**Bases de Datos II**  **Recuperatorio** **Primer Parcial** **30/09/24**

**TEMA 01**

**Alumno**: **…………………………………………………………………………………… L.U.** **………………**

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 profesores de la educación secundaria de los establecimientos educativos de la provincia de Santa Fe:

**Nodo 1: Profesores**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DNI | Apellidos y Nombres | Mail | Fecha de Nacimiento | Cód. del Establecimiento educativo |

8 bytes 60 bytes 40 bytes 8 bytes 6 bytes

Contiene: 59000 registros Longitud del registro: 122 bytes

**Nodo 2: Establecimientos educativos**

|  |  |  |  |
| --- | --- | --- | --- |
| Cód. del Establecimiento educativo | Nombre del Establecimiento educativo | Código Postal | Sitio web |

6 bytes 75 bytes 4 bytes 50 bytes

Contiene: 5500 registros Longitud del registro: 135 bytes

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

*Para cada profesor, obtener sus Apellidos y Nombres y el Nombre del Establecimiento educativo donde presta servicios.*

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 la posible solución a la consulta.
4. Determinar las instrucciones SQL necesarias, para la definición de las restricciones de integridad, reglas de negocio y triggers requeridos.
5. Crear la tabla **Usuarios** de la Dirección provincial de Energía de Corrientes, con los siguientes atributos: **nro. usuario, nro. de medidor, estado lectura anterior, fecha lectura anterior, estado lectura actual, fecha lectura actual, consumo kWh y cantidad de días facturados**, tener en cuenta lo siguiente:

* nro. usuario: Clave primaria
* Contemplar una restricción de integridad (**Constraint**) de nombre “Control\_Usuario” de tal manera que nro. Usuario, tenga una referencia externa a la tabla **Usuarios\_DPEC**(nro. Usuario)

1. Crear la tabla **Lectura\_Estados**, con los siguientes atributos: **nro. de medidor, lectura bimestral y fecha de la toma de lectura de estados**, considerar:

* nro. de medidor: Clave primaria

1. Crear un **trigger** de nombre **Actualizar\_estados**, que permita actualizar los atributos estado lectura actual y fecha lectura actual de la tabla **Usuarios**, respecto a los datos de la tabla Lectura\_Estados, además de los atributos consumo kWh y cantidad días facturados también de la tabla **Usuarios**, de la siguiente manera:

* estado lectura actual (tabla Usuarios) = lectura bimestral (tabla Lectura\_Estados)
* fecha lectura actual (tabla Usuarios) = fecha de la toma de lectura de estados (tabla Lectura\_Estados)

El evento que dispara el trigger es **INSERT** en la tabla Lectura\_Estados y su tiempo de acción es **AFTER**.

1. Crear un **trigger** de nombre **Consumo\_actual**, que permita actualizar los atributos consumo kwh y cantidad de días facturados de la tabla **Usuarios**, cada vez que se actualiza el atributo estado lectura actual de la misma tabla, de la siguiente manera:

* consumo kWh = estado lectura actual - estado lectura anterior
* cantidad de días facturados = fecha lectura actual – fecha lectura anterior

Tener en cuenta que los atributos estado lectura anterior y fecha lectura anterior, ya cuentan con la información de la anterior lectura realizada, no es necesario realizar ninguna operación sobre estos dos atributos.

El evento que dispara el trigger es **UPDATE** del atributo estado lectura actual de la tabla Usuarios y su tiempo de acción es **AFTER**.