PAUTA CONTROL 2 DE BASES DE DATOS
PRIMAVERA/2008

Problema 1

(1) El tiempo promedio que tarda un trabajador social en resolver sus casos + con cuántas personas trabaja.

Notas: hay ambigüedad en la pregunta, ya que no queda claro si las personas con que el trabajador social trabaja son clientes o trabajadores sociales (casos compartidos). También se han descartado los casos críticos para COUNT: trabajadores sin clientes ni colegas.

Caso 1: con el número de clientes

```
SELECT   A.RUT, COUNT(DISTINCT B.RUTcaso), AVG(fecha_cierre-fecha_apertura)
FROM     Trabajador_caso A, Caso B
WHERE    A.COD=B.COD
GROUP BY A.RUT ;
```

Caso 2: con el número de colegas

```
SELECT   A.RUT, COUNT(DISTINCT C.RUT)-1, AVG(fecha_cierre-fecha_apertura)
FROM     Trabajador_caso A, Caso B, Trabajador_caso C
WHERE    A.COD=B.COD AND A.COD=C.COD
GROUP BY A.RUT ;
```

Puntaje: descontar poco para errores pequeños (un par de décimas), como olvidar el -1 en el caso 2 o si escribió AVG(fecha_apertura-fecha_cierre), o sea, sacó el promedio al revés. Esos son detalles menores.
Descontar más (medio punto a un punto) en caso de mala agrupación (GROUP BY) o ausencia de ésta (mucho más grave: quitar un par de puntos), o un mal WHERE. Ignorar las tablas que pueden sobrar en el FROM, pero castigar si falta información.

(2) Los trabajadores sociales que trabajan con drogadictos y que tienen clientes en todas las comunas.

Notas: aquí no hay ambigüedad, pero hay un desafío: hacer algo parecido a una división. Sin embargo, la solución es mucho más fácil. Basta ver que el número de comunas de los clientes es igual al total de comunas: o sea, comparar COUNTs.

```
SELECT   A.RUT
FROM     Trabajador_caso A, Caso B, Trabajador_caso C, Caso D, Cliente E
WHERE    A.COD=B.COD AND B.descripcion LIKE '%droga%'
    AND  A.RUT=C.RUT AND C.COD=D.COD AND D.RUTcaso=E.RUTcaso
GROUP BY A.RUT
HAVING   COUNT(DISTINCT E.comuna) = ( SELECT COUNT(DISTINCT F.comuna)
                                      FROM Cliente F ) ;
```

Puntaje: es imperativo el incluir Caso.descripcion en WHERE, aunque el uso correcto de LIKE o SIMILAR (que compara con expresiones regulares a la Perl) no es trascendental. Si hay un mal uso, descontar un par de décimas. Pero descontar más si se escribió algo como Caso.descripcion = 'droga'.
Respecto a hacer la división, hay varias maneras. Pero siempre se debe descontar bastante (un punto o más) si se hizo comparación de COUNTs fuera del HAVING.

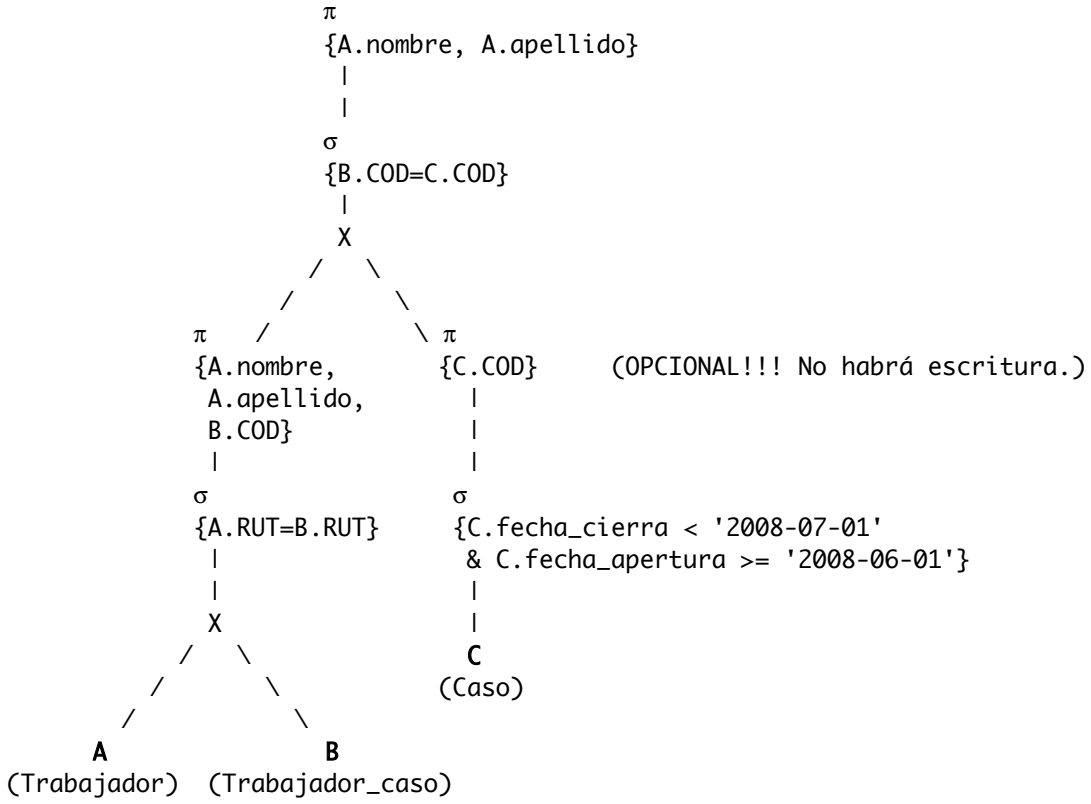
También es importante ver que las tuplas de drogadictos no deben interferir con las tuplas de comunas, ya que unas cuentan a una porción de la población de clientes mientras que la otra los considera a todos (para cada trabajador social).

Problema 2

Notas: El alumno es libre de permutar las tablas del FROM. Eso no debería influir en su nota. Las proyecciones en las ramas de lectura de tablas son omisibles, ya que se pueden ignorar en una ejecución en el vuelo (on-the-fly).

Puntaje: vale lo mismo para cada uno de los siguientes puntos:
- Convertir SQL a álgebra relacional / árbol de consulta;
- Tener las selecciones lo más cerca posible de las hojas;
- Proyectar correctamente. (Ojo con las notas.)

Ej:



Problema 3

Este problema es netamente conceptual. Para asignar nota, vea si el alumno tiene idea de cómo funciona la base de datos a nivel de algoritmos. La precisión no es importante.

Notas: Se dio indicación explícita a los alumnos para que indicaran sus supuestos. Ver si los supuestos son coherentes con los resultados escritos o si los supuestos pueden quedar implícitos (¿son intuitivos?).
También se indicó que cid = id. O sea, D.cid es lo mismo que D.id. (Problema de tipeo persistente...)

(1) Estimar ?X e ?Y.

Es un nested loop cuyo primer nivel ocurre en Diary y el segundo en el hash de Company, haciendo join natural en los id. Naturalmente, cada vez que hay un acierto C.id=D.id, hay que ir a la tabla Company a recuperar el resto de la tupla.

Una lectura literal nos lleva a estimar ?X = 1.03 + 8.14 = 9.17 y eso está OK. Ahora, el alumno pudo entender para el Index Scan que el costo era "por pasada", y debería haber estimado algo como ?X = 1.03 + 8.14 x N, donde N es el número de llamadas al segundo bucle (más o menos igual al número de páginas del primero). Una aproximación algo peor sería ?X = 1.03 x 8.14, o redondeado a ?X = 2 x 8.14, pero aquí se estaría obviando el costo de leer Diary, luego eso debería quedar como ?X = 1.03 + 1.03 x 8.14. (Nested Loop por páginas en vez de tuplas.)

Para ?Y, depende de lo que se entienda. Un match "todos con todos" da ?Y = 3 x 10 = 30, mientras si se asume que cada Company tiene sólo un Diary asociado, ?Y = 10. Un valor intermedio sería ?Y = 20, si se asume que la probabilidad de que cada Company relevante esté asociada a 1, 2 ó 3 Diary(s) es uniforme.

(2) Sin Index Scan

Se dice que el índice es el 10% de la tabla original, por lo que se podría magnificar el costo del segundo loop por 10, a manera de aproximar. Por supuesto, acá nos pasamos del costo, porque en Index Scan hay que visitar la tabla para recuperar algunas tuplas.

(3) Tablas ordenadas

Un join natural se implementa con pasadas secuenciales paralelas; luego el costo de Diary se mantiene y el de Company se reduce porque no hay bucles. Debido al tamaño de Diary, no deberían haber muchos cambios de costo, sólo decir algo como ?X = 1.03 + 81.4/1.03...

Puntaje: asignar puntaje a la coherencia o a las ideas tras las respuestas. Ver que el alumno entienda qué es un bucle anidado y un index scan (revisión completa del índice). Cada pregunta vale lo mismo.