

Carrera:

Ingeniería en Software

Asignatura:

Bases de Datos

Tarea 11. Consultas y Transacciones Distribuidas

Alumno(a)s:

Bryan Salvador Vargas López

Libni Uziel Gabriel Linares

Paul Alejandro Ramírez Zaragoza

Reyna Anayeli Manríquez Cisneros

Profesor:

Dr. Sergio Valadez Godínez

Fecha de entrega: 26/11/2021

Instrucciones:

Elaborar un documento digital que contenga:

- a) Descripción del proceso de construcción de consultas en bases de datos distribuidas

a) Descripción del proceso de construcción de consultas en bases de datos distribuidas

Las consultas distribuidas detienen acceso a datos de varios orígenes de datos heterogéneos. Estos orígenes de datos pueden estar almacenados en el mismo equipo o en equipos diferentes.

El procesamiento de consultas tiene varias etapas a seguir para resolver una consulta SQL, las características del modelo relacional permiten que cada motor de base de datos elija su propia representación que, comúnmente, resulta ser el álgebra relacional. Existen varios medios para calcular la respuesta a una consulta. En el caso del sistema centralizado, el criterio principal para determinar el costo de una estrategia específica es el número de acceso al disco. En un sistema distribuido es preciso tener en cuenta otros factores como son:

- El costo de transmisión de datos en la red.
- Repetición y fragmentación.
- Procesamiento de intersección simple.

El objetivo es convertir transacciones de usuario en instrucciones para manipulación de datos. No obstante, el orden en que se realizan las transacciones afecta grandemente la velocidad de respuesta del sistema. Así, el procesamiento de consultas presenta un problema de optimización en el cual se determina el orden en el cual se hace la menor cantidad de operaciones. Este problema de optimización es NP-difícil, por lo que en tiempos razonables solo se pueden obtener soluciones aproximadas.

En BDD se tiene que considerar el procesamiento local de una consulta junto con el costo de transmisión de información al lugar en donde se solicitó la consulta. El éxito creciente de la tecnología de bases de datos relacionales en el procesamiento de datos se debe, en parte, a la disponibilidad de lenguajes los cuales pueden mejorar significativamente el desarrollo de aplicaciones y la productividad del usuario final.

Ejemplo:

Use las OPENQUERY funciones OPENROWSET transact-SQL o transact-SQL.

Use una consulta con nombres de cuatro partes, incluido un nombre de servidor vinculado.

Por ejemplo:

```
sp_addlinkedserver 'mylinkedserver', 'product_name', 'myoledbprovider',  
'data_source','location', 'provider_string', 'catalog'
```

```
SELECT *
```

```
FROM OPENQUERY(mylinkedserver, 'select * from table1')
```

Puede usar la función `OPENROWSET` en una instrucción SQL Server para pasar consultas al `OPENROWSET OPENQUERY` servidor OLAP SELECT vinculado. La consulta se limita a la sintaxis abreviada admitida por servicios OLAP; sin embargo, la consulta puede incluir la sintaxis de SELECT expresiones multidimensionales (MDX). Una consulta que incluye MDX devuelve conjuntos de filas aplanados como se describe en la documentación de OLE DB. Para obtener más información acerca de la sintaxis admitida por SQL Server servicios OLAP, vea el tema `SELECT Supported SQL SELECT Syntax` en `Olap Services Books Online`.

Para consultar una base de datos de servidor OLAP local o remota desde SQL Server, debe instalar el proveedor OLE DB de MSOLAP en el equipo que ejecuta SQL Server. El proveedor OLE DB de MSOLAP se instala al instalar los componentes de cliente OLAP desde el SQL Server.

Ejemplo de OPENROWSET y OPENQUERY

En el siguiente ejemplo SQL código transact-SQL muestra cómo configurar y usar consultas distribuidas con un servidor OLAP con las `OPENQUERY` funciones `OpenRowset` y `OpenQuery`. Debe cambiar los nombres de origen de datos y el nombre del catálogo según corresponda.

SQLCopiar

```
-----
--OPENROWSET for OLAP Server
-----
```

```
SELECT a.*
FROM OpenRowset('MSOLAP','DATASOURCE=myOlapServer; Initial Catalog=FoodMart;',
'SELECT Measures.members ON ROWS,
[Product Category].members ON COLUMNS
FROM [Sales]') as a
go
```

```
-- Example of MDX with slicing --
```

```
SELECT a.*
FROM OpenRowset('MSOLAP','DATASOURCE=myOlapServer; Initial Catalog=FoodMart;',
'SELECT
{ Time.Year.[1997] } ON COLUMNS,
NON EMPTY Store.MEMBERS ON ROWS
FROM Sales
WHERE ( Product.[Product Category].[Dairy] )') as a
```

```
-----
-- Linked Server Examples with OPENQUERY
-----
```

```
EXEC sp_addlinkedserver
    @server='olap_server',
```

```
@srvproduct='',  
@provider='MSOLAP',  
@datasrc='server',  
@catalog='foodmart'
```

```
go
```

```
-- MDX in OPENQUERY --
```

```
SELECT *  
FROM OPENQUERY(olap_server,  
'SELECT  
{ Time.Year.[1997] } ON COLUMNS,  
NON EMPTY Store.MEMBERS ON ROWS  
FROM Sales  
WHERE ( Product.[Product Category].[Dairy])' )  
SELECT *
```

```
FROM OPENQUERY(olap_server, 'SELECT [customer], [quantity] FROM sales')
```

Solo se admite una forma limitada de SQL y solo se pueden especificar nombres de nivel o medida. Al ejecutar la consulta, recibe este mensaje de error:

Servidor: Msg 7399, Level 16, State 1, Line 1 OLE DB provider 'MSOLAP' reported an error. [Mensaje devuelto por el proveedor OLE/DB : El nombre de columna "cliente" no es válido. Solo se pueden especificar nombres de nivel o de medida.]

Una forma de corregir la consulta es usar lo siguiente:

```
SQLCopiar  
SELECT *  
FROM OPENQUERY(olap_server, 'SELECT [unit sales] FROM sales')
```

Sin embargo, pasar SQL instrucciones de ese formulario al servidor OLAP puede ser lento y puede que reciba un error de tiempo de espera en algunos equipos

Bibliografía

1.4. Procesamiento de consultas distribuidas. (2016, 2 diciembre). Tópicos de Bases de Datos.

Recuperado 26 de noviembre de 2021, de <https://topicdb.wordpress.com/1-4-procesamiento-de-consultas-distribuidas-2-2/>

R. (2021, 12 septiembre). Realizar una consulta distribuida con olap server - SQL Server. Microsoft

Docs. Recuperado 26 de noviembre de 2021, de <https://docs.microsoft.com/es-es/troubleshoot/sql/analysis-services/perform-distributed-query-olap>