**Bases de Datos II Primer Parcial**

**TEMA 01**

**Alumno: ……………………………………………………………………......… L.U. ……………… 23/09/24**

**1)** Considerando como criterio de optimización a la minimización del transporte de información entre los nodos distribuidos (minimizar el tráfico de red).

Sean las siguientes tablas de bases de datos distribuidas en varios nodos, pertenecientes a los alumnos de la Universidad del Nordeste:

**Nodo 1: Alumnos**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LU | Apellido y Nombres | Dirección | Fecha de Nacimiento | Cód. de Localidad |

6 bytes 40 bytes 40 bytes 8 bytes 4 bytes

Contiene: 14500 registros

Longitud del registro: 98 bytes

**Nodo 2: Localidades**

|  |  |  |
| --- | --- | --- |
| Cód. de Localidad | Nombre de la localidad | Código de Provincia |

4 bytes 45 bytes 2 bytes

Contiene: 4500 registros

Longitud del registro: 51 bytes

1. Determinar el tamaño de las relaciones.
2. Se desea obtener la siguiente consulta:

*Para cada alumno, obtener su Apellido y Nombres y el Nombre de la localidad a la cual pertenece.*

1. Teniendo en cuenta el **Nodo 3**, como nodo resultado (nodo en el cual hay que entregar el resultado de la consulta), obtener las alternativas de solución a la consulta de tipo reunión (inner join) y optar por aquella solución que resulte más adecuada de acuerdo al criterio considerado (“*minimizar el transporte de bytes, entre los distintos nodos distribuidos*”).
2. Si ahora consideramos al **Nodo 2**, como nodo resultado, obtener las alternativas de solución a la consulta de tipo reunión (inner join) y optar por aquella que resulte más óptima.
3. Para la consulta planteada, en el punto anterior (**Nodo 2 como resultado**), procesar la consulta distribuida usando el tipo de semireunión (semijoin), obtener las posibles soluciones a la consulta.

**2)** Determinar las instrucciones SQL necesarias, para la definición de las restricciones de integridad, reglas de negocio y triggers requeridos.

**a**- Crear la tabla **Clientes** que contiene información relativa a las cajas de ahorros en pesos de una entidad bancaria, con los siguientes atributos: Nro. de cuenta de caja de ahorros, CUIL, Apellido y Nombres, código de localidad, situación crediticia y saldo, teniendo en cuenta lo siguiente:

* Nro. de cuenta de caja de ahorros: Clave primaria
* Situación crediticia: puede contener los siguientes valores ‘A’, ‘M’, ‘B’
* Nro. de CUIL: tiene una integridad referencial con la tabla Afip(cuil)
* Cód. de localidad: tiene una integridad referencial con la tabla Provincia(localidad)

**b**- Crear la tabla **Debitos**, donde se registrarán aquellos importes que se debitarán de los saldos de las cajas de ahorros en pesos, sus atributos son: Nro. de cuenta de caja de ahorros, denominación, código de empresa e importe a debitar, teniendo en cuenta lo siguiente:

* Nro. de cuenta de caja de ahorros: Clave primaria
* Código de empresa: tiene una integridad referencial con la Empresa(cod\_empresa)
* Contemplar una restricción de integridad (Constraint) de nombre “Control\_Cuenta” de tal manera que nro. de cuenta de caja de ahorros, tenga una referencia externa a la tabla Clientes(nro\_caja\_ahorro)

**c**- Crear la tabla **Clientes\_de\_baja**, donde se registrarán aquellos clientes dados de baja, con los siguientes atributos: Nro. de cuenta de caja de ahorros, usuario y fecha de baja.

**d**- Crear un **trigger** de nombre **Baja\_clientesB**, que permita insertar en la tabla **Clientes\_de\_baja**, los datos de aquel cliente cuya situación crediticia sea igual a ‘B’, donde las columnas usuario y fecha de baja de esta tabla, se grabarán con las variables del sistema USER y SYSDATE.

El evento que dispara al trigger es **UPDATE** del atributo situación crediticia en la tabla Clientes, y su tiempo de acción es **BEFORE**.

**e**- Crear un **trigger** de nombre **Debitos\_pesos**, que permita descontar del saldo de la cuenta de caja de ahorros de la tabla Clientes, los importes de débitos que estén almacenados en la tabla Débitos para cada número de cuenta de caja de ahorros (saldo = saldo – importe a debitar).

El evento que dispara al trigger es **INSERT** de la tabla Debitos y su tiempo de acción es **AFTER**.