Base de Datos (75.15 / 75.28 / 95.05)

Evaluación Integradora - 13 de febrero de 2019

TEMA 20182C3					
SQL		**Proc.**		**DR**	
CyT		**NoSQL**		**DW**	
Corrigió: **Nota:**					

Padrón: _____

Apellido: _____

Nombre: _____

Cantidad de hojas: _____

☐ Aprobado ☐ Insuficiente

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. *(SQL)* Un club de golf mantiene en una base de datos un registro de todos sus hoyos, las personas afiliadas al club, las distintas partidas jugadas entre ellas y la puntuación de cada jugador para cada hoyo en cada una de las partidas:

   - `Hoyos(nro_hoyo, hándicap, par)`
   - `Socios(nro_socio, apellido, nombre)`
   - `Partidas(cod_partida, fecha, hora)`
   - `JugadoresPartidas(cod_partida, nro_socio)`
   - `Puntuaciones(cod_partida, nro_hoyo, nro_socio, puntuación)`

   Escriba una consulta SQL que liste el número de socio, apellido y nombre de las personas que hayan logrado alguna vez realizar al menos 3 *birdies* en una misma partida. Se dice que una persona realiza un *birdie* cuando obtiene una puntuación 1 unidad menor al par del hoyo (por ejemplo, si un hoyo tiene par 4 y el jugador lo completa en 3 golpes).

   Nota: En una *partida* de golf pueden jugar entre dos y cuatro personas, cada una de las cuales deberá completar una cantidad de *hoyos* (generalmente 9 ó 18). Cada hoyo de un campo de golf posee un *par* predeterminado –que es la cantidad de golpes esperada en que un profesional metería la pelota en dicho hoyo– y un *hándicap* predeterminado que mide la dificultad. Para cada persona y cada hoyo se registra la *puntuación* obtenida en la partida, definida como la cantidad de golpes realizados por la persona hasta meter la pelota en ese hoyo.

2. *(Procesamiento de Consultas)* Se quiere realizar la siguiente junta por igual ($\bowtie$) de dos relaciones $R(\underline{A}, B, \underline{C})$ y $S(\underline{C}, D)$, por el método de *sort-merge*:

$$R \bowtie_{R.B=S.D} S$$

a) Explique brevemente el origen de la siguiente fórmula de estimación del costo, describiendo los movimientos de lectura/escritura de bloques que se deberá realizar para completar la junta, suponiendo una cantidad mínima de 3 bloques de memoria disponibles.

$$cost(R * S) = B(R) + B(S) + 2 \cdot B(R) \cdot log_2\left(B(R)\right) + 2 \cdot B(S) \cdot log_2\left(B(S)\right)$$

b) Indique de cuál de las siguientes formas deberá ordenarse cada una de las relaciones para realizar dicha junta. Justifique su respuesta.

1) Ambas deberán ordenarse por el atributo en común, C.
2) Cada una deberá ordenarse por el atributo utilizado en la condición de junta. Es decir, R deberá ordenarse por B y S deberá ordenarse por D.
3) Cada una deberá ordenarse por su clave primaria. Es decir, R deberá ordenarse por A y S deberá ordenarse por C.

3. *(Diseño Relacional)* Dada una relación $R(A, B, C, D, E)$ con el siguiente conjunto de dependencias funcionales $F = \{A \rightarrow D, B \rightarrow CE, CD \rightarrow B, E \rightarrow D\}$, determine utilizando el algoritmo *chase* si la siguiente descomposición ρ de R preserva la información. Justifique su respuesta.

$$\rho = \{R_1(A, B, C), R_2(A, C, E), R_3(B, D)\}$$

Nota: Se dice que una descomposición $\rho = \{R_1(\bar{Z}_1), R_2(\bar{Z}_2), ..., R_n(\bar{Z}_n)\}$ de una relación $R(\bar{Z})$ preserva la información cuando *"para toda instancia de $R(\bar{Z})$ la junta natural de las proyecciones sobre cada $\bar{Z}_i$ es igual a dicha instancia."*

4. *(Concurrencia y transacciones)* Considere el siguiente solapamiento entre 4 transacciones:

$$\mathtt{b_{T_1}}; \mathtt{R_{T_1}(X)}; \mathtt{W_{T_1}(Y)}; \mathtt{c_{T_1}}; \mathtt{b_{T_2}}; \mathtt{b_{T_3}}; \mathtt{R_{T_2}(Y)}; \mathtt{R_{T_3}(Y)}; \mathtt{W_{T_2}(Y)}; \mathtt{W_{T_2}(Z)}; \mathtt{c_{T_2}}; \mathtt{c_{T_3}}; \mathtt{b_{T_4}}; \mathtt{R_{T_4}(Z)}; \mathtt{W_{T_4}(Y)}; \mathtt{c_{T_4}};$$

Se pide:

a) Construya el grafo de precedencias del solapamiento.

b) Determine si el solapamiento es serializable ó no, justificando su respuesta.

5. *(NoSQL)* La *Facultad de Ingeniería* quiere obtener un histograma del promedio con que egresan los alumnos, calculando para cada nota promedio (truncada a un dígito) la cantidad de alumnos que egresan con esa nota. Para ello dispone de una base de datos en *MongoDB*, en cuya colección *Graduados* se encuentran cargadas las notas finales de cada alumno en cada materia. Los documentos de la colección *Graduados* poseen la siguiente estructura:

```
{
  "_id": ObjectId("207ca8372c83b6b092d12512"),
  "padron": 97294,
  "apellido": "Martínez",
  "nombres": "Gonzalo",
  "notas": [
      { "codigo_materia": 7541,
        "nombre_materia": "ALGORITMOS Y PROGRAMACIÓN II",
        "nota_final": 8
      },
      { "codigo_materia": 6620,
        "nombre_materia": "ORGANIZACIÓN DE COMPUTADORAS",
        "nota_final": 9
      },
        ...
      ]
}
```

Escriba una consulta en *MongoDB* que permita obtener como resultado el histograma de notas. El formato del resultado deberá ser:

```
{
  "nota_promedio": 7.7,
  "cantidad_alumnos": 184
}
{
  "nota_promedio": 7.8,
  "cantidad_alumnos": 240
}
...
```

Para recordar parte de la sintaxis de *MongoDB* puede ayudarse con la siguiente consulta que cuenta la cantidad de alumnos egresados, agrupándolos por su número de padrón truncado por las decenas de mil (es decir, con padrón `8xxxx`, `9xxxx`, etc.). Observe que el operador `$trunc` no permite especificar la posición decimal en que se debe truncar el número.

```
[
  { $project: { padron_truncado: { $multiply:
                                   [{ $trunc: { $divide: ["$padron", 10000] } },
                                    10000] } } },
  { $group:   { _id: "$padron_truncado", cantidad_graduados: {$sum: 1} } }
]
```

6. *(Data Warehousing)* Explique el concepto de *cubo de datos* utilizado en Data Warehousing, y describa brevemente los tipos de operaciones que se pueden realizar sobre él.