A picture containing text, paper clip

Description automatically generatedA picture containing logo

Description automatically generated

INSTITUTO POLITÉCNICO NACIONAL

UNIDAD PROFESIONAL INTERDISCIPLINARIA EN INGENIERÍA Y TECNOLOGÍAS AVANZADAS

BASE DE DATOS DISTRIBUIDAS

descripción de planes de ejecución

de las consultas implementadas

calderon landeros sergio david

cruz tellez luisa michelle

En este documento se va a describir los planes de ejecución de las consultas implementadas. Se mostrará el diagrama de ejecución de los planes

**Consulta 3:**

Conocer qué encargado regional cuenta con más bonos

select Sales.SalesPerson.BusinessEntityID, Person.Person.FirstName, Person.Person.LastName

from Sales.SalesPerson join Person.Person

on Sales.SalesPerson.BusinessEntityID = Person.Person.BusinessEntityID

where Bonus= (select max(Bonus) from Sales.SalesPerson);

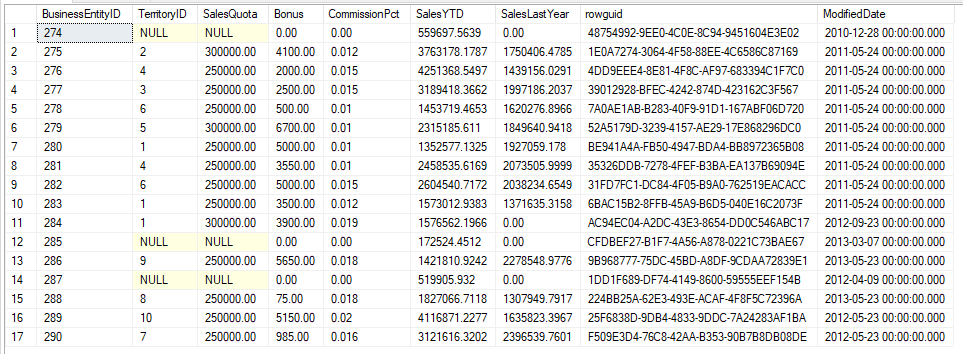
Diagram

Description automatically generated with medium confidence

**Análisis:**

Se puede apreciar que existen “Clustered Index” en las dos tablas que se van a unir para hacer la consulta, en este caso son Sales.SalesPerson unión con Person.Person. SQL Server utiliza el Nested Join debido a que es necesaria la unión par a par, ya que regresa un resultado de la unión por cada consulta de la primera tabla “Sales.SalesPerson”.

El Nested Loop que se aplica en esta consulta esta optimizado, debido a que una de las tablas es mas pequeña que la otra (*En Sales.SalesPerson existen solo 17 registros, los cuales se unen en combinación con Person.Person, la cual tiene mas de 19,000 registros*), y esto ocasiona un “bajo” costo de recursos por cada loop.



Índices ordenados en la Tabla Sales.SalesPerson

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Índices ordenados en la Tabla Person.Person

**Consulta 4:**

Listar el número total de personas que manejan cada tarjeta

select count(\*) as TotalXTarjeta, Sales.CreditCard.CardType

from Sales.CreditCard join Sales.PersonCreditCard

on Sales.PersonCreditCard.CreditCardID = Sales.CreditCard.CreditCardID

group by CardType;

Graphical user interface, application, Excel

Description automatically generated

**Análisis:**

Como bien sabemos el Hash Join es el método mas efectivo para poder unir cantidades enormes de datos no ordenados, lo cual pasa en este caso, ya que la unión de las tablas es mediante el dato “CreditCardId”, el cual se encuentra ordenado en la tabla “Sales.CreditCard”, pero este mismo no esta ordenado en la tabla “Sales.CreditCard”.

Para que estuviera optimizada la consulta, lo mejor seria que las dos tablas estuvieran ordenadas respecto a “CreditCardId”, pero es no es posible, ya que no podemos asegurar una relación 1:1 en la tabla “PersonCreditCard”, ya que no todas las tarjetas han sido asignadas a esta persona, o incluso una persona puede tener varias tarjetas.

Tabla

Descripción generada automáticamente

Parte de Sales.PersonCreditCard

Tabla

Descripción generada automáticamente

Índices ordenados de Sales.CreditCard

**Consulta 10:**

Listar las tiendas al cargo del encargado regional (SalesPerson) con ID 275

select Sales.Store.BusinessEntityID, Sales.Store.Name, Sales.SalesPerson.BusinessEntityID

from Sales.Store join Sales.SalesPerson

on Sales.SalesPerson.BusinessEntityID = Sales.Store.SalesPersonID

where Sales.SalesPerson.BusinessEntityID = 275;

Text

Description automatically generated

**Análisis:**

Dadas las dos tablas de las cuales se va a usar la unión, no es necesario optimizar la consulta, ya que estas tablas en cuestión ya se encuentran ordenadas, y se puede hacer una conexión 1:1 entre ellas.

Tabla

Descripción generada automáticamente con confianza media

Índices ordenados de BusinessEntityID en Sales.Store

Tabla

Descripción generada automáticamente

Índices ordenados de BusinessEntityID en Sales.SalesPerson

**Consulta 11:**

Listar el nombre de los encargados regionales

select Person.Person.FirstName, Person.Person.MiddleName, Person.Person.LastName

from Person.Person

where BusinessEntityID between 274 and 290;

Text

Description automatically generated

**Análisis:**

Como la consulta solo busca un rango de datos, los cuales se encuentran dentro de una lista ordenada, no hay mucho que se pueda hacer, debido a que ya está lo más optimizada posible.

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

ID ordenados en la tabla Person.Person

Consulta 12:

Eliminar cliente con número de cuenta AW00000001

delete from Sales.Customer where AccountNumber='AW00000001'

A picture containing Word

Description automatically generated

**Análisis:**

Como la consulta anterior, excepto que no se busca dentro de un rango, si no solo se busca un dato en específico, el cual es eliminado, de la tabla Sales.Customer, la cual ya se encuentra ordenada por CustomerID.

Imagen que contiene Tabla

Descripción generada automáticamente

Tabla después de eliminar al Cliente