INNOVA Research Journal, ISSN 2477-9024
(Febrero, 2018). Vol. 3, No.2.1 pp. 59-66
DOI: https://doi.org/10.33890/innova.v3.n2.2018.668
URL: http://revistas.uide.edu.ec/index.php/innova/index
Correo: innova@uide.edu.ec

Implementación de una base de datos heterogénea distribuida entre los SGBDs ORACLE, MySQL y PostgreSQL con replicación, mediante un script bash implementado en el sistema operativo CentOS usando software libre

Implementation of a heterogeneous database distributed among the ORACLE, MySQL and PostgreSQL SGBDs with replication, using a bash script implemented in the CentOS operating system using free software

Edwin Flores
Universidad de las Fuerzas Armadas-ESPE, Ecuador

Autor para correspondencia: egflores5@espe.edu.ec
Fecha de recepción: 15 de diciembre 2017 - Fecha de aceptación: 26 de febrero de 2018

Resumen
El presente artículo permite obtener una base de datos distribuida heterogénea con replicación en un ambiente CentOS en menos de 5 minutos, mediante un script bash, usando únicamente software libre. La idea se centra en tener un esclavo que será ORACLE y los nodos maestros estarán en MySQL y PostgreSQL, con la finalidad de reducir costos a la hora de diseñar e implementar una base de datos distribuida ya que se aprovecha al máximo la potencia de un gestor de pago, el cual es Oracle, con la gratuidad y buen desempeño de MySQL y PostgreSQL. El resultado de este artículo será un ODBC que configure los dos gestores sin gastar un solo centavo y en tiempo récord, y así poder obtener una base de datos distribuida conformada por distintos gestores, que trabajen como uno solo.

Palabras claves: heterogeneidad; MySQL; PostgreSQL; ORACLE; ODBC

Abstract
This article allows to obtain a heterogeneous distributed database with replication in a CentOS environment in less than 5 minutes, using a bash script, using only free software. The idea is focused on having a slave that will be ORACLE and the master nodes will be in MySQL and PostgreSQL, in order to reduce costs when designing and implementing a distributed database since the power of a manager is maximized of payment, which is Oracle, with the free and good performance of MySQL and PostgreSQL. The result of this article will be an ODBC that configures the two managers without spending a single cent and in record time, and thus be able to obtain a distributed database made up of different managers, who work as one.

Key words: heterogeneity; MySQL; PostgreSQL; ORACLE; ODBC

Introducción

Por lo general la heterogeneidad de una base de datos puede convertirse en una tarea difícil de lograr; y más aún cuando se quiere hacer que trabajen de manera distribuida, pero existen ciertas exigencias que hacen inevitable la coexistencia de distintos gestores de bases de datos o SGBDs en una misma organización, en la cual cambiar a un gestor de bases de datos determinado no resulta una buena opción. Por lo general las organizaciones adaptan los sistemas gestores de bases de datos según sus necesidades; conjugando el uso de gestores gratuitos y gestores de paga. Es allí donde nace la idea de poder conjugar dos o más gestores para que trabajen como uno solo.

La idea central de las bases de datos distribuidas es la integración lógica de varias bases de datos que se encuentran separadas físicamente, pero pueden almacenar y procesar en varios nodos distribuidos sobre una determinada red de computadores. Sin embargo, la distribución no aborda temas de heterogeneidad, pero tampoco los excluye (Cuadra & Castro, 2013). Para que un sistema administrador de base de datos distribuido pueda llamarse heterogéneo; debe utilizar al menos dos sistemas gestores de bases de datos distintos; estos a su vez presentan un problema en la trasferencia de los datos e información ya que cada SGBDs tiene su propio modelo de datos. (Cisneros, 1998)

Las principales dificultades a la hora de hablar de heterogeneidad de bases de datos; pasa principalmente por la difícil compatibilidad que surge entre los distintos gestores. Pues cada gestor está hecho para coexistir con bases de datos de su misma línea y marca. Es por ello que surgen distintos tipos de heterogeneidades de bases de datos que pueden ser clasificadas de la siguiente manera (Sheth, 1990):

- Heterogeneidad por diferencias en el SGBD.
- Heterogeneidad por diferencias en la semántica de los datos.

En el presente artículo se centra en la heterogeneidad de SGBD; este tipo de heterogeneidad es muy común en ciertas organizaciones que crecen sin una planificación en el área de sus sistemas de información, ya que cada departamento de la empresa puede tener requerimientos y necesidad diferentes a la hora de almacenar y procesar sus datos, en este caso se selecciona diferentes SGBDs. (Martínez, 2010)

Es necesaria esta coexistencia debido a que la actual sociedad de la información exige un acceso a la información de forma completa, esta sociedad de la información por lo general es distribuida y heterogénea, esto quiere decir que las diversas fuentes de datos e información deben trabajar de manera conjunta con el sistema que solicita la información y el principal problema de lograr lo mencionado es la interoperabilidad. (Muñoz & Obando, 2006)

UNIXODBC es una especificación abierta para proporcionar a los desarrolladores de aplicaciones una API predecible con la que puedan acceder a fuentes de datos como servidores SQL. Mediante este ODBC genérico se obtendrá la conexión entre los distintos gestores de bases de datos. (Harvey & Gorham, 2015)

En la actualidad existen diversos ODBC que permiten interconectar distintos gestores de bases de datos, pero estos no son gratuitos; un claro ejemplo es el ODBC de MySQL ofrecido por Devart cuyo precio es USD 449.95 (Devart, 2017). Sin embargo, solo ofrecen el ODBC, y la configuración sigue siendo un problema para el lograr un diseño de bases de datos distribuido.

Las escasas soluciones de heterogeneidad vienen dadas únicamente para el sistema operativo Windows, y es aún más escasa la información para sistemas operativos Linux. En este caso se eligió el sistema operativo CentOS; ya que se caracteriza por ser un sistema operativo para servidores, posee alta disponibilidad y su principal característica es su seguridad ya que es una bifurcación del sistema operativo Red Hat. (Smyth, 2010)

Los SGBD usados en el presente artículo son:

- ORACLE 12c: Es el motor de base de datos más usado en todo el mundo, es de licencia propietaria, además posee una alta disponibilidad, particionamiento, escalabilidad, seguridad, replicación y multiplataforma. (Heurtel, 2015)
- MySQL: Es la base de datos Open Source más usada a nivel mundial, se caracteriza por su rapidez en las operaciones y su buen rendimiento, posee una baja probabilidad de corromper los datos. (Edward, 2006)
- PostgreSQL: Es una base de datos SQL Open Source, ofrece estabilidad y confiabilidad, además de rendimiento y su gran capacidad de almacenamiento. También se caracteriza por ser multiplataforma. (Martin, 2011)

Pero la idea no solo se basa en bases de datos ya existentes, sino también se podrá considerar como una alternativa que abarate los costos a la hora del diseño e implementación de una base de datos distribuida, ya que se podrá reducir significativamente la compra de licencias de gestores de pago, debido a que se puede distribuir la cantidad de datos mediante gestores gratuitos como MySQL o PostgreSQL y lograr que funcionen como uno solo sin pagar un centavo. Ante esto, el objetivo principal del presente artículo es lograr la coexistencia de distintos gestores de bases de datos ya sea gratis o de pago con la finalidad de lograr una distribución y tratamiento de los datos de forma transparente de tal manera que el usuario ni siquiera note la diferencia de estar trabajando con una base de datos heterogénea.

Métodos

Para el presente artículo se empleó la metodología de investigación tecnológica, la cual brinda las pautas para resolver problemas de la realidad y tiene base empírica porque aplica los conocimientos teóricos de la ciencia a la práctica. (Espinoza, 2010)

Esta metodología fue aplicada de la siguiente manera:

Figura 1. Proceso aplicado de metodología de investigación tecnológica.

Hipótesis: Los gestores ORACLE, MySQL y PostgreSQL pueden funcionar de manera conjunta en una base de datos distribuida en un ambiente CentOS mediante un script de configuración.

Variable independiente; gestor de base de datos.

Variable dependiente: configuración del ODBC.

Población: Configuraciones posibles del ODBC para cada gestor de base de datos existente para la conexión con ORACLE.

Muestra: Configuraciones posibles del ODBC para los gestores MySQL y PostgreSQL para la conexión con ORACLE.

Análisis de datos: empleó la técnica de visualización de datos, esta consistirá en determinar si hay la coexistencia entre los gestores realizando una sentencia SQL desde el gestor ORACLE hacia las tablas que se encuentra replicando de MySQL y PostgreSQL devolviendo los datos existentes en aquellas tablas, de esta manera se podrá analizar si el script configuró correctamente la conexión y los gestores trabajan de manera conjunta y transparente.

Diseño de base de datos: Cada base de datos está diseñada según las necesidades específicas del entorno en el cual se va a utilizar, cada una de estas tendrá un modelo de datos, un contexto de datos, un nivel de abstracción y un formato de almacenamiento de datos determinado por el usuario; a esto se le llama heterogeneidad (Barroso, 2015). La arquitectura de un sistema de base de datos está influenciada directamente por el software que soporta la instalación del SGBD, lo que reflejará muchas de las características propias del sistema subyacente en el SGBD. (Gonzalez, 2000)

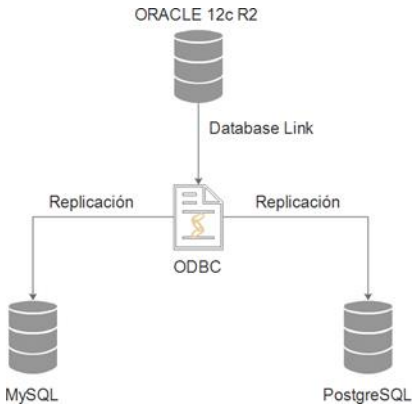

Figura 2. Arquitectura de base de datos empleada en el presente artículo

Metodología de Desarrollo de Software TDD: Es una metodología ágil de diseño e implementación de software. Su enfoque está orientado a pruebas de desarrollo. De forma resumida; se escribe las líneas de código y se comprueba su comportamiento, de esta manera se logra los objetivos deseados. Este tipo de metodología permite ir desarrollando pruebas constantes al código que se va desarrollando. Y da como resultado final un producto software en este caso será un script. (Jurado, 2010)

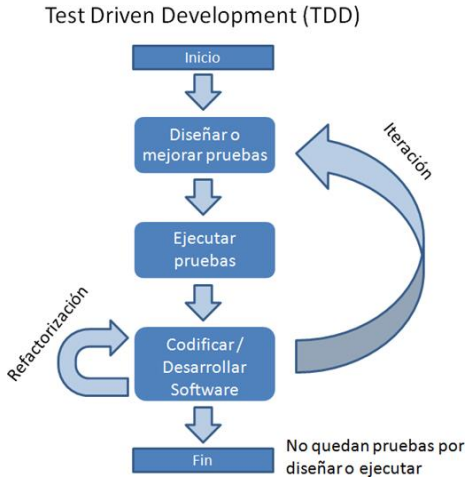

Figura 3. Proceso de la metodología ágil TDD

El script fue sometido a las siguientes pruebas:

- Realizar ping entre todos los nodos.
- Verifica la existencia de las variables de entorno de ORACLE.
- Permite la conexión de desde UnixODBC al nodo MySQL.
- Permite la conexión de desde UnixODBC al nodo PostgreSQL.
- Permite la conexión de desde el nodo ORACLE al nodo MySQL mediante el listener.
- Permite la conexión de desde el nodo ORACLE al nodo PostgreSQL mediante el listener.
- Realizar un SELECT desde el nodo ORACLE al nodo MySQL y al nodo PostgreSQL.
- Realizar un DELETE desde el nodo ORACLE al nodo MySQL y al nodo PostgreSQL.
- Realizar un INSERT desde el nodo ORACLE al nodo MySQL y al nodo PostgreSQL.

- Realizar un UPDATE desde el nodo ORACLE al nodo MySQL y al nodo PostgreSQL.
- Realizar sentencias DML en los nodos MySQL y PostgreSQL y visualizarlos en el nodo ORACLE.
- Verificar que se cumpla el tiempo de replicación.

Si el script pasa todas las pruebas antes mencionadas, se lo aceptará.

Resultados

Pseudocódigo del algoritmo que contiene el script.

1. Copiar UnixODBC en /usr/local/UnixODBC
2. Realizar un make
3. Realizar un make install
4. Instalar ODBC de MySQL.
5. Instalar ODBC de PostgreSQL.
6. Configurar el contenido odbc.ini de UnixODBC.
7. Probar conexión de MySQL con ./isql
8. Probar conexión de PostgreSQL con ./isql
9. Configurar el contenido de tnsnames.ora
10. Configurar el contenido de listener.ora
11. Configurar el contenido de initdg4odbc.ora
12. Configurar el contenido de initPostgres.ora
13. Crear el database link de MySQL en ORACLE.
14. Crear el database link de PostgreSQL en ORACLE.
15. Crear la vista materializada con el database link de MySQL.
16. Crear la vista materializada con el database link de PostgreSQL.

El algoritmo fue desarrollado en un script Bash, y es compatible con las distribuciones Linux basados en RPM y DEB.

Los resultados presentados a continuación se los logró empleado:

- ORACLE 12c en el sistema operativo CentOS.
- MariaDB versión 10.1.21
- PostgreSQL corriendo como contenedor en Docker.
- SQL Developer.

Para evidenciar los resultados obtenidos, se realizará una consulta desde ORACLE a la tabla existente en MySQL.

Discusión

Se puede decir que la hipótesis planteada es correcta y verdadera, ya que se logró la coexistencia de distintos gestores, realizando consultas desde ORACLE a los gestores MySQL y PostgreSQL.

Los resultados arrojaron lo siguiente:

- El script configuró correctamente la conexión de los gestores MySQL y PostgreSQL.
- Se logró la replicación programable en el tiempo que prefiera el usuario, en este caso fue de 1 segundo para evidenciar los resultados rápidamente.
- Mediante la sentencia SELECT se pudo mostrar desde ORACLE los registros existentes en la MySQL.
- Mediante la sentencia SELECT se pudo visualizar desde ORACLE los registros existentes en la PostgreSQL.
- Se logró generar un ambiente distribuido con distintos gestores, que brinden fiabilidad y rendimiento.
- Es script es portable y puede ser aplicable a cualquier distribución Linux que soporte RPM O DEB.

Conclusiones

La heterogeneidad de bases de datos distribuidas posee grandes problemas de implementación que han sido resueltos en el presente artículo mediante el empleo de UnixODBC y un script bash de configuración en un entorno CentOS, con los gestores ORACLE, MySQL y PostgreSQL, además de ofrecer una alternativa para disminuir costos a la hora de diseñar e implementar una base de datos distribuida gracias a la integración de gestores gratuitos con gestores de pago.

Toda la configuración se desarrolla en el nodo esclavo; el cual es ORACLE debido a que posee Database Link y se acopla de muy buena manera con el ODBC de Unix, además de poseer las vistas materializadas; las mismas que permiten realizar la replicación en un determinado tiempo.

Debido a ser un ODBC genérico, se puede incrementar la cantidad de gestores, únicamente se necesita el controlador del gestor a añadir y estar en la misma red.

El script propuesto en el presente artículo; reduce significativamente la inversión a la hora de diseñar una base de datos distribuida y promueve el empleo del software libre para poderlo aplicar en cualquier campo en la sociedad del conocimiento.

El script de configuración es un producto software innovador, ya que en el campo de la heterogeneidad las soluciones son muy escasa y costosas. De tal manera que se ofrece una solución a aquellas organizaciones que trabajan con distintos gestores de base de datos y desean que funcionen como uno solo, sin necesidad de migrar a un gestor determinado.

Bibliografía

Barroso, V. (2015). Explotación e Integración de Bases de Datos Heterogéneas para la. Universidad de Valladolid.

Cisneros, J. (1998). Panorama sobre base de datos. (Un enfoque práctico). UABC.

Cuadra, D., & Castro, E. (2013). Desarrollo de bases de datos:casos practicos desde el análisis a la. Ra-Ma.

Devart. (2017). Devart Web Site. Obtenido de https://www.devart.com/purchase.html Edward, L. (2006). MySQL. Peachpit Press.

Espinoza, C. (2010). Metodología de investigación tecnologica. Peru: Imagen Gráfica SAC.

Gonzalez, Ó. (2000). Arquitecturas de sistemas de bases de datos. Castilla: Universidad De Castilla La Mancha.

Harvey, P., & Gorham, N. (31 de 08 de 2015). UnixODBC. Obtenido de http://www.unixodbc.org/

Heurtel, O. (2015). Oracle 12c: administración. Ediciones ENI. Jurado, C. (2010). Diseño Ágil con TDD. Lulu.com.

Martin, S. (2011). PostgreSQL: Una poderosa base de datos libre. EAE.

Martínez, L. (2010). Diseño y construcción de bases de datos distribuidas heterogéneas sobre Oracle y SQL Server. Madrid: Universidad Carlos III de Madrid.

Muñoz, A., & Obando, A. (2006). Heterogeneidad de Datos y Posibles Soluciones. Bógota: Pontificia Universidad Javeriana.

Sheth, A. (1990). Federated Database Systems for Managing distributed, Heterogeneus, and Autonomus Database. ACM Computing.

Smyth, N. (2010). CentOS 5 Essentials. eBookFrenzy.