Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación
Segundo Semestre de 2017

IIC 2413 – Bases de Datos
Guía Interrogación 1

Pregunta 1: Álgebra Relacional

Para esta pregunta considere el siguiente esquema:

- `Persona(pid int, nombre string, instrumento string)`
- `Banda(bid int, nombre string, estilo string)`
- `Pertenece(pid int, bid int)`
- `Colabora(bid1 int, bid2 int, fecha date)`

La relación `Persona` tiene nombres de personas junto a su identificador y el instrumento que tocan. La relación `Banda` contienene nombres de bandas, junto a su identificador y el estilo. La relación `Pertenece` indica si una persona pertenece a una banda. La relación `Colabora` indica si una banda con id $bid1$ tocó en el recital de otra banda con id $bid2$ y la fecha en qué sucedió. Se piden las siguientes consultas en álgebra relacional (es posible usar rename):

- **(0.5 pts)** El nombre de cada banda junto al nombre de cada integrante.
- **(1 pt)** Todas las personas que han tocado en un recital de "Justin Bieber' y de "Black Sabbath".

Se dice que dos bandas con id $b1$ y $b2$ son enemigas durante un año a si todas las bandas que tocaron en un recital de $b1$ en ese año no tocaron en un recital de $b2$ y viceversa. En base a esto responda la siguiente consulta:

- **(1.5 pt)** Entregue todas las bandas enemigas durante el 2012.

Se define ahora el operador $enemies_a(Colabora)$, que recibe una relación binaria como "Colabora". A este operador se le indica un año a. El operador retorna una relación unaria que devuelve todos los bid de las bandas que fueron enemigas de una otra banda en el año a.

- **(2 pts)** Entregue una expresión en álgebra relacional que defina al operador. En base a su construcción, muestre si el operador es o no monótono.

Ahora, suponga que se encuentra definido el operador de la consulta anterior. Se le pide responder la siguiente consulta:

- **(1 pts)** Entregue las personas que pertenecen a todas las bandas enemigas del 2015 y del 2016.

Pregunta 2: SQL

En esta pregunta considera el siguiente esquema:

- `Jugador(jid int primary key, nombre varchar(20), eid int, eid foreign key references Equipo(eid))`
- `Equipo(eid int primary key, nombre varchar(20), creado_en date)`
- `Partido(eid_local int, eid_visita int, id_partido int primary key, eid_local foreign key references Equipo(eid), eid_visita foreign key references Equipo(eid))`
- `Gol(jid int, id_partido int, eid_favor int, goles int, primary key(jid, id_partido), id_partido foreign key references Partido(id_partido), eid_favor foreign key references Equipo(eid))`

La tabla `Jugador` contiene la información de jugadores y el id del equipo en que juegan. `Equipo` contiene el nombre de los equipos con su identificador. `Partido` tiene un id e indica el id del local y la visita junto a la fecha. Finalmente, `Goles` indica cuantos goles hizo cada jugador en qué partido y a favor de que equipo. Exprese las siguientes consultas en SQL utilizando el esquema anterior:

- **(1 pto)** Entrega todos los nombres y cantidad total de goles de Jugadores en orden ascendente de total.
- **(1 pto)** Entregue el nombre de cada jugador de "Palestino" junto con el número de partidos en que ha realizado goles y la cantidad total de goles.
- **(1 pto)** Entregue el partido con más autogoles. Un autogol es cada gol que `eid_favor` es distinto al id del equipo del mismo jugador.
- **(1.5 ptos)** Entregue todos los partidos que "Palestino" ha ganado por más de tres goles en los últimos dos meses. Asuma que la funcion `CURDATE()` entrega la fecha actual y que la resta de datos `dates` entregan la diferencia en dias.
- **(1.5 ptos)** Entregue los equipos con más de un 75 % de partidos ganados de local.

Pregunta 3: E/R + Vistas

Ha sido contactado por Uver para migrar datos de su sistema de transporte a la nueva herramienta *Uver&Banner*. Actualmente existen personas que ocupan los servicios de la Compañía. Estas personas tienen un RUT y un nombre. Existen tres posibles medios de transporte, que pueden ser UverX, UverPremium, UverBici. Todos deben almacenar la tarifa que cobran por cada 100 metros y por cada minuto, además de la patente. Además UverBici debe almacenar el número de bicicletas que puede transportar y UverPremium debe almacenar si es del tipo Sport o Limousine. Cada transporte tiene un solo chofer, pero cada chofer puede tener varios vehículos. Además el chofer debe tener almacenado la fecha de vencimiento de su licencia de conducir. Existen métodos de pago que pueden ser Efectivo, PayPal o Crédito. PayPal debe tener la información del nombre de usuario y un correo. Crédito debe tener el número de la tarjeta de crédito. Finalmente, se debe guardar la información de qué persona tomó qué transporte y qué método de pago utilizó, guardando además el horario y un multiplicador de tarifa (un número que va del 0 % al 100 %).

En concreto, le solicitan que:

- **(3 pts)** Cree un diagrama entidad relación que representie fielmente la situación del transporte público.
- **(1 pt)** Indique cómo quedaría el esquema. No es necesario ingresar los comandos en SQL, pero si es necesario indicar las llaves primarias.

Finalmente el chofer **Ash Ketchum** le dice que necesita el RUT de las personas que han utilizado sus UberX pagado con Efectivo o PayPal. Dado que no puede entregarle acceso a toda la base de datos, decide entregarle acceso a una vista con la información solicitada.

- **(2 pts)** Entregue el comando en SQL que crea la vista solicitada.

Hint: Es posible que quiera utilizar subclases. Si dos relacion B, C son subclase de otra relación A, una forma posible de traspasar esto a un esquema es almacenar en A todos los *id* de sus subclases. En caso de que yo quiera obtener la información específica de B o C, puedo acceder a ellas mediante un join.