



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
ESCUELA DE INGENIERÍA  
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2413 2016-2 Bases de Datos

## Examen

### Pregunta 1: SQL

a) [4 pts] Considere el siguiente esquema:

Computador (id, modelo, idproveedor, preciounidad)

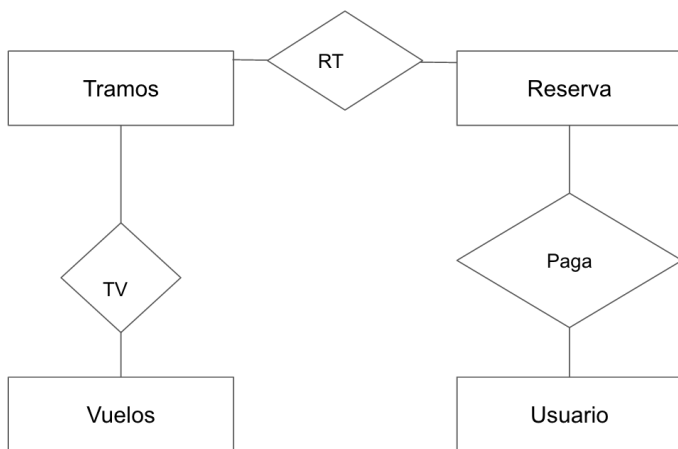
Proveedor (pid, nombre, ciudad, país)

Licitaciones (pid1, pid2, monto)

Donde cada licitación es peleada por dos proveedores y es de un monto fijo.

- [1 pto] Para cada licitación, queremos los nombres de los dos proveedores involucrados y el monto de la licitación.
- [1 pto] Nombre de los proveedores con computadores que cuesten sobre 500000.
- [2 ptos] Numero de proveedores por país, sin incluir China, ordenados de mayor a menor. Debe sólo incluir países con más de 5 proveedores.

b) [2 pts] Considere el siguiente diagrama:



Un usuario puede hacer Reservas, las que estan compuestas de distintos tramos (ida, vuelta, por ejemplo). Los vuelos se refiere a la información fija de la aerolínea, por ejemplo el vuelo 500 de LAN que vuela todos los días a las 10 am a Bélgica, los tramos estan asociados a un vuelo.

- Agregar los tipos de relaciones (1 : 1, 1 : N, N : N) al diagrama.
- Entregar las llaves primarias del esquema.

```

        Usuario (rut, nombre)
        Paga (rut, idreserva, monto)
    Reserva (idreserva, díadeemisión, medio)
        RT (idreserva, idtramo)
        Tramos (idtramo, día, asiento)
            TV (idtramo, idvuelo)
        Vuelos (idvuelo, numero, modeloavión, numasientos)

```

## Pregunta 2: Recursión

Un alumno obsesivo del ramo IIC2413 decidió modelar un asado de fin de año con un documento XML. El nodo raíz del documento XML es **Asado**, y tiene una cantidad no fija de hijos, que son nodos de tipo **Componente**. Estos poseen un atributo **id** y otro atributo **nombre**. Cada nodo **Componente** corresponde a un componente del asado, que también contiene una cantidad no fija de hijos de tipo **Componente**. Una posible instancia de este documento sería:

```

<Asado>
  <Componente id=1 nombre="Parrilla">
    <Componente id=3 nombre="Ruedas">
      <Componente id=5 nombre="Goma">
      </Componente>
    </Componente>
    <Componente id=6 nombre="Tornillos">
    </Componente>
  </Componente>
  <Componente id=2 nombre="Carne">
    <Componente id=4 nombre="Proteinas">
      <Componente id=7 nombre="Aminoacidos">
      </Componente>
    </Componente>
    <Componente id=8 nombre="Agua">
    </Componente>
  </Componente>
</Asado>

```

Sabemos además que en XPath la siguiente consulta entrega el nombre de todos los elementos que conforman a la componente  $i$  del asado:

```
//Componente[@id = i]/descendant::*/@nombre
```

El problema es que al parrillero no le gusta XML e insiste en que las bases de datos relacionales son superiores, por lo que creó un modelo relacional con una tabla **Componente** y otra tabla **Subcomponente**:

```

Componente(id int, nombre varchar(30))
Subcomponente(id1 int, id2 int, primary key(id1, id2),
              foreign key(id1) references Componente(id),
              foreign key(id2) references Componente(id))

```

La tabla componente posee todas las componentes según su id y su nombre. La tabla subcomponente contiene al par  $(i, j)$  si en el XML el nodo  $j$  es hijo del nodo  $i$ .

Entregue una consulta en SQL que compute en su modelo relacional la consulta en XPath descrita anteriormente.

T1	T2	T3	T4
$X$	$R(b)$ $W(a)$	$Y$ $W(b)$	$R(c)$
$Z$			$R(c)$

Table 1: Schedule pregunta 3 parte c

## Pregunta 3: Índices y transacciones

a) [1.5 pts] Explique la diferencia entre un índice Clustered (que sea Clustered File) y un índice Unclustered.

c) [2.5 pts] Sea la relación  $R(a, b, c, d)$  cuyo tamaño es de 4 millón de tuplas, en que cada página contiene en promedio  $P$  tuplas. Las tuplas de  $R$  están ordenados de manera aleatoria (cuando no hay índice). El atributo  $a$  es además un candidato a llave primaria, cuyos valores van del 0 al 7.999.999 (distribuidos uniformemente). Para cada una de las consultas a continuación, diga el número de I/O que se harán en cada uno de los siguientes casos:

- Analizar  $R$  sin ningún índice.
- Usar un  $B+Tree$  *Unclustered* sobre el atributo  $a$ . El árbol es de altura  $h$ . La cantidad de punteros **máxima** por pagina es de  $M$ . La ocupación de las hojas es del 50%.

Las consultas son:

1. Encontrar todas las tuplas de  $R$ .
2. Encontrar todas las tuplas de  $R$  tal que  $100 \leq a < 400$ .
3. Encontrar todas las tuplas de  $R$  tal que  $a = 100303$ .

Comente sus resultados e indique que índice prefiere en los distintos casos.

d) [2 pts] Sea el schedule del cuadro 1, de un valor para  $X$ ,  $Y$  y  $Z$  tal que representen acciones de escritura ( $W$ ) sobre variables ( $a$ ,  $b$ ,  $c$ ) para que el schedule sea:

- Serializable
- No Serializable

Justifique su respuesta.

## Pregunta 4: MongoDB

Piense en una base de datos en MongoDB con dos colecciones, una de usuarios de una red social y otra de compras. Se muestra a continuación una instancia de documento para cada colección:

```
// Usuario
{
  "id_usuario": 1,
  "name": "Florencia Barrios",
  "desc": "Me encanta jugar hockey!"
}

// Compra
{
  "id_usuario": 1,
  "id_compra": 1,
  "fecha": "10-09-2016",
  "compras": [
    {
      "producto": "Bastón de Hockey",
      "tipo": "Deporte",
      "valor": 20000
    },
    {
      "producto": "Café con leche",
      "tipo": "Comida",
      "valor": 1000
    }
  ]
}
```

A un usuario le puede corresponder más de un documento de la colección compra. Se pide que entregue las siguiente consulta en MongoDB:

- Para cada persona que en su descripción contenga la frase “Guitarrista de Metallica” y no la frase “Guitarrista de Megadeth” indique el nombre de la persona junto al id de cada compra, junto al promedio de los valores de los productos adquiridos en esa compra.

Si va a utilizar un índice indique cómo lo creó. Puede hacer uso de JavaScript o Python en caso de ser necesario.

## Pregunta 5: XML

Considere los documentos XML cuya estructura es como la siguiente:

```
<div id="cities">
  <country nombre="Chile">
    <city nombre="Santiago">
      <habitantes>6.158.080</habitantes>
      <superficie>154.032</superficie>
    </city>
    <city nombre="Viña del Mar">
      <habitantes>311.399</habitantes>
      <superficie>121.600.000</superficie>
    </city>
    <city nombre="Punta Arenas">
      <habitantes>123.401</habitantes>
      <superficie>17.846.300.000</superficie>
    </city>
  </country>
</div>
```

```

    </city>
  </country>
<country nombre="Argentina">
  <city nombre="Buenos Aires">
    <habitantes>2.890.151</habitantes>
    <superficie>203,3</superficie>
  </city>
  <city nombre="Mendoza">
    <habitantes>114.822</habitantes>
    <superficie>54.000.000</superficie>
  </city>
</country>
<country nombre="Brasil">
  <city nombre="Brasilia">
    <habitantes>2.977.216</habitantes>
    <superficie>5.802.000.000</superficie>
  </city>
  <city nombre="Rio de Janeiro">
    <habitantes>6.453.682</habitantes>
    <superficie>1.260.029.215</superficie>
  </city>
</country>
<country nombre="Perú">
  <city nombre="Lima">
    <habitantes>7.605.742</habitantes>
    <superficie>2.672</superficie>
  </city>
</country>
</div>

```

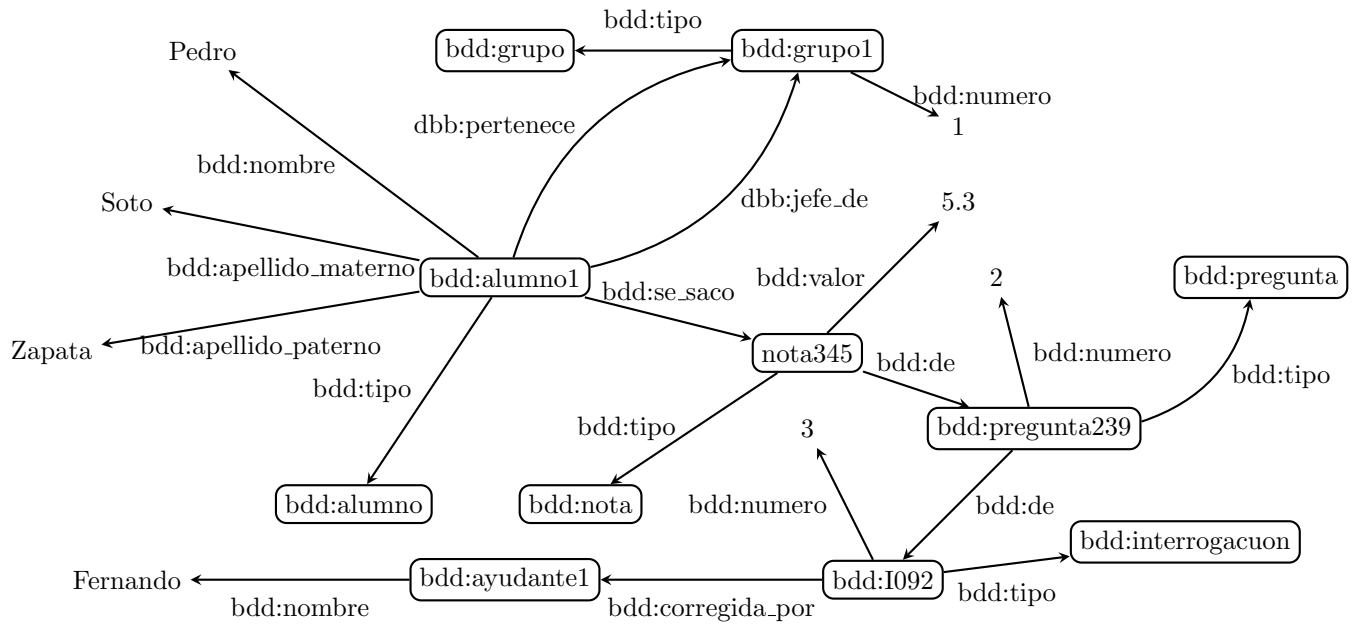
Estos documentos pueden tener muchos más países y cada país puede tener muchas ciudades. Escriba las siguientes consultas en XPath:

- Obtenga todos las ciudades con más de 1.000.000 de habitantes.
- Obtenga los países que tienen dos o más ciudades con una superficie entre 200 y 54.000.000  $km^2$  (inclusive). Puede usar el operador `count`.
- Obtenga los países que tienen al menos una ciudad que se llama igual en otro país. No puede utilizar el operador `count`.

## Pregunta 6: SPARQL

Lo crea o no, los ayudantes y profesor del ramo manejan toda la información del curso mediante una base de datos en formato RDF. Esta incluye tanto la información de alumnos (nombre, apellidos, número de alumno), la composición de grupos, las notas de las interrogaciones y la distribución de corrección de preguntas de los ayudantes. ¿No lo creen aún? A continuación se muestra una porción de los datos enfocados en un alumno en particular. Se asumirá que se cuenta con el siguiente prefijo común para todas las entidades:

PREFIX bdd: <http://bases.ing.puc.cl/2016-2/>



El diagrama indica que el alumno Pedro Zapata Soto pertenece al grupo 1 de proyecto y además es jefe de este. Además Pedro se sacó un 5.3 en la pregunta 2 de la interrogación 3 que fue corregida por Fernando.

a) [5 pts] A partir del ejemplo anterior, escriba las siguientes consultas en SPARQL:

- [1 pts] Obtenga todos los tipos de dato que contiene la base de datos.
- [2 pts] Obtenga el nombre completo de todos los alumnos del curso y si son jefes de un grupo, el número del grupo del cual son jefes.
- [2 pts] Obtenga solamente el nombre de aquellos alumnos que sólo han tenido azules en notas de preguntas.

b) [1 pts] Explique con palabras, para cada consulta del ítem anterior, si existe una consulta similar en SQL.