

OPERADORES DE BASES DE DATOS TEMPORALES Y SU IMPLEMENTACIÓN

Ángela María Castaño Restrepo¹
Jorge Hugo Uribe Restrepo²
Francisco Javier Moreno Arboleda³

Recibido: 16/02/2007

Aceptado: 16/09/2008

RESUMEN

Actualmente, los Sistemas de Gestión de Bases de Datos (SGBD) comerciales no soportan adecuadamente las bases de datos temporales. Por ejemplo, el planteamiento de consultas temporales es difícil de expresar. En este artículo se discuten los aspectos referentes a la implementación de operadores temporales con el fin de facilitar la gestión de los datos temporales en un SGBD. Además, se presenta un caso de estudio para ejemplificar los operadores temporales implementados.

Palabras clave: *bases de datos temporales, modelos de datos temporales, operadores temporales.*

TEMPORARY DATABASE OPERATORS AND THEIR IMPLEMENTATION

ABSTRACT

Nowadays commercial Database Management Systems (DBMS) do not offer an appropriate support for temporary databases. For example, the formulation of temporary consultations is a difficult task. In this paper, we discuss an implementation of temporary operators in order to facilitate the management of temporary data in a DBMS. In addition, we present a case study that illustrates the implemented temporary operators.

Key words: *temporary databases, temporary data models, temporary operators*

1 Estudiante de Ingeniería de Sistemas. Universidad de Antioquia. acastano@pragma.com.co

2 Estudiante de Ingeniería de Sistemas. Universidad de Antioquia. huribe@dharmasoftware.com

3 MsC, Estudiante de Doctorado en Ingeniería de Sistemas, Becario de Colciencias. Universidad Nacional Sede Medellín. fjmoreno@unal.edu.co

I. INTRODUCCIÓN

Paralelo al crecimiento experimentado por la tecnología, han surgido nuevas propuestas y aplicaciones que implican el manejo de información dependiente del tiempo. Se desprende de allí la idea de almacenamiento histórico, que consiste en almacenar toda la historia de los estados asumidos por un objeto del mundo real a través del tiempo. Las bases de datos (BD) que almacenan este tipo de información se conocen en la literatura como BD históricas, BD orientadas al tiempo o BD temporales (BDT) (Jensen et al., 1994).

Uno de los principales problemas que existen para la implementación de BDT es que el planteamiento de restricciones y consultas temporales en los Sistemas de Gestión de Bases de Datos (SGBD) comerciales es una tarea ardua y compleja, ya que éstos no implementan ninguna característica temporal especial.

Aunque desde mediados de la década de 1970 se inició la investigación en el área de las BDT, la comunidad investigadora está muy dividida en la forma de abordar el problema (Date, 2001). Algunos investigadores favorecen un enfoque orientado a la definición de operadores y lenguajes de consulta temporales especializados, otros se inclinan hacia la concepción de operadores de propósito más general que proporcionen una base para el desarrollo de un enfoque especializado para la gestión de los datos temporales. Sin embargo, ninguna de las propuestas presentadas hasta el momento ha sido implementada en un SGBD comercial.

En este trabajo se implementan algunos operadores temporales, véase Sección 2.2, en un SGBD comercial. Se discuten asimismo los aspectos relacionados con la implementación de tales operadores. El objetivo es mostrar que con el uso de estos operadores, el planteamiento de consultas y restricciones temporales en SGBD comerciales se facilita.

El artículo está estructurado así: la Sección 2 presenta los operadores y tipos de datos implementados y un caso de estudio, en el que se modela e implementa una BDT. En la Sección 3 se presentan los resultados. Finalmente, en la Sección 4 se concluye el artículo y se plantea el trabajo futuro.

2. MÉTODO

Para la solución propuesta se definieron tipos de datos para soportar la representación del tiempo en el que los datos son válidos en el mundo real; este tiempo es llamado *tiempo válido*. Se implementaron algunos operadores temporales clasificados así: operadores de Allen, operadores de totales sobre intervalos, operadores de totales relacionales y operadores relacionales temporales. Para la ejemplificación de estos operadores, primero se estudiaron los principales modelos conceptuales de diseño de BDT y se eligió uno de ellos; luego, tomando como base el modelo seleccionado, se construyó un caso de estudio que fue transformado a un modelo relacional; finalmente, con los operadores implementados, se realizaron consultas sobre el caso de estudio.

La implementación se hizo en el SGBD Oracle. Se seleccionó dicho SGBD debido a que proporciona mecanismos como tipos y métodos, procedimientos, funciones y paquetes que facilitan, tanto la definición de los tipos de datos necesarios para el manejo del tiempo, como la implementación de los operadores temporales.

2.1. Tipos de datos

En la solución propuesta se definen tipos de datos que sirven para dar soporte a conceptos temporales tales como *instante*, *intervalo* y *elemento temporal*. Se define un instante como un punto particular en el eje del tiempo, un intervalo como el tiempo entre dos instantes y un elemento temporal se puede considerar como una unión finita de intervalos de tiempo.

El tipo de datos que modela los instantes consta de un atributo llamado *punto* que sirve para representar un punto de tiempo con alguna *granularidad*.

La *granularidad* puede entenderse como la precisión con la que se representa la información variante en el tiempo de una aplicación. Por ejemplo, si la información que va a ser gestionada por una aplicación varía mensualmente, es conveniente una granularidad mensual ya que no es relevante conocer el día, la hora, los minutos o los segundos en que varía dicha información. Así, la granularidad depende de la aplicación.

El tipo de datos usado para modelar los intervalos está conformado por dos atributos tipo instante, *i* y *f*, que representan el instante inicial y el instante final del intervalo, respectivamente.

Finalmente, el tipo de datos para los elementos temporales no es más que una tabla o arreglo que puede contener cualquier cantidad de elementos de tipo intervalo.

2.2. Operadores temporales

Entre los operadores más útiles para cualquier tipo de datos están los operadores de comparación. Allen (1983) propone un conjunto de operadores temporales lógicos para comparar intervalos de tiempo conocidos como operadores de Allen.

En este trabajo se implementaron los operadores de Allen EQUALS, BEFORE, AFTER, MEETS, OVERLAP, BEGINS, ENDS, MERGES, INCLUDES e INCLUDED_IN. Todos estos operadores reciben como parámetros dos intervalos y devuelven un valor de verdad, es decir, operan bajo una signatura:

Operador Temporal: $I \times I \rightarrow Boolean$

Donde I es un conjunto de intervalos.

$I = \{ i_1, i_2, \dots, i_n \}$.

EQUALS es verdadero, si los dos intervalos son iguales; BEFORE es verdadero, si el primer intervalo está antes del segundo; AFTER es verdadero si el primer intervalo está después del segundo; MEETS es verdadero cuando los dos intervalos son contiguos; OVERLAP es verdadero cuando los dos intervalos se solapan; BEGINS indica si el primer intervalo es el comienzo del segundo; ENDS indica si el primer intervalo es el final del segundo; MERGES es verdadero si los dos intervalos se pueden fusionar en uno solo, es decir, son contiguos o se solapan; INCLUDES indica si el primer intervalo contiene al segundo e INCLUDED_IN indica si el primer intervalo está dentro del segundo. En la tabla 1 se muestra el comportamiento de estos operadores, donde I_1 e I_2 son intervalos, $I_1 = [i_1, f_1]$ e $I_2 = [i_2, f_2]$; i_1 , i_2 , f_1 y f_2 representan instantes de tiempo.

Tabla 1. Operadores temporales de Allen.

Operación	Representación gráfica
I1 EQUALS I2	
I1 BEFORE I2	
I1 AFTER I2	
I1 MEETS I2	
I1 OVERLAP I2	
I1 BEGINS I2	
I1 ENDS I2	
I1 MERGES I2	
I1 INCLUDES I2	
I1 INCLUDED_IN I2	

Fuente: elaboración propia

Además de los operadores de Allen, se han propuesto dos operadores conocidos como operadores de totales (Date et al., 2003) que, aunque no son muy útiles en la práctica, son necesarios para definir otros operadores temporales que se describen más adelante.

Los operadores de totales son EXPAND y COLLAPSE. A diferencia de los operadores de Allen, EXPAND y COLLAPSE se aplican a conjuntos de intervalos en lugar de a un par de intervalos y producen como resultado otro conjunto de intervalos.

El funcionamiento de los operadores de totales es el siguiente: EXPAND recibe como parámetro un conjunto de intervalos y devuelve el conjunto de todos los intervalos cuyo punto inicial está contenido en algún intervalo del conjunto que recibe como parámetro y, además, el punto final es igual al punto inicial.

COLLAPSE recibe como parámetro un conjunto de intervalos y devuelve como resultado el conjunto mínimo de intervalos que tiene la misma forma expandida (obtenida con el operador EXPAND) que el conjunto original. EXPAND y COLLAPSE operan bajo una signatura:

Operador Temporal: $I \times I \times \dots \times I \rightarrow I \times I \times \dots \times I$
 $I^m \rightarrow I^n$

Donde $n \geq m$ para el operador EXPAND y $m \geq n$ para el operador COLLAPSE.

La tabla 2 presenta ejemplos de estos operadores. Por simplicidad se representan los puntos de tiempo de la forma $p1$, $p2$, $p3$, etcétera y se supone que el punto 1 ($p1$) precede inmediatamente al punto 2 ($p2$), $p2$ precede inmediatamente a $p3$ y así sucesivamente.

Tabla 2. Operadores temporales de totales.

Operación	Conjunto X	Conjunto de salida
EXPAND(X)	$\{[p1:p1], [p3:p5], [p4:p6]\}$	$\{[p1:p1], [p3:p3], [p4:p4], [p5:p5], [p6:p6]\}$
COLLAPSE(X)	$\{[p1:p1], [p3:p5], [p4:p6]\}$	$\{[p1:p1], [p3:p6]\}$

Fuente: elaboración propia

Debido a que los lenguajes de consulta de las BD relacionales generan nuevas relaciones a partir de otras, para los operadores EXPAND y COLLAPSE se requieren sus contrapartes relacionales. Los operadores de totales relacionales PACK y UNPACK (Date et al., 2003), son las versiones relacionales de los operadores COLLAPSE y EXPAND respectivamente.

Los operadores PACK y UNPACK se comportan así:

PACK recibe como parámetros una relación que contiene al menos un atributo de tipo intervalo y el nombre del atributo de tipo intervalo sobre el que se va a hacer la operación. Retorna una relación, con el mismo encabezado, obtenida al aplicar la operación COLLAPSE sobre el atributo especificado; UNPACK se comporta de manera análoga pero aplica la operación EXPAND sobre el atributo especificado. Estos operadores trabajan bajo una signatura:

Operador Temporal: $R \rightarrow R$

Donde $R = \{ r_1, r_2, \dots, r_z \}$ es un conjunto de relaciones r_i , $1 \leq i \leq z$, cada una posee un atributo de tipo intervalo sobre el que se hace la operación correspondiente.

Sea r una relación que tiene un atributo tipo intervalo llamado *durante*, las expresiones: PACK r ON *durante* y UNPACK r ON *durante*, son ejemplos de los operadores PACK y UNPACK respectivamente. En las figuras 1 y 2 se presentan estos ejemplos.

Relación R		PACK R ON DURANTE	
S#	DURANTE	S#	DURANTE
S2	[p2:p4]	S2	[p2:p5]
S2	[p3:p5]	S4	[p2:p6]
S4	[p2:p5]	S4	[p9:p10]
S4	[p4:p6]	S4	[p16:p16]
S4	[p9:p10]		
S4	[p16:p16]		

Fuente: elaboración propia.

Figura 1. Ejemplo operador PACK.

Relación R		UNPACK R ON DURANTE	
S#	DURANTE	S#	DURANTE
S2	[p2:p4]	S2	[p2:p2]
S4	[p3:p5]	S2	[p3:p3]
S4	[p2:p5]	S2	[p4:p4]
S4	[p9:p10]	S2	[p5:p5]
S4	[p16:p16]	S4	[p2:p2]
		S4	[p3:p3]
		S4	[p4:p4]
		S4	[p5:p5]
		S4	[p6:p6]
		S4	[p9:p9]
		S4	[p10:p10]
		S4	[p16:p16]

Fuente: elaboración propia

Figura 2. Ejemplo operador UNPACK

Los operadores PACK y UNPACK son la base para definir los operadores relacionales temporales U_UNION, U_MINUS, U_INTERSECT, U_RESTRICT y U_JOIN (Date et al., 2003). Estos operadores funcionan análogamente a sus contrapartes relacionales aunque poseen algunas diferencias:

- Las relaciones sobre las que se realizan las operaciones deben tener al menos un atributo tipo intervalo,
- Aplican el operador UNPACK sobre las relaciones recibidas como parámetro (usando

el atributo de tipo intervalo recibido como parámetro) antes de hacer la operación y

- Usan el operador PACK sobre el resultado de la operación (usando el atributo de tipo intervalo recibido como parámetro) antes de mostrar su resultado.

La signature bajo la que trabajan los operadores relacionales temporales es:

Operador temporal: $R \times R \rightarrow R$

En la figura 3 se presenta un ejemplo del operador U_MINUS. Obsérvese la diferencia entre el resultado obtenido con MINUS y el resultado obtenido con U_MINUS.

Relación A		Relación B	
S#	DURANTE	S#	DURANTE
S2	[p2:p4]	S2	[p3:p3]
S4	[p2:p6]	S4	[p2:p4]
S4	[p16:p20]	S4	[p16:p20]

A MINUS B		A U MINUS (DURANTE) B	
S#	DURANTE	S#	DURANTE
S2	[p2:p4]	S2	[p2:p2]
S4	[p2:p6]	S2	[p4:p4]
		S4	[p5:p6]

Fuente: elaboración propia

Figura 3. Ejemplo del operador U_MINUS

El operador U_MINUS es equivalente a las siguientes operaciones:

PACK ((UNPACK A ON (DURANTE))

MINUS (UNPACK B ON (DURANTE))) ON (DURANTE)

2.3. Caso de estudio

El caso de estudio consiste en el historial médico de los pacientes atendidos en un centro de salud. Cada vez que un paciente asiste a una consulta se recolecta la información acerca de los síntomas que padece y el médico receta los medicamentos necesarios para el tratamiento. Aquí se

cuenta con información de carácter temporal. Por ejemplo, los síntomas que padecen los pacientes, en general, se presentan sólo por un período. Similarmente, los medicamentos suelen ser recetados en determinadas dosis que deben ser consumidas durante cierto tiempo. Por otro lado, al ser de interés la información médica y personal de un paciente a lo largo de su vida, los pacientes son considerados también de carácter temporal.

Además de los aspectos temporales mencionados en el párrafo anterior existen otros aspectos que podrían ser modelados como temporales, por ejemplo, el tiempo de vida de los médicos o la duración de una consulta, entre otros. Sin embargo, estos aspectos son modelados como no temporales o no son tenidos en cuenta.

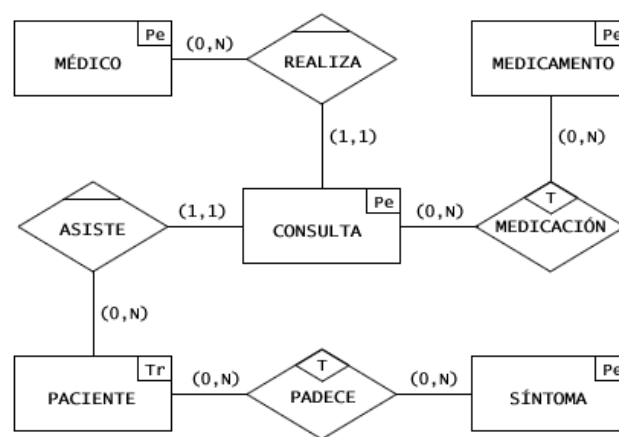
Para modelar el caso de estudio, se examinaron once extensiones temporales al modelo Entidad Relación (ER) (Chen, 1976). Si se usa el modelo ER para representar aspectos temporales, los diagramas resultantes suelen ser complejos y difíciles de entender. Por lo tanto, se han propuesto modelos que agregan al modelo ER la expresividad necesaria para representar la semántica de la información que varía con el tiempo. En (Jensen y Gregersen, 1999) se presenta un estudio de diez de estos modelos: Modelo Entidad Relación Temporal (*Temporal Entity Relationship Model*, TERM) (Klopprogge, 1981), Modelo de Relaciones Atributos Claves y Entidades (*Relationships, Attributes, Keys, and Entities Model*, RAKE) (Ferg, 1985), Modelo para Objetos con Atributos y Relaciones Temporales (*Model for Objects with Temporal Attributes and Relationships*, MOTAR) (Narasimhalu, 1988), Modelo EER Temporal (*Temporal EER Model*, TEER) (Elmasri y Wu, 1990), Modelo EER Temporal Semántico (*Semantic Temporal EER*, STEER) (El-Assal et al., 1990), Modelo ER Temporal (*Temporal ER Model*, TER) (Touzovich, 1991), Modelo Entidad-Relación-Tiempo (*Entity-Relation-Time*, ERT) (Loucopoulos et al., 1992), Modelo EER Temporal (*Temporal EER Model*, TempEER) (Lai et al., 1994),

Modelo TempRT (Kraft, 1996) y Modelo ER Temporal con Objetos Complejos más Identidad de Objetos (*Temporal ER with Complex Objects plus Object Identity*, TERC+) (Zimányi et al., 1997). Además en (Antunes et al., 1997) se presenta el modelo ER Temporal (*Temporal ER*, TempER).

Se seleccionó el modelo TempER (Antunes et al., 1997) debido a los siguientes factores:

- Notación gráfica clara y simple,
- Se diferencian fácilmente las entidades temporales de las no temporales,
- Soporta tiempo válido,
- Permite expresar restricciones como la cardinalidad de las relaciones y
- Ofrece un método de transformación al modelo relacional.

En la figura 4 se presenta el diagrama TempER correspondiente al caso de estudio.



Fuente: elaboración propia

Figura 4. Diagrama TempER.

En los diagramas TempER las entidades se representan con rectángulos que tienen un recuadro en la parte superior derecha. Si una entidad es temporal (transitoria), el recuadro contiene las letras “Tr”; si es no temporal (perenne), contiene las letras “Pe”. Las relaciones se representan con rombos. Si una relación es temporal, el rombo que la representa

contiene otro rombo (pequeño) en la parte superior con la letra “T”; si es no temporal, el rombo que la representa contiene una línea horizontal en la parte superior. Los atributos no se muestran explícitamente en el diagrama sino que se especifican a través de un diccionario de datos asociado con el diagrama.

La descripción de un atributo presenta los siguientes elementos: el nombre del atributo, si es temporal o no temporal y el dominio de los valores que pueden ser asociados con dicho atributo. En la tabla 3 se presenta el diccionario de datos correspondiente al diagrama de la figura 4.

Tabla 3. Diccionario de datos para el modelo TempER.

ENTIDAD	PACIENTE	ENTIDAD	MÉDICO
ATRIBUTOS		ATRIBUTOS	
Cédula	STRING No temporal	Cédula	STRING No temporal
Nombre	STRING No temporal	Nombre	STRING No temporal
Fecha de Nacimiento	DATE No temporal	Especialidad	STRING No temporal
Identificador: (Cédula)		Identificador: (Cédula)	
ENTIDAD	CONSULTA	ENTIDAD	SÍNTOMA
ATRIBUTOS		ATRIBUTOS	
Id	NUMBER No temporal	Id	NUMBER No temporal
Lugar	STRING No temporal	Nombre	STRING No temporal
Fecha	DATE No temporal	Identificador: (Id)	
Identificador: (Id)			
ENTIDAD	MEDICAMENTO	RELACIÓN	MEDICACIÓN
ATRIBUTOS		ATRIBUTOS	
Id	NUMBER No temporal	Dosis	STRING Temporal
Nombre	STRING No temporal		
Identificador: (Id)			
RELACIÓN	PADECE	Las demás relaciones no tienen atributos.	
ATRIBUTOS			
Descripción	STRING Temporal		

Fuente: elaboración propia

2.4. Consulta temporal

Después de la implementación de los operadores temporales y del caso de estudio, se plantean consultas que se basan en la información temporal. Por ejemplo: “Obtener los pacientes que han padecido el síntoma S5 ó el síntoma S6 junto con el tiempo en el que los han padecido”.

El hecho que un paciente padezca algún síntoma durante algún periodo se encuentra almacenado en la tabla PADECE. En la tabla 4 se presentan datos de muestra para dicha tabla.

En la tabla 5 se presentan diferentes planteamientos de la consulta discutida en esta sección. En la primera columna se muestra el planteamiento realizado según lo propuesto en Date et al. (2003) (las llaves indican proyección);, en la segunda columna, se muestra cómo se plantearía la consulta

si se hubieran podido implementar los operadores relacionales temporales como operadores *nativos* del sistema; finalmente, en la tercera columna se muestra el planteamiento de la consulta con los operadores tal como fueron implementados.

Tabla 4. Datos de ejemplo para la tabla PADECE.

Paciente	Sintoma	TV
P1	S1	[1984-11-22, 1985-12-22]
P2	S2	[1980-06-14, 1980-06-20]
P3	S3	[2005-10-01, 2005-12-01]
P1	S4	[1982-04-01, 1982-10-01]
P2	S5	[1981-03-10, 1981-03-11]
P3	S6	[2003-02-15, 2003-02-25]
P4	S7	[2001-03-20, 2001-04-01]
P5	S5	[2005-12-01, 2005-12-10]
P2	S6	[1981-03-10, 1981-06-11]

Fuente: elaboración propia

Tabla 5. Planteamientos de la consulta.

Planteamiento teórico	Planteamiento “ideal” en SQL	Planteamiento logrado en el SQL de ORACLE
U_RESTRICT padece ON (tv) WHERE síntoma= 'S5' OR síntoma='S6' {paciente, tv}	SELECT paciente, U_RESTRICT(tv) AS tv FROM padece WHERE síntoma='S5' OR síntoma='S6'	SELECT paciente, intervalo(fecha.i, fecha.f) tv FROM (SELECT paciente, U_RESTRICT(tv) f FROM padece WHERE síntoma = 'S5' OR síntoma = 'S6' GROUP BY paciente) t, TABLE(t.f) fecha;

Fuente: elaboración propia

En la tabla 6 se muestra el resultado de la consulta tomando como base los datos de la tabla 4.

Tabla 6. Resultado de la consulta.

PACIENTE	TV
P2	[1981-03-10, 1981-06-11]
P3	[2003-02-15, 2003-02-25]
P5	[2005-12-01, 2005-12-10]

Fuente: elaboración propia

La consulta discutida en esta sección se puede plantear en SQL sin usar los operadores temporales, aunque de forma muy laboriosa. También se puede implementar en un lenguaje procedural, como PL/SQL de Oracle, pero igualmente se llega a soluciones muy complejas.

3. RESULTADOS

En este trabajo fue posible implementar operadores temporales en el SGBD Oracle, aunque fue necesario modificar el planteamiento de algunos de ellos. Posteriormente se plantearon consultas que requieren información temporal.

A pesar de las facilidades ofrecidas por Oracle, no fue posible implementar los operadores temporales PACK y UNPACK, de manera que se comportaran como operadores *nativos* del sistema. Sin embargo, fue posible simular su comportamiento usando la función TABLE de Oracle, que permite mostrar el resultado de los operadores EXPAND y COLLAPSE como una relación.

Los operadores relacionales temporales se implementaron como operadores nativos a manera de funciones de agregación definidas por el usuario, aunque fue necesario modificar su planteamiento ya que, como se mencionó en el párrafo anterior, no fue posible implementar los operadores PACK y UNPACK como operadores nativos del sistema.

Tomando como ejemplo una consulta sencilla, se formuló con y sin usar los operadores implementados. Se encontró una gran diferencia en la cantidad de código necesaria, lo que demostró la expresividad de los operadores implementados.

4. DISCUSIÓN

Fue posible comprobar que el planteamiento de consultas temporales usando los operadores implementados es más simple, ya que, incluso, para una consulta temporal sencilla, se presenta una gran reducción en la codificación necesaria para su implementación respecto al planteamiento de la consulta sin usar dichos operadores. Esta diferencia se debe a que en los operadores está inmersa gran parte de la lógica de la consulta y cuando no se usan éstos, la lógica debe expresarse en la formulación de la consulta.

Con respecto a la implementación de los operadores temporales, las limitantes encontradas podrían ser resueltas si Oracle ofreciera algún mecanismo que permitiera la creación de operadores relacionales definidos por el usuario, análogamente a como se incorporaron las funciones de grupo definidas por el usuario a partir de Oracle 9i, (Oracle, 2002). Otra alternativa sería implementar los operadores en un SGBD de código abierto, en el que sea posible modificar su código fuente de tal manera que permitiera la creación de operadores relacionales definidos por el usuario.

REFERENCIAS

- ALLEN, J. F., (1983). Maintaining Knowledge about Temporal Intervals. CACM, 16: 842-843.
- ANTUNES, D., HEUSER, C. y EDELWEISS, N., (1997). TempER: uma Abordagem para Modelagem Temporal de Banco de Dados. Revista de Informática Teórica e Aplicada, 4: 49-86.
- CHEN, P., (1976). The Entity-Relationship Model – Toward a Unified View of Data. ACM Transaction on Database Systems, 1: 9-36.
- DATE, C. J., (2001). Introducción a los Sistemas de Bases de Datos. México. Pearson Educación. 730-768.
- DATE, C. J. DARWEN, H. y LORENTZOS, N. A., (2003). Temporal Data and the Relational Model. San Francisco. Morgan Kaufmann Publishers.
- EL-ASSAL, I., ELMASRI, R. y KOURAMAJIAN, V., (1990). Semantics of Temporal Data in an Extended ER Model. Proceedings of the 9th International Conference on the Entity-Relationship Approach, 249-264.
- ELMASRI, R. y WUU, G., (1990). A Temporal Model and Query Language for ER databases. Proceedings of the Sixth International Conference on Data Engineering, 76-83.
- FERG, S., (1985). Modeling the Time Dimension in an Entity-Relationship Diagram. Proceedings of the 4th International Conference on the Entity-Relationship Approach, 280-286.
- JENSEN, C. S., CLIFFORD, J., ELMASRI, R., GADIA, S. K., HAYES, P. J. y JAJODIA, S., (1994). A Consensus Glossary of Temporal Database Concepts. ACM SIGMOD Record, 23: 52-64.

- JENSEN, C. S. y GREGERSEN, H., (1999). Temporal Entity Relationship Models – A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 11: 464-497.
- KLOPPROGGE, M. R., (1981). TERM: An Approach to Include the Time Dimension in the Entity-Relationship Model. *Proceedings of the Second International Conference on the Entity Relationship Approach*, 473-508.
- KRAFT, P., 1996. Temporale Kvaliteter i ER Modeller. The Aarhus School of Business, Department of Information Science.
- LAI, V., KUILBOER, J-P. y GUYNES, J. L. (1994). Temporal Databases: Model Design and Commercialization Prospects. *Data Base*, 25: 6-18.
- LOUCOPOULOS, P., THEODOULIDIS, C. y WANGLER, B. (1992). The Entity Relationship Time Model, Conceptual Modelling, Databases, and CASE: An Integrated View of Information Systems Development. John Wiley & Sons Inc. 87-115.
- NARASIMHALU, A. (1988). A Data Model for Object-Oriented Databases with Temporal Attributes and Relationships. Reporte técnico, National University of Singapore.
- ORACLE Corporation. (2002). Oracle 9i Application Developer's Guide Object-Relational Features. En: http://downloadwest.oracle.com/docs/cd/B10501_01/appdev.920/a96594/whatsnew.htm. Recuperado el 03 de noviembre de 2006.
- TAUZOVIH, B. (1991). Toward Temporal Extensions to the Entity-Relationship Model. *Proceedings of the 10th International Conference on the Entity Relationship Approach*, 163-179.
- ZIMÁNYI, E., PARENT, C., SPACCAPIETRA, S. y PIROTTE, A. (1997). TERC+: A Temporal Conceptual Model. *International Symposium on Digital Media Information Base*.