

Base de Datos (75.15 / 75.28 / 95.05)

Evaluación Integradora - 28 de febrero de 2018

| | | | | | | |
|-------------------------------|--|-------------|--|--------------|--|--|
| TEMA 20172C5 | | | | | | Padrón: _____ Apellido: _____ Nombre: _____ Cantidad de hojas: _____ <input type="checkbox"/> Aprobado <input type="checkbox"/> Insuficiente |
| SQL | | DR | | DML | | |
| 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. (*SQL*) Dadas las siguientes relaciones que guardan información sobre los alumnos de una facultad y sus calificaciones:

- Alumnos(padrón, apellido, nombre)
- Notas(padrón, materia, fecha, nota)

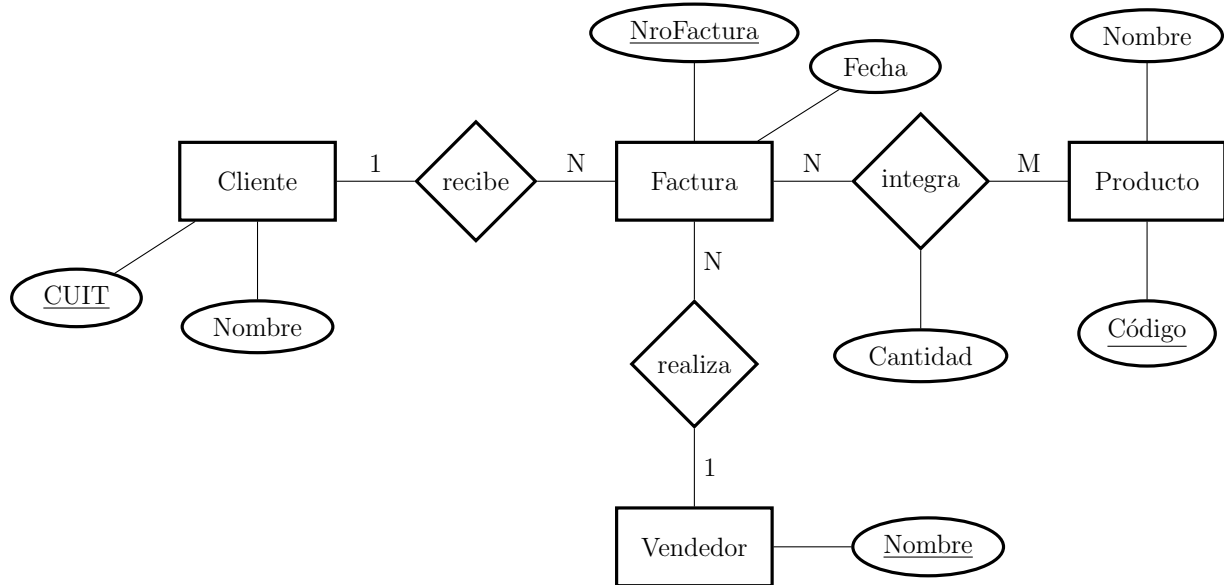
Considere la siguiente consulta SQL que busca para cada alumno su mejor nota histórica:

```
SELECT DISTINCT a.padron, a.apellido, a.nombre, n.nota as mejor_nota
FROM Alumnos a, Notas n
WHERE a.padron = n.padron
AND n.nota >= ALL (SELECT n2.nota
                   FROM Alumnos a2, Notas n2
                   WHERE a2.padron=a.padron AND a2.padron=n2.padron);
```

Esta consulta incluye una subconsulta *correlacionada* (es decir, que hace referencia a atributos de las tablas de la consulta externa). Típicamente la existencia de una subconsulta correlacionada genera un mayor costo de procesamiento porque la misma debe reejecutarse para cada una de las tuplas obtenidas en la consulta externa, a menos que el motor de base de datos logre optimizar la ejecución para evitarlo.

Escriba una consulta SQL equivalente a la anterior, que no contenga subconsultas correlacionadas.

2. (*Diseño Relacional*) El siguiente diagrama Entidad-Interrelación registra las ventas de un supermercado, identificando a los clientes, los productos vendidos en cada factura y el vendedor que realizó la operación:



Se pide:

- Considerando la relación universal $U(\text{CUIT}, \text{nombre_cliente}, \text{nro_factura}, \text{fecha}, \text{nombre_vendedor}, \text{cod_producto}, \text{nombre_producto}, \text{cantidad})$, proponga un conjunto de dependencias funcionales F que capture todas las dependencias funcionales existentes de acuerdo al diagrama Entidad-Interrelación presentado.
- Aplicando el algoritmo de descomposición a Forma Normal Boyce-Codd (FNBC), halle una descomposición a FNBC del esquema universal dado.
- Indique si la descomposición que obtuvo preserva todas las dependencias funcionales halladas en el punto *a*). Justifique su respuesta.

Nota: Se le sugiere utilizar la siguiente notación abreviada para los atributos: $U(C, NC, NF, F, NV, CP, NP, Q)$.

3. (*Lenguajes de Manipulación de Datos*) Dadas las siguientes relaciones que almacenan información sobre tenistas, torneos y los tenistas vencedores de cada uno:

- Tenistas(nombre_tenista, país, altura, diestro)
- Torneos(nombre_torneo, tipo_torneo)
- Campeones(nombre_tenista, nombre_torneo, modalidad, año)

Considere la consulta ‘*Listar los nombres de los tenistas que ganaron al menos 5 torneos de tipo **Grand Slam** en su vida*’. Indique si esta consulta es expresable en el lenguaje del Cálculo Relacional de Tuplas (CRT), si es expresable en el lenguaje del Álgebra Relacional con los operadores básicos, y si es expresable en lenguaje SQL. Justifique su respuesta.

Nota: No es necesario que traduzca la consulta a ninguno de estos lenguajes.

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

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

- a) Dibuje el grafo de precedencias del solapamiento.
 - b) Indique si el solapamiento es serializable. Justifique su respuesta.
5. (*Recuperación*) Un SGBD implementa el algoritmo de recuperación UNDO/REDO con checkpoint activo. Luego de una falla, el sistema encuentra el siguiente archivo de log:

```
01 (BEGIN, T1);
02 (WRITE T1, A, 10, 15);
03 (BEGIN, T2);
04 (WRITE T2, B, 20, 40);
05 (WRITE T1, C, 30, 35);
06 (BEGIN CKPT, T1, T2);
07 (WRITE T2, D, 40, 50);
08 (COMMIT, T2);
09 (WRITE T1, B, 40, 60);
10 (BEGIN, T3);
11 (WRITE T3, E, 15, 30);
12 (END CKPT);
13 (WRITE T3, D, 50, 15);
```

Explique cómo se llevará a cabo el procedimiento de recuperación, indicando qué cambios deben ser realizados en disco y en el archivo de log.

Nota: Los registros WRITE tienen el siguiente formato: (WRITE, T_i , X, v_{old} , v_{new}), en donde T_i identifica a la transacción, X identifica al ítem, v_{old} representa su valor anterior y v_{new} representa su nuevo valor.

6. (*NoSQL*) Una red social almacena las publicaciones de sus usuarios en una base de datos MongoDB. El siguiente documento JSON ejemplifica la estructura almacenada por cada publicación, que incluye: un identificador para la publicación, el día de publicación, el texto, un vector con los *hashtags* (los mismos que fueron incluidos en el texto), y algunos datos sobre el usuario directamente embebidos en el documento.

```
1  {
2    "_id": ObjectId("563c42d26b69693c4e61f98d"),
3    "pub_id": 672424855848215452,
4    "created_at": Date("2018-02-06"),
5    "text": "Solo un poco de agua, todavia sirve, #TodaviaSirve #Simpsons",
6    "hashtags": ["TodaviaSirve", "Simpsons"],
7    "user": {
8      "id": 318720170,
9      "name": "Juani",
10     "followers_count": 517,
11     "friends_count": 378,
12     "created_at": Date("2016-10-30")
13   }
14 }
```

Con el objetivo de encontrar los temas del momento en la red social (*trending topics*), un analista ejecutó la siguiente consulta en MongoDB que busca los 3 *hashtags* más usados (en términos de cantidad de publicaciones en que aparecieron) desde ayer hasta ahora, listando su nombre y la cantidad de usos que tuvieron:

```
[
  { $match: { created_at: { $gte: new Date("2018-02-27") } } },
  { $unwind: "$hashtags" },
  { $group: { _id: "$hashtags", cantidad: { $sum: 1 } } },
  { $sort: { cantidad: -1 } },
  { $limit: 3 }
]
```

Uno de los *hashtags* de mayor actualidad resultó ser #LollaAR. Para comprender cómo evolucionó el interés por este tema en la red, escriba una consulta en MongoDB que devuelva la cantidad de usos por día del *hashtag* #LollaAR, ordenada por fecha. El formato de los documentos resultantes deberá ser:

```
1  {
2    "_id": Date("2018-02-05"),
3    "cantidad_usos": 23
4  }
5  {
6    "_id": Date("2018-02-06"),
7    "cantidad_usos": 38
8  }
9  ...
```