**Bases de Datos II Primer Parcial**

**TEMA 02**

**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 abonados de una empresa de telefonía celular:

**Nodo 1: Abonados**

|  |  |  |  |
| --- | --- | --- | --- |
| Nro. abonado | Apellido y Nombres | Dirección | Tipo de servicio |

8 bytes 40 bytes 30 bytes 3 bytes

Contiene: 15700 registros

Longitud del registro: 91 bytes

**Nodo 2: Servicios**

|  |  |  |
| --- | --- | --- |
| Tipo de servicio | Denominación del servicio | Costo del servicio |

3 bytes 40 bytes 7 bytes

Contiene: 4800 registros

Longitud del registro: 50 bytes

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

*Para cada abonado, obtener su Apellido y Nombres y la Denominación del servicio contratado.*

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** de 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
* Débitos: puede contener los siguientes valores de 3, 5, 7 ó 9
* 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 **Créditos**,donde se registrarán aquellos importes que se acreditarán en 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 de crédito, teniendo en cuenta lo siguiente:

* Nro. de cuenta de caja de ahorros: Clave primaria
* Código de empresa: tiene una integridad referencial con la Empresa(cód\_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 **Nuevos\_clientes**,donde se registrarán aquellos clientes que han sido dados de alta, con los siguientes atributos: Nro. de cuenta de caja de ahorros, cuil, usuario y fecha de alta.

**d**- Crear un **trigger** de nombre **Alta\_clientesA**, que permita insertar en la tabla **Nuevos\_clientes**, los datos de aquel cliente dado de alta y cuya situación crediticia sea igual a ‘A’, donde las columnas usuario y fecha de alta de esta tabla, se grabarán con las variables del sistema USER y SYSDATE.

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

**e**- Crear un **trigger** de nombre **Nuevo\_saldo**, que permita ir actualizando el saldo del Nro. de cuenta de caja de ahorros de la tabla Clientes con el importe de crédito de la tabla Créditos (debería de incrementarse el saldo de la caja de ahorros de la tabla Clientes con el importe de crédito proveniente de la tabla Créditos).

El evento que dispara al trigger es **INSERT** de la tabla Créditos, y su tiempo de acción es **BEFORE**.