

Tema 02 - Resolución correcta - Primer Parcial Bases Datos II 23/09/24

Ejercicio 1:

A –

Abonados – Tamaño de la relación =  $15700 \text{ registros} * 81\text{bytes} = 1,271,700$

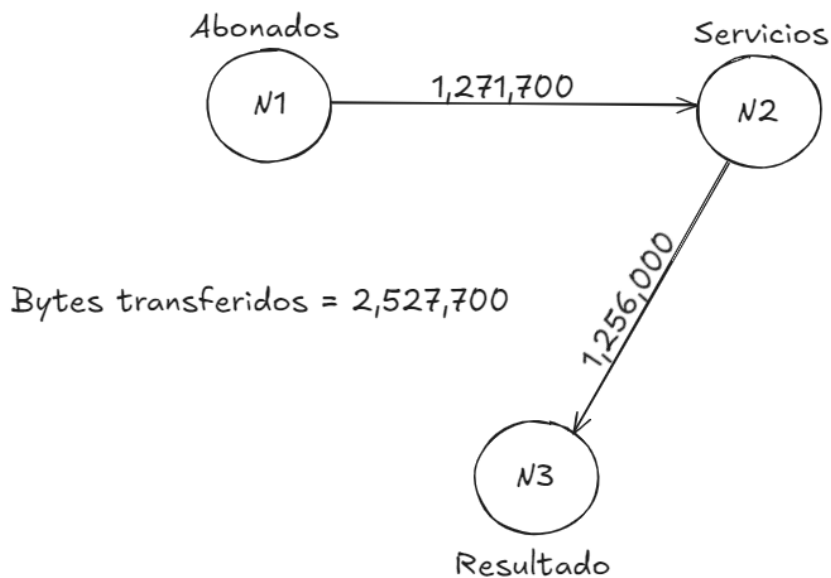
Servicios – Tamaño de la relación =  $4800 * 50\text{bytes} = 240,000$

Tamaño del resultado de la consulta =  $(40 + 40) * 15700 = 1,256,000$

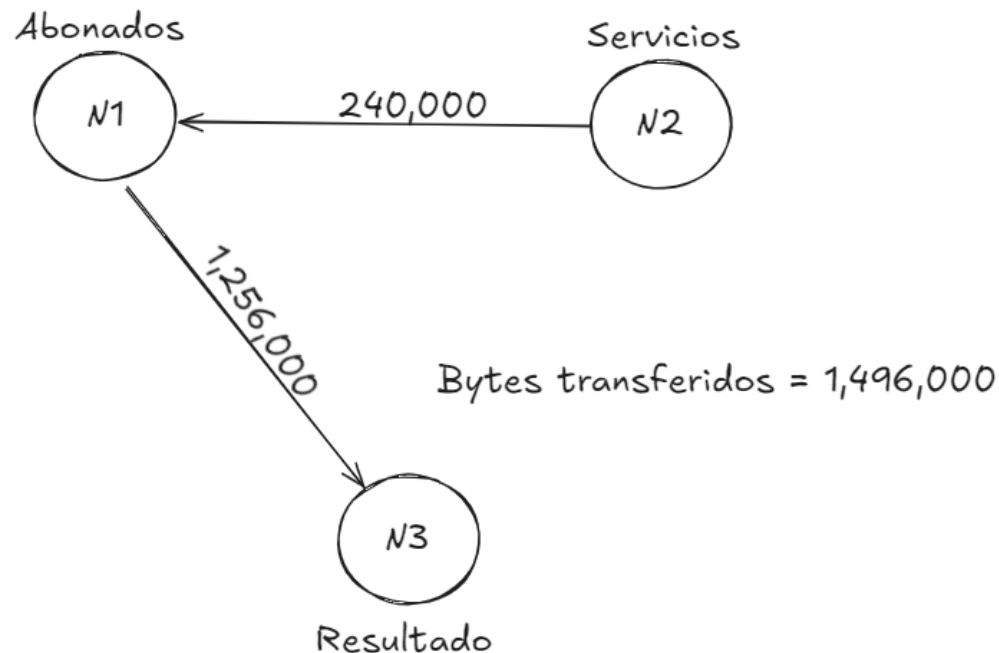
B –

i)

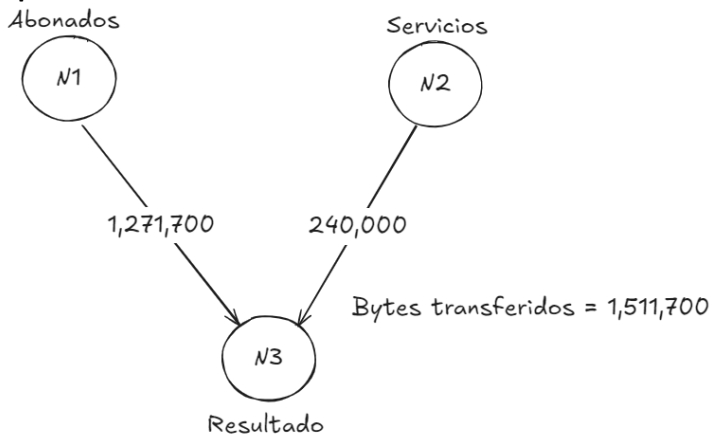
Opción 1:



Opción 2:



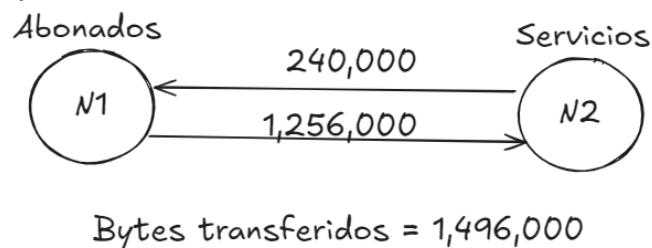
### Opción 3:



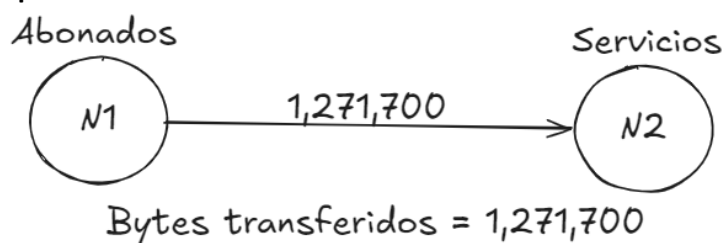
La solución mas apropiada que minimiza el transporte de bytes entre los distintos nodos distribuidos es la **Opción 1**.

ii)

### Opción 1



### Opción 2

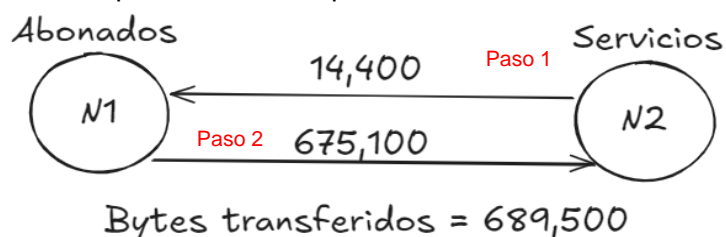


La solución mas optima es la **Opción 2**

iii)

Paso 1: 3 bytes \* 4800 tuplas = 14,400 bytes transferidos Transferencia de Nodo 2 a Nodo 1 solo atributo de enlace por el total de tuplas de la relación Servicios.

Paso 2: (3bytes(tipo servicio) + 40bytes (apellido y nombres)) \* 15700 = 675100 bytes transferidos Transferencia de Nodo 1 a Nodo 2 solo los tamaños de los atributos de enlace-tipo de servicio + apellido y nombres por el total de tuplas de la relación Abonados.



Por lo tanto, la solución obtenida en el ítem iii es aún más óptima que la evaluada en ítem ii.

2-

```
CREATE TABLE Clientes (  
    nro_caja_ahorro INT PRIMARY KEY,  
    cuil CHAR(11),  
    apellido VARCHAR(50),  
    nombres VARCHAR(50),  
    cod_localidad INT,  
    situacion_crediticia CHAR(1) CHECK (situacion_crediticia IN ('A', 'M', 'B')),  
    saldo DECIMAL(10, 2),  
    debitos INT CHECK (debitos IN (3, 5, 7, 9)),  
    CONSTRAINT fk_cuil FOREIGN KEY (cuil) REFERENCES Afip(cuil),  
    CONSTRAINT fk_cod_localidad FOREIGN KEY (cod_localidad) REFERENCES Provincia(localidad)  
);
```

```
CREATE TABLE Creditos (  
    nro_caja_ahorro INT PRIMARY KEY,  
    denominacion VARCHAR(50),  
    cod_empresa INT,  
    importe_credito DECIMAL(10, 2),  
    CONSTRAINT fk_cod_empresa FOREIGN KEY (cod_empresa) REFERENCES Empresa(cod_empresa),  
    CONSTRAINT Control_Cuenta FOREIGN KEY (nro_caja_ahorro) REFERENCES Clientes(nro_caja_ahorro)  
);
```

```
CREATE TABLE Nuevos_clientes (  
    nro_caja_ahorro INT,  
    cuil CHAR(11),  
    usuario VARCHAR(50),  
    fecha_alta DATE  
);
```

```
CREATE TRIGGER Alta_clientesA  
AFTER INSERT ON Clientes  
FOR EACH ROW  
WHEN (NEW.situacion_crediticia = 'A')  
BEGIN  
    INSERT INTO Nuevos_clientes (nro_caja_ahorro, cuil, usuario, fecha_alta)  
    VALUES (NEW.nro_caja_ahorro, NEW.cuil, USER, SYSDATE);  
END;
```

```
CREATE TRIGGER Nuevo_saldo  
BEFORE INSERT ON Creditos  
FOR EACH ROW  
BEGIN  
    UPDATE Clientes  
    SET saldo = saldo + NEW.importe_credito  
    WHERE nro_caja_ahorro = NEW.nro_caja_ahorro;  
END;
```