

# Tema 11

Integración de la información.

SQL PLSQL y Java.

# Integración de la información

- A medida que las compañías evolucionan se hace necesario compartir su información en diferentes o múltiples bases de datos y aplicaciones.
- Hay tres formas de resolver este problema.
  - Consolidar la información en una sola BD.
  - Federar la información (federating inf).
  - Compartir la información (sharing inf).

# Federating Information

- Se apoya en SQL distribuido.
- Permite acceder a diferentes BD como si toda la información estuviera en una sola.
- Mantiene la transparencia de la ubicación de la información.
- Emplea enlaces de BD para permitir a los usuarios acceder a BD remotas.
- El optimizador de consultas permite ejecutarlas en el servidor remoto y solamente devolver los datos pedidos
  - De esta forma se disminuye el trafico de la red.

# Sharing Information

- Para compartir la información se debe replicar.
  - Replicar la información en 2 o mas BD.
- Este proceso soluciona la escalabilidad y disponibilidad y además mejora el rendimiento de los sitios webs.
- La principal herramienta para compartir la información son los Oracle streams.

# Sharing Information

- Oracle streams permite la propagación y administración de datos, transacciones y eventos, ya sea dentro de una misma base de datos como entre dos ó más BD.
- Oracle streams suministra componentes que permiten a los usuarios controlar una serie de parámetros como:
  - Qué información se introduce.
  - Cómo se enruta.
  - Qué utilidad se les da a los mensajes creados en el enrutamiento.
  - Etc.

# Arquitectura Oracle Streams

- La arquitectura de Oracle streams es muy flexible.
- Contienen tres elementos o fases básicas:



# Replicación con Oracle streams

- Oracle streams es una tecnología para compartir información que automáticamente determina qué datos son relevantes y la comparte con aquellos que los necesiten.
- Este comportamiento incluye captura y administración de mensajes en la BD, incluyendo también los mensajes que encapsulan cambios DML y DDL, y su propagación a otras BD y otras aplicaciones.

# Captura de cambios DML y DDL

- Configurar Oracle streams comienza con la especificación de un objeto o un grupo de objetos a ser replicados.
- Usando el mecanismo de captura implícita se capturan y replican a una o mas BD los cambios hechos a estos objetos.



# Propagacion de los cambios en una red

- Los cambios guardados en el área de almacenamiento (staging) pueden enviarse a otras áreas de almacenamiento en otras BD.
- Los cambios realizados en cualquier BD pueden ser publicados y propagados a distintas BD en la red.
- Los administradores eligen qué cambios se propagan a cada destino y también pueden especificar las rutas a seguir.

# Resolución de conflictos y aplicación de los cambios

- Los mensajes en el área de almacenamiento se consumen por procesos que los aplican a una o más BD replicadas.
- Las BD remotas en un entorno de replicación están abiertas en modo lectura/escritura y no necesariamente tienen que ser copias idénticas de la BD fuente.
- Las BD remotas pueden actualizarse por otros medios.

# Captura de los cambios en los datos

- La captura de los cambios en la información es una característica desarrollada sobre la infraestructura de Oracle streams.
- Dicha característica identifica y captura datos que han sido añadidos, actualizados o eliminados de las tablas relacionales de la BD de Oracle, y hace que los cambios estén disponibles para ser usados por las herramientas y aplicaciones ETL.

# Entornos heterogéneos

- Oracle streams es una solución para compartir información, soportando además la replicación heterogénea entre Oracle y otros sistemas RDBMS no-Oracle.
- El empleo de Oracle Transparent Gateway (OTG), permite que los cambios DML realizados en una BD Oracle se apliquen a otros RDBMS no-Oracle.
- Para implementar la captura y aplicación de los cambios DML de un sistema Oracle a otro no-Oracle, el sistema funciona como un proxy que ejecuta el proceso de consumo y aplicación que normalmente se ejecutaría sobre un sistema Oracle replicado.

# Vistas materializadas

- Oracle streams puede operar con vistas materializadas, las cuales se pueden usar para mantener copias de datos actualizables.
- Estas vistas materializadas pueden definirse para contener una copia completa de una tabla o un subconjunto de filas de la tabla maestra que satisfagan un criterio de selección.
- Pueden existir también vistas multinivel donde una vista esta basada en la otra.

# Integración de sistemas no-Oracle

- Oracle ofrece dos soluciones para comunicarse entre sistemas Oracle y no-Oracle.
  - Generic Connectivity
  - Oracle Transparent Gateway (OTG).
- Ambos pueden usarse para conectarse de forma síncrona usando SQL distribuido.
- Además, OTG puede comunicarse de forma asíncrona a través de Oracle Streams.
- En general OTG obtiene un mejor rendimiento.

SQL

# Instrucciones SQL incrustadas

- Objetivo:
  - Incorporar dentro de un procedimiento sentencias:
    - DDL.
    - DML.
    - Instrucciones de control de transacciones.
- Permiten:
  - Definir, asignar y liberar los cursores (DECLARE CURSOR, OPEN, CLOSE).
  - Especificar una base de datos y conectarse a Oracle (DECLARE DATABASE, CONNECT).
  - Asignar nombres de las variables (DECLARE).
  - Inicializar los descriptores (DESCRIBE).
  - Especificar cómo manejar errores y avisos (WHENEVER).
  - Analizar y ejecutar sentencias SQL (PREPARE, EXECUTE, EXECUTE IMMEDIATE).
  - Recuperar los datos de la base de datos (FETCH).



# Cursor

- Es un controlador para un area SQL de memoria privada en la cual se guarda una sentencia analizada sintacticamente, su conjunto respuesta y otras informaciones.
- Se emplea para el procesamiento individual de las filas devueltas por el sistema gestor de base de datos a una consulta.
- Por norma general, los lenguajes de programación son procedimentales y no disponen de ningún mecanismo para manipular conjuntos de datos en una sola instrucción.
- Debido a ello, las filas deben ser procesadas de forma secuencial por la aplicación.
- Puede verse como un iterador sobre la colección de filas de la respuesta.

# Cursor

- Operaciones:
  - Se crea utilizando la sentencia DECLARE CURSOR para asociar la sentencia al cursor.
    - Es obligatorio asignarle un nombre.
      - DECLARE *cursor\_name* CURSOR FOR SELECT... FROM...
  - Antes de ser utilizado, el cursor debe abrirse con una sentencia OPEN que obtiene el conjunto respuesta.
    - OPEN *cursor\_name*
  - La sentencia FETCH avanza el cursor a la siguiente fila del conjunto respuesta y transfiere la información de la fila a variables de la aplicación.
    - FETCH *cursor\_name* INTO...
  - El último paso consiste en cerrar el cursor utilizando la sentencia CLOSE para desasociarlo del conjunto respuesta.
    - CLOSE *cursor\_name*
- Una vez un cursor está cerrado puede volverse a abrir, lo cual implica que la consulta se vuelve a evaluar y se obtiene un nuevo conjunto respuesta.

# SQL compartido

- Oracle automáticamente avisa a la base de datos cuando una aplicación envía sentencias SQL similares, haciendo que compartan el area privada de SQL.
- Se reduce el uso de memoria en el servidor y aumenta el rendimiento del sistema.

# Análisis sintáctico

- Es una etapa en el procesamiento de una sentencia SQL.
- Cuando una aplicación emite una sentencia SQL, se realiza una petición de análisis sintáctico a Oracle que:
  - Comprueba la validez sintáctica y semántica.
  - Determina si el proceso de ejecución de la sentencia tiene privilegios para ejecutarlo.
  - Asigna un área privada de SQL.

# El optimizador

- Todas las sentencias SQL usan el optimizador.
- Es una parte de Oracle que determina la estrategia óptima para llevar a cabo una sentencia SQL.
- Se puede influir en las elecciones del optimizador estableciendo consejos (HINTS) y recopilando estadísticas debidamente actualizadas.
- El paquete PL/SQL “DBMS\_STATS” permite recopilar estadísticas para el optimizador.

PL/SQL

# Procedural Language/SQL (PL/SQL)

- PL/SQL es un lenguaje de programación procedimental, robusto, portátil y seguro, que permite SQL incrustado.
- Se pueden definir y ejecutar unidades de programa, tales como:
  - Procedimientos.
  - Funciones.
  - Paquetes.
- Las unidades de programa PL/SQL se clasifican en:
  - Bloques anónimos.
  - Procedimientos almacenados.

# PL/SQL

- Un bloque anónimo es un bloque que aparece en la aplicación y ni tiene nombre ni se almacena en la base de datos.
  - Oracle compila el bloque PL/SQL y coloca la versión compilada en la zona compartida de la SGA, pero no lo almacena.
- Un procedimiento almacenado es un bloque PL/SQL de Oracle que se almacena en la base de datos y puede ser llamado por su nombre desde una aplicación.
  - Oracle también permite crear y almacenar en la BD funciones y paquetes.
- Las unidades de programa se pueden ejecutar, ya sea con el interprete de ejecución o con la ejecución nativa.



# Intérprete de ejecución

- En versiones anteriores al Oracle 9i, los códigos fuente PL/SQL se compilaban siempre en una representación bytecode, que eran ejecutados en la maquina virtual implementada como parte del servidor de base de datos Oracle (interprete de ejecución).
- En las versiones posteriores a Oracle 9i, ya se puede elegir entre:
  - Intérprete de ejecución.
  - Ejecución nativa (para la plataforma subyacente).

# Ejecución nativa

- Para un mayor rendimiento se compila el código fuente de las unidades de programa almacenadas en la BD directamente a código objeto para la plataforma en cuestión.

# Motor de PL/SQL

- El motor de PL/SQL es la herramienta que se emplea para definir, compilar y ejecutar unidades de programa PL/SQL.
- Este motor es un componente especial de muchos productos de Oracle, incluyendo el servidor de bases de datos Oracle.

# Sentencias SQL dinámicas

- Se pueden ejecutar sentencias SQL dinámicas cuyo texto completo no se conoce hasta el momento de la ejecución de la instrucción.
- Se pueden escribir procedimientos almacenados y bloques anónimos PL/SQL que incluyan sentencias SQL dinámicas de dos maneras:
  - Incorporar sentencias SQL dinámicas en el bloque PL/SQL.
  - Emplear el paquete de DBMS\_SQL.

# Ejecución

- Se puede ejecutar un procedimiento o una función de las siguientes maneras:
  - Utilizando un cliente de Oracle.
  - Llamando al procedimiento explícitamente en el código de una aplicación.
  - Llamando al procedimiento explícitamente en el código de otro procedimiento o disparador.

# Control de la seguridad

- Los procedimientos almacenados controlan la seguridad de una de las dos maneras siguientes:
  - Heredando los derechos del usuario que lo crea (definer's rights).
  - Heredando los derechos del usuario que lo invoca (invoker's rights).

# Funciones tabla

- Son funciones que producen como resultado un conjunto de filas.
- Se pueden emplear en lugar de las tablas normales en la cláusula FROM de las sentencias SQL.
- Su tratamiento puede ser paralelizado.

# Procedimientos independientes

- Son procedimientos almacenados que no están definidos en el contexto de un paquete.



# PL/SQL: Colección

- Grupo ordenado de elementos del mismo tipo.
- Cada elemento tiene un índice único que determina su posición en la colección.
- Se pueden usar colecciones como:
  - Varrays (arrays).
  - Tablas anidadas (conjuntos y bolsas).
  - Arrays asociativos (tablas hash y otras clases de tablas de búsqueda desordenadas).

# PL/SQL: Registros

- Contienen campos que pueden tener diferentes tipos de datos y cada uno con su nombre único.
- Los elementos son diferentes pero están lógicamente relacionados.
- Permite tratar los datos como una unidad lógica.

# Java

# Estrategia de Oracle para aplicaciones Java

- El entorno de programación Java consiste en:
  1. La máquina virtual de Java (JVM) de Oracle.
  2. Procedimientos almacenados de Java.
    - Los procedimientos almacenados de Java están estrechamente integrados con PL/SQL.
      - Se puede llamar a un procedimiento almacenado de Java desde PL/SQL.
      - Se puede llamar a procedimientos PL/SQL desde un procedimiento almacenado de Java.
  3. Los datos SQL pueden ser accedidos a través de la interfaz de programación de JDBC.
  4. Las herramientas y scripts para ayudar al desarrollo

# Java DataBase Connectivity (JDBC)

- JDBC es una API destinada a desarrolladores de Java para acceder a datos SQL.
- Está disponible en el cliente y el servidor, por lo que puede utilizar el mismo código en cualquier lugar.
- Permite el acceso a los objetos y tipos definidos en la BD desde los programas de Java a través de SQL dinámico.
- El SQL dinámico significa que la sentencia SQL embebida que se va a ejecutar no se conoce antes de que se ejecute la aplicación, y se requiere de la construcción de la declaración.

# JDBC

Oracle ofrece tres drivers JDBC distintos:

- JDBC Thin driver.
- JDBC Oracle Call Interface (OCI).
- JDBC del lado del servidor interno del controlador.

# JDBC Thin driver

- Se puede utilizar el controlador JDBC Thin para escribir aplicaciones de 100% puro Java y applets que acceden a datos Oracle SQL.
- El driver JDBC Thin es especialmente adecuado en aplicaciones Web y applets, ya que se puede descargar dinámicamente desde una página Web como cualquier otro applet de Java.

# JDBC Oracle Call Interface

- El driver JDBC OCI accede a bibliotecas en el cliente en código nativo Oracle (no Java), proporcionando un conjunto más completo de funcionalidad y mayor rendimiento en comparación con el driver JDBC Thin, a un costo de un tamaño mucho más grande y la instalación del lado del cliente.



# JDBC del lado del servidor

- Se utiliza cuando el código Java se ejecuta en el servidor.
- Permite que las aplicaciones Java que se ejecutan en la JVM del servidor accedan a los datos definidos a nivel local con JDBC (en el mismo equipo y en el mismo proceso).
- Aumenta el rendimiento debido a su capacidad para utilizar las librerías subyacentes Oracle RDBMS directamente, sin la sobrecarga de una conexión de red entre el código Java y SQL.

# SQLJ

- Permite a los desarrolladores usar tipos de datos de objetos en los programas de Java.
- Los desarrolladores pueden utilizar JPublisher para asignar objetos Oracle en las clases de Java para ser utilizados en la aplicación.
- Proporciona acceso a los objetos del servidor mediante sentencias SQL embebidas en el código Java.
- SQLJ proporciona en tiempo de compilación la comprobación de tipos de objetos y colecciones de las sentencias SQL.
- La sintaxis se basa en un estándar ANSI.

# Open Database Connectivity (ODBC)

# ODBC

- Es un protocolo de acceso a la BD que permite conectarse a una BD y luego preparar y ejecutar sentencias SQL en la BD.
- En unión con el driver ODBC, una aplicación puede acceder a cualquier fuente de datos, como por ejemplo, las hojas de Excel.
- Usando los drivers de los datos específicos de la fuente, un programa ODBC puede acceder cualquier fuente de datos sin tener que realizar desarrollo extra.