

Trabajo Práctico Bases de Datos Distribuidas y Replicación de Datos

El siguiente trabajo práctico tiene como objetivo introducir al alumno en el conocimiento de las bases de datos distribuidas y en la replicación de datos.

Descripción del laboratorio para implementar el siguiente trabajo:

Se conectó una red LAN virtual entre tres máquinas virtuales, una máquina con Windows, una máquina con Linux Debian y otra máquina con Linux Debian.

En los tres sistemas operativos tenemos el servicio del motor de bases de datos Firebird, el cual nos permite usar EXECUTE STATEMENT ¹para ejecutar SQL en una máquina remota.

Desarrollo del trabajo práctico:

En este documento se presentan ejemplos para que el alumno los implemente, se familiarice con el EXECUTE STATEMENT, y repase el uso de STORED PROCEDURE Y TRIGGER, ya que estos elementos son necesarios para desarrollar el trabajo práctico.

Se pide:

- 1) Crear la Base de Datos FACTURACIÓN en las tres máquinas virtuales.
- 2) Trabajando en Firebird sobre Windows, replicar la creación de las tablas CLIENTE, FACTURA, DETALLE, PRODUCTOS, SUCURSAL en las máquinas Debian.
- 3) Trabajando en Firebird Windows, insertar, modificar y eliminar filas en las tablas, y cada actualización se debe replicar en las bases de datos de las máquinas Debian.
- 4) Supongamos ahora que los clientes pertenecen a una sucursal determinada y solo queremos almacenar en una máquina Debian que podríamos llamar Debian_1 los clientes de la sucursal 1 y en la otra máquina Debian que podríamos llamar Debian_2 los clientes de la sucursal 2 (Repasar fragmentación horizontal y vertical de replicación de datos).
- 5) Consultas a juancarlosjromer@gmail.com

¹¹ En EXECUTE STATEMENT leer la diferencia entre with autonomous transaction y with common transaction.

Conexión Remota entre Firebird en Windows contra Firebird en Debian

Ejemplo 1

stored procedure ejecutado en Firebird sobre Windows

Crea una tabla en Firebird sobre Debian

```

1  -----
2  3 --drop procedure CREARTABLA_T2_deb3
3  4 -----
4  5 SET TERM ^ ;
5  6 create PROCEDURE CREARTABLA_T2_deb3
6  7 AS
7  8     declare lcComando varchar(512) ;
8  9 BEGIN
9  10     lcComando = 'create table T2_deb3 (
10 11         cod integer not null primary key,
11 12         nombre varchar (32) not null);';
12 13     EXECUTE STATEMENT :lcComando WITH COMMON TRANSACTION
13 14     ON EXTERNAL '192.168.1.105/3050:/var/lib/firebird/2.5/data/db2serverdebian3'
14 15     AS USER 'SYSDBA' PASSWORD 'masterkey' ;
15 16 END^
16 17 SET TERM ; ^
17 18 -----
18 19 GRANT EXECUTE ON PROCEDURE CREARTABLA_T2_deb3 TO SYSDBA;
19 20 -----
20 21 execute procedure CREARTABLA_T2_deb3
21 22 -----|

```

Ejemplo 2

stored procedure ejecutado en Firebird sobre Windows

Inserta una fila en una tabla en Firebird sobre Debian

```

1  3 --drop procedure INSERTAR_SERVER_en_win_t2_deb3;
2  4 -----
3  5 SET TERM ^ ;
4  6 create PROCEDURE INSERTAR_SERVER_en_win_t2_deb3(atcod integer, atnombre varchar(32))
5  7 AS
6  8 DECLARE VARIABLE lcComando VARCHAR(200);
7  9 BEGIN
8  10     lcComando = 'INSERT INTO T2_deb3 (cod, nombre) VALUES(' ||
9  11     :atcod || ',' || ASCII_CHAR(39) || :atnombre || ASCII_CHAR(39) || ')';
10 12
11 13     EXECUTE STATEMENT :lcComando WITH COMMON TRANSACTION
12 14     ON EXTERNAL '192.168.0.118/3050:/var/lib/firebird/2.5/data/db2serverdebian3'
13 15     AS USER 'SYSDBA' PASSWORD 'masterkey' ;
14 16 END^
15 17 SET TERM ; ^
16 18 -----
17 19 GRANT EXECUTE ON PROCEDURE INSERTAR_SERVER_en_win_t2_deb3 to sysdba;
18 20 -----
19 21 execute procedure INSERTAR_SERVER_en_win_t2_deb3(400,'cuatrocientos');
20 22 commit;
21 23 -----|
22 24 -----

```

Ejemplo 3

stored procedure ejecutado en Firebird sobre Windows

Recorre una tabla y inserta sus filas en una tabla en Firebird sobre Debian

```

3  --Drop procedure COPIAR_CONTENIDO_DE_UNA_TABLA_1
4  -----
5  SET TERM ^ ;
6  create PROCEDURE COPIAR_CONTENIDO_DE_UNA_TABLA_1
7  RETURNS ( ATCOD Integer, ATNOMBRE Varchar(32) )
8  AS DECLARE VARIABLE lcComando VARCHAR(200);
9  BEGIN
10
11     FOR SELECT COD , NOMBRE FROM T1_deb3 INTO :ATCOD , :ATNOMBRE
12     DO begin
13         lcComando = 'INSERT INTO T1_deb3 (cod, nombre) VALUES(' ||
14             atcod || ',' || ASCII_CHAR(39) || atnombre || ASCII_CHAR(39) || ')';
15         EXECUTE STATEMENT :lcComando WITH COMMON TRANSACTION
16         ON EXTERNAL '192.168.1.105/3050:/var/lib/firebird/2.5/data/db2serverdebian3' AS
17         USER 'SYSDBA' PASSWORD 'masterkey' ;
18     suspend;
19     END
20 END^
21 SET TERM ; ^
22
23 GRANT EXECUTE ON PROCEDURE COPIAR_CONTENIDO_DE_UNA_TABLA_1 TO SYSDBA;
24 -----
25 select * from t1_deb3;
26 -----
27 insert into t1_deb3 (cod,nombre) values (30,'treinta');
28 insert into t1_deb3 (cod,nombre) values (31,'treinta uno');
29 insert into t1_deb3 (cod,nombre) values (32,'treinta dos');
30 insert into t1_deb3 (cod,nombre) values (33,'treinta tres');
31 -----
32 --execute procedure COPIAR_CONTENIDO_DE_UNA_TABLA_1;
33 SELECT * FROM COPIAR_CONTENIDO_DE_UNA_TABLA_1;
34 commit;
35 -----

```

Ejemplo 4

trigger after delete que se ejecuta en una tabla en Firebird sobre Windows

Replica el delete en una tabla en Firebird sobre Debian

```

1  --Ejemplo 4
2  --drop trigger TRG_AD_T1_deb3;
3  -----
4  SET TERM ^ ;
5  create TRIGGER TRG_AD_T1_deb3 for t1_deb3 ACTIVE AFTER DELETE POSITION 0
6  AS
7  declare lcComando varchar(132) ;
8  --declare atcod integer ;
9  BEGIN
10     --atcod      = OLD.COD ;
11     --lcComando = 'delete from T1_deb3 where COD = ' || :atcod ;
12     lcComando = 'delete from T1_deb3 where COD = ' || OLD.COD ;
13     EXECUTE STATEMENT :lcComando WITH AUTONOMOUS TRANSACTION
14     ON EXTERNAL '192.168.0.118/3050:/var/lib/firebird/2.5/data/db2serverdebian3'
15     AS USER 'SYSDBA' PASSWORD 'masterkey' ;
16 END^
17 SET TERM ; ^
18 -----
19 delete from t1_deb3 where cod = 1;
20 commit ;
21 select * from t1_deb3 ;
22

```

Ejemplo 5

trigger after insert que se ejecuta en una tabla en Firebird sobre Windows
 Replica el insert en una tabla en en Firebird sobre Debian

```

1
2 --Ejemplo 5
3 --drop trigger TRG_AI_t1_deb3
4
5 SET TERM ^ ;
6 create TRIGGER TRG_AI_t1_deb3 for t1_deb3 ACTIVE
7 AFTER INSERT POSITION 0
8 AS
9 declare lcComando varchar(132) ;
10 declare atcod integer ;
11 declare atnombre varchar(32) ;
12 BEGIN
13   atcod = NEW.COD ;
14   atnombre = NEW.NOMBRE ;
15   lcComando = 'INSERT INTO T1_deb3(cod, nombre) VALUES(' ||
16               :atcod || ',' || ASCII_CHAR(39) || :atnombre || ASCII_CHAR(39) || ')';
17
18   EXECUTE STATEMENT
19     :lcComando WITH COMMON TRANSACTION
20   ON EXTERNAL
21     '192.168.0.118/3050:/var/lib/firebird/2.5/data/db2serverdebian3'
22   AS
23     USER 'SYSDBA' PASSWORD 'masterkey' ;
24 END^
25 SET TERM ; ^
26 -----
27 insert into t1_deb3 (cod,nombre) values (111,'111');
28

```

Ejemplo 6

trigger after update que se ejecuta en una tabla en Firebird sobre Windows
 Replica el update en una tabla en Firebird sobre Debian

```

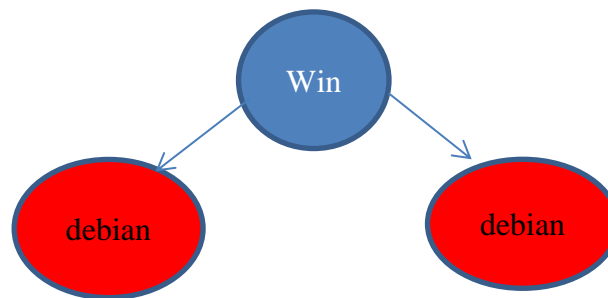
1 --Ejemplo 6
2
3 SET TERM ^ ;
4 create TRIGGER TRG_AU_T1_deb3 for t1_deb3 ACTIVE
5 AFTER UPDATE POSITION 0
6 AS
7 declare lcComando varchar(132) ;
8 declare atcod integer ;
9 declare atnombre varchar(32) ;
10
11 BEGIN
12
13   atcod = NEW.COD ;
14   atnombre = NEW.NOMBRE ;
15
16   lcComando = 'update T1_deb3 set nombre = ' || ASCII_CHAR(39) || :atnombre || ASCII_CHAR(39)
17               || ' where COD = ' || :atcod ;
18
19   EXECUTE STATEMENT
20     :lcComando WITH COMMON TRANSACTION
21   ON EXTERNAL
22     '192.168.0.118/3050:/var/lib/firebird/2.5/data/db2serverdebian3'
23   AS
24     USER 'SYSDBA' PASSWORD 'masterkey';
25 END^
26 SET TERM ; ^
27
28
29 select * from t1_deb3
30
31 update T1_deb3 set nombre = 'ciento once' where cod = 111

```

Ejemplo 7

stored procedure que se ejecuta en Firebird sobre Windows

Replica en dos servidores Firebird sobre debian la creación de una tabla.



```

SQL
crea crear_tabla
File Edit View History Statement Grid
1 -- drop procedure crear_tabla
2
3 SET TERM ^ ;
4 create PROCEDURE crear_tabla(tabla varchar(32))
5 AS
6 declare lcComando varchar(512) ;
7 BEGIN
8   lcComando = 'create table ' || :tabla || ' (
9               cod integer not null primary key,
10              nombre varchar (32) not null);';
11   EXECUTE STATEMENT
12   :lcComando WITH COMMON TRANSACTION
13   ON EXTERNAL
14   '192.168.0.37/3050:/var/lib/firebird/2.5/data/db2serverdebian3'
15   AS
16   USER 'SYSDBA' PASSWORD 'masterkey' ;
17
18 EXECUTE STATEMENT
19 :lcComando WITH COMMON TRANSACTION
20 ON EXTERNAL
21 '192.168.0.36/3050:/var/lib/firebird/2.5/data/db2serverdebian3'
22 AS
23 USER 'SYSDBA' PASSWORD 'masterkey' ;
24
25
26 END^
27 SET TERM ; ^
28 -----
29 GRANT EXECUTE
30 ON PROCEDURE crear_tabla TO SYSDBA;
31 -----
32 execute procedure crear_tabla('personas');
33 -----
34
35
  
```

Ejemplo 8

stored procedure que se ejecuta en Firebird sobre Windows (azul) y pone en ejecución un store procedure en un servidor Firebird sobre Debian (verde) que permite la creación de una tabla en un servidor Firebird sobre Debian (rojo)



```

1 -- drop procedure crear_tabla_diferida
2 SET TERM ^ ;
3 create PROCEDURE crear_tabla_diferida(tabla varchar(32))
4 AS
5 declare lcComando varchar(512) ;
6 BEGIN
7   lcComando = 'execute procedure crear_tabla('' || :tabla || '')';
8
9 EXECUTE STATEMENT
10   :lcComando WITH COMMON TRANSACTION
11   ON EXTERNAL
12   '192.168.0.36/3050:/var/lib/firebird/2.5/data/db2serverdebian3'
13 AS
14   USER 'SYSDBA' PASSWORD 'masterkey' ;
15
16
17 END^
18 SET TERM ; ^
19 -----
20 GRANT EXECUTE
21 ON PROCEDURE crear_tabla_diferida TO SYSDBA;
22 -----
23 execute procedure crear_tabla_diferida('productos');
24 -----
25
26
  
```

Stored procedure en el servidor Firebird en Debian intermedio (verde).

```

SET TERM ^ ;
ALTER PROCEDURE CREAR_TABLA ( TABLA Varchar(32) )
AS declare lcComando varchar(512) ;
BEGIN
  lcComando = 'create table ' || :tabla || ' ( cod integer not null primary key,
              nombre varchar (32) not null);';
  EXECUTE STATEMENT :lcComando WITH COMMON TRANSACTION
  ON EXTERNAL '192.168.0.38/3050:/var/lib/firebird/2.5/data/db2serverdebian3'
  AS
    USER 'SYSDBA' PASSWORD 'masterkey' ;
END^
SET TERM ; ^
  
```