREATE VIEW v AS

    SELECT col1 FROM t WHERE col1 > 4;

SELECT * FROM v WHERE col1 < 6;

same effect as:

SELECT * FROM t

    WHERE col1 > 4 AND col1 < 6;
Views can be Manipulated with

**INSERT/UPDATE/DELETE**

```sql
CREATE TABLE t (col1 INT, col2 INT);
CREATE VIEW v AS SELECT * FROM t;

INSERT INTO v VALUES (1, 2);

UPDATE v SET col1 = col1 + 10;

DELETE FROM v WHERE col1 < 5;
```
CREATE VIEW Syntax

CREATE [OR REPLACE]
[ALGORITHM =
     {MERGE | TEMPTABLE | UNDEFINED}]
VIEW view_name
[(column_list)]
AS select_statement
[WITH [CASCADED | LOCAL] CHECK OPTION]
WITH CHECK OPTION

CREATE VIEW v1 AS
   SELECT col1 FROM t WHERE col1 <> 'A'
   WITH CHECK OPTION;

mysql> INSERT INTO v1 VALUES ('A');
ERROR 1369 (HY000): CHECK OPTION failed 'db.v1'

mysql> UPDATE v1 SET col1 = 'A';
ERROR 1369 (HY000): CHECK OPTION failed 'db.v1'
WITH LOCAL CHECK OPTION

CREATE VIEW v2 AS

    SELECT * FROM v1 WHERE col1 <> 'B'
    WITH LOCAL CHECK OPTION;

mysql> INSERT INTO v2 VALUES ('B');
ERROR 1369 (HY000): CHECK OPTION failed 'db.v2'

mysql> INSERT INTO v2 VALUES ('A');
Query OK, 1 row affected (0.01 sec)
WITH LOCAL CHECK OPTION

CREATE VIEW v2 AS
  SELECT * FROM v1 WHERE col1 <> 'B'
WITH LOCAL CHECK OPTION;

mysql> INSERT INTO v2 VALUES ('B');
ERROR 1369 (HY000): CHECK OPTION failed 'db.v2'

mysql> INSERT INTO v2 VALUES ('A');
Query OK, 1 row affected (0.01 sec)
WITH CASCADED CHECK OPTION

CREATE VIEW v3 AS
    SELECT * FROM v2 WHERE col1 <> 'C'
    WITH CASCADED CHECK OPTION;

mysql> INSERT INTO v3 VALUES ('C');
ERROR 1369 (HY000): CHECK OPTION failed ... 
-- because v3's condition doesn't allow 'C'
WITH CASCADED CHECK OPTION

CREATE VIEW v3 AS
    SELECT * FROM v2 WHERE col1 <> 'C'
WITH CASCADED CHECK OPTION;

mysql> INSERT INTO v3 VALUES ('B');
ERROR 1369 (HY000): CHECK OPTION failed ... -- because v2's condition doesn't allow 'B'
and v3 depends on v2
WITH CASCADED CHECK OPTION

CREATE VIEW v3 AS
    SELECT * FROM v2 WHERE col1 <> 'C'
    WITH CASCADED CHECK OPTION;

mysql> INSERT INTO v3 VALUES ('A');
ERROR 1369 (HY000): CHECK OPTION failed ...  
-- because v1's condition doesn't allow 'A' 
and v3 depends on v1 through v2
ALTER VIEW Syntax

ALTER

[ALGORITHM =
  {MERGE | TEMPTABLE | UNDEFINED}]

VIEW view_name

[(column_list)]

AS select_statement

[WITH [CASCADED | LOCAL] CHECK OPTION]
DROP VIEW Syntax

DROP VIEW

[IF EXISTS]

view_name [, view_name ...]

[RESTRICT | CASCADE]
Legal: DROP Multiple Views

mysql> DROP VIEW view1;
Query OK ...

mysql> DROP VIEW view2, view3, view4;
Query OK ...

- DROP privilege required for each view.
No Error with IF EXISTS

mysql> DROP VIEW view5;
ERROR 1051 (42S02): Unknown table 'view5'

mysql> DROP VIEW IF EXISTS view5;
Query OK, 0 rows affected, 1 warning
RESTRICT & CASCADE: No Difference

DROP VIEW view1;

DROP VIEW view1 RESTRICT;

DROP VIEW view1 CASCADE;

• All three statements take the same action.
SHOW

- DESCRIBE
- SHOW COLUMNS
- SHOW TABLE STATUS
- SHOW TABLES
- SHOW GRANTS FOR user
- SHOW CREATE TABLE
SHOW COLUMNS / DESCRIBE (I)

CREATE TABLE table1 (  
coll INT, PRIMARY KEY (coll));

CREATE VIEW view1 AS SELECT * FROM table1;
SHOW COLUMNS / DESCRIBE (II)

```sql
mysql> DESCRIBE table1;
+-------------------------+-------+------+-----+---------+------------+
| Field  | Type    | Null | Key | Default |
+---------+---------+------|-----+---------+------------+
| col1    | int(11) |      | PRI | 0       |
+---------+---------+------|-----+---------+------------+

mysql> DESCRIBE view1;
+-------------------------+-------+------+-----+---------+------------+
| Field  | Type    | Null | Key | Default |
+---------+---------+------|-----+---------+------------+
| col1    | int(11) |      |     | 0       |
+---------+---------+------|-----+---------+------------+
```
SHOW TABLE STATUS

mysql> SHOW TABLE STATUS\G
...
************************** 2. row **************************
    Name: v
  Engine: NULL
Version: NULL
...
Comment: view
SHOW TABLES

mysql> SHOW FULL TABLES;
+-------------------------+
| Tables_in_db | table_type |
+-------------------------+
| table1 | BASE TABLE |
| view1 | VIEW |
+-------------------------+
SHOW CREATE TABLE/VIEW

mysql> SHOW CREATE VIEW v\G
 **************** 1. row ****************
          View: v
Create View: CREATE ALGORITHM=UNDEFINED VIEW `db`.`v` AS select `db`.`t`.`col1` AS `col1`, `db`.`t`.`col2` AS `col2` from `db`.`t`
INFORMATION_SCHEMA.VIEWS

mysql> SELECT * FROM
   -> INFORMATION_SCHEMA.VIEWS\G

************************** 1. row **************************

TABLE_CATALOG: NULL
TABLE_SCHEMA: db
TABLE_NAME: v1
VIEW_DEFINITION: select `db`.`t1`.`c1` AS `c1`
                 from `db`.`t1`
CHECK_OPTION: LOCAL
IS_UPDATABLE: YES
Canonical Form (I)

```
SET SQL_MODE = 'ANSI';
CREATE VIEW v AS SELECT 'a' || 'b';

mysql> SHOW CREATE VIEW v
+-------------------+
| View              |
+-------------------+
| v                 |
+-------------------+
| Create View: CREATE VIEW "db"."v" AS select concat(_latin1'a', _latin1'b') AS `a' || 'b` |
+-------------------+
```
### Canonical Form (II)

```
mysql> SELECT * FROM v; -- sql_mode = 'ANSI'
+------------+
| 'a' || 'b' |
+------------+
| ab         |
+------------+

mysql> SELECT * FROM v; -- sql_mode = ''
+------------+
| 'a' || 'b' |
+------------+
| ab         |
+------------+
```
Canonical Form (III)

```sql
SET SQL_MODE = 'ANSI';
CREATE VIEW v AS SELECT 'a' || 'b';
```

```
mysql> SHOW CREATE VIEW v
+------------------------+
| View                   |
+------------------------+
| CREATE VIEW "db"."v" AS select concat(_latin1'a',_latin1'b') AS `a' || 'b` |
+------------------------+
```

Environment Changes (I)

CREATE TABLE t (c CHAR(5));
INSERT INTO t VALUES ('tom'), ('dick'), ('harry');
CREATE VIEW v AS SELECT group_concat(c) FROM t;

mysql> SELECT * FROM v;
+-----------------+
| group_concat(c) |
+-----------------+
| tom,dick,harry  |
+-----------------+
Environment Changes (II)

```sql
mysql> SET @group_concat_max_len = 4;

mysql> SELECT * FROM v;
+-----------------+
<table>
<thead>
<tr>
<th>group_concat(c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>tom,</td>
</tr>
</tbody>
</table>
+-----------------+
```
New Privileges

GRANT SHOW VIEW ON *.* TO peter;

GRANT CREATE VIEW ON *.* TO peter;

REVOKE CREATE VIEW ON *.* FROM peter;

REVOKE SHOW VIEW ON *.* FROM peter;
Upgrading to MySQL 5.0

The `mysql_fix_privilege_tables` script does this:

```
UPDATE user SET
  Create_view_priv=Create_priv,
  Show_view_priv=Create_priv
WHERE user <> ''
  AND "Already Had Create_view_priv" = FALSE;
```
CREATE VIEW Privilege

GRANT CREATE VIEW ON database1.view1 to joe;

GRANT CREATE VIEW ON database1.* TO joe;

GRANT CREATE VIEW ON *.* TO joe;
CREATE VIEW Privilege

GRANT CREATE VIEW ON database1.view1 to joe;

GRANT CREATE VIEW ON database1.* TO joe;

GRANT CREATE VIEW ON *.* TO joe;
CREATE VIEW Privilege

GRANT CREATE VIEW ON database1.view1 to joe;

GRANT CREATE VIEW ON database1.* TO joe;

GRANT CREATE VIEW ON *.* TO joe;
CREATE VIEW Privilege

GRANT CREATE VIEW ON database1.view1 to joe;

GRANT CREATE VIEW ON database1.* TO joe;

GRANT CREATE VIEW ON *.* TO joe;
Other Privileges Needed (I)

Standard SQL requirement:

- SELECT privilege on all columns in view's SELECT definition

MySQL requirement:

- Any privilege on all columns in view's SELECT definition
Other Privileges Needed (II)

CREATE VIEW v1 AS

SELECT c1, c2 FROM t1
WHERE c3 IN (SELECT c3 FROM t3);  

- User must have SELECT|UPDATE|etc. on c1, c2, and c3 of t1.
- User must have SELECT|UPDATE|etc. on c3 of t3.
- User must have CREATE VIEW.
Privileges Needed to Use a View

General rule:
• Same as for tables.

Special rule:
• If you have privilege on a view, you don't need a privilege on its underlying tables.
SHOW VIEW Privilege – Why?

Why is the privilege needed?
SHOW VIEW Privilege (I)

1. To alleviate confusion:

CREATE VIEW view1 AS SELECT * FROM table1;

mysql> EXPLAIN SELECT * FROM view1;

+----+-------------+--------+-------+ ... 
| id | select_type | table  | type  | ... 
+----+-------------+--------+-------+ ... 
|1   | PRIMARY     | table1 | index | ... 
+----+-------------+--------+-------+ ...
SHOW VIEW Privilege (I)

2. To close a security loop-hole:

CREATE VIEW view1 AS SELECT * FROM table1;

mysql> EXPLAIN SELECT * FROM view1;

+----+-------------+--------+-------+
| id | select_type | table  | type  |
+----+-------------+--------+-------+
| 1  | PRIMARY     | table1 | index |
+----+-------------+--------+-------+
SHOW VIEW Privilege (II)

CREATE VIEW view1 AS SELECT * FROM table1;

EXPLAIN SELECT * FROM view1;
-- Works if user has privileges on table1

OR if user has SHOW VIEW privilege on view1.
What is an updatable view?
• Can accept UPDATE, DELETE
• Can sometimes accept INSERT

Effect:
• As if you changed the data in the underlying base table
What’s NOT Updatable

1. UNION ALL
2. UNION
3. DISTINCT / DISTINCTROW or grouping
4. Subquery in the select list
5. Non-updatable table in the FROM clause
6. Subquery table same as outer table
7. ALGORITHM = TEMPTABLE
View with UNION ALL

CREATE VIEW v AS
    SELECT * FROM t1
    UNION ALL
    SELECT * FROM t2;

1. UNION ALL = not an updatable view
View with UNION

CREATE VIEW v AS
  SELECT * FROM t1
UNION
  /* or EXCEPT or INTERSECT */
  SELECT * FROM t2;

2. UNION =
not an updatable view
View with Grouping or DISTINCT

CREATE VIEW v AS

SELECT DISTINCT col1 FROM t1;

CREATE VIEW v AS

SELECT AVG(c1), c2 FROM t1
GROUP BY c2 HAVING c2 > 10;

3. DISTINCT / DISTINCTROW or grouping = not an updatable view
CREATE VIEW v AS
    SELECT
        t1.*,
        (SELECT c1 FROM v2 WHERE c1 = 1)
    FROM t1;

4. Subquery in the select list = not an updatable view
View on Non-updatable Table

CREATE VIEW view1 AS
    SELECT DISTINCT * FROM table1;

CREATE VIEW view2 AS SELECT * FROM view1;

5. Non-updatable table in the FROM clause = not an updatable view
View with Subquery Table same as Outer Table

CREATE VIEW v AS
  SELECT * FROM table1
WHERE col1 IN
  (SELECT MAX(col1) FROM table1);

6. Subquery table same as outer table = not an updatable view
View with

ALGORITHM = TEMPTABLE

CREATE ALGORITHM = TEMPTABLE VIEW v AS
SELECT * FROM t;

7. ALGORITHM = TEMPTABLE =
not an updatable view
“Delete-able” Views

mysql> CREATE TABLE t (c1 INT, c2 INT);
Query OK ...

mysql> CREATE VIEW v AS SELECT c1 FROM t;
Query OK ...

mysql> DELETE FROM v;
Query OK ...
View Updatability Flag (I)

Set at CREATE VIEW time.

- YES (true) if UPDATE, DELETE and similar operations are legal for the view;
- otherwise NO (false).
View Updatability Flag (II)

mysql> SELECT * FROM
   -> INFORMATION_SCHEMA.VIEWS\G
************ 1. row **************
   
   TABLE_CATALOG: NULL
   TABLE_SCHEMA: db
   TABLE_NAME: v1
   VIEW_DEFINITION: select `db`.`t1`.`c1` AS `c1` from `db`.`t1`
   CHECK_OPTION: LOCAL
   IS_UPDATABLE: YES
The “Insertable Into” Criteria

INSERT and view must:
1. contain all columns, in the underlying table, that have no default value

2. not contain a derived column in the select list (i.e. only simple column references allowed)

3. not contain a duplicated column in the select list (e.g. SELECT a AS a1, a AS a2 FROM t; not allowed)
“Insertable Into” Criterion (I)

Must contain all columns, in the underlying table, that have no default value

```sql
SET SQL_MODE='traditional';
CREATE TABLE t (c1 INT, c2 INT NOT NULL);
CREATE VIEW v AS SELECT c1 FROM t;

UPDATE v SET c1 = 5;  /* OK */
INSERT INTO v VALUES (1);  /* not OK */
```
“Insertable Into” Criterion (I)

Must contain all columns, in the underlying table, that have no default value

SET SQL_MODE='traditional';
CREATE TABLE t (c1 INT, c2 INT NOT NULL);
CREATE VIEW v AS SELECT * FROM t;

UPDATE v SET c1 = 5; /* OK */
INSERT INTO v (c1) VALUES (1); /* not OK */
“Insertable Into” Criterion (II)

Must not contain a derived column in the select list (i.e. only simple column references allowed)

CREATE TABLE t (c1 INT, c2 CHAR);
CREATE VIEW v AS
    SELECT c1 + 1, UPPER(c2), c1 FROM t;

UPDATE v SET c1 = 5; /* succeeds */
INSERT INTO t (c1) VALUES (5); /* fails */
“Insertable Into” Criterion (III)

Must not contain a duplicated column in the select list (e.g. SELECT a AS a1, a AS a2 FROM t; not allowed)

CREATE TABLE t (a int);
CREATE VIEW v AS
  SELECT a AS a1, a AS a2 FROM t;

UPDATE v SET a1 = 1; /* succeeds */
INSERT INTO v (a1) VALUES (1); /* fails */
Joined Views may be Updatable

CREATE VIEW v (t1_c1, t2_c1) AS
SELECT * FROM t1, t2;
Joined Views may be Updatable

CREATE VIEW v (t1_c1, t2_c1) AS
    SELECT * FROM t1, t2;

mysql> INSERT INTO v (t1_c1) VALUES (1);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO v (t2_c1) VALUES (2);
Query OK, 1 row affected (0.00 sec)
Joined Views may be Updatable

CREATE VIEW v (t1_c1, t2_c1) AS
  SELECT * FROM t1, t2;

mysql> UPDATE v SET t1_c1 = 2;
Query OK, 2 rows affected (0.00 sec)
Rows matched: 2  Changed: 2  Warnings: 0

• UPDATE v ... must be transformable to UPDATE t1, t2 ...
Using Views

What are views for?
Use of Views (I)

1. Views can be effective copies of base tables.

CREATE VIEW v AS SELECT * FROM t;

SELECT * FROM v;
instead of
SELECT * FROM t;
Use of Views (II)

2. Views can provide a synonym for a too-long table name.

CREATE VIEW cols AS
    SELECT * FROM INFORMATION_SCHEMA.COLUMNS
    WHERE TABLE_SCHEMA = DATABASE();

SELECT * FROM cols;
Use of Views (III)

3. Views can be used in place of constraint checks.

CREATE TABLE t (col1 INT, col2 INT);

CREATE VIEW v AS SELECT col1, col2
    FROM t WHERE
    (col1 is NOT NULL OR col2 IS NOT NULL)
    AND col1 BETWEEN 1 AND 5
    WITH CHECK OPTION;
Use of Views (III)

3. Views can be used in place of constraint checks.

CREATE TABLE t (col1 INT, col2 INT);

CREATE VIEW v AS SELECT col1, col2
    FROM t WHERE
    (col1 IS NOT NULL OR col2 IS NOT NULL)
    AND col1 BETWEEN 1 AND 5
WITH CHECK OPTION;
Use of Views (III)

3. Views can be used in place of constraint checks.

CREATE TABLE t (col1 INT, col2 INT);

CREATE VIEW v AS SELECT col1, col2
   FROM t WHERE
      (col1 is NOT NULL OR col2 IS NOT NULL)
   AND col1 BETWEEN 1 AND 5
WITH CHECK OPTION;
Use of Views (III)

3. Views can be used in place of constraint checks.

CREATE TABLE t (col1 INT, col2 INT);

CREATE VIEW v AS SELECT col1, col2
FROM t WHERE
  (col1 is NOT NULL OR col2 IS NOT NULL)
AND col1 BETWEEN 1 AND 5
WITH CHECK OPTION;
Use of Views (IV)

4. Views can be used instead of a subquery.

```sql
SELECT * FROM
(SELECT * FROM t2 UNION SELECT * FROM t3) AS t2_t3;

CREATE VIEW t2_t3 AS
SELECT * FROM t2 UNION SELECT * FROM t3;

SELECT * FROM t2_t3;
```
Use of Views (V)

5. Table Restructuring: Views can be used to hide columns.

CREATE TABLE t (            
    col1 INT, col2 CHAR(5), COL3 DATE);

CREATE VIEW v AS SELECT col1, col3 FROM t;

GRANT SELECT ON v TO some_user;
Use of Views (VI)

6. Data Security:
Views can be used to hide rows.

CREATE TABLE t (
    col1 INT, col2 CHAR(5), COL3 DATE);

CREATE VIEW v AS SELECT * FROM t
    WHERE col1 BETWEEN 10 AND 100;

GRANT SELECT ON v TO some_user;
Source

- sql/sql_view.cc
- mysql-test/t/view.test
Thank You

For further information:

• White Paper:  
  http://dev-mysql.com/tech-resources/articles/mysql-views.pdf

• Forum:  