

Grupo - 09
Jose Alejandro Silva Ariza
Javier Esteban Camargo Urrego

Iteración 3

1. Análisis:

Si lo requiere, ajuste el modelo del mundo (modelo conceptual: diagrama de clases UML) propuesto en la iteración 2 que representó el caso de estudio. Indique cuáles clases del modelo del mundo fueron actualizadas o creadas en esta iteración.

Modelo conceptual en PDF adjunto

2. Diseño de la aplicación:

- a. A partir del diseño existente, analice el impacto que representa la introducción de los nuevos requerimientos y restricciones a nivel del modelo conceptual. Realice los cambios necesarios en su modelo relacional para respetar las reglas de negocio y asegurar la calidad del mismo. Tenga en cuenta los comentarios recibidos en la sustentación del taller 2. Documente el diseño y las decisiones tomadas para crear los elementos de la base de datos que da el respaldo de persistencia a la aplicación, a partir del modelo conceptual. Incluya un listado con las tablas generadas en la base de datos, utilizando los estándares establecidos, disponibles en la wiki del curso (sección tutoriales). Este listado debe incluir el nombre de la tabla, el nombre y el tipo de dato de sus campos así como los nombres de restricciones de llaves primarias, llaves foráneas y de chequeo. Sea claro en mencionar explícitamente los cambios relevantes entre su diseño entregado con el taller anterior y este.**

CAPACIDADDEPRODUCCION

TIEMPO	NUMPRODUCTOSPRODUCIDOS	CANTIDAD_MATERIA_PRIMA	CODIGO_ESTACION_PROD
CK	NN,ck	NN,CK	PK,NN

ETAPAS_DE_ESTACION

ID_ESTACION	ID_ETAPA
PK1,NN,FK(ESTACION.ID)	PK2,NN, FK(ESTACION.ID)

MATERIALESDEPRODUCTOS

MATERIAL	PRODUCTO	CANTIDAD_MATERIA
PK1,NN,FK(MATERIAL.ID)	PK2,NN, FK(PRODUCTO.NOMBRE)	NN,CK

PEDIDOS_MATERIAL

MATERIAL	ID_PROVEEDOR	ID_PEDIDO	TIEMPO_ENTREGA
PK1,NN,FK(MATERIAL.ID)	PK2,NN, FK(PROVEEDOR.NOMBRE)	NN,CK,PK3	NN,CK

COSTO	CANTIDAD
NN,CK	NN,CK

PEDIDOSCLIENTE

ID_CLIENTE	NOM_PRODUCTO	UNIANDEADES_REQ
PK1,NN,FK(CLIENTE.ID)	PK2,NN, FK(PRODUCTO.NOMBRE)	NN,CK

En general, no se hicieron cambios a las tablas de las iteraciones anteriores, se crearon fue nuevas relaciones de muchos a muchos, por consiguiente la gran mayoría de tablas son respecto a estas relaciones que demandaban crear las tablas anteriores. Con respecto a la de capacidad, esta fue creada para aclarar ciertos atributos que referían a una estación de producción pero que a su vez corresponden a una nueva clase. Se puede observar en el uml como ciertas relaciones en general de los productos son sus materiales y los clientes con sus productos cambiaron, ahora un cliente puede tener varios productos así como los productos pueden tener varios cliente; los materiales pueden tener varios productos y los productos varios materiales, por ese motivo es que se crearon las tablas de MATERIALESDEPRODUCTOS y de PEDIDOSCLIENTE. Con respecto a los pedidos de material, se incluyó una nueva clase en el uml para facilitar el manejo de estos, por consiguiente se creó la respectiva tabla.

- b. Valide que su modelo se encuentra en BCNF y que no presenta anomalías de inserción, borrado o actualización con respecto a las reglas de negocio.**

Realizado y revisado en la BD.

- c. Documente la lógica de los nuevos requerimientos a desarrollar, descritos en la sección de caso de estudio de este documento. En este punto se requiere definir los mecanismos que utiliza para garantizar las propiedades ACID del requerimiento.**

RFC6

Se genera la sentencia SQL, con el tipo de consulta y el parámetro deseado, a partir de este se creara la consulta buscando en todas las tablas implicadas, si la consulta requiere un tiempo mayor a 10 segundos significa que el recurso puede estar siendo modificado por otro usuario o que tiene algún tipo de candado, si se supera este tiempo, se cancela la solicitud para evitar cualquier tipo de abrazo mortal

RFC6

Se genera la sentencia SQL, con el tipo de consulta y el parámetro deseado, a partir de este se crea la consulta buscando en todas las tablas implicadas, si la consulta requiere un tiempo mayor a 10 segundos significa que el recurso puede estar siendo modificado por otro usuario o que tiene algún tipo de candado, si se supera este tiempo, se cancela la solicitud para evitar cualquier tipo de abrazo mortal

RFC7

Se genera la sentencia SQL, con el tipo de consulta y el parámetro deseado, a partir de este se crea la consulta buscando en todas las tablas implicadas, si la consulta requiere un tiempo mayor a 10 segundos significa que el recurso puede estar siendo modificado por otro usuario o que tiene algún tipo de candado, si se supera este tiempo, se cancela la solicitud para evitar cualquier tipo de abrazo mortal

RF13

En la interfaz, se envía un Cliente con sus diferentes atributos al mundo, a partir de la información que tiene este cliente. El cliente tiene asociado un producto, que si esta en modificación por parte de alguien más, no se realiza la operación, una vez traspasado este filtro, existen dos opciones, que el producto a ser cancelado no tiene ningún tipo de unidades en espera de ser producidas o que tengan varias unidades en espera de ser producidas. Para el caso de que no tenga ningún tipo de unidades en espera de ser producidas, se cancela el pedido existente, este pedido se encuentra en la tabla de PEDIDOSCLIENTE, primero se define el autocommit como false, el nivel de aislamiento como TRANSACTION_READ_COMMITTED y por último un savePoint, guardado en una variable global antes de ejecutar la transacción, lo anterior se hace para evitar en primer medida cualquier tipo de intentos de modificaciones y garantizar que a lo largo de la cancelación del pedido, no haya fantasmas ni lecturas erradas.

En el otro caso, para borrar los materiales en reserva y las unidades en reserva del producto, estos números vienen incluidos en la consulta pasada, para mantener el estado actual de la base de datos, estas transacciones son similares a la de cancelar un pedido, solo que se hacen en las tablas de Producto y Materiales, reduciendo las unidades consistentemente; adicionalmente, se deja al final la eliminación del producto, ya que es la transacción de mayor envergadura. Con respecto a la atomicidad, si existe algún tipo de problemas al momento de realizar cualquier mini-traduccion, esta definido en el requerimiento que se paren todas las transacciones, luego recurra al rollback completo de todos los Savepoint guardados globalmente y finalmente se le reporta un mensaje al usuario avisando que no se pudo realizar la eliminación correspondiente. Con respecto a la durabilidad, estas transacciones, tienen su respectivo commit, enfocado a que si todos los cambios son consistentes, prevalezcan en la base de datos.

RF15

Para evitar errores de persistencia es necesario desactivar la opción de autocommit así es seguro que se guardará y actualizará solo las transacciones necesarias en la base de datos. Aparte de esto se revisa que la transacción sea menor a un tiempo determinado, pues si es mayor significa que otro usuario está intentando hacer la misma transacción con exactamente los mismos datos y la transacción es cancelada, de esta manera se evitan los abrazos mortales en este requerimiento.

También se hace el uso de rollbacks y savepoints, pues de esta manera si llegara a existir un error de cualquier tipo, la conexión vuelve a un estado (savepoint) en el que estaba funcionando de manera óptima. Aunque existe más de una transacción en este método es necesario que funcionen todas o el método no funcionará.

Debido a que se deben completar todas o ninguna de las transacciones de este método, la durabilidad se realiza mediante un commit después de lograr completar todas las transacciones.

RF16

Para evitar errores de persistencia es necesario desactivar la opción de autocommit así es seguro que se guardará y actualizará solo las transacciones necesarias en la base de datos. Aparte de esto se revisa que la transacción sea menor a un tiempo determinado, pues si es mayor significa que otro usuario esta intentando hacer la misma transacción con exactamente los mismos datos y la transacción es cancelada, de esta manera se evitan los abrazos mortales en este requerimiento.

También se hace el uso de rollbacks y savepoints, pues de esta manera si llegara a existir un error de cualquier tipo, la conexión vuelve a un estado (savepoint) en el que estaba funcionando de manera óptima. Aunque existe más de una transacción en este método es necesario que funcionen todas o el método no funcionará.

Debido a que se deben completar todas o ninguna de las transacciones de este método, la durabilidad se realiza mediante un commit después de lograr completar todas las transacciones.

RF17

Para evitar errores de persistencia es necesario desactivar la opción de autocommit así es seguro que se guardará y actualizará solo las transacciones necesarias en la base de datos. Aparte de esto se revisa que la transacción sea menor a un tiempo determinado, pues si es mayor significa que otro usuario esta intentando hacer la misma transacción con exactamente los mismos datos y la transacción es cancelada, de esta manera se evitan los abrazos mortales en este requerimiento.

También se hace el uso de rollbacks y savepoints, pues de esta manera si llegara a existir un error de cualquier tipo, la conexión vuelve a un estado (savepoint) en el que estaba

funcionando de manera óptima. Aunque existe más de una transacción en este método es necesario que funcionen todas o el método no funcionará.

Debido a que se deben completar todas o ninguna de las transacciones de este método, la durabilidad se realiza mediante un commit después de lograr completar todas las transacciones.

3. Construcción de la aplicación:

a. Ajuste las tablas creadas en Oracle de acuerdo a las decisiones del punto anterior.

Realizado en SQL Developer.

b. Poble las tablas con información suficiente para poder realizar pruebas.

i. Diseñe los datos que le permitan verificar adecuadamente las reglas de negocio. Note que es más importante generar adecuadamente los datos, que obtener un número muy grande de ellos.

ii. Puede escribir un programa de generación automática de datos acorde al diseño establecido para los mismos.

iii. Para la población de las tablas utilice herramientas de carga masiva como SQLLoader o las disponibles en SQLDeveloper. Consulte el tutorial disponible en la wiki del curso sobre SQLLoader

Realizado en SQLDeveloper.

c. Desarrolle o ajuste las clases correspondientes a los nuevos requerimientos, de forma que complete o modifique los requerimientos funcionales y cumpla con las restricciones de negocio. En particular, ajuste lo necesario para satisfacer las condiciones ACID sobre las operaciones de negocio. Realice los cambios sobre las clases que corresponden a servlets, fachada, y ConsultaDAO.

i. Desarrolle o ajuste los servlets correspondientes a la interfaz de la aplicación.

ii. Cambios en fachadas y clases .

iii. Cambios en ConsultaDAO. Todos los requerimientos tienen el mismo valor. Los bonos serán acreditados con un máximo de 2% sobre el valor total de la nota.

Realizado en el proyecto java.

d. Verifique el comportamiento transaccional de los requerimientos que implican actualización, inserción o borrado de información.

Hecho.

e. Diseñe un esquema de respaldo de datos. En máximo una página describa los procesos y la infraestructura que usted propone para que el negocio al cual corresponde la aplicación desarrollada pueda tener un esquema confiable de disponibilidad y recuperación de datos.

Se utilizará una bitácora llamada log, ubicada en memoria secundaria y primaria, con el propósito de garantizar el respaldo de la información relevante, por medio de un constante actualización de un archivo de texto, en el cual queden plasmadas, todas las transacciones, ya sean exitosas o no, y a su vez las sentencias correspondientes a cada una.

Este log, utilizará el concepto de checkpoint, en el cual cada vez que haya un proceso de modificación de la BD, tendrá que generar un LSN para cada registro y cuando esta pagina sea transferida a disco, garantizar que se tendrá un completo manejo de todas las transacciones, esto a su vez enfocado a que toda la información sea consistente, en especial al momento de presentar fallas o cualquier dificultad.

Para todo tipo de transacción que implique algún tipo de modificación sobre la base de datos se creará un Checkpoint de tipo Fuzzy, en especial para mantener un registro de las últimas transacciones que han tenido lugar; ya que si se llegase a tener algún tipo de problema con cualquier transacción, podría saber hasta cual transacción llegó y por consiguiente pasar a hacer un rollback un redo. Ya que muchas transacciones requieren realizar ciertas modificaciones, se busca generar un Checkpoint por cada transacción que genera una modificación, si estas transacciones corresponden a una mucho mayor, se utilizarán diferentes estampillas que marquen toda la transacción global, y a su vez, si esta no se pudo completar, garantizar que todas las pequeñas transacciones hagan el rollback correspondiente.

Por último, ya que la información de ProdAndes es crucial para diferentes empresas, habrán diferentes respaldos físicos, se utilizará 1 disco espejo para guardar el estado actual de la BD, a su vez cada mes se hará un backup completo de la BD. Por medio de estos mecanismo y de la bitácora, se garantizará que toda la información sea durable, que a pesar de que se presenten fallas, exista un mecanismo de respaldo en el cual queden plasmadas todas la transacciones, para evitar inconsistencias con los diferentes datos y muchos menos modificaciones no deseadas de la información.

f. Implemente los escenarios de prueba para que le permiten asegurar el correcto funcionamiento de la aplicación y la corrección y calidad de los datos en la base de datos i. En un archivo Excel documente claramente, para cada caso de uso, cuáles son los datos que le permiten realizar las pruebas, tanto para los casos de terminación exitosa como los fallidos. Indique cuáles son las respuestas esperadas que corresponden a los datos de prueba.

Adjunto en archivo excel.

g. Automatice los procesos asociados al desarrollo y despliegue utilizando Ant. Haga el despliegue de la aplicación en JBoss.

Realizado desde Java.