

Guía de ejercicios de SQL

Profesor Luis Rojas Pino

Ayudante Abraham Marianjel

Modelamiento de la información

[P1] Corretaje de propiedades

Enunciado Sean las siguientes tablas de una base de datos de una corredora de propiedades:

```
Arrendatario (RUT, Nombre, Apellido)
Arrienda (RUT, Id_casa, Deuda) Ojo: Deuda >=0 (si es 0, no hay deuda)
Telefonos (RUT, Fono)
Dueño (RUT, Nombre, Apellido)
Casa (Id_casa, RUT, Nro, Calle, Comuna)
```

Al respecto, conteste las siguientes preguntas:

1. Los arrendatarios que arriendan la casa ubicada en la calle Carrera n° 1024, Santiago.
2. ¿Cuánto le deben a María Pérez?
3. ¿Cuál es la deuda total para cada dueño?
4. Liste todas las personas de la base de datos
5. Indique los dueños que poseen tres o más casas. Con y sin agregación.
6. Liste los dueños que tengan deudores en todas sus casas.

1. Este es el tipo más sencillo de consulta posible.
2. Se supondrá que María Pérez hay una sola.
3. Aquí es necesario agrupar la información, así la suma se hará dentro de cada grupo indicado. Entregué toda la información necesaria en el SELECT, aunque con el RUT del dueño bastaría (si en el trabajo le piden algo así, entregue todo).
4. Las personas de la BD son los arrendatarios y los dueños. Para entregar ambos, hay que realizar una unión. Nota: para realizar una unión, los esquemas deben ser compatibles (atributos con mismo nombre y dominio). Afortunadamente, éste es el caso.
5. Hay dos maneras de hacer esto: con agregación y sin ésta. El caso sin agregación (menos evidente en general) consiste en hacer un *join* de tres tablas.

Sin agregación

Con agregación: en este caso, es necesario utilizar HAVING. HAVING es el WHERE pero para funciones agregadas. En el HAVING sólo pueden aparecer funciones agregadas y constantes.

6. Jugando con la semántica vemos que un dueño con deudores en todas sus casas equivale a un dueño con deuda en todas sus casas. Y el complemento de eso son los dueños con casas sin deudas.

Otra manera consiste en exigir que la deuda de cada casa del dueño sea positiva. En este caso, una consulta anidada exigiendo igualdad sobre ALL basta.

[P2] Sistema de e-learning

Enunciado *Noob Saiborg*² está desarrollando un sistema de evaluación automático como parte de un paquete de e-learning comercial. En este sistema, los *tests* consisten de 1 ó más preguntas con alternativas. Para hacer el sistema flexible, y para aumentar la dificultad de los problemas, uno de los requerimientos es que las preguntas -de las que consisten los tests- tengan una o más soluciones. (Por ejemplo, una pregunta del tipo “¿Cuál de las siguientes propiedades presenta el paradigma OOP?” requiere la selección de varias alternativas.) Actualmente, Noob S. cuenta con el siguiente modelo de datos relacional para modelar los tests:

```
test(tnum, titulo, curso, descripcion, autor)
preg(tnum, pnum, enunciado)
pregalt(tnum, pnum, alt, texto, correcta)    //correcta es booleana
resp(user, tnum, pnum, alt)
alumno(user, nombre, apellido)
curso(user, curso)
contesta(user, tnum)
cursos(curso, nombre)
```

En este sencillo (primitivo) modelo de datos, Noob S. necesita resolver los siguientes problemas con SQL:

1. Sobrescribir el test X con otro test Y. (X,Y son tnum)
2. Conocer el número de tests por curso.
3. Conocer los cursos sin tests.
4. Determinar los tests con falencias. Un test tiene falencias si no tiene preguntas, si su primera pregunta (pnum) no está numerada 1, si las preguntas no son consecutivas (ej. 1,2,4,5,8), si hay preguntas con 1 ó menos alternativas³, si todas las alternativas son verdaderas o si todas las alternativas son falsas.
5. Corregir los test cuya única falencia radica en la numeración de las preguntas.
6. Cuántos alumnos hay en cada curso.
7. Qué alumnos han contestado tests que no les corresponden (de cursos que no cursan).
8. Obtener el puntaje no normalizado de cada rendición de test. El puntaje no normalizado ha sido definido (requerimiento) como: $P = \text{buenas} - \text{malas}/4$. Si un alumno no contesta en una pregunta exactamente lo mismo que se ha definido como correcto, la pregunta cuenta como mala a menos que el alumno haya omitido.
9. Obtener el puntaje normalizado, o sea, de 1,0 a 7,0.

Solución Para resolver este problema es necesario saber cómo escribir consultas y cómo modificar datos con SQL.

[1] Sobrescribir el test X con el test Y se hace borrando el test X y copiando los datos del test Y con el tnum X. Haremos el proceso completo de borrado de un test, considerando si el test fue contestado, y copiaremos los datos de Y.

[2] El número de tests por curso tiene la forma (curso, número). Es una simple y vil agrupación con count(*).

[3] Una simple sustracción.

[4] Cada exigencia del enunciado puede ser consultada por separado y luego unida para obtener los tnum que corresponden a tests inválidos.

El resto de las preguntas se dejan como ejercicios propuestos

