# Replicación SymmetricDS en MySql 5.5

## Objetivo

- Implementar mecanismo de replicación basado en SymmetricDS entre dos servidores MySql 5.5

## Requisitos

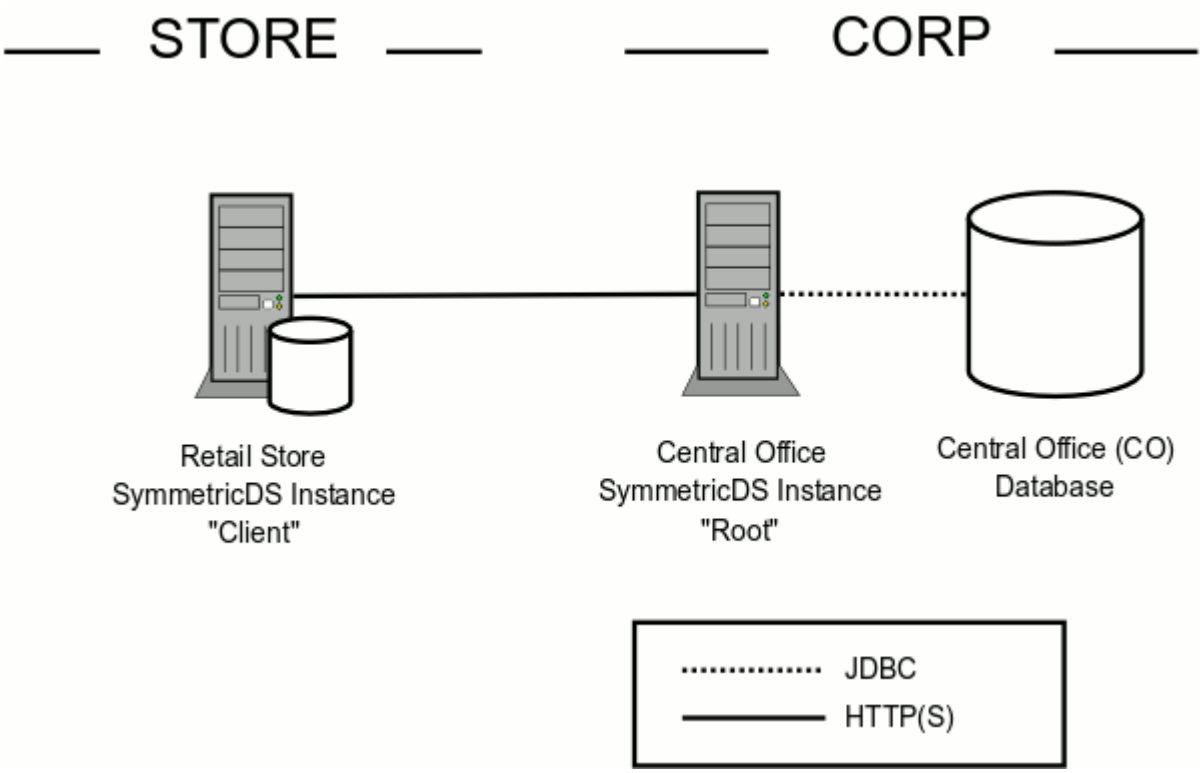
- Lectura y comprensión de los apuntes [2], [3]
- Comprender el concepto de log de transacciones, técnicas de backup y recuperación.
- Que el alumno cuente con conocimientos básicos de sistemas operativos Unix / Linux: file system, network file system, shell scripts, etc.
- Contar con servidor Mysql versión 5.5 instalado con usuario y password de administrador de la base de datos.
- Acceso a máquina Linux Debian amd64 funcionando, no se requiere usuario administrador (root) para realizar tareas administrativas en el equipo o cambios de configuración[1].
- Java JRE instalado en equipo, este tutorial fue probado sobre java versión "1.7.0_25", Java(TM) SE Runtime Environment (build 1.7.0_25-b15), Java HotSpot(TM) 64-Bit Server VM (build 23.25-b01, mixed mode)

## Introducción

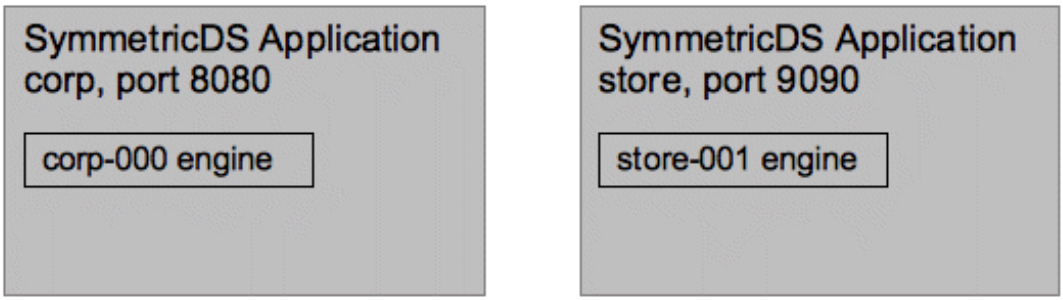
SymmetricDS (SDS) es un producto basado en Java que permite la replicación de distintas bases de datos (mientras se posea una driver JDBC para conectarlas), ha sido probado en bases de datos tables como: Oracle, MySQL, MariaDB, PostgreSQL, MS SQL Server (including Azure), IBM DB2, H2, HSQLDB, Derby, Firebird, Interbase, Informix, Greenplum, SQLite (including Android), Sybase ASE, and Sybase ASA (SQL Anywhere), etc. El producto es open-source, permite una replicación multi-master (varios servidores maestros), replicación asincrónica, múltiples suscriptores en forma uni-direccional o bi-direccional. Utiliza tecnologías web y de base de datos para replicación de datos cercanos al tiempo real. Es un software pensado para replicar en multiples nodos, incluso a través de redes con escaso ancho de banda.

El siguiente ejemplo, implementado a partir de [1], trata sobre la sincronización de dos bases de datos mysql con esquemas similares entre dos nodos SDS. Un nodo representa a la oficina central de una empresa de ventas de productos, denominada nodo root o corp y que posee múltiples sucursales denominadas nodo “client” o “store”. Para facilitar la práctica, se trabajará con una sola sucursal y se simulará, tanto la oficina central como la sucursal en un mismo servidor, acorde como se indica en la figura:

1 A lo sumo para iniciar y detener el servicio de mysql.




Se instalan 2 copias separadas de SDS que representan 2 diferentes servers (store server y corp server). Cada SDS se encarga de una base de datos y cada copia actúa como un simple nodo SDS. Esta es la configuración mas simple, también puede hacerse una configuración en donde 1 instancia de SDS se encargue de 2 base de datos, esto se llama "multi-homing". Para probar esto en una misma pc, se correrán 2 copias de SDS en distintos puertos tcp/ip (8080 para corp server y 9090 para store server):




La aplicación captura los cambios en los productos (numero, descripción, precio, etc) en cada sucursal (en este caso store server, store-001 en puerto 9090) se capturan los cambios que se hacen en las transacciones de ventas hechas en el corp server (corp-000, puerto 8080),

tales como la fecha-hora de la venta, los productos vendidos. El precio de los productos es enviado solo al store server únicamente cuando se trate de productos de dicho store.

La configuración de ejemplo hace que el cliente (store server) siempre inicie la comunicación con root (corp server). El cliente conectará al servidor root en forma periódica para traer datos del servidor y el cliente también enviará al servidor root los cambios que captura en los productos cuando éstos estén disponibles.

# 1 Instalación

Asegurarse de tener instalados los siguientes paquetes de software:

```
$ aptitude install openjdk-7-jre[2]
$ aptitude install mysql-server-5.5[3]
$ aptitude install mysql-server-core-5.5
$ aptitude install mysql-common
$ aptitude install mysql-client-5.5
$ aptitude install mysql-gui-tools-common
```

-En ningún momento se utilizó el usuario root, pues se trata de descomprimir archivos en area de usuario linux sin privilegios adicionales y ejecución de una aplicación Java

-Descargar de http://www.symmetricds.org/download Plataforma Server/Desktop symmetric-3.6.11-server.zip

-Crear 2 directorios para representar los 2 servidores: sym-corp (corp server, root) y sym-store001 (store server, client). Los nombre de los directorios son arbitrarios, puede usar los que Ud desee, pero recuerde identificar a cada server.

-Descomprimir symmetric-3.6.11-server.zip en ambos directorios (en mi caso, descomprimí a partir del contenido de la carpeta symmetric-3.6.11 para el que path no sea tan extenso[4]).

-Copiamos los archivos de configuración de ejemplo que vienen con SDS par facilitar la configuración a los directorios de ambos servidores (estando parado en el directorio padre de sym-corp y sym-store001), hacer:

```
$ cp sym-corp/samples/corp-000.properties sym-corp/engines
$ cp sym-corp/samples/store-001.properties sym-store001/engines
```

2 También es posible tener instalado el paquete openjdk-7-jdk o versión superior.

3 Podría ser también una versión superior, se debe contar con el usuario administrador de mysql, en este caso, el usuario se llama root y password nbuser . No confundir este usario mysql con el usuario administrador de linux.

4 Atencion! En mi caso no se descomprimieron los sub-directorios /engines, /logs,/patches,/tmp que están dentro de los directorios sym-corp y sym-store001. Esos directorios están vacíos, crearlos en forma manual antes de avanzar al paso siguiente.

-Utilice un editor de texto y edite ambos archivos de configuración desde el sub-directorio engines: corp-000.properties y store-001.properties; observe que en el primer archivo el external id es 000 (external.id=000) y en el segundo el external id es 001 (external.id=0001)[5]

-Modifique ambos archivos de la siguiente forma para configurarlos, de forma tal, que el servidor corp utilice la base de datos mysql corp y el servidor store utilice la base de datos mysql store001. Ambas bases de datos creadas en el servidor local (localhost) mysql 5.5 con usuario administrador mysql root y password nbuser (en mi caso):

ejemplo para archivo corp-000.properties[6]:

```
# The class name for the JDBC Driver
db.driver=com.mysql.jdbc.Driver

# The JDBC URL used to connect to the database
db.url=jdbc:mysql://localhost/corp?tinyInt1isBit=false

# The user to login as who can create and update tables
db.user=root

# The password for the user to login as
db.password=nbuser
```

ejemplo para archivo store-001.properties, aquí se agrega además información para que el cliente pueda contactar al servidor corp:

```
# The class name for the JDBC Driver
db.driver=com.mysql.jdbc.Driver

# The JDBC URL used to connect to the database
db.url=jdbc:mysql://localhost/store001?tinyInt1isBit=false

# The user to login as who can create and update tables
db.user=root

# The password for the user to login as
db.password=nbuser

# The HTTP URL of the root node to contact for registration
registration.url=http://localhost:8080/sync/corp-000[7]
```

5 Linea de texto 65 en ambos archivos.

6 Atencion!! verifique cuidadosamente en ambos archivos que todas las lineas de conexiones a bases de datos/url's/etc estén comentadas, pues sino el comando dbimport fallará, verifique los mensajes que el mismo emite (se resaltarán en negritas, los mensajes que deben hacer referencia a una base de datos mysql y no otra, como sucedió en mi caso).

7 El formato general para la url SDS es http://{hostname}:{port}/sync/{engine.name}

# 2 Creación de bases de datos para replicación

-creamos 2 bases de datos mysql para corp y store (acorde con la url a las bd's indicada en el paso anterior):

```
$ mysql -u root -p
Enter password: nbuser
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 45
Server version: 5.5.40-0+wheezy1 (Debian)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights
reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

mysql> create database corp;
Query OK, 1 row affected (0.05 sec)

mysql> create database store001;
Query OK, 1 row affected (0.01 sec)

mysql> exit;
Bye
grchere@debian2:~/UNLu/11078/apuntes/unidad.iv/symmetricds$
```

## 2.1 Creando las tablas y los datos de las bases de datos

-Abrir una consola en el directorio sym-corp/sample y ejecutar el siguiente comando para crear las tablas de ventas, productos, precios, etc.:

```
$../bin/dbimport --engine corp-000 --format XML
/home/grchere/UNLu/11078/apuntes/unidad.iv/symmetricds/sym-
corp/samples/create_sample.xml [8]
```

la salida del comando, puede ser algo similar a esto (se remarca en negrita los mensajes que considero importantes para evitar errores que no estan indicados en [1], cualquier referencia a otra base de datos que no sea mysql, seguramente se trata de alguna linea no comentada dentro del archivo corp-000.properties):

```
Log output will be written to ../logs/symmetric.log
[] - AbstractCommandLauncher - Option: name=engine, value={corp-000}
```

8 El comando dbimport no localiza al archivo create_sample.xml que se encuentra en el directorio actual, para resolver ese problema se tuvo que indicar la ruta (path) completa al mismo. También es posible que deba indicar el switch –force al comando dbimport en caso de que no cree las tablas, de la forma: ../bin/dbimport –force … .

```
[] - AbstractCommandLauncher - Option: name=format, value={XML}
[] - JdbcDatabasePlatformFactory - Detected database 'MySQL', version '5',
protocol 'mysql'
[] - AbstractDatabaseWriter - Did not find the item table in the target
database
[] - DefaultDatabaseWriter - About to create table using the following
definition: <?xml version="1.0"?>
<!DOCTYPE database SYSTEM "http://db.apache.org/torque/dtd/database">
<database name="dbimport">
      <table name="item">
            <column name="item_id" primaryKey="true" required="true"
type="INTEGER"/>
            <column name="name" type="VARCHAR" size="100"/>
      </table>
</database>
[] - AbstractDatabaseWriter - Did not find the item_selling_price table in
the target database
[] - DefaultDatabaseWriter - About to create table using the following
definition: <?xml version="1.0"?>
<!DOCTYPE database SYSTEM "http://db.apache.org/torque/dtd/database">
<database name="dbimport">
      <table name="item_selling_price">
            <column name="item_id" primaryKey="true" required="true"
type="INTEGER"/>
            <column name="store_id" primaryKey="true" required="true"
type="VARCHAR" size="5"/>
            <column name="price" required="true" type="DECIMAL"
size="10,2"/>
            <column name="cost" type="DECIMAL" size="10,2"/>
            <foreign-key name="fk_price_item_id" foreignTable="item">
                  <reference local="item_id" foreign="item_id"/>
            </foreign-key>
      </table>
</database>
[] - AbstractDatabaseWriter - Did not find the sale_transaction table in
the target database
[] - DefaultDatabaseWriter - About to create table using the following
definition: <?xml version="1.0"?>
<!DOCTYPE database SYSTEM "http://db.apache.org/torque/dtd/database">
<database name="dbimport">
      <table name="sale_transaction">
            <column name="tran_id" primaryKey="true" required="true"
type="INTEGER"/>
            <column name="store_id" required="true" type="VARCHAR"
size="5"/>
            <column name="workstation" required="true" type="VARCHAR"
size="3"/>
            <column name="day" required="true" type="VARCHAR" size="10"/>
            <column name="seq" required="true" type="INTEGER"/>
      </table>
</database>
[] - AbstractDatabaseWriter - Did not find the sale_return_line_item table
in the target database
[] - DefaultDatabaseWriter - About to create table using the following
definition: <?xml version="1.0"?>
<!DOCTYPE database SYSTEM "http://db.apache.org/torque/dtd/database">
```

```
<database name="dbimport">
      <table name="sale_return_line_item">
            <column name="tran_id" primaryKey="true" required="true"
type="INTEGER"/>
            <column name="item_id" required="true" type="INTEGER"/>
            <column name="price" required="true" type="DECIMAL"
size="10,2"/>
            <column name="quantity" required="true" type="INTEGER"/>
            <column name="returned_quantity" type="INTEGER"/>
            <foreign-key name="fk_srli_tran_id"
foreignTable="sale_transaction">
                  <reference local="tran_id" foreign="tran_id"/>
            </foreign-key>
            <foreign-key name="fk_srli_item_id" foreignTable="item">
                  <reference local="item_id" foreign="item_id"/>
            </foreign-key>
      </table>
</database>
grchere@debian2:~/UNLu/11078/apuntes/unidad.iv/symmetricds/sym-
corp/samples$
```

- comprobamos que el comando anterior haya creado las tablas correspondientes dentro de la base de datos mysql corp:

```
$  mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 61
Server version: 5.5.40-0+wheezy1 (Debian)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights
reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| affablebean        |
| corp               |
| mysql              |
| performance_schema |
| store001           |
| test               |
+--------------------+
7 rows in set (0.00 sec)
```

```
mysql> use corp;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----------------------+
| Tables_in_corp        |
+-----------------------+
| item                  |
| item_selling_price    |
| sale_return_line_item |
| sale_transaction      |
+-----------------------+
4 rows in set (0.00 sec)

mysql> describe item;
+---------+--------------+------+-----+---------+-------+
| Field   | Type         | Null | Key | Default | Extra |
+---------+--------------+------+-----+---------+-------+
| item_id | int(11)      | NO   | PRI | NULL    |       |
| name    | varchar(100) | YES  |     | NULL    |       |
+---------+--------------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql> describe item_selling_price;
+----------+---------------+------+-----+---------+-------+
| Field    | Type          | Null | Key | Default | Extra |
+----------+---------------+------+-----+---------+-------+
| item_id  | int(11)       | NO   | PRI | NULL    |       |
| store_id | varchar(5)    | NO   | PRI | NULL    |       |
| price    | decimal(10,2) | NO   |     | NULL    |       |
| cost     | decimal(10,2) | YES  |     | NULL    |       |
+----------+---------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> describe sale_return_line_item;
+-------------------+---------------+------+-----+---------+-------+
| Field             | Type          | Null | Key | Default | Extra |
+-------------------+---------------+------+-----+---------+-------+
| tran_id           | int(11)       | NO   | PRI | NULL    |       |
| item_id           | int(11)       | NO   | MUL | NULL    |       |
| price             | decimal(10,2) | NO   |     | NULL    |       |
| quantity          | int(11)       | NO   |     | NULL    |       |
| returned_quantity | int(11)       | YES  |     | NULL    |       |
+-------------------+---------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> describe sale_transaction;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| tran_id     | int(11)     | NO   | PRI | NULL    |       |
| store_id    | varchar(5)  | NO   |     | NULL    |       |
| workstation | varchar(3)  | NO   |     | NULL    |       |
| day         | varchar(10) | NO   |     | NULL    |       |
```

```
| seq         | int(11)     | NO   |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
5 rows in set (0.01 sec)

mysql> exit;
Bye
root@debian2:/home/grchere#
```

- Ahora creamos las tablas especificas de SDS que guardan la información de configuración para sincronización, dentro del mismo directorio sym-corp/samples, hacemos:

```
$ ../bin/symadmin --engine corp-000 create-sym-tables
```

la salida del comando es muy extensa, algo como esto:

```
….
[corp-000] - MySqlSymmetricDialect - DDL applied: CREATE TABLE
`sym_trigger_router_grouplet`(
    `grouplet_id` VARCHAR(50) NOT NULL,
    `trigger_id` VARCHAR(128) NOT NULL,
    `router_id` VARCHAR(50) NOT NULL,
    `applies_when` CHAR(1) NOT NULL,
    `create_time` DATETIME NOT NULL,
    `last_update_by` VARCHAR(50) NULL,
    `last_update_time` DATETIME NOT NULL,
    PRIMARY KEY (`grouplet_id`, `trigger_id`, `router_id`, `applies_when`)
)
[corp-000] - MySqlSymmetricDialect - DDL applied: ALTER TABLE
`sym_conflict`
ADD CONSTRAINT `sym_fk_cf_2_grp_lnk` FOREIGN KEY (`source_node_group_id`,
`target_node_group_id`) REFERENCES `sym_node_group_link`
(`source_node_group_id`, `target_node_group_id`)
[corp-000] - MySqlSymmetricDialect - DDL applied: ALTER TABLE
`sym_file_trigger_router`

….

[corp-000] - MySqlSymmetricDialect - Just installed
sym_transaction_id_post_5_1_23
[corp-000] - ConfigurationService - Auto-configuring config channel
[corp-000] - ConfigurationService - Auto-configuring reload channel
[corp-000] - ConfigurationService - Auto-configuring heartbeat channel
[corp-000] - ConfigurationService - Auto-configuring default channel
[corp-000] - ConfigurationService - Auto-configuring dynamic channel
[corp-000] - ConfigurationService - Auto-configuring filesync channel
[corp-000] - ConfigurationService - Auto-configuring filesync_reload
channel
[corp-000] - AbstractSymmetricEngine - Done initializing SymmetricDS
database

grchere@debian2:~/UNLu/11078/apuntes/unidad.iv/symmetricds/sym-
corp/samples$
```

- Cargamos los datos ejemplo de productos y transacciones en el servidor root (bd mysql corp), haciendo (siempre sobre directorio sym-corp/samples)[9]:

```
$ ../bin/dbimport --engine corp-000
/home/grchere/UNLu/11078/apuntes/unidad.iv/symmetricds/sym-
corp/samples/insert_sample_mysql.sql
```

la salida del comando, es algo como esto:

```
Log output will be written to ../logs/symmetric.log
[] - AbstractCommandLauncher - Option: name=engine, value={corp-000}
[] - JdbcDatabasePlatformFactory - Detected database 'MySQL', version '5',
protocol 'mysql'
$
```

-Ya esta lista la bd corp, servidor root; ahora debemos crear las tablas de ejemplo en la bd store, servidor cliente para que éste reciba los datos que provienen del servidor root. Ahora nos paramos dentro del sub-directorio /sym-store001/samples, abrimos una consola allí y hacemos el siguiente comando para crear las tablas de ejemplo vacías[10]:

```
$ ../bin/dbimport --engine store-001 --format XML
/home/grchere/UNLu/11078/apuntes/unidad.iv/symmetricds/sym-
store001/samples/create_sample.xml
```

la salida del comando es muy extensa, algo como esto:

```
Log output will be written to ../logs/symmetric.log
[] - AbstractCommandLauncher - Option: name=engine, value={store-001}
[] - AbstractCommandLauncher - Option: name=format, value={XML}
[] - JdbcDatabasePlatformFactory - Detected database 'MySQL', version '5',
protocol 'mysql'
[] - AbstractDatabaseWriter - Did not find the item table in the target
database
[] - DefaultDatabaseWriter - About to create table using the following
definition: <?xml version="1.0"?>
<!DOCTYPE database SYSTEM "http://db.apache.org/torque/dtd/database">
<database name="dbimport">
      <table name="item">
            <column name="item_id" primaryKey="true" required="true"
type="INTEGER"/>
            <column name="name" type="VARCHAR" size="100"/>
```

9 Para otras bases de datos que no sean mysql, se puede usar el script insert_sample.sql, debido a que mysql tienen una sintaxis diferente. Nuevamente, debemos indicar el path completo al script sql para que funcione dbimport.

10 Atencion!! antes chequee nuevamente el archivo store-001.properties que no contenga ninguna linea descomentada que no corresponda; por ejemplo en mi caso, estaba descomentada la linea 37: db.driver=org.h2.Driver y ello provoca error en este comando. Nuevamente, también puede agregar el switch –force en caso de error, como se explicó antes.

```
      </table>
</database>
[] - AbstractDatabaseWriter - Did not find the item_selling_price table in
the target database
[] - DefaultDatabaseWriter - About to create table using the following
definition: <?xml version="1.0"?>
<!DOCTYPE database SYSTEM "http://db.apache.org/torque/dtd/database">
<database name="dbimport">
      <table name="item_selling_price">
            <column name="item_id" primaryKey="true" required="true"
type="INTEGER"/>
            <column name="store_id" primaryKey="true" required="true"
type="VARCHAR" size="5"/>
            <column name="price" required="true" type="DECIMAL"
size="10,2"/>
            <column name="cost" type="DECIMAL" size="10,2"/>
            <foreign-key name="fk_price_item_id" foreignTable="item">
                  <reference local="item_id" foreign="item_id"/>
            </foreign-key>
      </table>
</database>
[] - AbstractDatabaseWriter - Did not find the sale_transaction table in
the target database
[] - DefaultDatabaseWriter - About to create table using the following
definition: <?xml version="1.0"?>
<!DOCTYPE database SYSTEM "http://db.apache.org/torque/dtd/database">
<database name="dbimport">
      <table name="sale_transaction">
            <column name="tran_id" primaryKey="true" required="true"
type="INTEGER"/>
            <column name="store_id" required="true" type="VARCHAR"
size="5"/>
            <column name="workstation" required="true" type="VARCHAR"
size="3"/>
            <column name="day" required="true" type="VARCHAR" size="10"/>
            <column name="seq" required="true" type="INTEGER"/>
      </table>
</database>
[] - AbstractDatabaseWriter - Did not find the sale_return_line_item table
in the target database
[] - DefaultDatabaseWriter - About to create table using the following
definition: <?xml version="1.0"?>
<!DOCTYPE database SYSTEM "http://db.apache.org/torque/dtd/database">
<database name="dbimport">
      <table name="sale_return_line_item">
            <column name="tran_id" primaryKey="true" required="true"
type="INTEGER"/>
            <column name="item_id" required="true" type="INTEGER"/>
            <column name="price" required="true" type="DECIMAL"
size="10,2"/>
            <column name="quantity" required="true" type="INTEGER"/>
            <column name="returned_quantity" type="INTEGER"/>
            <foreign-key name="fk_srli_tran_id"
foreignTable="sale_transaction">
                  <reference local="tran_id" foreign="tran_id"/>
            </foreign-key>
```

```
            <foreign-key name="fk_srli_item_id" foreignTable="item">
                  <reference local="item_id" foreign="item_id"/>
            </foreign-key>
      </table>
</database>
grchere@debian2:~/UNLu/11078/apuntes/unidad.iv/symmetricds/sym-
store001/samples$
```

- comprobamos que el comando anterior haya creado las tablas correspondientes dentro de la base de datos mysql store001:

```
$mysql -u root -p
Enter password: XXXXXX

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 102
Server version: 5.5.40-0+wheezy1 (Debian)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights
reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| affablebean        |
| corp               |
| mysql              |
| performance_schema |
| store001           |
| test               |
+--------------------+
7 rows in set (0.00 sec)

mysql> use store001
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----------------------+
| Tables_in_store001    |
+-----------------------+
| item                  |
| item_selling_price    |
| sale_return_line_item |
```

```
| sale_transaction      |
+-----------------------+
4 rows in set (0.00 sec)

mysql> describe item;
+---------+--------------+------+-----+---------+-------+
| Field   | Type         | Null | Key | Default | Extra |
+---------+--------------+------+-----+---------+-------+
| item_id | int(11)      | NO   | PRI | NULL    |       |
| name    | varchar(100) | YES  |     | NULL    |       |
+---------+--------------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql> describe item_selling_price;
+----------+---------------+------+-----+---------+-------+
| Field    | Type          | Null | Key | Default | Extra |
+----------+---------------+------+-----+---------+-------+
| item_id  | int(11)       | NO   | PRI | NULL    |       |
| store_id | varchar(5)    | NO   | PRI | NULL    |       |
| price    | decimal(10,2) | NO   |     | NULL    |       |
| cost     | decimal(10,2) | YES  |     | NULL    |       |
+----------+---------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> describe sale_return_line_item;
+-------------------+---------------+------+-----+---------+-------+
| Field             | Type          | Null | Key | Default | Extra |
+-------------------+---------------+------+-----+---------+-------+
| tran_id           | int(11)       | NO   | PRI | NULL    |       |
| item_id           | int(11)       | NO   | MUL | NULL    |       |
| price             | decimal(10,2) | NO   |     | NULL    |       |
| quantity          | int(11)       | NO   |     | NULL    |       |
| returned_quantity | int(11)       | YES  |     | NULL    |       |
+-------------------+---------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> describe sale_transaction;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| tran_id     | int(11)     | NO   | PRI | NULL    |       |
| store_id    | varchar(5)  | NO   |     | NULL    |       |
| workstation | varchar(3)  | NO   |     | NULL    |       |
| day         | varchar(10) | NO   |     | NULL    |       |
| seq         | int(11)     | NO   |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> exit;
Bye
$
```

# 3 Arrancando SymmetricDS

Se arrancarán ambos nodos SDS y se observará la salida en el log.

-Abrir dos consolas, una en cada directorio de cada nodo ( /sym-corp/samples y /sym-store001/samples)

-En la consola abierta en /sym-corp/samples, hacer:

```
$ ../bin/sym --engine corp-000 --port 8080
```

la salida del comando es muy extensa, similar a esto:

```
../bin/sym --engine corp-000 --port 8080
Log output will be written to ../logs/symmetric.log
[startup] - AbstractCommandLauncher - Option: name=engine, value={corp-000}
[startup] - AbstractCommandLauncher - Option: name=port, value={8080}
[startup] - SymmetricWebServer - About to start SymmetricDS web server on
host:port default:8080
[startup] - / - Initializing Spring root WebApplicationContext
[corp-000] - JdbcDatabasePlatformFactory - Detected database 'MySQL',
version '5', protocol 'mysql'
[corp-000] - MySqlSymmetricDialect - The DbDialect being used is
org.jumpmind.symmetric.db.mysql.MySqlSymmetricDialect
….

[corp-000] - TriggerRouterService - Synchronizing triggers
[corp-000] - MySqlSymmetricDialect - Creating SYM_ON_I_FOR_SYM_LD_FLTR_CRP
trigger for corp.sym_load_filter
…

[corp-000] - DataGapPurgeJob - Starting job.purge.datagaps with cron
expression: 0 0 0 * * *
[corp-000] - StatisticFlushJob - Starting job.stat.flush with cron
expression: 0 0/5 * * * *
[corp-000] - SyncTriggersJob - Starting job.synctriggers with cron
expression: 0 0 0 * * *
[corp-000] - HeartbeatJob - Starting job.heartbeat on periodic schedule:
every 900000ms with the first run at Tue Oct 28 14:07:38 ART 2014
```

el proceso crea todos los triggers necesarios para la replicación acorde con la configuración y queda a la espera en el puerto 8080 para realizar sincronización.

- En la consola abierta en /sym-store001/samples, hacer:

```
$ ../bin/sym --engine store-001 --port 9090
```

la salida del comando es muy extensa, crea triggers, tablas SDS, comienza a hacer pooling del servidor root intentando su registro, pero como aun no esta abierto el registro, da un error de autorización (HTTP 403).

## 4 Registrando un nodo

El nodo que registra es el nodo root (bd corp), debemos abrir la registración para el nodo cliente (bd store001) para que se pueda enviar y recibir datos desde el nodo root.

-Abrir una nueva terminal en la carpeta del nodo root, /sym-corp/samples, hacer el siguiente comando para realizar el registro del nodo store-001:

```
$ ../bin/symadmin --engine corp-000 open-registration store 001
```

la salida del comando es similar a esto:

```
Log output will be written to ../logs/symmetric.log
[] - AbstractCommandLauncher - Option: name=engine, value={corp-000}
[corp-000] - JdbcDatabasePlatformFactory - Detected database 'MySQL',
version '5', protocol 'mysql'
[corp-000] - MySqlSymmetricDialect - The DbDialect being used is
org.jumpmind.symmetric.db.mysql.MySqlSymmetricDialect
[corp-000] - StagingManager - The staging directory was initialized at the
following location:
/home/grchere/UNLu/11078/apuntes/unidad.iv/symmetricds/sym-corp/tmp/corp-
000
[corp-000] - ClusterService - This node picked a server id of debian2
[corp-000] - ExtensionPointManager - Found 9 extension points that will be
registered
[corp-000] - RegistrationService - Just opened registration for external id
of 001 and a node group of store and a node id of 001
Opened registration for node group of 'store' external ID of '001'
```

- Si abre el archivo log del nodo store-001 ubicado en /sym-store001/logs/symmetric.log , deberá encontrar una linea similar a esta:

```
2014-10-28 14:23:37,878 INFO [store-001] [RegistrationService] [store-001-
job-3] Successfully registered node [id=001]
```

que indica el registro satisfactorio del nodo.

## 5 Enviando la carga inicial de datos

Se enviará la carga inicial de datos al nodo store-001, ejecutando un comando administrativo desde el nodo corp, utilizando una terminal abierta sobre el nodo corp, en sub-directorio samples, hacer:

```
$ ../bin/symadmin --engine corp-000 reload-node 001
```

la salida del comando es similar a esto:

```
Log output will be written to ../logs/symmetric.log
[] - AbstractCommandLauncher - Option: name=engine, value={corp-000}
[corp-000] - JdbcDatabasePlatformFactory - Detected database 'MySQL',
version '5', protocol 'mysql'
[corp-000] - MySqlSymmetricDialect - The DbDialect being used is
org.jumpmind.symmetric.db.mysql.MySqlSymmetricDialect
[corp-000] - StagingManager - The staging directory was initialized at the
following location:
/home/grchere/UNLu/11078/apuntes/unidad.iv/symmetricds/sym-corp/tmp/corp-
000
[corp-000] - ClusterService - This node picked a server id of debian2
[corp-000] - ExtensionPointManager - Found 9 extension points that will be
registered
Successfully enabled initial load for node 001
```

los archivos logs de ambos servidores se actualizan indicando la transferencia de datos realizada.

# 6 Modificando Datos y probando Replicación

## 6.1 Pulling data

Vamos a cambiar datos en la base de datos corp de la oficina central para observar si dichos datos se replican al nodo store-001. Para ello abriremos una sesión de sql interactivo con mysql y ejecutaremos algunos comandos para insertar un item, con distinto precio (en store 001 y store 002) como se indica a continuación:

```
$ mysql -u root -p
Enter password: XXXXXXX
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 124
Server version: 5.5.40-0+wheezy1 (Debian)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights
reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

mysql> use corp;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> insert into `item` (`item_id`, `name`) values (110000055, `Soft
Drink`);
ERROR 1054 (42S22): Unknown column 'Soft Drink' in 'field list'
```

```
mysql> insert into `item` (`item_id`, `name`) values (110000055, 'Soft
Drink');
Query OK, 1 row affected (0.07 sec)

mysql> insert into `item_selling_price` (`item_id`, `store_id`, `price`)
values (110000055, '001', 0.65);
mysql> insert into `item_selling_price` (`item_id`, `store_id`, `price`)
values (110000055, '002', 1.00);
Query OK, 1 row affected (0.07 sec)

Query OK, 1 row affected (0.02 sec)

mysql> commit;
Query OK, 0 rows affected (0.00 sec)

mysql> exit;
Bye
$
```

*Atención: Observe que los nombres de tablas y campos en mysql deben ir entre signos ` ` (no comillas simples ni dobles), mientras que los datos de tipo carácter van entre comillas simples.*

Una vez que los datos han sido *comiteados*, los cambios son capturados por SDS y encolados en el nodo store 001 para que éste los tome (esta configurado para que lo haga una vez por minuto); la acción queda registrada en el log de ambos nodos. Como sólo se ha configurado el nodo 001 , sólo se recibirá allí el item correspondiente a dicho nodo. El item del nodo 002 (que aun no esta implementado) no será enviado.

### 6.2 Pushing data

Vamos a simular una venta en el store 001 para que SDS envíe (push) la transacción de vena a la oficina central corp

-Abrimos una sesión de sql interactivo con mysql y ejecutaremos algunos comandos para hacer una venta en el nodo store 001, como se indica a continuación:

```
mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 53
Server version: 5.5.40-0+wheezy1 (Debian)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights
reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| affablebean        |
| corp               |
| mysql              |
| performance_schema |
| store001           |
| test               |
+--------------------+
7 rows in set (0.05 sec)

mysql> use store001;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> insert into `sale_transaction` (`tran_id`, `store_id`,
`workstation`, `day`, `seq`) values (1000, '001', '3', '2007-11-01', 100);
Query OK, 1 row affected (0.02 sec)

mysql> insert into `sale_return_line_item` (`tran_id`, `item_id`, `price`,
`quantity`) values (1000, 110000055, 0.65, 1);
Query OK, 1 row affected (0.08 sec)

mysql> commit;
Query OK, 0 rows affected (0.00 sec)

mysql> exit;
Bye
```

Una vez que la transacción ha sido *comiteada*, los cambios en los datos son capturados por SDS y encolados para que el nodo store 001 los envie (push) a la oficina central (corp).

-Verifique el log de ambos nodos para comprobar la transferencia de los datos. El nodo store 001 esta configurado para enviar datos a corp cada 1 minuto.

Por ejemplo, aquí vemos algunas lineas del log de store 001, asociado con las operaciones anteriores:

```
2014-10-29 13:00:06,620 INFO [store-001] [PullService] [store-001-pull-1]
Pull data received from corp:000:000.  1 rows and 1 batches were processed
2014-10-29 13:00:11,587 INFO [store-001] [RouterService] [store-001-job-14]
Routed 1 data events in 258 ms
2014-10-29 13:00:16,363 INFO [store-001] [PushService] [store-001-push-1]
Push data sent to corp:000:000
2014-10-29 13:00:16,496 INFO [store-001] [PushService] [store-001-push-1]
Pushed data to corp:000:000. 1 data and 1 batches were processed
2014-10-29 13:06:07,490 INFO [store-001] [RouterService] [store-001-job-15]
Routed 1 data events in 133 ms
2014-10-29 13:06:10,518 INFO [store-001] [PushService] [store-001-push-1]
Push data sent to corp:000:000
2014-10-29 13:06:10,607 INFO [store-001] [PushService] [store-001-push-1]
Pushed data to corp:000:000. 1 data and 1 batches were processed
2014-10-29 13:07:49,349 INFO [store-001] [RouterService] [store-001-job-14]
Routed 1 data events in 265 ms
2014-10-29 13:07:51,584 INFO [store-001] [PushService] [store-001-push-1]
Push data sent to corp:000:000
2014-10-29 13:07:51,695 INFO [store-001] [PushService] [store-001-push-1]
Pushed data to corp:000:000. 1 data and 1 batches were processed
```

### 6.3 Verificando los batch de salida

Hemos enviado (push) y recibido (pull) datos, ahora veremos como obtener información aerca de los datos enviados en batch. El nodo que envía crea un batch y el nodo que recibe lo recibe y notifica dicha recepción.

En SDS las tablas son agrupadas en “canales”, si un batch en un determinado canal esta en error, ese batch se reintentará en cada sincronización en ese canal hasta que el batch ya no esté en error. Una vez superado esto, los otros batches para ese canal serán enviados. Esto se hace para asegurar que el orden de envio de los batches sea el mismo en el que ocurrieron en el nodo origen. Un canal no se bloquea cuando hay error en otro canal.

-Abramos una sesión sql interactiva en corp o store 001 para verificar los cambios que fueron capturados, haciendo:

**select * from sym_data order by data_id desc;**

la salida del comando es extensa, se acortó para mostrar los datos enviados y recibidos en la oficina central:

```
$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 112
```

```
Server version: 5.5.40-0+wheezy1 (Debian)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights
reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

mysql> use corp
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from sym_data order by data_id desc;
+---------+--------------------+------------
+-------------------------------------------------------------------------
-------------------------------------------------------------------------
-------------------------------------------------------------------------
-------------------------------------------------------------------------
---------------------------------------+-----------------+----------
+-----------------+-----------------+----------------+----------------
+---------------+-----------+---------------------+
| data_id | table_name         | event_type | row_data
| pk_data         | old_data | trigger_hist_id | channel_id      |
transaction_id | source_node_id | external_data | node_list | create_time
|
+---------+--------------------+------------
+-------------------------------------------------------------------------
-------------------------------------------------------------------------
-------------------------------------------------------------------------
-------------------------------------------------------------------------
---------------------------------------+-----------------+----------
+-----------------+-----------------+----------------+----------------
+---------------+-----------+---------------------+
|      46 | sym_node_host      | U          |
"001","debian2","192.168.54.163","grchere","Linux","amd64","3.2.0-4-
amd64","4","125015472","200802304","863109120","1.7.0_25","Oracle
Corporation","mysql-connector-java-5.1.30 ( Revision:
alexander.soklakov@oracle.com-20140310090514-
8xt1yoht5ksg2e7c )","3.6.11","-03:00","2014-10-29 13:15:06","2014-10-29
13:00:05","2014-10-28 14:23:37" | "001","debian2" | NULL     |
4 | heartbeat       | 79.60          | 001            | NULL          |
NULL      | 2014-10-29 13:15:16 |
|     45 | sym_node_host      | U          |
"000","debian2","192.168.54.163","grchere","Linux","amd64","3.2.0-4-
amd64","4","36886424","203620352","863109120","1.7.0_25","Oracle
```

```
Corporation","mysql-connector-java-5.1.30 ( Revision:
alexander.soklakov@oracle.com-20140310090514-
8xt1yoht5ksg2e7c )","3.6.11","-03:00","2014-10-29 13:14:46","2014-10-29
12:59:46","2014-10-28 14:07:28"  | "000","debian2" | NULL     |
4 | heartbeat       | 79.0           | NULL           | NULL          |
NULL      | 2014-10-29 13:14:46 |
|      44 | sym_node_host      | U          |
"001","debian2","192.168.54.163","grchere","Linux","amd64","3.2.0-4-
amd64","4","81162584","143327232","863109120","1.7.0_25","Oracle
Corporation","mysql-connector-java-5.1.30 ( Revision:
alexander.soklakov@oracle.com-20140310090514-
8xt1yoht5ksg2e7c )","3.6.11","-03:00","2014-10-29 13:00:05","2014-10-29
13:00:05","2014-10-28 14:23:37"  | "001","debian2" | NULL     |
4 | heartbeat       | 42.20          | 001            | NULL          |
NULL      | 2014-10-29 13:00:16 |
|      43 | sym_node_host      | U          |
"000","debian2","192.168.54.163","grchere","Linux","amd64","3.2.0-4-
amd64","4","87526456","145358848","863109120","1.7.0_25","Oracle
Corporation","mysql-connector-java-5.1.30 ( Revision:
alexander.soklakov@oracle.com-20140310090514-
8xt1yoht5ksg2e7c )","3.6.11","-03:00","2014-10-29 12:59:46","2014-10-29
12:59:46","2014-10-28 14:07:28"  | "000","debian2" | NULL     |
4 | heartbeat       | 42.0           | NULL           | NULL          |
NULL      | 2014-10-29 12:59:46 |
|      42 | sym_node_host      | U          |
"000","debian2","192.168.54.163","grchere","Linux","amd64","3.2.0-4-
amd64","4","43656976","92012544","863109120","1.7.0_25","Oracle
Corporation","mysql-connector-java-5.1.30 ( Revision:
alexander.soklakov@oracle.com-20140310090514-
8xt1yoht5ksg2e7c )","3.6.11","-03:00","2014-10-28 16:52:39","2014-10-28
14:07:28","2014-10-28 14:07:28"   | "000","debian2" | NULL     |
4 | heartbeat       | 405.162        | NULL           | NULL          |
NULL      | 2014-10-28 16:52:39 |
|      41 | sym_node_host      | U          |
"001","debian2","192.168.54.163","grchere","Linux","amd64","3.2.0-4-
amd64","4","16580952","67305472","863109120","1.7.0_25","Oracle
Corporation","mysql-connector-java-5.1.30 ( Revision:
alexander.soklakov@oracle.com-20140310090514-
8xt1yoht5ksg2e7c )","3.6.11","-03:00","2014-10-28 16:44:41","2014-10-28
14:23:37","2014-10-28 14:23:37"   | "001","debian2" | NULL     |
4 | heartbeat       | 392.190        | 001            | NULL          |
NULL      | 2014-10-28 16:44:54 |
|      40 | sym_node_host      | U          |
"000","debian2","192.168.54.163","grchere","Linux","amd64","3.2.0-4-
amd64","4","50106056","107675648","863109120","1.7.0_25","Oracle
Corporation","mysql-connector-java-5.1.30 ( Revision:
alexander.soklakov@oracle.com-20140310090514-
8xt1yoht5ksg2e7c )","3.6.11","-03:00","2014-10-28 16:37:39","2014-10-28
14:07:28","2014-10-28 14:07:28"  | "000","debian2" | NULL     |
```

```
4 | heartbeat       | 383.54         | NULL           | NULL          |
NULL      | 2014-10-28 16:37:39 |
|      39 | sym_node_host      | U          |
"001","debian2","192.168.54.163","grchere","Linux","amd64","3.2.0-4-
amd64","4","18285352","78053376","863109120","1.7.0_25","Oracle
Corporation","mysql-connector-java-5.1.30 ( Revision:
alexander.soklakov@oracle.com-20140310090514-
8xt1yoht5ksg2e7c )","3.6.11","-03:00","2014-10-28 16:29:41","2014-10-28
14:23:37","2014-10-28 14:23:37"   | "001","debian2" | NULL     |
4 | heartbeat       | 370.64         | 001            | NULL          |
NULL      | 2014-10-28 16:29:45 |
|      38 | sym_node_host      | U          |
"000","debian2","192.168.54.163","grchere","Linux","amd64","3.2.0-4-
amd64","4","58176000","114229248","863109120","1.7.0_25","Oracle
Corporation","mysql-connector-java-5.1.30 ( Revision:
alexander.soklakov@oracle.com-20140310090514-
8xt1yoht5ksg2e7c )","3.6.11","-03:00","2014-10-28 16:22:39","2014-10-28
14:07:28","2014-10-28 14:07:28"  | "000","debian2" | NULL     |
4 | heartbeat       | 354.162        | NULL           | NULL          |
NULL      | 2014-10-28 16:22:39 |
|      37 | sym_node_host      | U          |
"001","debian2","192.168.54.163","grchere","Linux","amd64","3.2.0-4-
amd64","4","48500560","88866816","863109120","1.7.0_25","Oracle
Corporation","mysql-connector-java-5.1.30 ( Revision:
alexander.soklakov@oracle.com-20140310090514-
8xt1yoht5ksg2e7c )","3.6.11","-03:00","2014-10-28 16:14:41","2014-10-28
14:23:37","2014-10-28 14:23:37"   | "001","debian2" | NULL     |
4 | heartbeat       | 341.172        | 001            | NULL          |
NULL      | 2014-10-28 16:14:46 |
|      36 | sym_node_host      | U          |
"000","debian2","192.168.54.163","grchere","Linux","amd64","3.2.0-4-
amd64","4","22814672","127860736","863109120","1.7.0_25","Oracle
Corporation","mysql-connector-java-5.1.30 ( Revision:
alexander.soklakov@oracle.com-20140310090514-
8xt1yoht5ksg2e7c )","3.6.11","-03:00","2014-10-28 16:07:39","2014-10-28
14:07:28","2014-10-28 14:07:28"  | "000","debian2" | NULL     |
4 | heartbeat       | 331.54         | NULL           | NULL          |
NULL      | 2014-10-28 16:07:39 |
|      35 | sym_node_host      | U          |
"001","debian2","192.168.54.163","grchere","Linux","amd64","3.2.0-4-
amd64","4","59096912","101449728","863109120","1.7.0_25","Oracle
Corporation","mysql-connector-java-5.1.30 ( Revision:
alexander.soklakov@oracle.com-20140310090514-
8xt1yoht5ksg2e7c )","3.6.11","-03:00","2014-10-28 15:59:40","2014-10-28
14:23:37","2014-10-28 14:23:37"  | "001","debian2" | NULL     |
4 | heartbeat       | 314.64         | 001            | NULL          |
NULL      | 2014-10-28 15:59:46 |
|      34 | sym_node_host      | U          |
"000","debian2","192.168.54.163","grchere","Linux","amd64","3.2.0-4-
amd64","4","43918544","140247040","863109120","1.7.0_25","Oracle
```

```
Corporation","mysql-connector-java-5.1.30 ( Revision:
alexander.soklakov@oracle.com-20140310090514-
8xt1yoht5ksg2e7c )","3.6.11","-03:00","2014-10-28 15:52:39","2014-10-28
14:07:28","2014-10-28 14:07:28"  | "000","debian2" | NULL     |
4 | heartbeat       | 296.162        | NULL           | NULL          |
NULL      | 2014-10-28 15:52:39 |
|      33 | sym_node_host      | U          |
"001","debian2","192.168.54.163","grchere","Linux","amd64","3.2.0-4-
amd64","4","58797128","115671040","863109120","1.7.0_25","Oracle
Corporation","mysql-connector-java-5.1.30 ( Revision:
alexander.soklakov@oracle.com-20140310090514-
8xt1yoht5ksg2e7c )","3.6.11","-03:00","2014-10-28 15:44:40","2014-10-28
14:23:37","2014-10-28 14:23:37"  | "001","debian2" | NULL     |
4 | heartbeat       | 279.172        | 001            | NULL          |
NULL      | 2014-10-28 15:44:47 |
|      32 | sym_node_host      | U          |
"000","debian2","192.168.54.163","grchere","Linux","amd64","3.2.0-4-
amd64","4","25792568","154402816","863109120","1.7.0_25","Oracle
Corporation","mysql-connector-java-5.1.30 ( Revision:
alexander.soklakov@oracle.com-20140310090514-
8xt1yoht5ksg2e7c )","3.6.11","-03:00","2014-10-28 15:37:39","2014-10-28
14:07:28","2014-10-28 14:07:28"  | "000","debian2" | NULL     |
4 | heartbeat       | 265.54         | NULL           | NULL          |
NULL      | 2014-10-28 15:37:39 |
|      31 | sym_node_host      | U          |
"001","debian2","192.168.54.163","grchere","Linux","amd64","3.2.0-4-
amd64","4","71700616","129368064","863109120","1.7.0_25","Oracle
Corporation","mysql-connector-java-5.1.30 ( Revision:
alexander.soklakov@oracle.com-20140310090514-
8xt1yoht5ksg2e7c )","3.6.11","-03:00","2014-10-28 15:29:40","2014-10-28
14:23:37","2014-10-28 14:23:37"  | "001","debian2" | NULL     |
4 | heartbeat       | 249.73         | 001            | NULL          |
NULL      | 2014-10-28 15:29:48 |
|      30 | sym_node_host      | U          |
"000","debian2","192.168.54.163","grchere","Linux","amd64","3.2.0-4-
amd64","4","86846784","165216256","863109120","1.7.0_25","Oracle
Corporation","mysql-connector-java-5.1.30 ( Revision:
alexander.soklakov@oracle.com-20140310090514-
8xt1yoht5ksg2e7c )","3.6.11","-03:00","2014-10-28 15:22:39","2014-10-28
14:07:28","2014-10-28 14:07:28"  | "000","debian2" | NULL     |
4 | heartbeat       | 231.162        | NULL           | NULL          |
NULL      | 2014-10-28 15:22:39 |
|      29 | sym_node_host      | U          |
"001","debian2","192.168.54.163","grchere","Linux","amd64","3.2.0-4-
amd64","4","26294576","146669568","863109120","1.7.0_25","Oracle
Corporation","mysql-connector-java-5.1.30 ( Revision:
alexander.soklakov@oracle.com-20140310090514-
8xt1yoht5ksg2e7c )","3.6.11","-03:00","2014-10-28 15:14:40","2014-10-28
14:23:37","2014-10-28 14:23:37"  | "001","debian2" | NULL     |
```

```
4 | heartbeat       | 211.181        | 001            | NULL          |
NULL      | 2014-10-28 15:14:49 |
|      28 | sym_node_host      | U          |
"000","debian2","192.168.54.163","grchere","Linux","amd64","3.2.0-4-
amd64","4","126645792","176947200","863109120","1.7.0_25","Oracle
Corporation","mysql-connector-java-5.1.30 ( Revision:
alexander.soklakov@oracle.com-20140310090514-
8xt1yoht5ksg2e7c )","3.6.11","-03:00","2014-10-28 15:07:38","2014-10-28
14:07:28","2014-10-28 14:07:28" | "000","debian2" | NULL     |
4 | heartbeat       | 195.45         | NULL           | NULL          |
NULL      | 2014-10-28 15:07:39 |
|      27 | sym_node_host      | U          |
"001","debian2","192.168.54.163","grchere","Linux","amd64","3.2.0-4-
amd64","4","31991928","161873920","863109120","1.7.0_25","Oracle
Corporation","mysql-connector-java-5.1.30 ( Revision:
alexander.soklakov@oracle.com-20140310090514-
8xt1yoht5ksg2e7c )","3.6.11","-03:00","2014-10-28 14:59:40","2014-10-28
14:23:37","2014-10-28 14:23:37"  | "001","debian2" | NULL     |
4 | heartbeat       | 180.73         | 001            | NULL          |
NULL      | 2014-10-28 14:59:51 |
|      26 | sym_node_host      | U          |
"000","debian2","192.168.54.163","grchere","Linux","amd64","3.2.0-4-
amd64","4","142825456","190316544","863109120","1.7.0_25","Oracle
Corporation","mysql-connector-java-5.1.30 ( Revision:
alexander.soklakov@oracle.com-20140310090514-
8xt1yoht5ksg2e7c )","3.6.11","-03:00","2014-10-28 14:52:38","2014-10-28
14:07:28","2014-10-28 14:07:28" | "000","debian2" | NULL     |
4 | heartbeat       | 161.162        | NULL           | NULL          |
NULL      | 2014-10-28 14:52:38 |
|      25 | sym_node_host      | U          |
"001","debian2","192.168.54.163","grchere","Linux","amd64","3.2.0-4-
amd64","4","124832912","173342720","863109120","1.7.0_25","Oracle
Corporation","mysql-connector-java-5.1.30 ( Revision:
alexander.soklakov@oracle.com-20140310090514-
8xt1yoht5ksg2e7c )","3.6.11","-03:00","2014-10-28 14:44:40","2014-10-28
14:23:37","2014-10-28 14:23:37" | "001","debian2" | NULL     |
4 | heartbeat       | 141.190        | 001            | NULL          |
NULL      | 2014-10-28 14:44:53 |
|      24 | sym_node_host      | U          |
"000","debian2","192.168.54.163","grchere","Linux","amd64","3.2.0-4-
amd64","4","131639880","204668928","863109120","1.7.0_25","Oracle
Corporation","mysql-connector-java-5.1.30 ( Revision:
alexander.soklakov@oracle.com-20140310090514-
8xt1yoht5ksg2e7c )","3.6.11","-03:00","2014-10-28 14:37:38","2014-10-28
14:07:28","2014-10-28 14:07:28" | "000","debian2" | NULL     |
4 | heartbeat       | 127.29         | NULL           | NULL          |
NULL      | 2014-10-28 14:37:38 |
|      23 | item_selling_price | I          | "110000055","002","1.00",
| NULL            | NULL     |             20 | item            | 124.0
| NULL          | NULL          | NULL      | 2014-10-28 14:37:38 |
```

```
|      22 | item_selling_price | I          | "110000055","001","0.65",
| NULL            | NULL     |              20 | item            | 124.0
| NULL           | NULL          | NULL      | 2014-10-28 14:37:38 |
|      21 | item                | I          | "110000055","Soft Drink"
....
$46 rows in set (0.01 sec)

mysql> exit;
Bye
```

Cada fila representa una fila de datos que fue modificada. El data_id se incrementa secuencialmente. Cada fila representa una tupla que ha sido cambiada. data_id se incrementa secuencialmente event_type es "I" para insert, "U" para update, o "D" para delete. Para insert y update, los valores de datos capturados se indican en row_data. Para update y delete, los valores de clave primaria son indicados en inpk_data.

-para saber los cambios incluidos en cada batch, usar data_id:

**select * from sym_data_event where data_id = ?;**

Los batchs son creados con la necesidad de enrutarlos a los nodos como parte de un job en background, llamado route job (job de enrutamiento). Como parte del route job, los cambios en los datos son asignados al batch usando un batch_id, el cual se utilizar para seguir la pista del mismo y sincronizar los datos. El vinculo entre los batchs y los datos se mantienen en la tabla sym_data_event.

-Verificar que los datos cambiados han sido puestos en un batch, enviados a su destino y recibidos, usando el batch_id:

**select * from sym_outgoing_batch where batch_id = ?;**

Los batchs inicialmente tiene el estado "NE" cuando son nuevos y aun no han sido enviados. Una vez que se ha recibido el Ok de su envio (acknowledges) el estado del batch cambia a "OK" or "ER" en caso de fallo. Si hay fallo, el campo error_flagse pone con valor 1, si luego de reintentos, cambia su condición de error, se modificarán estos valores de forma apropiada.

Comprendiendo estas 3 tablas (mas otra más que se verá en la siguiente sección), es clave para diagnosticar cualquier problema de sincronización que Ud pueda encontrar

### 6.4 Verificando los batch de entrada

El nodo que recepciona mantiene la pista de los batches y su Ok de recepción y registra estatdísticas acerca de la carga de datos. Los batches duplados son salteados (acción por defecto), pero este comportamiento puede ser cambiado a través de la propiedad incoming.batches.skip.duplicates .

-Para explorar los batch de entrada, en cualquier nodo ejecute el siguiente comando sql:

**select * from sym_incoming_batch where batch_id = ?;**

Un batch representa una colección de cambios cargados por el nodo. El nodo que envia el batch es registrado y el estado (status) del batch puede ser "OK" or "ER" en caso de error .

**Felicitaciones!. El ejemplo de replicación esta terminado.**

## 7. Arrancando y Deteniendo la Replicación

Se puede detener la replicación simplemente interrumpiedo los procesos que se ejecutan en cada terminal. Y se puede volver a arrancar los procesos, de la misma forma que se indicó anteriormente, para el nodo corp, en sub-directorio /samples, hacer:

```
$ ../bin/sym --engine corp-000 --port 8080
```

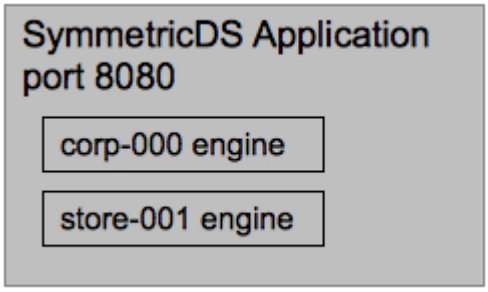
en nodo node 001, sub-directorio /samples, hacer:

```
$ ../bin/sym --engine store-001 --port 9090
```

## 8. SDS “Multi-Homing” o “Multi-homed”

En este ejemplo de replicación, hemos puesto un archivo de propiedades en cada sub-directorio /engines de cada aplicación SDS instalada; cuando ésta se ejecuta, atiende a cada nodo, acorde con dicho archivo de propiedades y es responsable por una determinada base de datos.

SDS es capaz de arrancar mas una aplicación SDS al mismo tiempo. Cuando SDS arranca, verifica en el subdirectorio /engines todos los archivos *.properties que se encuentran en el mismo y ejecutará una aplicación SDS por cada archivo de propiedades encontrado. El switch –engine que fue utilizado anteriormente impide este comportamiento y sólo carga un único archivo de propiedades. Cuando un mismo SDS corre multiples archivos de propiedades para multiples bases de datos, esto se denomina “multi-homed” o “multi-homing” SDS, acorde con el siguiente gráfico:




El tutorial anterior podría haberse hecho "multi-homed", instalando una única copia de SDS, para ello sólo se requieren los siguientes pasos:

1. Instale una unica copia de SDS (en vez de dos, como hicimos anteriormente). No se necesita un directorio para representar los 2 servidores.

2. En vez de copiar el archivo de propiedades en cada directorio de cada servidor, ahora se copian dos archivos de propiedades en el mismo subdirectorio /engines.

3. Todos los comandos del tutorial, ahora van a correr desde el mismo directorio /samples.

4. Cuando Ud arranca SDS, no debe indicar un determinado *engine* , ya que vamos a arrancar ambos *engines* al mismo tiempo. El comando se ejecutará desde /samples y será el siguiente:

```
../bin/sym --port 8080
```

   Ya no usamos el puerto 9090, SDS ahora escucha en el puerto 8080 para obtener cualquier trafico de datos importante, ya sea para store 001 como para corp.

5. Cualquier otro SDS que Ud arranque, todos los otros comandos que ejecute, deberán indicar el switch --engine para indicar el nodo al cual direccionar los comandos SDS al nodo correspondiente, para abrir la registración, *setear* el servidor corp, hacer la carga inicial del servidor store 001, etc.

# 9. Anexos

## 9.1 Estructura de directorios utilizadas en este tutorial

En negritas se indican los directorios utilizados en este tutorial:

```
symetricds
|-- sym-corp
|   |-- bin
|   |-- conf
|   |-- databases
|   |   |-- firebird
|   |   |   |-- x32
|   |   |   `-- x64
|   |   `-- interbase
|   |-- doc
|   |   |-- quick-start
|   |   |   |-- html-single
|   |   |   |   |-- css
|   |   |   |   `-- images
|   |   |   |       |-- admons
|   |   |   |       `-- callouts
|   |   |   `-- pdf
|   |   `-- user-guide
|   |       |-- html
|   |       |   |-- css
|   |       |   `-- images
|   |       |       |-- admons
|   |       |       `-- callouts
|   |       |-- html-single
|   |       |   |-- css
|   |       |   `-- images
|   |       |       |-- admons
|   |       |       `-- callouts
|   |       `-- pdf
|   |-- engines
|   |-- lib
|   |-- logs
|   |-- patches
|   |-- samples
|   |-- security
|   |-- tmp
|   |   |-- corp-000
|   |   |   `-- outgoing
|   |   |       `-- 000
|   |   |           `-- 001
|   |   `-- jetty-0.0.0.0-8080-web-_-any-
|   |       `-- jsp
|   `-- web
|       `-- WEB-INF
|           `-- lib
`-- sym-store001
    |-- bin
    |-- conf
    |-- databases
    |   |-- firebird
    |   |   |-- x32
    |   |   `-- x64
    |   `-- interbase
    |-- doc
```

```
    |   |-- quick-start
    |   |   |-- html-single
    |   |   |   |-- css
    |   |   |   `-- images
    |   |   |       |-- admons
    |   |   |       `-- callouts
    |   |   `-- pdf
    |   `-- user-guide
    |       |-- html
    |       |   |-- css
    |       |   `-- images
    |       |       |-- admons
    |       |       `-- callouts
    |       |-- html-single
    |       |   |-- css
    |       |   `-- images
    |       |       |-- admons
    |       |       `-- callouts
    |       `-- pdf
    |-- engines
    |-- lib
    |-- logs
    |-- patches
    |-- samples
    |-- security
    |-- tmp
    |   |-- jetty-0.0.0.0-9090-web-_-any-
    |   |   `-- jsp
    |   `-- store-001
    |       `-- filesync_incoming
    |           `-- 001
    |               `-- 000
    `-- web
        `-- WEB-INF
            `-- lib
```

### 9.2 Archivo de propiedades corp-000.properties del nodo corp, almacenado en /sym-corp/engines:

```
#
# Licensed to JumpMind Inc under one or more contributor
# license agreements.  See the NOTICE file distributed
# with this work for additional information regarding
# copyright ownership.  JumpMind Inc licenses this file
# to you under the GNU General Public License, version 3.0 (GPLv3)
# (the "License"); you may not use this file except in compliance
# with the License.
#
# You should have received a copy of the GNU General Public License,
# version 3.0 (GPLv3) along with this library; if not, see
# <http://www.gnu.org/licenses/>.
#
# Unless required by applicable law or agreed to in writing,
# software distributed under the License is distributed on an
```

```
# "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
# KIND, either express or implied.  See the License for the
# specific language governing permissions and limitations
# under the License.
#

engine.name=corp-000

# The class name for the JDBC Driver
db.driver=com.mysql.jdbc.Driver
#db.driver=oracle.jdbc.driver.OracleDriver
#db.driver=org.postgresql.Driver
#db.driver=org.apache.derby.jdbc.EmbeddedDriver
#db.driver=org.hsqldb.jdbcDriver
#db.driver=net.sourceforge.jtds.jdbc.Driver
#db.driver=com.ibm.db2.jcc.DB2Driver
#db.driver=com.informix.jdbc.IfxDriver
#db.driver=org.firebirdsql.jdbc.FBDriver
#db.driver=interbase.interclient.Driver
#db.driver=org.sqlite.JDBC
#db.driver=com.sybase.jdbc4.jdbc.SybDriver db.driver=org.h2.Driver

# The JDBC URL used to connect to the database
db.url=jdbc:mysql://localhost/corp?tinyInt1isBit=false
#db.url=jdbc:mysql://localhost/corp
#db.url=jdbc:oracle:thin:@127.0.0.1:1521:corp
#db.url=jdbc:postgresql://localhost/corp?
protocolVersion=2&stringtype=unspecified
#db.url=jdbc:derby:corp;create=true
#db.url=jdbc:hsqldb:file:corp;shutdown=true
#db.url=jdbc:jtds:sqlserver://localhost:1433;useCursors=true;bufferMaxMemor
y=10240;lobBuffer=5242880
#db.url=jdbc:db2://localhost/corp
#db.url=jdbc:informix-
sqli://localhost:9088/corp:INFORMIXSERVER=ol_ids_1150_1
#db.url=jdbc:firebirdsql:localhost:/var/lib/firebird/data/databasename
#db.url=jdbc:interbase://localhost//opt/interbase/data/corp.gdb
#db.url=jdbc:sqlite:corp.sqlite
#db.url=jdbc:sybase:Tds:localhost:5000/databasename
#db.url=jdbc:h2:corp;AUTO_SERVER=TRUE;LOCK_TIMEOUT=60000

# The user to login as who can create and update tables
db.user=root

# The password for the user to login as
db.password=nbuser

registration.url=
sync.url=http://localhost:8080/sync/corp-000

# Do not change these for running the demo
group.id=corp
external.id=000

# Don't muddy the waters with purge logging
job.purge.period.time.ms=7200000
```

```
# This is how often the routing job will be run in milliseconds
job.routing.period.time.ms=5000
# This is how often the push job will be run.
job.push.period.time.ms=10000
# This is how often the pull job will be run.
job.pull.period.time.ms=10000
```

### 9.3 Archivo de propiedades store-001.properties del nodo store 001, almacenado en /sym-store001/engines:

```
#
# Licensed to JumpMind Inc under one or more contributor
# license agreements.  See the NOTICE file distributed
# with this work for additional information regarding
# copyright ownership.  JumpMind Inc licenses this file
# to you under the GNU General Public License, version 3.0 (GPLv3)
# (the "License"); you may not use this file except in compliance
# with the License.
#
# You should have received a copy of the GNU General Public License,
# version 3.0 (GPLv3) along with this library; if not, see
# <http://www.gnu.org/licenses/>.
#
# Unless required by applicable law or agreed to in writing,
# software distributed under the License is distributed on an
# "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
# KIND, either express or implied.  See the License for the
# specific language governing permissions and limitations
# under the License.
#

engine.name=store-001

# The class name for the JDBC Driver
db.driver=com.mysql.jdbc.Driver
#db.driver=oracle.jdbc.driver.OracleDriver
#db.driver=org.postgresql.Driver
#db.driver=org.apache.derby.jdbc.EmbeddedDriver
#db.driver=org.hsqldb.jdbcDriver
#db.driver=net.sourceforge.jtds.jdbc.Driver
#db.driver=com.ibm.db2.jcc.DB2Driver
#db.driver=com.informix.jdbc.IfxDriver
#db.driver=org.firebirdsql.jdbc.FBDriver
#db.driver=interbase.interclient.Driver
#db.driver=org.sqlite.JDBC
#db.driver=com.sybase.jdbc4.jdbc.SybDriver
#db.driver=org.h2.Driver

# The JDBC URL used to connect to the database
db.url=jdbc:mysql://localhost/store001?tinyInt1isBit=false
#db.url=jdbc:oracle:thin:@127.0.0.1:1521:store001
#db.url=jdbc:postgresql://localhost/store001?
protocolVersion=2&stringtype=unspecified
```

```
#db.url=jdbc:derby:store001;create=true
#db.url=jdbc:hsqldb:file:store001;shutdown=true
#db.url=jdbc:jtds:sqlserver://localhost:1433/store001;useCursors=true;buffe
rMaxMemory=10240;lobBuffer=5242880
#db.url=jdbc:db2://localhost/store001
#db.url=jdbc:informix-
sqli://localhost:9088/store001:INFORMIXSERVER=ol_ids_1150_1
#db.url=jdbc:firebirdsql:localhost:/var/lib/firebird/data/databasename
#db.url=jdbc:interbase://localhost//opt/interbase/data/store001.gdb
#db.url=jdbc:sqlite:store001.sqlite
#db.url=jdbc:sybase:Tds:localhost:5000/databasename
#db.url=jdbc:h2:store001;AUTO_SERVER=TRUE;LOCK_TIMEOUT=60000

# The user to login as who can create and update tables
db.user=root

# The password for the user to login as
db.password=nbuser

# The HTTP URL of the root node to contact for registration
registration.url=http://localhost:8080/sync/corp-000

# Do not change these for running the demo
group.id=store
external.id=001

# This is how often the routing job will be run in milliseconds
job.routing.period.time.ms=5000
# This is how often the push job will be run.
job.push.period.time.ms=10000
# This is how often the pull job will be run.
job.pull.period.time.ms=10000
```

### 9.4 Archivo SymmetricDS utilizado:

```
$ ls -l symmetric-3.6.11-server.zip
-rw-r----- 1 grchere grchere 61396052 oct 28 08:50 symmetric-3.6.11-
server.zip
```

## Referencias

[1] symmetricDS, Quick Start Guide, disponible en http://www.symmetricds.org/doc/3.6/quick-start/html-single/quick-start.html

[2] Cherencio, G., “Soluciones de Replicación en PostgreSQL 9.1”, apunte de asignatura 11078 Base de Datos II, UNLu, disponible en http://www.grch.com.ar/docs/bdd/apuntes/unidad.iii/11078-Soluciones%20de%20replicación.pdf

[3] Cherencio, G. “Procedimientos habituales y tips en PostgreSQL 9.1”, apunte de asignatura 11078 Base de Datos II, UNLu, disponible en http://www.grch.com.ar/docs/bdd/apuntes/unidad.iii/11078-Procedimientos%20y%20tips.pdf

Atte. Guillermo Cherencio
11078 Base de Datos II
11077 Base de Datos I
División Sistemas
Departamento de Ciencias Básicas
UNLu