

## UNO. Anuncios.

Ya tengo la información para calcular el Quiz que corresponde a asistencia. Por problemas de sincronización entre las compu, mientras una estaba en el taller, se perdieron como 2 días de asistencia en clases remotas, así como las actualizaciones que hice por razones de ausencias o tardías justificadas.

Hoy en la tarde, solicito que me envíen mensaje para enviar mi registro de asistencia y la nota para ese Quiz, si tienen reclamo por justificación de ausencia o tardía no registrada, favor indicarlo.

Espero para fin de semana, tener revisado el quiz de asistencia, y el quiz queharemos este viernes. Para dar una nota final de promedio de quices y nota previa definitiva.

Ultimo quiz (este viernes), se evalúan la clase anterior y la presente. Preguntas de falso y verdadero, no más de 10 preguntas.

Revisión del proyecto queda para miércoles 26 y jueves 27.

Examen 2do para el 3 de diciembre (3 horas: 25 min primera parte 1hora para SP, 1 hora para modelado, resto del tiempo para repasar las respuestas del examen para que las pueda ver bien el profe.) Es seguro que proporcione mas de 3 horas. 930 am a 1230 pm.

Taller de modelado y codificación para el viernes 28 de nov, 930 am.

Con respecto de la evaluación del profesor:

Agradezco si al final del curso, me envían un email o msj, sobre cosas que puedo mejorar como profe. No tengo formación de educador, aunque no es justificación para hacer las cosas mal, ni para ser irresponsable. He preferido definirme como ‘mentor’, tal como lo aclare al inicio del curso, señalar un camino para que cada persona aprenda por sí misma. Yo sé que es lo que debo mejorar para ser un mejor educador. El email lo envían cuando ya se han dado las notas finales, se agradece la sinceridad, aunque duela.

Agradezco que antes de escribir las recomendaciones para mejorar, (esto es opcional) se escriban o al menos piense en los aspectos positivos de mi persona como profesor, lo cual es buena práctica de realizar en general cuando se tiene una conversación franca con la persona con la cual se tiene un conflicto, sirve para apaciguar la pasión o emoción o los sentimientos negativos, de manera que la conversación es mas fluido y lo argumentos mas balanceados.

Análisis FODA : fortalezas, oportunidades, debilidades y amenazas.

DOS. Calculo relacional y algebra relacional.

Este tema es importante por razones de cultura general acerca de teoría de BD. Esto lo evaluamos en el Q ultimo y en preguntas teóricas del 2do examen.

Calculo relacional utiliza lenguaje declarativo, y el algebra es lenguaje procedimental, el sql es una combinación de ambos.

El algebra utiliza operadores de conjuntos, típicamente: unión, intersección, diferencia, producto cartesiano, más otros operadores tales como select (filtro de filas), el project (filtro de columnas), división, el equijoin (similar la join de SQL), los outer join (similares a los de SQL) y el natural join (que no existe en SQL).

El select de SQL filtra columnas (\*) para todas las columnas), en el algebra filtra filas (where de SQL). Es desafortunado que el Select en SQL tiene semántica diferente que en algebra relacional.

Concepto de esquema en modelo relacional no es el concepto de esquema en SQL. Otra circunstancia desafortunada, se utiliza la misma palabra para describir hechos en contextos diferentes. Debe ser que E. Codd (teoría de BD relacionales) y .C. J. Date (desarrollo R el primer SABD, el lenguaje SQL y escribió los primeros libros de texto), no se conocían o nunca hablaron entre si, propusieron nombres iguales para conceptos diferentes.

En SQL y suponiendo una BD de recursos humanos, el esquema ‘pla’ es para todas las tablas de planilla: ‘pla.planillasemanal’, ‘pla.deducciones’, el esquema por default es ‘dbo’, ‘inc.EnfermedadIncapacidad’, inc es el esquema para incapacidades.

Esquema en modelo relacional, es la estructura (nombres de columna), tal que las relaciones son una instancia de una esquema, el concepto de relación en el SQL, su concepto equivalente es tabla.

Suponemos este esquema:

empleado (id, nombre, fecha Nacimiento, idDepto, Salario)

Depto (id, nombre)

“Los nombres y salario de todos empleados”

⇒ Filtrar columnas de la tabla empleados. Mediante operador anotado con el símbolo Pi en griego.  $\pi_{<\text{lista de atributos}>} <\text{nombre de tabla}>$ . El cual se expresa como “Project de <nombre de tabla> sobre <lista de atributos>”

$\pi_{<\text{nombre, salario}>} \text{ empleado}$



$\Pi$  empleados  
 Nombre, salario

En SQL:

```
Select E.Nombre, E.Salario
from dbo.Empleado E
```

Los nombres de empleado y salario de aquellos empleados con salario > 30000.

Lo resolvemos con subconsultas (o resultados parciales) y haciendo uso del sigma, operación para hacer filtro de filas, que usa este formato:  
 $\langle \text{símbolo sigma} \rangle_{\langle \text{predicado} \rangle} \langle \text{nombre de tabla} \rangle$

$T_0 \rightarrow \sigma$  empleados  
 salario > 3000

$R \rightarrow \Pi_{\text{Nombre, Salario}} T_1$

```
;With t1 (Select * from empleado where Salario>3000)
select t1.Nombre, t1.Salario
from t1
```

$R \rightarrow \Pi_{\text{Nombre, Salario}} (\sigma_{\text{Salario} > 3000} \text{empleados})$

```
select E.Nombre, E.Salario
from dbo.Empleado E
where E.salario>3000
```

--- nota al margen

En TP si pueden usar el with con CTE. Pues es una manera practica que sustituye el uso de tablas temporales o tablas variable. NO pueden usar lo que no se haya ilustrado en clase de alguna manera, por ejemplo: cursores, merge o using.

El output si es aceptable que lo usen.

---

Aunque se pudo hacer así:

$R \bowtie \Pi_{\text{Nombre, salario}} (\sigma_{\text{salario} > 3000} \text{ empleado})$

Los nombres y salario de empleado cuyo salario > 3000 que trabajan en departamento "ventas".

Las subconsultas o consultas parciales heredan el esquema las tablas base.

La X es producto cartesiano.

A X B en algebra relacional, equivale en SQL a : Select \* from A, B

$$T_1 \leftarrow \sqrt{\sum_{\text{Xbunse} = \text{"verfas"} \text{ do}}^{} \text{disto}}$$

$$T_2 \leftarrow \text{Employees} \times T_1$$

$$T_3 \leftarrow \sqrt{T_2}$$

(Employees, rd Dentos =  $T_2 \cdot 1d$ )  
and (Employees Salario > 300)

$$R \leftarrow \overline{\pi}_{\text{Nubre,Salario}} T_3$$

Sin usar tablas intermedias, lo anterior en SQL se escribiría de esta manera:

```
Select E.Nombre, E.Salario  
From dbo.empleado E  
Inner join dbo.Depto D on E.idDept= D.Id  
where e.salario > 30000
```

Otra manera de resolver esta consulta es usando join:

$R \bowtie_{\text{Nombre, Salario}} T$  (empleados,  $\bowtie$ , Deptos)  
Empleados, idDept = Deptos, id  
and Empleados, salario > 30000

$\bowtie$   
 $\bowtie$  / predicado

El equijoin se anota así:  $A |X|_{<\text{predicado}>} B$

El cual es equivalente a  $\langle \sigma \rangle_{<\text{predicado}>} (A \times B)$

Se puede usar Union, diferencia e intersección.

$\cup$  Union     $\cap$  intersección     $-$  diferencia

En SQL hay unión y diferencia, no he visto en ningún dialecto la intersección, que puede implementarse combinando unión y diferencia.

$$\begin{aligned} R &\leftarrow A \cup B \\ R &\leftarrow A \cap B \\ R &\leftarrow A - B \end{aligned}$$

Trabaja igual que en SQL, las tablas deben ser compatibles en cantidad de atributos, y el binding de los atributos debe ser compatible, el tipo de primer atributo en A debe ser compatible con el tipo del 1er atributo en B, el tipo de 2do atributo en A debe ser compatible con el tipo del 2do atributo en B, and so on.

El concepto de binding supongo lo verán en curso de lenguajes de programación.

Natural join:  $A * B$ , que es un join con un predicado de igualdad implícito entre todos los atributos que se llaman igual. Es útil para documentar las anomalías en 5NF.

En sql tenemos producto cartesiano, join, outer join, unión, diferencia, select y Project, no he visto que haya un equivalente para natural join o división.

La división es útil para consultas que su descripción tengan la palabra ‘para todos’. ‘Liste los nombres de vendedores que tienen venas en todos los cantones de Heredia’, este tipo de consultas son difíciles en SQL, no le son naturales, la que indican ‘en todas’ o ‘para todo’, aquí es útil el operador de división que no he visto en ningún dialecto de SQL (Oracle en los años 90 y hace un par de años, MySql, postgre y MSSQL)

**Calculo relacional**, se basa en la declaración de una variable tuplo (o variable dominio), que se declara a través la operación pertenece y se cuantifica a través los operadores de ‘existe’ o ‘para todo’.

La idea en calculo relacional es definir la variable tuplo que recorre la tabla (conjunto) que resultado de la consulta. Aquí el concepto de tabla se usa como equivalente al concepto de conjunto o relación utilizado en contexto de teoría de BD relacionales.

Para declarar una variable tuplo (algo así como el alias en SQL). T pertenece a conjunto, ejemplo: ‘<operador existe>T and T e Empleado’, e ‘::=’ ‘pertenece a’

El cuantificador de existencia se especifica Existe T,  $\exists T$

El cuantificador universal ‘es para todo’:  $\forall T$

‘Los nombres de empleados y su salario’

Especifico utilizando notación de conjuntos el conjunto resultado de la consulta:

La “/” se lee “tal que”

{t.nombre, t.salario / Existe t and t e Empleado}

‘Los nombres de empleados y su salario, de aquellos que ganan mas de 30000’

{t.nombre, t.salario / Existe t and t e Empleado and t.salario>30000}

‘Los nombres de empleados, el nombre de depto donde trabaja, y su salario, de aquellos que ganan más de 30000’

Para esta consulta, en el conjunto resultado debemos integrar atributos que provienen de empleado (nombre y salario) y atributos de depto (nombre). Es inevitable crear una variable tuplo para la tabla resultados, a diferencia de los ejemplos anteriores en que hemos reutilizado en el resultado la variable tuplo que definimos para empleado, aque debemor declarar una variable tuplo para el conjunto resultado (llamaremos R)

```
{R. nombreEmpleado R. nombreDept, R.salario /  
    existe R          # declaramos variables tupo R  
    and existe T and T e Empleado      # variable tuplo en empleado  
    and existe D and D e Depto        # variable tuplo en depto  
    and T.salario>30000            #filtramos empleados  
    and T.idDept = D.id            #vinculamos tablas  
}  
}
```

En anterior consulta hay que hacer renombrado de atributos, hay manera de hacerlos en calculo relacional, no lo ilustraremos.

‘Los nombres de empleados, el nombre de depto donde trabaja, y su salario, de aquellos que ganan mas de 30000, y su departamento no es ventas’

```
{R.nombreEmpleado, R.nombreDept, R.salario /  
    existe R          # declaramos variables tupo  
    and existe T and T e Empleado  
    and existe D and D e Depto
```

```
        and T.salario>3000          #filtramos empleados  
        and T.idDept = D.id         #vinculamos tablas  
        and D.nombre <> 'ventas'    #filtro en departamento  
    }
```

Pregunta típica de verdadero o falso en quiz o examen. Se propone una especificación: “los nombres de empleados tal que x and z....”, y luego se propone una solución en calculo relacional o algebra relacional, y la pregunta es: la solución en calculo relacional es la correcta según especificación? Falso o verdadero.

Respecto de normalización, lo típico es que se propone un esquema y se pregunta, ese esquema esta en ‘X’ NF?, falso o verdadero.

---

Preguntas de Normalizacion empleado(id PK, nombre, fechanac, genero), esta en 5NF?

Suponiente este esquema empleado(id PK, nombre, fechanac, genero) y depto (id, nombre), indique F o V, si la solución en Calculo relacional es congruente con es especificación: “los nombres de empleados que x y y”

<expresión en CR>

DOS Concurrencia y recuperación.

Para profundizar en este tema, va a hacer falta una clase. Faltamos a 7 clases durante el semestre, 2 por permiso ('compensadas' con clases asincrónicas, 3 por incapacidad debido a enfermedad y 2 feriados 15 de agosto y 29 de octubre).

Sin embargo, hemos venido conversando del tema a través de todo el semestre. El profe hace uso del acrónimo ACID (atómico, consistente, isolated, durable) referido a transacciones de BD para explorar este tema.

Hay transacciones distribuidas, o sea que tienen que asegurar el ACID en una operación que implica varias bases de datos, para ello existe Commit de 2 fases, commit de 3 fases, cuando la transacción corre en BD distribuidas, mi problema, del profe, actualmente.

La concepción de transacción de BD está muy relacionada con la concurrencia y la recuperación de errores.

En esta teoría, tiene como precondición que la transacción de BD, o de forma mas general el SP, ejecuta de manera correcta si corre sola. Se dan errores cuando el código corre de manera concurrente, o sea que son errores que produce el contexto de ejecución, tal que; cuando se produce un error, el sistema debe hacer un rollback si es que el error sucede dentro de una transacción. Ya que el error no es culpa del código, el sistema debe reiniciar la transacción (ese reinicio es ‘la recuperación del error’) en otras palabras se tiene que forzar a una recuperación del proceso, o sea ‘volver a correr la transacción’ luego de un error de concurrencia, del cual NO es responsable el código) y que debe ser transparente para el usuario final, quien nunca debe saber ni sospechar que hubo un error.

Los errores debido problemas de concurrencia son frecuentes. Por ejemplo: Dueño de empresa, con tarjeta corporativa está pagando un viaje de avión (el proceso de autorización envia un msj de confirmación por email, el proceso es complejo y no es inmediato), al mismo tiempo se esta pagando una planilla de 300 empleados, y al mismo tiempo un empleado esta pagando el recibo de luz, en esta situación puede suceder un interbloqueo, o sea un deadlock, que son comunes en los picos de actividad y también son comunes debido a mala programación. Buena práctica: objetos deben bloquearse en mismo orden y en orden de importancia.

Hay otros problemas que son lecturas sucias, actualización perdida, y otros como problemas como, por ejemplo: ‘registro fantasma’ que son problemas derivados de los 2 primeros. En los libros de texto, usualmente se documentan 5 problemas (interbloqueo, lecturas sucias, actualización perdida, registro fantasma y <quinto problema no llega el nombre a la muy utilizada y casi ‘llena’ y maltratada memoria del profe>).

El profe esta recuperando la sinapsis entre neuronas, para mejorar sus capacidades memorísticas y cognitivas, a través de una terapia ancestral, natural (a través de un ser viviente) a la vez que novedoso, no se puede decir en público.