

Grand Tour of the `information_schema`

Roland Bouman

Curriculum Developer (Training department)

Sun | MySQL

Diagram:

http://www.xcdsql.org/MySQL/information_schema/5.1/MySQL_5_1_INFORMATION_SCHEMA.html

Objectives

- Overview of the information_schema
- Discuss a few common queries
- Explain a few gotcha's
- Demo some applications
- Answer your questions

Experiences?

- Information schema users?
- 5.0
- 5.1
 - plugin writers?
- 6.0

What is the information_schema?

- Built-in database
 - 'virtual'
 - accessible to all users
 - contains 'tables' (actually, system views)
- Alternative to SHOW commands
 - SHOW commands are a MySQL proprietary extension
 - information_schema defined in the SQL standard (ISO 9075)
- Metadata: data about data
 - Data about databases and their contents
- Dynamic status data
 - Data about the server status
 - Data about the connections

Information schema contents

–Tables etc.

- COLUMNS
- ENGINES
- TABLES
- VIEWS

–Constraints, Indexes

- KEY_COLUMN_USAGE
- REFERENTIAL_CONSTRAINTS
- STATISTICS
- TABLE_CONSTRAINTS

–Partitioning

- PARTITIONS
- FILES (only NDB)

–Privileges

- COLUMN_PRIVILEGES
- SCHEMA_PRIVILEGES
- TABLE_PRIVILEGES
- USER_PRIVILEGES

–Server settings and status

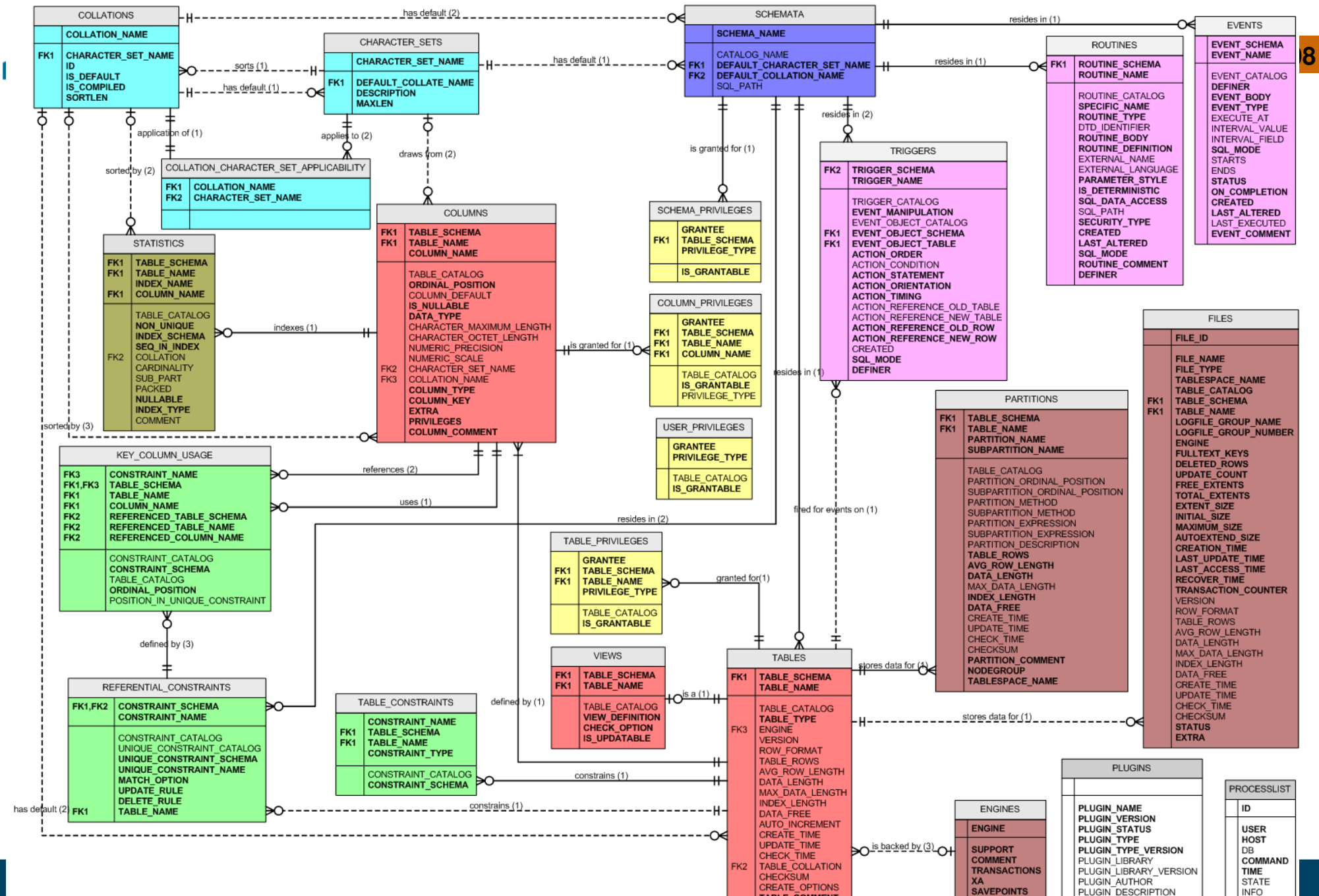
- GLOBAL_STATUS
- GLOBAL_VARIABLES
- PLUGINS
- PROCESSLIST
- SESSION_STATUS
- SESSION_VARIABLES

–Character Set Support

- CHARACTER_SETS
- COLLATIONS
- COLLATION_CHARACTER_SET_APPLICABILITY

–Routines etc.

- EVENTS
- ROUTINES
- TRIGGERS



Schemata:

Field	Type	Null	Key	Default	Extra
CATALOG_NAME	varchar(512)	YES		NULL	
SCHEMA_NAME	varchar(64)	NO			
DEFAULT_CHARACTER_SET_NAME	varchar(64)	NO			
DEFAULT_COLLATION_NAME	varchar(64)	NO			
SQL_PATH	varchar(512)	YES		NULL	

- In MySQL SCHEMA == DATABASE
- 'PRIMARY KEY': SCHEMA_NAME
- CATALOG_NAME: not applicable
 - all %CATALOG% columns are always NULL
- SQL_PATH: not applicable
 - always NULL

Tables:

Field	Type	Null	Key	Default	Extra
TABLE_CATALOG	varchar(512)	YES		NULL	
TABLE_SCHEMA	varchar(64)	NO			
TABLE_NAME	varchar(64)	NO			
TABLE_TYPE	varchar(64)	NO			
ENGINE	varchar(64)	YES		NULL	
VERSION	bigint(21) unsigned	YES		NULL	
ROW_FORMAT	varchar(10)	YES		NULL	
TABLE_ROWS	bigint(21) unsigned	YES		NULL	
AVG_ROW_LENGTH	bigint(21) unsigned	YES		NULL	
DATA_LENGTH	bigint(21) unsigned	YES		NULL	
MAX_DATA_LENGTH	bigint(21) unsigned	YES		NULL	
INDEX_LENGTH	bigint(21) unsigned	YES		NULL	
DATA_FREE	bigint(21) unsigned	YES		NULL	
AUTO_INCREMENT	bigint(21) unsigned	YES		NULL	
CREATE_TIME	datetime	YES		NULL	
UPDATE_TIME	datetime	YES		NULL	
CHECK_TIME	datetime	YES		NULL	
TABLE_COLLATION	varchar(64)	YES		NULL	
CHECKSUM	bigint(21) unsigned	YES		NULL	
CREATE_OPTIONS	varchar(255)	YES		NULL	
TABLE_COMMENT	varchar(80)	NO			

Columns

Field	Type	Null	Key	Default	Extra
TABLE_CATALOG	varchar(512)	YES		NULL	
TABLE_SCHEMA	varchar(64)	NO			
TABLE_NAME	varchar(64)	NO			
COLUMN_NAME	varchar(64)	NO			
ORDINAL_POSITION	bigint(21) unsigned	NO		0	
COLUMN_DEFAULT	longtext	YES		NULL	
IS_NULLABLE	varchar(3)	NO			
DATA_TYPE	varchar(64)	NO			
CHARACTER_MAXIMUM_LENGTH	bigint(21) unsigned	YES		NULL	
CHARACTER_OCTET_LENGTH	bigint(21) unsigned	YES		NULL	
NUMERIC_PRECISION	bigint(21) unsigned	YES		NULL	
NUMERIC_SCALE	bigint(21) unsigned	YES		NULL	
CHARACTER_SET_NAME	varchar(64)	YES		NULL	
COLLATION_NAME	varchar(64)	YES		NULL	
COLUMN_TYPE	longtext	NO		NULL	
COLUMN_KEY	varchar(3)	NO			
EXTRA	varchar(27)	NO			
PRIVILEGES	varchar(80)	NO			
COLUMN_COMMENT	varchar(255)	NO			

Query CREATE TABLE

```

SELECT      CONCAT ( 'CREATE TABLE ', t.TABLE_NAME, ' (\\n '
              , GROUP_CONCAT (
                  c.COLUMN_NAME, ' ', c.COLUMN_TYPE
                  , IF (c.IS_NULLABLE = 'NO', ' NOT NULL', '')
              )
              ORDER BY c.ORDINAL_POSITION
              SEPARATOR '\\n, '
              )
              , '\\n) ENGINE = ', t.ENGINE
              )
FROM        information_schema.TABLES t
INNER JOIN  information_schema.COLUMNS c
USING      (TABLE_SCHEMA, TABLE_NAME)
WHERE      t.TABLE_SCHEMA = SCHEMA ()
AND        t.TABLE_TYPE    = 'BASE TABLE'
GROUP BY   t.TABLE_SCHEMA, t.TABLE_NAME

```

STATISTICS (index columns)

Field	Type	Null	Key	Default	Extra
TABLE_CATALOG	varchar(512)	YES		NULL	
TABLE_SCHEMA	varchar(64)	NO			
TABLE_NAME	varchar(64)	NO			
NON_UNIQUE	bigint(1)	NO		0	
INDEX_SCHEMA	varchar(64)	NO			
INDEX_NAME	varchar(64)	NO			
SEQ_IN_INDEX	bigint(2)	NO		0	
COLUMN_NAME	varchar(64)	NO			
COLLATION	varchar(1)	YES		NULL	
CARDINALITY	bigint(21)	YES		NULL	
SUB_PART	bigint(3)	YES		NULL	
PACKED	varchar(10)	YES		NULL	
NULLABLE	varchar(3)	NO			
INDEX_TYPE	varchar(16)	NO			
COMMENT	varchar(16)	YES		NULL	

STATISTICS -> INDEXES

```
SELECT    TABLE_SCHEMA, TABLE_NAME, INDEX_NAME
,          INDEX_TYPE
,          IF (NON_UNIQUE, 'NO', 'YES')           AS IS_UNIQUE
,          GROUP_CONCAT (
            COLUMN_NAME
            ORDER BY SEQ_IN_INDEX
          )                                       AS COLUMNS
FROM      information_schema.STATISTICS
GROUP BY  TABLE_SCHEMA, TABLE_NAME, INDEX_NAME
```

TABLE_CONSTRAINTS

Field	Type	Null	Key	Default	Extra
CONSTRAINT_CATALOG	varchar(512)	YES		NULL	
CONSTRAINT_SCHEMA	varchar(64)	NO			
CONSTRAINT_NAME	varchar(64)	NO			
TABLE_SCHEMA	varchar(64)	NO			
TABLE_NAME	varchar(64)	NO			
CONSTRAINT_TYPE	varchar(64)	NO			

- CONSTRAINT_TYPE:
 - FOREIGN KEY
 - PRIMARY KEY
 - UNIQUE
- Natural identifier:
 - TABLE_SCHEMA
 - TABLE_NAME
 - CONSTRAINT_TYPE
 - CONSTRAINT_NAME

KEY_COLUMN_USAGE

Field	Type	Null	Key	Default	Extra
CONSTRAINT_CATALOG	varchar(512)	YES		NULL	
CONSTRAINT_SCHEMA	varchar(64)	NO			
CONSTRAINT_NAME	varchar(64)	NO			
TABLE_CATALOG	varchar(512)	YES		NULL	
TABLE_SCHEMA	varchar(64)	NO			
TABLE_NAME	varchar(64)	NO			
COLUMN_NAME	varchar(64)	NO			
ORDINAL_POSITION	bigint(10)	NO		0	
POSITION_IN_UNIQUE_CONSTRAINT	bigint(10)	YES		NULL	
REFERENCED_TABLE_SCHEMA	varchar(64)	YES		NULL	
REFERENCED_TABLE_NAME	varchar(64)	YES		NULL	
REFERENCED_COLUMN_NAME	varchar(64)	YES		NULL	

- Natural identifier???

CONSTRAINT COLUMNS:

SELECT

```

CONCAT (IF (CONSTRAINT_TYPE != 'PRIMARY KEY'
            , CONCAT ('CONSTRAINT ', con.CONSTRAINT_NAME, ' '), '')
, CONSTRAINT_TYPE, ' (' , GROUP_CONCAT (col.COLUMN_NAME ORDER BY ORDINAL_POSITION), ') '
, IF (con.CONSTRAINT_TYPE != 'FOREIGN KEY', ''
      , CONCAT ('REFERENCES ', col.REFERENCED_TABLE_SCHEMA, '.', col.REFERENCED_TABLE_NAME
        , ' (' , GROUP_CONCAT (col.REFERENCED_COLUMN_NAME ORDER BY ORDINAL_POSITION), ') '
      )
    )
)
FROM      information_schema.TABLE_CONSTRAINTS con
INNER JOIN information_schema.KEY_COLUMN_USAGE  col
ON        con.TABLE_SCHEMA      = col.TABLE_SCHEMA
AND       con.TABLE_NAME        = col.TABLE_NAME
AND       con.CONSTRAINT_NAME   = col.CONSTRAINT_NAME
AND ( (    con.CONSTRAINT_TYPE = 'FOREIGN KEY'
          AND col.REFERENCED_TABLE_SCHEMA IS NOT NULL)
      OR (    con.CONSTRAINT_TYPE != 'FOREIGN KEY'
          AND col.REFERENCED_TABLE_SCHEMA IS NULL)
      )
GROUP BY con.TABLE_SCHEMA, con.TABLE_NAME, con.CONSTRAINT_TYPE, con.CONSTRAINT_NAME

```

Some Applications

- Schema documentation
- Query wizard
- Foreign key violation checker
 - <http://forge.mysql.com/tools/tool.php?id=11>
- Emulating foreign key constraints
- Checking for redundant indexes
 - <http://forge.mysql.com/tools/tool.php?id=45>
- Creating FEDERATED tables
 - <http://forge.mysql.com/tools/tool.php?id=54>

Application: FEDERATED tables

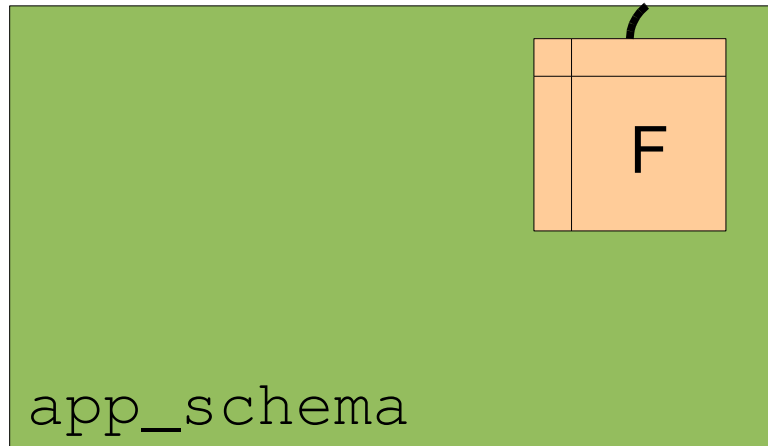
- A FEDERATED table is a 'peephole' to a remote table:
- Local table has same definition as the remote table

```
CREATE TABLE federated_table (  
    ID INT NOT NULL PRIMARY KEY  
) ENGINE = FEDERATED  
CONNECTION='mysql://user@host:3306/federated/test_table'
```

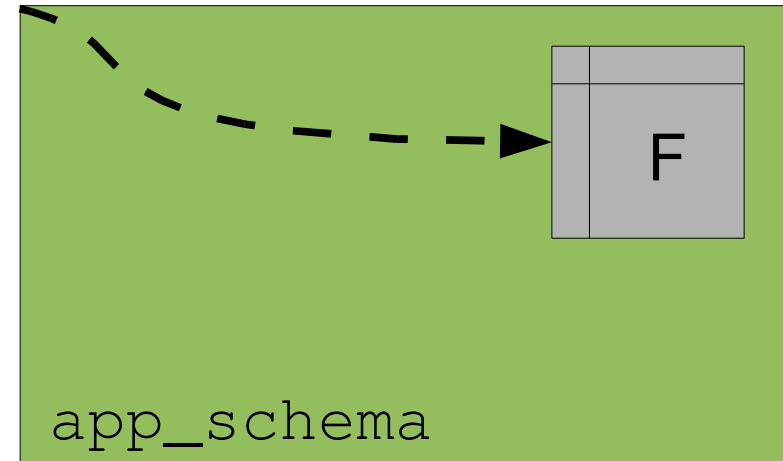
- If we have the remote user, host and port, we can use the local information_schema tables to generate DDL to create TEMPORARY FEDERATED tables of remote information_schema tables
- Using dynamic SQL (PREPARE syntax) we can execute generated DDL and thus actually create these tables
- using these federated information schema tables, we can generate DDL to create any desired federated table

Application: FEDERATED tables

Local

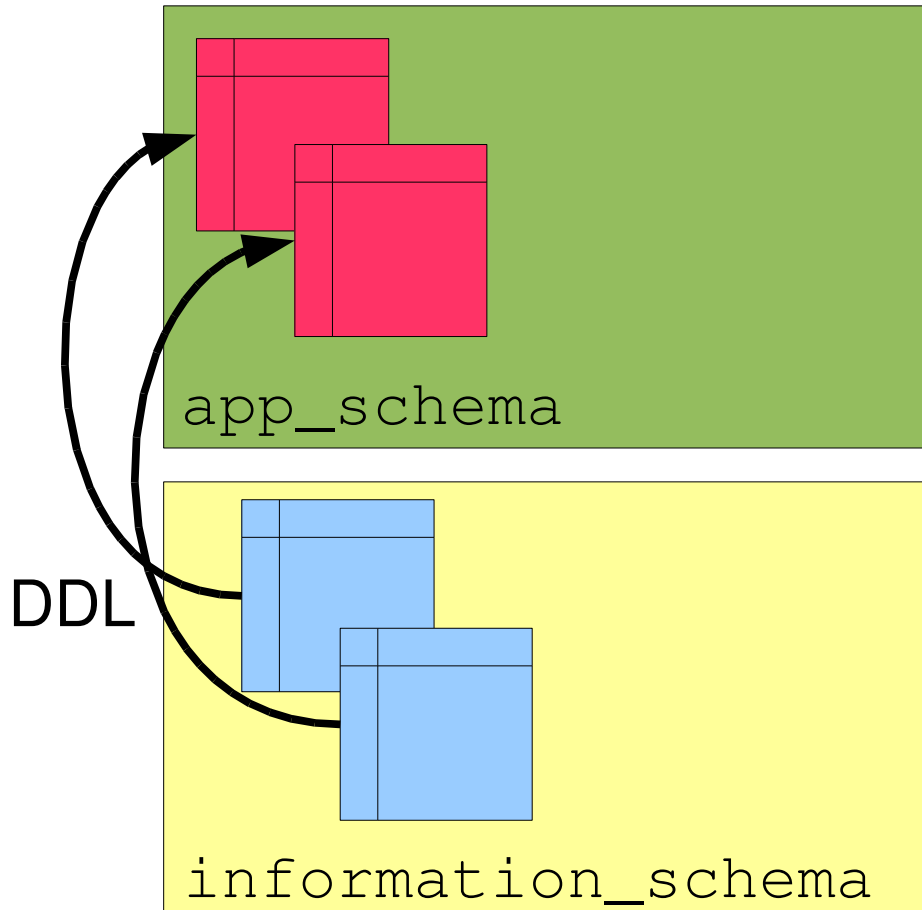


Remote



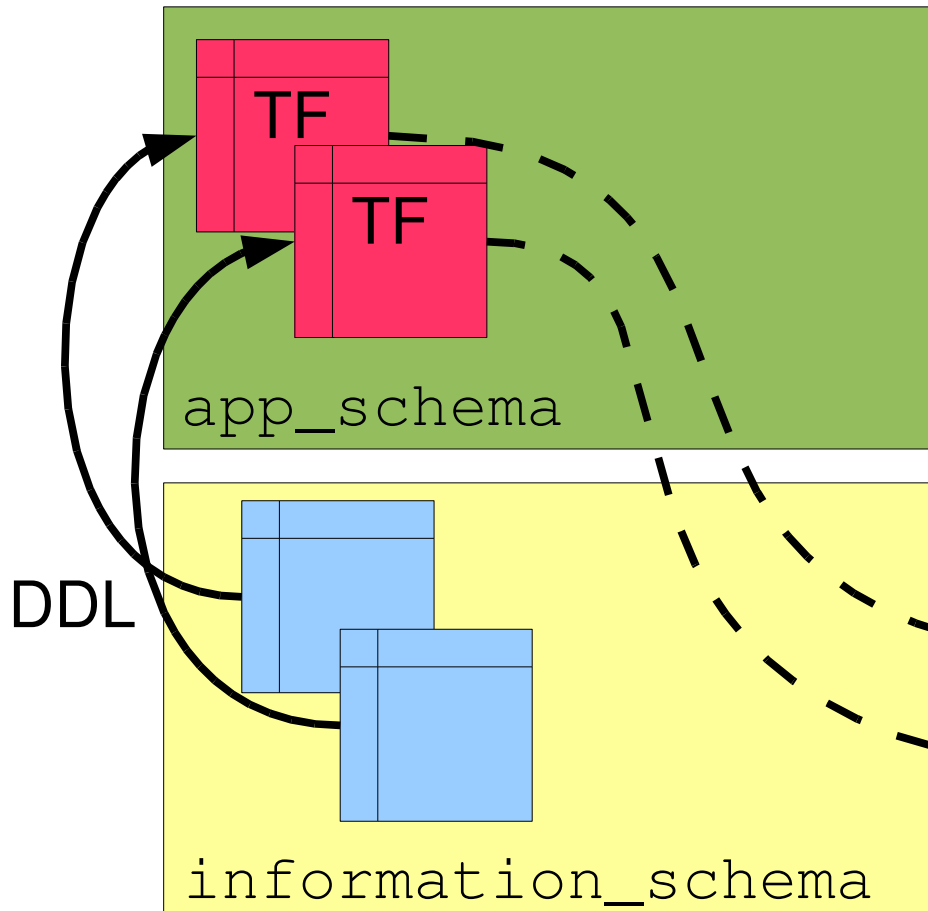
Application: FEDERATED tables

Local

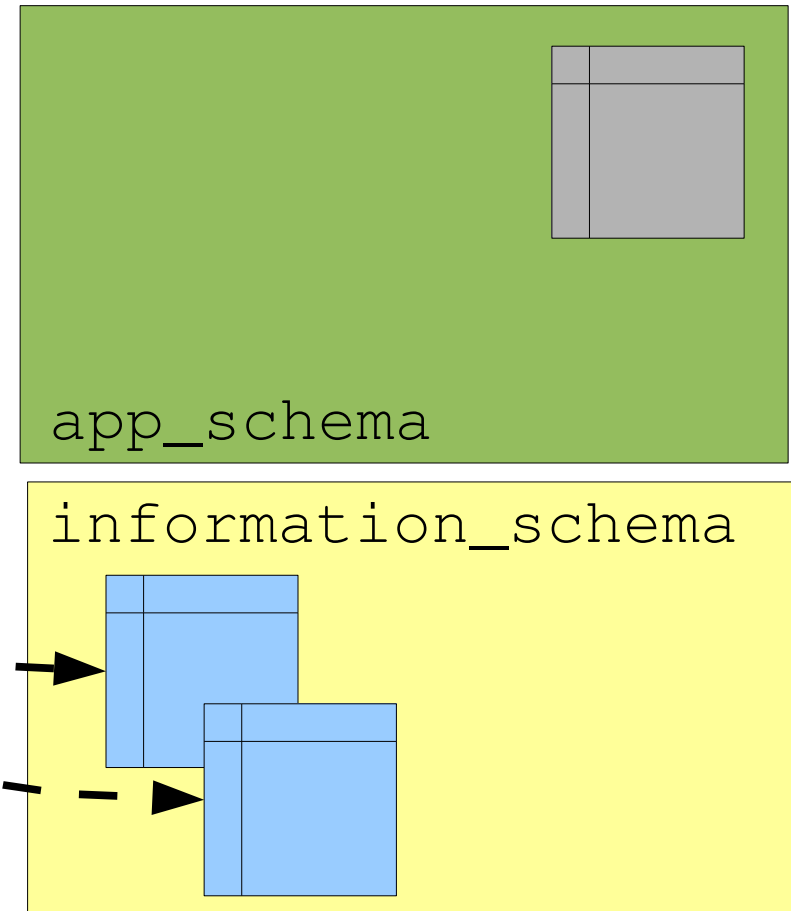


Application: FEDERATED tables

Local

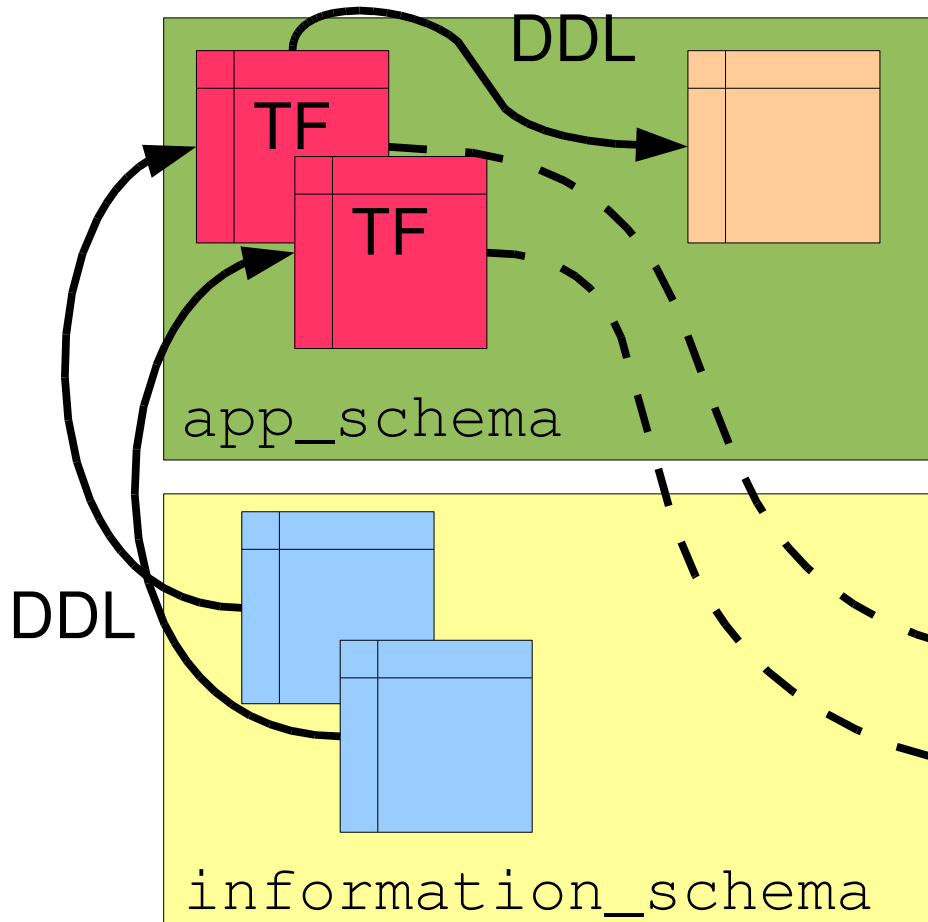


Remote

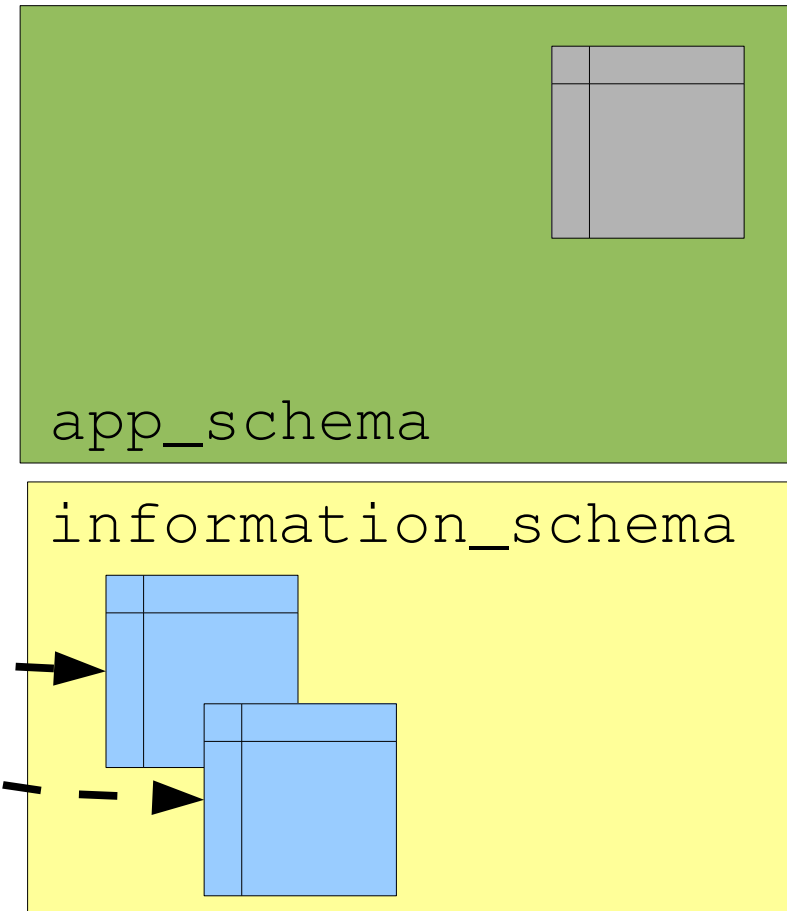


Application: FEDERATED tables

Local

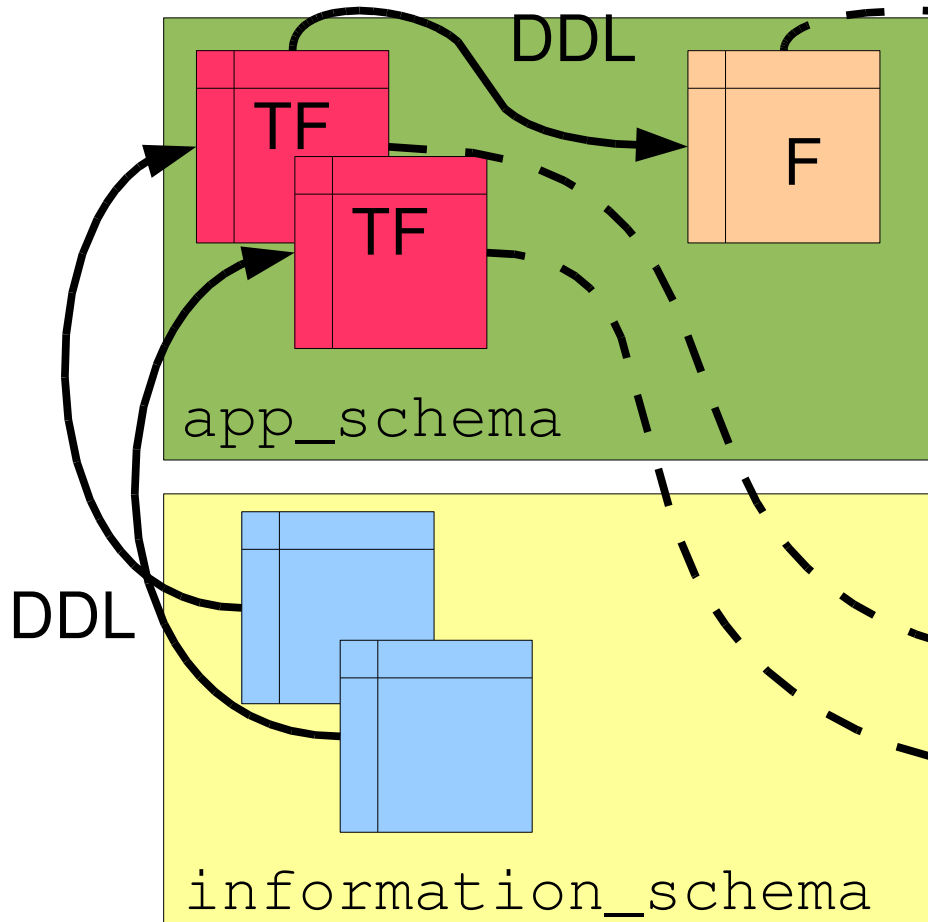


Remote

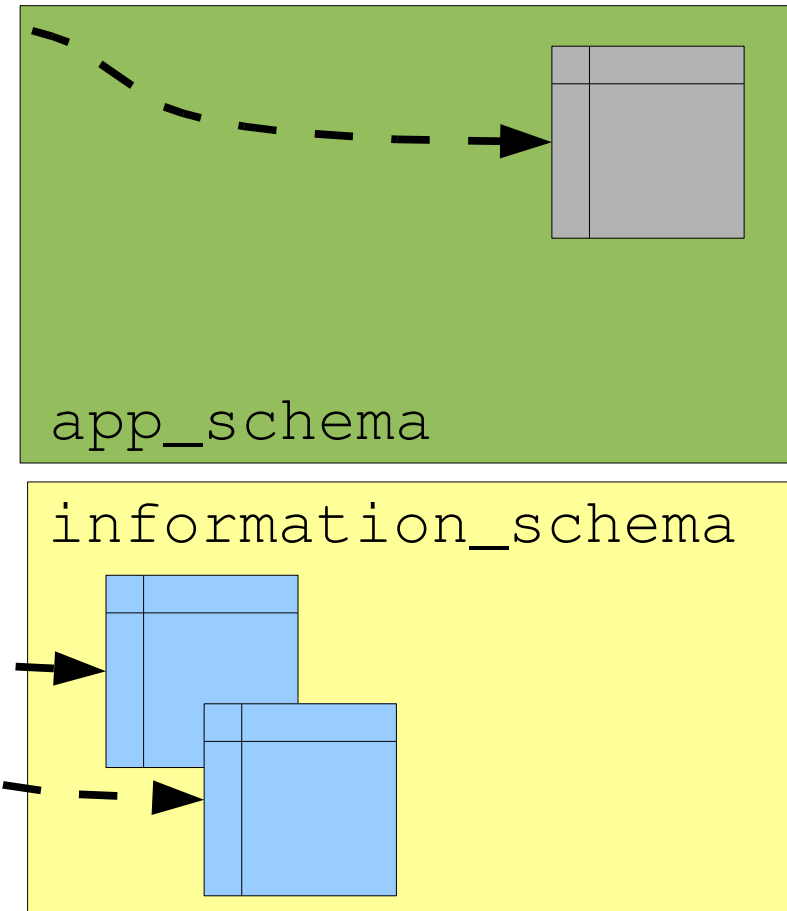


Application: FEDERATED tables

Local



Remote



Application: FEDERATED tables

```
PROCEDURE p_create_federated_table (  
    p_remote_host          VARCHAR (32)  
, p_remote_port          INT UNSIGNED  
, p_remote_user          VARCHAR (16)  
, p_remote_password      VARCHAR (32)  
, p_remote_schema        VARCHAR (64)  
, p_remote_table         VARCHAR (64)  
, p_local_schema         VARCHAR (64)  
, p_local_table          VARCHAR (64)  
)
```

- <http://forge.mysql.com/tools/tool.php?id=54>