

Base de Datos (75.15 / 75.28 / 95.05)

Evaluación Integradora - 11 de diciembre de 2019

TEMA 20192C1						Padrón: _____ Apellido: _____ Nombre: _____ Cantidad de hojas: _____ <input type="checkbox"/> Aprobado <input type="checkbox"/> Insuficiente
CRT		Proc.		Esp.		
CyT		Rec.		NoSQL		
Corrigió: Nota:						

Criterio de aprobación: El examen está compuesto por 6 ítems, cada uno de los cuales se corrige como B/B-/Reg/Reg-/M. Se aprueba con nota mayor o igual a 4(cuatro), equivalente a desarrollar el 60 % del examen correctamente.

1. (*Cálculo Relacional de Tuplas*) El *Club del Vino* posee una base de datos con todos los vinos que tiene disponibles en su bodega. Las siguientes tablas de dicha base mantienen información sobre la cantidad de botellas disponibles de cada vino, la bodega y cosecha de la que el mismo proviene, y su composición. La composición de un vino se define como el porcentaje de cada cepa que el mismo contiene.

- Bodegas(nombre_bodega, localidad)
- Vinos(nombre_vino, nombre_bodega, color, acidez, porc_alcohol)
- Cepas(nombre_cepa, color)
- ComposicionesVinos(nombre_vino, nombre_cepa, porcentaje)
- BotellasDisponibles(nombre_vino, cosecha, cant_disponible, precio)

Un socio acaba de pedir una botella de vino '*Pinot Noir*'. Escriba una expresión en Cálculo Relacional de Tuplas que devuelva todos los vinos varietales *Pinot Noir* de los que existan unidades disponibles, indicando el nombre del vino, la bodega de la que proviene, el año de cosecha y el precio.

Nota: Un vino se considera un varietal cuando contiene un porcentaje igual o mayor al 85 % de su cepa principal. Esto no implica que no pueda contener porcentajes minoritarios de otras cepas.

2. (*Procesamiento de consultas*) Describa brevemente el método de *junta hash GRACE*, justificando su costo e indicando qué hipótesis supone sobre la cantidad de memoria disponible para garantizar dicho costo.

3. (*Bases de Datos Espaciales*) El *Cuerpo de Expertos en Bases de Datos* de la Policía (CUE.BA.DA) está tras la pista de un asesino serial que ha cometido varios crímenes en el país. El listado de asesinatos cometidos se encuentra en una tabla **Crímenes**(nro_crimen, lugar_crimen, fecha_crimen), en donde el atributo lugar_crimen es un atributo geométrico de tipo **Point**. Por otra parte, la Policía tiene acceso a una tabla con todas las conexiones registradas desde teléfonos móviles a cualquier antena del país: **Conexiones**(nro_línea, fecha_conexión, antena_conexión), en donde el atributo antena_conexión también es un atributo geométrico de tipo **Point**, y representa las coordenadas de la antena a la que se conectó el dispositivo móvil. En la *Tabla 1* se detalla la estructura de ambas tablas.

El objetivo es encontrar a los 10 números de línea que tienen más chances de pertenecer al asesino, de acuerdo al modelo que se describe a continuación:

- a) Primero se encontrará para cada usuario y para cada asesinato la conexión más tardía registrada antes de dicho asesinato, y la primera conexión realizada desde el instante de dicho asesinato en adelante.

Escriba una consulta SQL que encuentre para cada usuario y para cada asesinato, la conexión inmediatamente anterior y la de dicho instante o inmediatamente posterior. La tabla resultante deberá tener las siguientes columnas:

ConexionesCercanas(nro_línea, nro_crimen, lugar_crimen, fecha_crimen, antena_anterior, fecha_anterior, antena_posterior, fecha_posterior).

- b) En segundo paso se encontrará para cada usuario la posición estimada \hat{x}_u en el instante de cada asesinato c . Para ello se interpolarán la posición anterior x_a y la posterior x_p de acuerdo al instante en que se registraron (t_a y t_p) y al instante en que ocurrió el crimen, t_c :

$$\hat{x}_u = x_a + \frac{t_c - t_a}{t_p - t_a}(x_p - x_a)$$

Observe que cuando el crimen ocurrió muy cerca del instante t_a , dicha fórmula estima que la posición del usuario era x_a , mientras que si el crimen ocurrió muy cerca del instante t_p la fórmula estima que el usuario se encontraba en x_p .

```
CREATE TABLE Crímenes (
  nro_crimen INT PRIMARY KEY,
  lugar_crimen GEOMETRY(POINT),
  fecha_crimen DATETIME
);

CREATE TABLE Conexiones (
  nro_línea INT,
  fecha_conexión DATETIME,
  antena_conexión GEOMETRY(POINT),
  PRIMARY KEY (nro_línea, fecha_conexión)
);
```

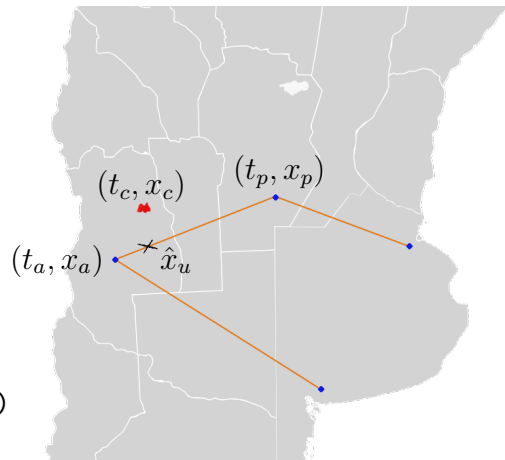


Tabla 1: (*Izq.*) Estructura de las tablas iniciales. (*Der.*) Esquema de un posible recorrido realizado por un usuario, y sus conexiones inmediatas anterior (t_a) y posterior (t_p) a un crimen c .

Escriba una consulta SQL que encuentre, a partir de la tabla `ConexionesCercanas`, la posición estimada de cada usuario en cada crimen. La tabla resultante deberá tener las siguientes columnas:

`UbicacionesEst(nro_línea, nro_crimen, lugar_crimen, posicion_estimada)`.

Para resolver este punto, considere el uso de las siguientes dos funciones de SQL/MM:

- `ST_MakeLine(g1, g2)`: Permite construir un `Line` a partir de dos `Points`.
- `ST_Line_Interpolate_Point(g, fraction)`: Interpola los extremos de una línea a partir de la fracción indicada en la ecuación anterior. Devuelve como resultado un `Point`.

- c) Finalmente se calculará para cada usuario la distancia promedio a la que se estima que se encontraba en el instante de cada crimen.

Escriba una consulta SQL que encuentre, a partir de la tabla `UbicacionesEst`, la distancia promedio de cada usuario a los crímenes, devolviendo (`nro_línea`, `distancia_promedio`) para aquellos 10 usuarios cuya distancia promedio es más chica. Para este punto considere utilizar la función `ST_Distance(g1, g2)`, que calcula la distancia entre dos geometrías.

- d) (*Bonus track*) ¿Cree que sería más útil indexar un atributo espacial o un atributo temporal? Proponga un índice concreto para una de las tablas, que crea que podría optimizar significativamente las consultas, justificando su respuesta. Indique si sería un índice simple o compuesto, y si sería primario, secundario o de clustering.

4. (*Concurrencia y transacciones*) Dado el siguiente solapamiento de transacciones:

$b_{T_1}; R_{T_1}(X); b_{T_2}; R_{T_2}(Y); W_{T_2}(Z); c_{T_2}; b_{T_3}; R_{T_3}(Z); R_{T_1}(Y); R_{T_3}(X); c_{T_3}; W_{T_1}(Y); W_{T_1}(X); c_{T_1}$

- a) Dibuje el grafo de precedencias del solapamiento.
- b) Indique si el solapamiento es serializable. Justifique su respuesta.
- c) Indique si el solapamiento es recuperable. Justifique su respuesta.

5. (*Recuperación*) Un SGBD utiliza un *log* de tipo REDO con *checkpoint* activo. Indique si cada una de las siguientes afirmaciones sobre el funcionamiento del mismo es verdadera ó falsa, justificando su respuesta.

- a) El registro de *log* correspondiente a un *write_item(X)* debe ser volcado a disco después de volcar el nuevo valor de *X* modificado por dicha instrucción a disco.
- b) El registro de *log* correspondiente al *commit* de una transacción T_i debe ser volcado a disco después de volcar el nuevo valor de todo ítem *X* modificado por T_i a disco.
- c) En caso de recuperación y necesidad de *rollback* de una transacción T_i que no había llegado a hacer su *commit*, es necesario reescribir en disco para cada ítem modificado por T_i su valor previo a la ejecución de T_i .
- d) La existencia de un registro (END CKPT) garantiza que todas las transacciones que tienen su (BEGIN, T_i) antes de dicho (END CKPT) ya tienen sus modificaciones volcadas a disco.

6. (*NoSQL*) Explique en qué consiste el concepto de *agregado* descrito por Martin Fowler y proveniente del *Domain Driven Design*. Indique qué relevancia tiene en el contexto de las bases de datos NoSQL, y ejemplifique de qué forma es implementado en algún tipo de ellas.