

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE CHALCO

HOJA DE EVIDENCIA DE EVALUACIÓN SUMATIVA

Carrera: Ingeniería en Sistemas Computacionales

Asignatura: Administración de Base de Datos

Nombre del(a) alumno(a):

Actividad correspondiente a : (periodo, unidad, avance, etc.)

- Cadena Gonzales Luis Raúl
- Cortes Vásquez Gustavo
- Espinosa Sánchez Daniel Antonio

Nombre del profesor: Virginia Aguilar Guerrero

Grupo: 4602

Fecha: 24/04/2020

Resultado: (nivel de desempeño/ calificación)

Firma del Alumno: _____

Actividad: Investigar la arquitectura interna de el SGBD asignado por equipo y colocar la imagen de dicha estructura. Solo colocar la imagen y generar el documento en pdf uno por equipo y subirlo en la carpeta del equipo, fecha de entrega el lunes 27 de Abril del presente año a mas tardar a las 11:59 o 23:59.

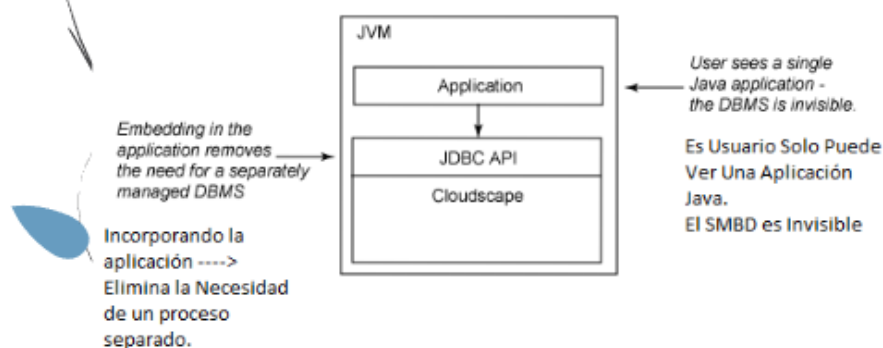
Alumno	SGBD Asignado
MUNGUIA OLIVO ANGEL ADRIAN	Apache derby
SANTOS PEREZ JOSE MANUEL	Apache derby
TREJO PEREZ ADRIAN	Apache derby
FLORES ALCALA ROBERTO ALDAI	Cassandra
JUAREZ ROSALES KAREN CITLALI	Cassandra
SORIANO LOPEZ ALBERTO	Cassandra
LOPEZ PEREZ JULIO CESAR	Couch DB
MARTINEZ ALVAREZ GERARDO	Couch DB
MEJIA MARTINEZ RICARDO	Couch DB
ARCOS JACOME ELVIA VANESSA	Cubrid
ESCUADERO VILLA JOAQUIN GUSTAVO	Cubrid
RAMOS RAMIREZ MARIA FERNANDA	Cubrid
GARCIA CATARINO JOSE MANUEL	DB2
RAMOS GALICIA VALERIA ALEJANDRA	DB2
ALVAREZ MUÑOZ LEONARDO DAVID	Elasticsearch
AMARO OLAYA JOSE ALFREDO	Elasticsearch
LOPEZ RAMIREZ LIZBETH	Elasticsearch
CORONA LOPEZ RAFAEL	Fox Pro
ESTRADA HERNANDEZ JUAN CARLOS	Fox Pro
CADENA GONZALEZ LUIS RAUL	Apache derby
CORTES VASQUEZ GUSTAVO	Apache derby
ESPINOSA SANCHEZ DANIEL ANTONIO	Apache derby
MUJICA HERNANDEZ BRIAN EDUARDO	Neo4j
VAZQUEZ BAUTISTA DANIEL	Neo4j
AGUIRRE VELAZQUEZ LUIS RAYMUNDO	Redis
CALDERON PEREZ ANA KAREN	Redis
MEDINA GARCIA JOSE	Redis
ROMERO ROSALES ALOYSIUS GABRIEL	Riak
VICTORIA MUÑOZ ADRIAN ALFREDO	Riak

Apartados	Descripción		
	Classic	SuperClassic	SuperServer
Procesos	Hay que enviar la señal apropiada al proceso padre httpd que se esté ejecutando. Hay dos maneras de enviar estas señales. En primer lugar, puede usar el comando de Unix kill que envía señales directamente a los procesos. Puede que tenga varios procesos httpd ejecutándose en su sistema, pero las señales deben enviarse solamente al proceso padre, cuyo pid está especificado en la directiva PidFile. Esto quiere decir que no debe necesitar enviar señales a ningún proceso excepto al proceso padre. Hay tres señales que puede enviar al proceso padre: TERM, HUP, y USR1, que van a ser descritas a continuación.	Esto puede hacer que se produzca el error "bind: Address already in use" (después de usarHUP) o el error "long lost child came home!" (después de usar USR1). En el primer caso se trata de un error irrecuperable, mientras que en el segundo, solo ocurre que el servidor pierde un slot del scoreboard. Por lo tanto, sería aconsejable usar reinicios graceful, y solo hacer reinicios normales de forma ocasional.	implicada en la segunda y subsiguientes peticiones en una conexión HTTP persistente (KeepAlive). Puede ser que el servidor termine después de leer la línea de petición, pero antes de leer cualquiera de las cabeceras de petición. Hay una solución que fue descubierta demasiado tarde para la incluirla en versión 1.2. En teoría esto no debe suponer ningún problema porque el cliente KeepAlive ha de esperar que estas cosas pasen debido a los retardos de red y a los timeouts que a veces dan los servidores.
Cache	Los datos de caché almacenados por el servidor derby incorporado se almacenan en {DENODO_HOME} / metadata / db / cache, principalmente como archivos * .dat. También hay un procedimiento almacenado incorporado que le informará sobre el contenido de un caché, CACHE_CONTENTS , que recuperará información sobre qué patrones de consulta se almacenan en el caché para una base de datos y / o vista determinada.		
Conexiones locales	Hay un adaptador de recursos genérico que funciona con cualquier gestor de bases de datos JDBC (Java Database Connectivity), pero solo soporta transacciones locales. Los otros son adaptadores de recursos específicos de cada fabricante, optimizados para un gestor de bases de datos determinado.		
	Apache Derby	Adaptador de recursos de transacción local de cliente TranQL para Apache Derby	tranql-connector-derby-client-local-<versión>.rar
	Apache Derby	Adaptador de recursos de transacción XA de cliente TranQL para Apache Derby	tranql-connector-derby-client-xa-<versión>.rar
	Apache Derby	Adaptador de recursos local incorporado TranQL para Apache Derby	tranql-connector-derby-embed-local-<versión>.rar
Acceso simultaneo	Las bases de datos controlan ellas mismas que dos usuarios no escriban a la vez el mismo registro (primero lo haría uno y después el otro)Lo que deberías controlar tú es que si un usuario selecciona un registro para modificarlo, otro usuario no		

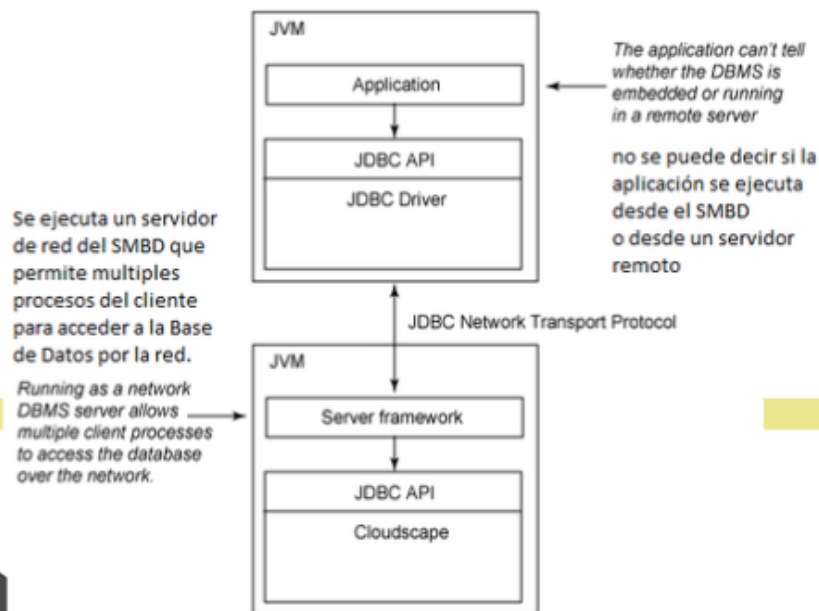
	<p>pudiese entrar a la vez en el mismo registro para modificar también. Generalmente cada base de datos tiene sus mecanismos de bloqueo (selects con locks...)</p>
Multiprocesador/multinúcleo	<p>Apache inicia varios subprocesos y cada petición es atendida por uno de estos; cuando termina con esta petición este subproceso podría atender a otro cliente o ser terminado, según al valor de MaxRequestsPerChild.</p> <p>Es el modo más estable, ya que un error crítico solo afectaría a una petición. Este es el único modo en que se pueden usar módulos / extensiones que no sean Thread-Safe.</p> <p>Requiere más recursos (Memoria RAM y CPU) para atender cierto número de peticiones simultaneas, respecto a otras configuraciones. Esto limita drásticamente la escalabilidad del servidor.</p> <p>Favorece el uso intensivo de PHP. Los aceleradores de PHP no son Thread-Safe, pero al usarlos junto a Prefork podemos justificar el mayor uso de php (o páginas sin ningún tipo de caché, aparte del acelerador en sí).</p> <p>Prefork es la configuración predeterminada en la mayoría de instalaciones.</p>

Arquitectura general:

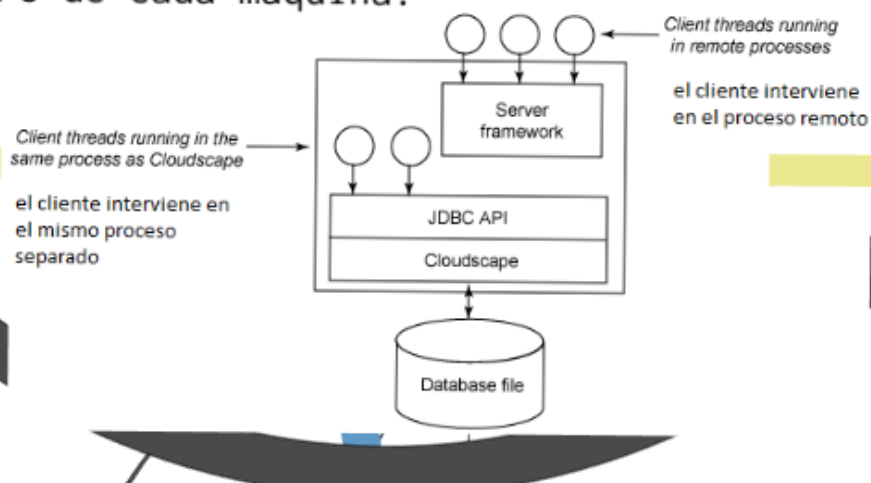
El motor de base de datos es integrable. Esto significa que en lugar de correr como un proceso separado, el software de motor de base de datos puede ser parte de la aplicación para que la misma y el motor de base de datos se ejecuten en la JVM (Máquina Virtual Java). Con un motor de base de datos integrada, la aplicación utiliza la API JDBC (Base de Datos Java de Conectividad) para acceder a la base de datos.



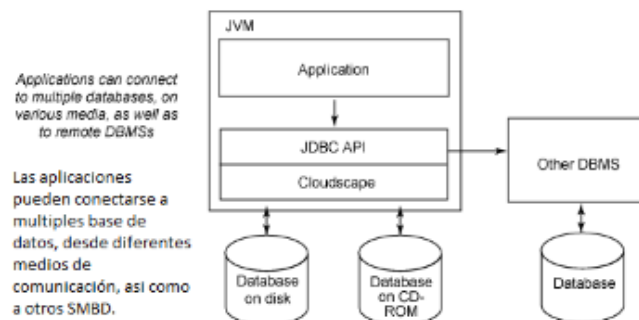
CLIENTE/SERVIDOR: Es Posible implementar Derby como un tradicional Cliente / Servidor de Base de Datos. Derby incluye un servidor de red, pero tambien puede ejecutar una infraestructura del servidor de su elección.



PERSISTENCIA DE DATOS: Derby proporciona la persistencia de datos mediante el almacenamiento de datos en archivos de discos. Un motor de Derby puede manejar uno o más archivos de base de datos, pero cada archivo de base de datos, solo se puede acceder por un único motor de Derby. Los Datos en el disco de archivos de base de datos, se almacenan de una forma portátil, por lo que las bases de datos pueden ser fácilmente transportadas de una máquina a otra, independientemente de la arquitectura de CPU de cada máquina.



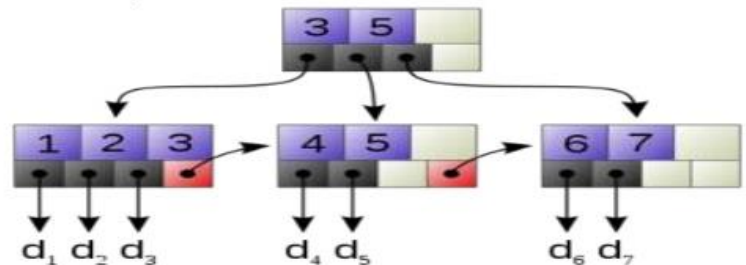
FLEXIBILIDAD: Derby proporciona una gran flexibilidad para los diseñadores del sistema. Cada Instancia de Derby puede gestionar múltiples base de datos, las cuales pueden estar en diversos medios de comunicación, y no hay nada para detener la aplicación de la conexión a otros SMBD.



CRITERIOS FUNCIONALES. INTEGRIDAD DE LOS DATOS

◦ ORGANIZACIÓN DE ARCHIVOS

En Apache Derby la organización de archivos es de tipo INDEXADA SECUENCIAL, debido a que implementa índices secundarios de SQL como Árbol B, además implementa el algoritmo estándar de Árbol B+, ya que los nodos de la hoja, Los nodos internos sólo contienen claves y punteros. Todas las hojas se encuentran en el mismo, más bajo nivel. Los nodos hoja se encuentran unidos entre sí como una lista enlazada para permitir búsqueda secuencial. Apache Derby está constituida por los árboles B+, ya que conservan la propiedad de acceso aleatorio rápido y permiten además un recorrido secuencial rápido. En un árbol B+ todas las claves se encuentran en hojas, duplicándose en la raíz y nodos interiores aquellas que resulten necesarias para definir los caminos de búsqueda. A continuación se muestra la estructura del Árbol B+:



◦ Bibliografía:

- <http://db.apache.org>

CRITERIOS FUNCIONALES. SEGURIDAD.

- Derby proporciona características de seguridad para el despliegue de bases de datos a sitios remotos o móviles, incluyendo la autenticación utilizando una variedad de esquemas, la autorización del usuario, soporte para ejecutar con un Administrador de seguridad de Java 2, y el cifrado de base de datos.

Acción	Usuarios readOnlyAccess	Usuarios FullAccess
Ejecución de sentencias SELECT	X	X
Leer las propiedades de base de datos	X	X
Carga de clases base de datos de archivos jar	X	X
La ejecución de INSERT, UPDATE o DELETE		X
Ejecución de sentencias DDL		X
Añadir o sustituir los archivos jar		X
Configuración de las propiedades de base de datos		X

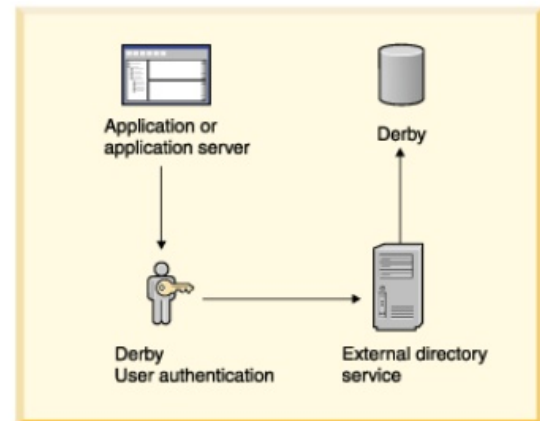
CRITERIOS FUNCIONALES. SEGURIDAD.

• Autenticación y Métodos

Derby proporciona soporte para la autenticación de usuario, esto significa que Derby se autentica en un nombre y una contraseña antes de permitir que el acceso del usuario al sistema. Una vez Derby autentica al usuario, que otorga al usuario el acceso al sistema de Derby, pero no necesariamente el acceso a la base de datos hecha en la solicitud de conexión. En el sistema de Derby, el acceso a una base de datos está determinada por la autorización del usuario.

Derby es compatible con una simple función a modo de mantener una lista interna de los usuarios autorizados. También es compatible con la integración de sistemas externos de autenticación de usuario, por ejemplo, se puede configurar para comprobar con un servicio de autenticación externo. Lightweight Directory Access Protocol (LDAP) se presta apoyo, y los planes personalizados son compatibles con los definidos por el usuario de Java Naming and Directory Interface (JNDI) clases.

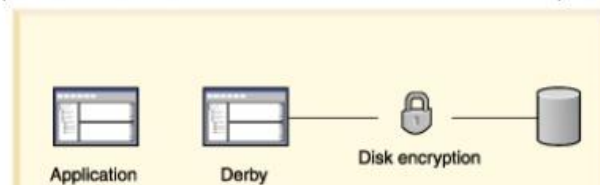
Al no tener los nombres de usuario y contraseñas mantiene en la base de datos de medios administrativos generales menos para la transferencia de nombres en la base de datos. Esto es especialmente importante en aplicaciones de servidor desplegado.



CRITERIOS FUNCIONALES. SEGURIDAD.

Otro mecanismo de seguridad Derby, es el cifrado de disco, protege los datos cuando el destinatario no puede saber cómo proteger los datos. Es útil para implementar bases de datos en un entorno integrado. Normalmente, los sistemas de bases de datos cifrar y descifrar datos en el transporte por la red, utilizando sistemas estándar del sector. Este sistema funciona bien para bases de datos cliente / servidor, el servidor se supone que en un entorno de confianza y segura, gestionada por un administrador del sistema. Además, el destinatario de los datos es de confianza y deben ser capaces de proteger los datos. El único riesgo viene cuando el transporte de datos a través del cable, y cifrado de datos ocurre durante el transporte de la red solamente.

Esta función de cifrado de datos proporciona la capacidad de almacenar datos de los usuarios de forma encriptada. El usuario que arranca la base de datos debe proporcionar una contraseña de arranque.



CRITERIOS TECNICOS. Manejo de Datos.

DICCIONARIO / DIRECTORIO DE DATOS (D/D)

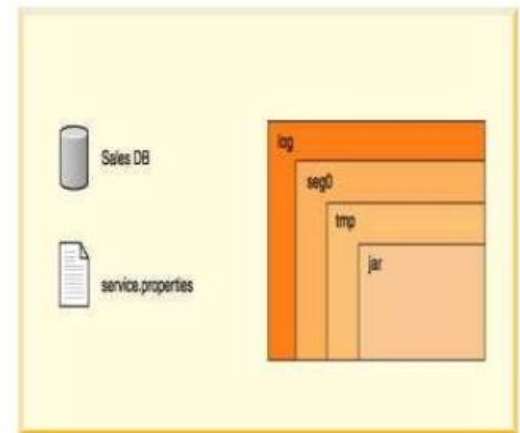
Un diccionario de Datos contiene las características lógicas de los datos que se van a utilizar en el sistema que estamos programando. Es decir, almacenan la información sobre los datos relacionado con el origen, descripción, relación con otros datos, uso, responsabilidad y formato. En el SDBD Apache Derby, el D/D contiene objetos como tablas, columnas índices y archivos jar. Además, almacena en archivos que residen en un directorio del mismo nombre que la base de datos. también suelen vivir en los directorios del sistema, conteniendo archivos y directorios tales como:

log directory: este directorio contiene los archivos que componen el registro de transacciones de bases de datos, de uso interno para la recuperación de datos

seg0 directory: contiene un archivo para cada tabla de usuario, tabla de sistema, y el índice

tmp directory: es un directorio temporal utilizado por una variedad de sentencias

jar directory: en este directorio se almacenan los archivos jar.



CRITERIOS TECNICOS. INTEGRIDAD DE LOS DATOS.

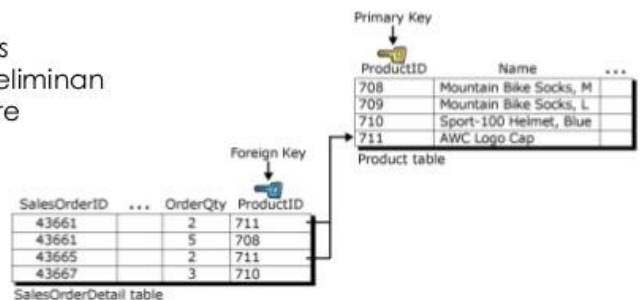
Integridad de Dominio: La integridad de dominio viene dada por la validez de las entradas para una columna determinada, es decir, se encarga de proteger los datos en una columna de la base de datos.

Integridad Referencial: La integridad referencial protege las relaciones definidas entre las tablas cuando se crean o se eliminan filas. La integridad referencial se basa en las relaciones entre claves externas y claves principales.

Ejemplo:

Nombre de la base de datos: AdventureWorks

Tablas de la base de datos: SalesOrderDetail, Product



La integridad referencial se basa en la relación entre la clave externa (**ProductID**) de la tabla **SalesOrderDetail** y la clave principal (**ProductID**) de la tabla **Product**. Esta relación garantiza que un pedido de ventas no pueda nunca hacer referencia a un producto que no existe en la tabla **Product**. Como también se puede observar gráficamente;□

Bibliografía:

- <http://db.apache.org>

Bibliográficas:

<https://prezi.com/fauewxnrycoc/apache-derby/>

<https://es.slideshare.net/CastellanoPGJ/apache-derby-41902929>

<https://prezi.com/fauewxnrycoc/apache-derby/>

<https://prezi.com/iykmp1vhmpgo/base-de-datos-apache-derby/>

https://www.ibm.com/support/knowledgecenter/es/SSRULV_9.3.0/com.ibm.tivoli.itws.doc_9.3/common/src_al/awsal_DBcfg.htm

http://www.juntadeandalucia.es/empleo/recursos/material_didactico/especialidades/materialdidactico_administrador_servidores/Content/3-servicios_web/5-ApacheWebServer.pdf

<https://httpd.apache.org/docs/2.4/es/stopping.html>

<http://descargas.pntic.mec.es/mentor/visitas/Apache.pdf>

<http://www.maestrosdelweb.com/entendiendo-los-modos-multiproceso-de-apache/>