



estudios de postgrado
en computación



Bases de datos avanzadas

Universidad de Los Andes

Postgrado en Computación

Prof. Isabel M. Besembel Carrera

*Unidad II. Sesiones 27 y 28. Bases de Datos
Temporales.*

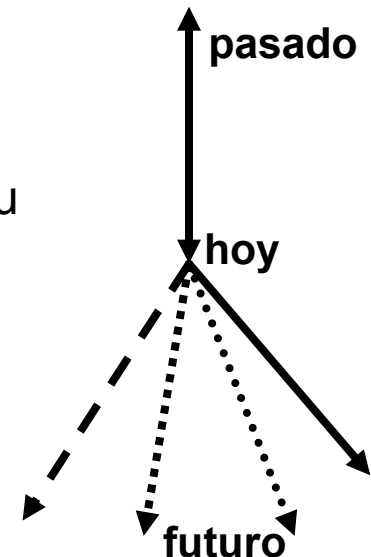
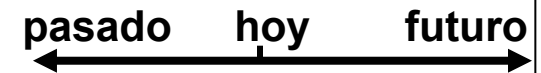


Introducción

- Los objetos y sus relaciones existen en periodos de tiempo definidos
- Los eventos ocurren en puntos específicos del tiempo
- Las bases de datos convencionales representan un estado de la misma en un momento particular
- Pueden ser consideradas como una fotografía de los datos de la organización.
- Los atributos temporales se manipulan aisladamente por los programas de la aplicación y el SMBD los considera como valores del tipo fecha

El dominio temporal

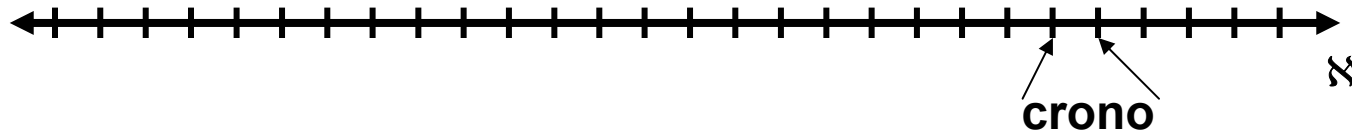
- *Estructura:* Si se considera una sola dimensión temporal, dos modelos estructurales son:
 - ✓ Lineal: el tiempo avanza desde el pasado hacia el futuro de forma totalmente ordenada.
 - ✓ Ramificado: el tiempo es lineal desde el pasado hasta hoy, donde se divide en varias líneas, cada una representando una secuencia potencial de eventos. Su estructura global es un árbol cuya raíz es hoy.
- Agregando un axioma de ordenamiento total se obtiene un modelo lineal totalmente ordenado y también se pueden agregar axiomas para definir la densidad de la línea de tiempo.



Modelo lineal

➤ Discreto:

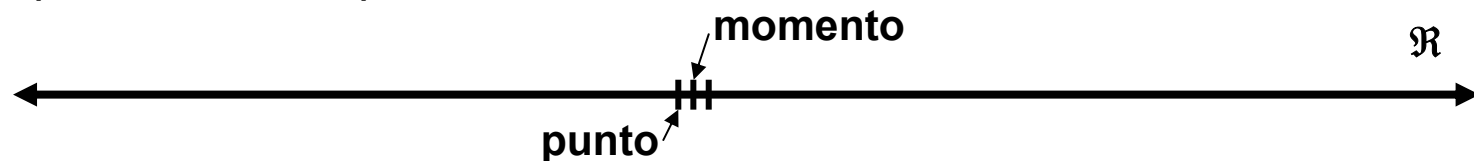
- ✓ es isomórfico con los números naturales, cada punto en el tiempo tiene un sucesor simple.
- ✓ Cada número natural corresponde a una unidad de tiempo indescomponible que tiene una duración arbitraria y que se denomina un **crono** (la duración más pequeña de tiempo que puede ser representada en el modelo).
- ✓ No es un punto, sino un segmento de la línea del tiempo.



Modelo lineal

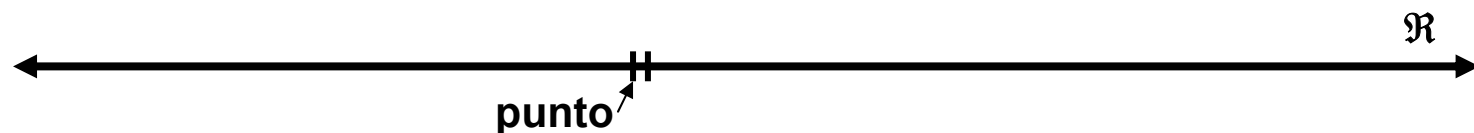
➤ Denso:

- ✓ es isomórfico con los números reales o con los racionales, entre cada dos puntos del tiempo existe otro momento.



➤ Continuo:

- ✓ es isomórfico con los números reales, es decir que es denso pero sin brechas intermedias.
- ✓ Cada punto en el tiempo se corresponde con un número real.





El dominio temporal

- Espacio y tiempo son métricas que tienen la noción de distancia, la cual satisface las propiedades:
 - ✓ La distancia entre dos puntos cualesquiera no es negativa
 - ✓ La distancia entre dos puntos distintos cualesquiera no es cero
 - ✓ La distancia entre A y B, $\text{dist}(A, B) = \text{dist}(B, A)$
 - ✓ $\text{dist}(A, C) \leq \text{dist}(A, B) + \text{dist}(B, C)$



El dominio temporal

- Mediante axiomas se pueden colocar los límites del tiempo y con ello la distancia
 - ✓ Se puede colocar un rango: Big Bang 14 ± 4 años atrás - Big Crunch.
- Se puede diferenciar entre
 - ✓ tiempos absolutos (10p.m. 3 de Marzo de 1994) y
 - ✓ tiempos relativos (3 horas)
- Los tiempos absolutos estarán supeditados a otro tiempo
 - 12p.m. del 1 de Enero del año 1.
- Los relativos difieren de la distancia en que ellos pueden tener signo
 - -3 horas

Dimensionalidad

- En el contexto de las BD, hay dos dimensiones del tiempo:
 - ✓ **Tiempo válido:** tiempo en el cual el hecho se dio o se dará en la realidad independientemente del tiempo en que se registró en la BD.
 - puede estar o no limitado (normalmente en el pasado)
 - ✓ **Tiempo transaccional:** tiempo en que el hecho se almacenó en la BD
 - es un intervalo que dice cuando se insertó y cuando se eliminó el hecho de la BD.
 - está siempre limitado en el pasado y en el futuro
- Estas dos dimensiones son ortogonales, pero no homogéneas

Modelo de datos

- **Fotografía:** es aquel que no soporta ninguna dimensión.
- **Histórico:** soporta únicamente la dimensión del tiempo válido.
- **Tiempo transaccional:** soporta únicamente la dimensión transaccional.
- **Bitemporal:** soporta las dos dimensiones
- El tiempo transaccional comienza cuando la BD se crea.
 - ✓ Los cambios de estado de la BD se registran con su tiempo transaccional actual, por lo cual él y las relaciones bitemporales son solamente de anexión solamente, manteniendo un crecimiento monotónico.
 - ✓ Este es semánticamente diferente del tiempo válido
- **Tiempo definido por el usuario:** su semántica es conocida por el usuario y no es interpretada por el Sistema Manejador de Bases de Datos Temporales (SMBDT)



El tiempo en el SMBDT

- Información históricamente indeterminada (No se exactamente cuando) solo aplica al tiempo válido
- Tiempo válido está fijado de antemano por el SMBDT, en la mayoría es el segundo.
- Tiempo transaccional está siempre determinado.
- La granularidad del transaccional es el intervalo más pequeño entre ellos.
- Representación física del tiempo en el SMBDT: impone una granularidad y un rango
 - ✓ 8 bytes
 - resolución de 1 segundo y un rango de 36 mil millones de años o
 - resolución de 1 microsegundo y un rango de 17400 años.

- **Modelo basado en eventos**
- **Evento:** es la información fundamental desde la cual el SMBDT puede derivar cualquier estado de la BD.
- **Mantiene una base de eventos que puede ser consultada con cualquier tiempo y la respuesta es deducida con las reglas de derivación**



Modelo basado en tiempo

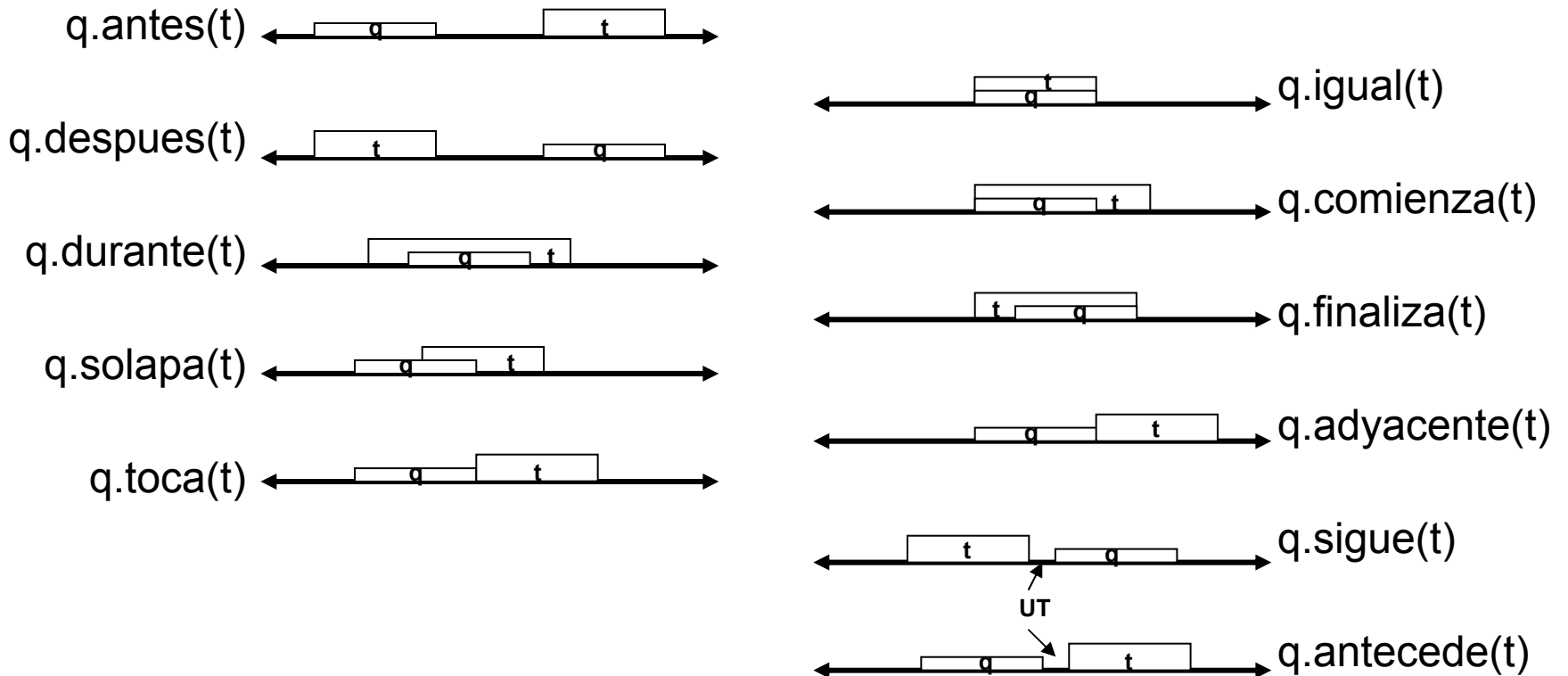
➤ Tipos:

- ✓ **Basado en puntos:** Consideran la línea de tiempo como densa, completa, ilimitada y real.
 - Cada intervalo es un par ordenado de puntos
- ✓ **Basada en intervalos:** Considera la línea de tiempo lineal y discreta con el intervalo como su primitiva.
 - No permite puntos temporales y soporta lógica temporal basada en intervalos y cálculo de predicados de primer orden
- ✓ **Mixto:** Considera puntos como intervalos de longitud cero y una línea de tiempo lineal y totalmente ordenada

Operadores temporales

operador	$b \cap b$	$i \cap i$	$b \cap i$	$i \cap b$	i-s
q.antes(t)	\emptyset	\emptyset	\emptyset	\emptyset	$q < t$
q.despues(t)	\emptyset	\emptyset	\emptyset	\emptyset	$q > t$
q.durante(t)	\emptyset	$\neg \emptyset$	$\neg \emptyset$	\emptyset	
q.solapa(t)	\emptyset	$\neg \emptyset$	$\neg \emptyset$	$\neg \emptyset$	
q.toca(t)	$\neg \emptyset$	\emptyset	\emptyset	\emptyset	
q.igual(t)	$\neg \emptyset$	$\neg \emptyset$	\emptyset	\emptyset	
q.comienza(t)	$\neg \emptyset$	$\neg \emptyset$	\emptyset	$\neg \emptyset$	
q.finaliza(t)	$\neg \emptyset$	$\neg \emptyset$	$\neg \emptyset$	\emptyset	
q.adyacente(t)	\emptyset	\emptyset	\emptyset	\emptyset	$t.i - q.s = UT \wedge q.i - t.s = UT$
q.siguiendo(t)	\emptyset	\emptyset	\emptyset	\emptyset	$q.i - t.s = UT$
q.antecede(t)	\emptyset	\emptyset	\emptyset	\emptyset	$t.i - q.s = UT$

Operadores temporales

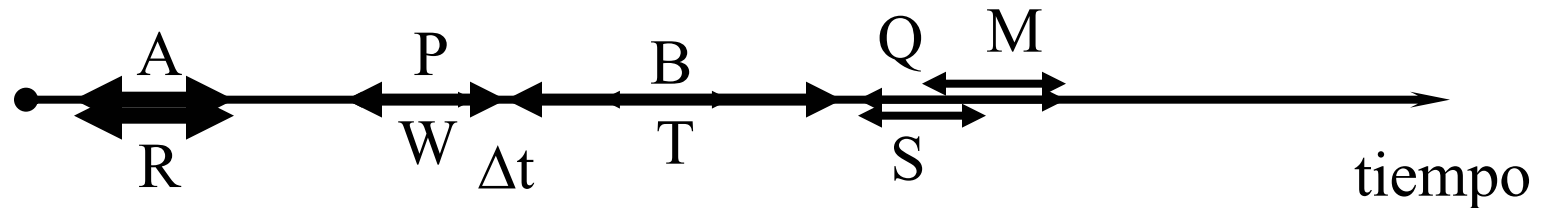


Relaciones temporales

✎ Relaciones entre dos objetos temporales en una dimensión

A antes de T
T después de A
B durante T
S solapa M
W toca T

A igual a R
S comienza Q
M termina Q
Q adyacente a T
Q sigue a T
W precede a T





Tiempo definido por el usuario

- Soportado por la mayoría de los SMBDR como un dominio asociado a los atributos.
- El tipo *Date* tiene una granularidad de 1 día y *Time* de 1 segundo.
DateTime.(SQL)
- En TOODM (Rose y Segev-91) *Time* es una clase del sistema con *Absolute* y *Relative* como subclases

➤ Representación de los atributos:

- ✓ Valor atómico:
 - Ejm: “José”
- ✓ Conjunto:
 - Ejm: {“José”, “Manuel”}
- ✓ Valor atómico funcional: Los valores son funciones del tiempo válido al dominio del atributo.
 - Ejm: 1993 → “José”, 1994 → “Manuel”
- ✓ Pares ordenados: (Valor, tiempo).
 - Ejm: (“José”, 1993) (“Manuel”, 1994)
- ✓ Tripletas: (Valor, desde, hasta)
 - Ejm: (“José”, Enero 1993, Marzo 1994)
- ✓ Conjunto de tripletas:
 - Ejm: { (“José”, Enero 1993, Marzo 1994), (“Manuel”, Abril 1994, Junio 1995)}

Tiempo válido

- SMBDR consideran dichos valores como atómicos
- SMBDOO pueden soportarlo:
 - ✓ Directamente a través del modelo: La semántica del tiempo válido la soporta
 - ✓ el usuario cuando especifica el esquema y la consulta. Ejm: OODAPLEX

Ejm: *type empleado is object*

function nombre(e : empleado @ n : string)

function dpto(e : empleado @ f : (t : time @ d : departamento))

function proyectos(e : empleado @ f : (t : time @ d : {proyecto}))

for the e in extent(empleado) where nombre(e) = 'Carlos'

for the p in extent(proyecto) where nombre(p) = 'SMBDOO'

for each t where p in proyectos(e) (t)

dpto(e) (t)

end

end

end

- No hay soporte temporal en el sistema



Tiempo válido

- Usando otras extensiones que a su vez sirven para soportar el tiempo:
 - ✓ Sciore extendió el modelo EXTRA incluyendo la palabra clave *versioned* que indica cuando un atributo tiene versiones.
 - ✓ Se incluye una clase especial denominada *annotation* y se extendió el lenguaje de consultas EXCESS para permitir consultas sobre todas las versiones (inall).
 - ✓ Los predicados del *where* pueden usarse para seleccionar la versión deseada.
 - ✓ Para soportar las anotaciones se asocia una variable escondida a cada variable convencional

Tiempo válido

Ejm: *class empleado inherits object =*

variables

nombre : *string*;

dpto : *HistoryFn*, departamento;

proyectos : *HistoryFn*, {proyecto};

end empleado;

((empleado *select* : [:e | e.nombre = 'Carlos']). -dpto escoger:

(proyecto. -nombre cuando : [: n | n = 'SMBDOO'])

) .value

- . - selecciona la variable histórica asociada con los métodos escoger y cuando.
- Aunque está la clase anotación y la histórica, la optimización de las consultas no manipula el tiempo directamente, ya que ello es hecho a través de métodos definidos por el usuario.



Incorporación del tiempo

- En el modelo de datos: Algunos lenguajes incluyen cláusulas y/o operadores para manejar el tiempo válido.

- Ejm: TOSQL, TOOSQL y OOSTSQL.

```
SELECT e.dpto, e.dpto.vt
```

```
FROM e : empleado, p : proyecto
```

```
WHERE e.nombre = 'Carlos'
```

```
WHEN e.dpto.vt.DURING(e.proyectos.vt) AND p.nombre='SMBDOO' IN e.proyecto
```

En OOSTSQL:

```
SELECT dpto(e)
```

```
RESTRICTED TO [['SMBDOO' IN proyecto-nombre(proyecto(e))]]
```

```
FROM empleado e
```

```
WHERE nombre(e) = 'Carlos'
```



Tiempo transaccional

- Se usa para deshacer una transacción y para soportar las versiones.
- En versionamiento la línea de tiempo es ramificada.
- Permite saber si hubo modificaciones en los datos de la base y cuando se realizaron tales modificaciones.
- Las consultas serán sobre los valores de los datos en el pasado y los errores no pueden ser corregidos, ya que en ese tiempo no se sabía que eran errores.
- Versionamiento de la extensión:
 - ✓ versiones sobre las instancias o sobre los atributos.
- Versionamiento del esquema:
 - ✓ versiones sobre las definiciones de los objetos.
- Si hay versionamiento de las extensiones, se puede o no soportar el versionamiento del esquema.

Versionamiento

- **Versionamiento de la extensión:** Tres enfoques
 - ✓ Usar el modelo directamente:
 - Como el usuario implementa su soporte temporal, el tiempo transaccional puede acomodarse para anexar nuevos valores cada vez que el valor del atributo cambie. Ejm: OODAPLEX
 - ✓ Extensiones que pueden ser usadas para soportar la temporalidad:
 - Con la clase anotación puede soportar revisiones (*RevisionFn*) y alternativas (*AlternativesFn*).
 - ✓ Incorporación del tiempo al modelo:
 - Normalmente se diferencia entre los objetos versionados y los que no lo son.
 - En TOODM, un atributo del tipo TS[] es un conjunto de triplas con el ObjId del valor histórico, el valor del atributo y su tiempo



Versionamiento (ORION)

- Se usa el versionamiento de los objetos, indicando si la clase es versionable o no.
- Las versiones se enumeran automáticamente por el sistema se asocian con nombres, tiempos y estatus.
- Se soportan versiones transcientes que pueden ser actualizadas en cualquier momento, versiones de trabajo que pueden ser compartidas pero no actualizadas y versiones liberadas que no pueden ser actualizadas ni eliminadas.
- Se soportan también referencias genéricas y específicas.
- Genéricas apuntan a un objeto con todas sus versiones y se usan para retardar la asociación de un atributo con una versión dada.
- Específicas apuntan a una versión fija del objeto.
- Las versiones forman un digrafo acíclico (dga) asociado a cada objeto.
- Se mantienen dos relaciones:
 - la *versión-de* entre un objeto versionable y todas sus versiones, y
 - la relación *derivada-de* entre una versión y las versiones derivadas de ella

Versionamiento

- **Versionamiento del esquema:** En la evolución del esquema, éste puede cambiar en base a las necesidades de la aplicación.
 - ✓ Varios esquemas pueden ser válidos dependiendo del tiempo transaccional que se considere.
- Su implementación se puede hacer según dos enfoques:
 - ✓ **Inmediata:** se modifican todas las instancias para incluir el cambio
 - ✓ **Tardía:** No se modifica inmediatamente lo que hay en disco, sino cuando se referencia algún objeto y sea traído a memoria principal donde se transforma para que esté de acuerdo al esquema válido en ese momento.
- En ambos casos, sólo un esquema es válido en ese momento.
- Cuando se va hacia atrás en el tiempo, se presentan los datos en base al esquema válido para ese momento.



Lenguaje de consulta relacional temporal

- El lenguaje TQUEL es una extensión del lenguaje Quel que soporta temporalidad.
- Consultas sobre la dimensión del tiempo válido.
RANGE OF c IS Ciudad
RETRIEVE c.nombre, c. numHab
WHERE c.país = "Venezuela" WHEN c SOLAPA |May 1, 1990| AS OF PRESENT
- Consultas sobre la dimensión del tiempo transaccional.
RANGE OF c IS Ciudad
RETRIEVE c.nombre, c. numHab
WHERE c.país = "Venezuela" AS OF BEGIN OF |May 1, 1990|
- Calendario Gregoriano con los nombres en Inglés que es el calendario por omisión del sistema

Ejemplo prototipo

- Recupere los nombres de las personas que sufrieron accidentes en el solapamiento del 1/1/2001, 3/3/2001 y 11/2/2001,4/4/2001?
 - ✓ { 1.nom_Per; 10/Accidente,1/Persona ; SOLAPA ((#1/1/2001#,#3/3/2001#) AND (#11/2/2001#,#4/4/2001#)) AND 10.Ced_per = 1.Ced_Per }
- ¿Indique los nombres de las personas que sufrieron accidente antes del 1/4/2001?
 - ✓ { 1.nom_Per; 10/Accidente,1/Persona ; ANTES #1/3/01# AND 10.Ced_per = 1.Ced_Per }

