RelaX

**Introducción**

Todo modelo de datos tiene 3 herramientas:

-donde vamos a almacenar nuestros datos (como van a estar organizados). Los modelos relaciones son modelos basados en tablas, o en relaciones.

-restricciones que estań acompañando a los datos que están organizados de esta forma.

-conjunto de operaciones que nos proporciona le modelo para poder hacer recuperación de datos.

\*consultar a una base de datos relacional siempre nos va a devolver como respuesta una relación (por ser el modelo relacional un *modelo cerrado*).

Existen dos formalismos matemáticos muy cercanos a sql, que nos van a permitir especificar una consulta:

-*álgebra relacional*: tiene un enfoque procedural.

-*cálculo relacional de tuplas*: lenguaje declarativo, más que describir las operaciones que se requiere para obtener un conjunto de datos, tenemos que describir el tipo de información que necesitamos sin preocuparme por el procedimiento específico que me va a permitir obtenerla.

ID dataset utilizado en RelaX: **62414bd8eaf6ea30b329c838c565cca9**

**Ejercicios**

-- **Ejemplo1:** Representa la tabla "Préstamo":

-- prestamo

-- **Ejemplo 2:** Obtén todos los clientes cuyo estado='OAXACA':

-- σ estado='OAXACA' (cliente)

**\***En este caso no necesitamos efectuar ninguna relación porque la misma tabla de cliente tiene el atributo de estado.

-- **Ejemplo 3:** Obtén todos los préstamos cuya sucursal sea MERIDA.

-- σ nombresucursal = 'MERIDA' (prestamo ⨝ sucursal)

**\***Esto se puede hacer gracias a que préstamo y sucursal comparte el atributo numsucursal.

**\***Siempre que vamos a hacer recuperación de un conjunto de tuplas, utilizamos el operador “*sigma*” o de “*selección completa*” **σ**.

**\***Siempre que vamos a hacer una consulta sobre dos tablas que se relacionan por un atributo (en este caso las tablas préstamo y sucursal se relacionan por el atributo numsucursal), entonces nos convendrá utilizar el operador “*natural join*” **⨝**.

**-- Ejemplo 4:** Obtén todos los clientes que tengan cuenta, préstamos o ambos.

-- π nombrecliente, numcta (cliente ⨝ (ctacliente ∪ prestatario))

**\***Se deberá usar el operador "PI" **π** para obtener solo los atributos que yo le indique a continuación (a esto se le llama *proyección*) de esos clientes los cuales serán el nombre y el numcta.

**\***Para poder utilizar la “unión” **∪** las tablas implicadas deben tener una *relación conformable*, es decir, que tengan *mismo grado,* *mismo nombre de atributos* (aunque en este ejemplo no se cumpla porque en consultas sql se acepta siempre y cuando las otras dos condiciones se cumplan) y mismo tipos de datos (*mismo dominio*).

**\***Para obtener clientes que tienen cuenta pondríamos sólo ctacliente:

-- ctacliente

**\***Para los que tienen prestamos solo prestatario.

-- prestatario

-- **Ejemplo 5:** Obtén la relación de los clientes del banco que tienen una cuenta pero no tienen preśtamos.

\*Esto lo resolvemos utilizando el operador **"-"**. Si en un principio escribimos ctacliente-prestatario primero mira el numcta de ctacliente y luego busca ese nº en numprestamo de prestatario para ver si también existe, pero como estos atributos tienen formatos distinto de código, no lo encontraría nunca y el programa se creería que no existe ningún cliente que tenga cuenta y no tenga préstamo, por esta razón deberemos de hacer una proyección especificando el atributo que se debe comparar de cada tabla el cual es idcliente. Por otro lado, esto solo nos arrojaría el resultado de los id de los clientes que es el único atributo en común de ambas tablas, por eso deberemos utilizar el natural join para obtener todos los datos de esos clientes. entonces haríamos:

-- cliente ⨝ (π idcliente ctacliente - π idcliente prestatario).

\*Si fuera de mi interés encontrar el número de cuenta que tienen, entonces tendremos que utilizar el operador “*join externo*” o “*join left”* **⟕,** que nos proporciona los datos tanto los que estén en la columna de la izquierda como los que estén en la izquierda y en la derecha

-- π nombrecliente, numcta (cliente ⨝ ((ctacliente ⟕ prestatario) - (ctacliente ⨝ prestatario)))