

3

Autorizo a la Director General de Bibliotecas de la UNAM a difundir en formato electrónico e impreso el contenido de mi trabajo recapital.

NOMBRE: Antonio Benito Patricio

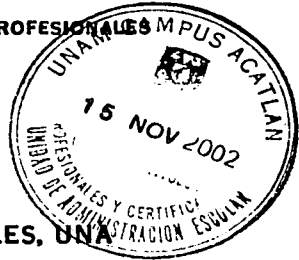
FECHA: 16- Noviembre - 2002

FIRMA: [Signature]



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
"ACATLAN"



"BASES DE DATOS TEMPORALES, UNA HERRAMIENTA PARA LA TOMA DE DECISIONES"



T E S I S
QUE PARA OBTENER EL TITULO DE
LICENCIADO EN MATEMATICAS APLICADAS Y
COMPUTACION
PRESENTA:

ANTONIO BENITO PATRICIO

ASESOR: M. EN C. JUDITH JARAMILLO LOPEZ



NOVIEMBRE 2002

TESIS CON FALLA DE ORIGEN



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

PAGINACION DISCONTINUA

Agradecimiento

Antes de mencionar a las personas que son y han sido una parte muy especial a lo largo de mi vida; quiero decirle a la persona más importante para mí, que eso que tanto deseaba y en lo que puse tanto empeño para lograrlo y sobre todo amor, se ha vuelto realidad. Aunque ya no estés conmigo para celebrarlo quiero que sepas que sin tí no lo hubiera logrado. Esta meta de mi vida esta dedicada especialmente a tí y te encuentres donde te encuentres se que estarás disfrutando de ésto tanto o más que yo. Muchas gracias Ina por haber cambiado el rumbo de mi vida.

Le agradezco a mis padres Chivis, Juanita y Toño por todo lo que han hecho para que haya llegado hasta este momento, en la forma en la que han contribuido con consejos, regaños y sobre todo el amor que pusieron para que sea la persona que soy.

A mis hermanos Alicia, Carlos, Fernando, Giovanna, René que son una parte muy importante de mi vida, con quienes he pasado buenos y malos momentos les agradezco porque en el momento que los he necesitado han estado ahí. Sólo puedo decirles gracias y que los quiero mucho.

Les doy las gracias a mis amigos que me han apoyado en diferentes etapas de mi vida, que han sido parte importante a lo largo de la misma y sobre todo que me quieren con mis virtudes y defectos. En especial a Roberto Colin y su familia, quienes estuvieron conmigo en una etapa difícil de mi vida. A todos los demás no se ofendan pero sería muy largo mencionar el porqué son todos y cada uno de ustedes importantes en mi vida: Adriana Quintana, Fabiola Alcántara, Fabiola Ferrer, Gabriela Ortega, Marilú Mata, Adolfo Pablo, Damián Vázquez, Emilio Quiroz, Héctor Martínez, Jonathan Gutiérrez, Juan Pablo Correa, Uriel Lira y Victor Vega.

A mi asesora Judith Jaramillo le agradezco el tiempo que le dedicó de principio a fin al presente trabajo y en especial por la amistad que me ha brindado.

A mis profesores les agradezco el tiempo que dedicaron en mi educación académica en especial a: Alfonso Verdín y Jorge García, porque ellos fueron un paso más allá del simple hecho de ser mis profesores, me enseñaron la importancia de valores que ellos como personas poseen y para mí son un ejemplo a seguir.

A mis compañeros del Centro Cómputo que colaboraron en la elaboración del presente trabajo en especial a: Anabel Moreno, Ángeles Cabrera, Lupita Fernández y Fernando Manzanares gracias a todos por la ayuda que me han brindado.

A la Universidad le agradezco el haber permitido que fuera uno de sus hijos, por la educación que me proporcionó y por dejar equipo de cómputo bajo mi custodia, que permitió la elaboración de las pruebas necesarias para la demostración de la hipótesis de esta tesis.

TESIS CON
FALLA DE ORIGEN

M
10

Introducción	i
Hipótesis	ii
Justificación	ii
Objetivo	ii
CAPITULO I <i>Las Bases de datos temporales</i>	1
1.1 Definición de una base de datos temporal	1
1.2 Fundamentos matemáticos de las bases de datos temporales.....	2
1.2.1 Operadores dentro de las Series de Tiempo	3
1.2.2 Categorías de las Series de Tiempo	5
1.2.3 Componentes de una Serie de Tiempo	5
1.2.4 Modelos de Series de Tiempo	6
1.2.5 Alcance de las Series de Tiempo en la Bases de Datos Temporales	10
1.3 Espacio de diseño del modelo temporal	11
1.3.1 Estructura Temporal	11
1.3.2 Representación Temporal	13
1.3.3 Orden Temporal	13
1.3.4 Historia Temporal	13
1.4 Tipos de bases de datos temporales	14
1.4.1 Bases de datos Snapshot	14
1.4.2 Bases de datos Históricas	15
1.4.3 Bases de datos Rollback	16
1.4.4 Bases de datos Bitemporales	17
CAPITULO II <i>El elemento tiempo en las bases de datos</i>	18
2.1 El tiempo en los manejadores de bases de datos	18
2.1.1 El tiempo en el modelo relacional	18
2.1.2 Las series de tiempo en el modelo objeto relacional	25
2.2 Series de tiempo en un manejador de bases de datos (Oracle 8i)	27
2.2.1 Calendario	27
2.2.2 Tipos de series que utiliza el manejador Oracle 8i	29
2.2.3 La arquitectura que presentan las series de tiempo en Oracle 8i	29
2.2.4 Reglas para la consistencia en las series de tiempo	32
CAPITULO III <i>El papel de las bases de datos temporales en la toma de decisiones</i>	33
3.1 Concepto de Información	33
3.1.1 Cualidades de la Información	33
3.1.2 Importancia de la Información	34
3.2 Teoría de decisiones	35
3.2.1 Criterios para la toma de decisiones	35
3.3 Como actúan las bases de datos temporales en la toma de decisiones	36
3.4 Factores a analizar para la implementación de una base de datos temporal	41
CAPITULO IV <i>Aplicación de una base de datos temporal</i>	42
4.1 Planteamiento del Problema	42
4.2 Propuesta	42
4.3 Alcance	42
4.4 Desarrollo	43

TESIS CON
FALLA DE ORIGEN

4.4.1 Esquema Temporal de la base de datos del Área de Cursos del Centro de Cómputo ...	45
4.4.2 Elaboración de los elementos de series de tiempo que requiere el Área de Cursos	46
4.5 Pruebas de funcionamiento de la base de datos temporal del Área de Cursos	55
Conclusiones	63
Glosario	65
Apéndice A <i>Elementos necesarios para la utilización de las series de tiempo</i>	68
Apéndice B <i>Diccionario de datos</i>	69
Referencias Bibliográficas	90

TESIS CON
FALLA DE ORIGEN

Introducción

El hombre desde sus inicios se ha visto en la necesidad de almacenar información, con la finalidad de dar a conocer sus avances y conocimientos a generaciones posteriores. Ejemplo de esto lo podemos ver con las pinturas rupestres, en las que el hombre primitivo daba cuenta de sus actividades y al mismo tiempo enseñaba la forma en la que se efectuaban actividades como la cacería de algunos animales, o los egipcios que almacenaban su conocimiento en pergaminos.

También se sabe que la pérdida de información genera atrasos, como ejemplo citaremos a la Biblioteca de Alejandría en la que se perdieron siglos de avances, debido a que en este lugar se encontraba una gran parte del conocimiento de ese entonces. En la era moderna la información se ha ido almacenando en diversos medios como: papeles en archiveros, cintas magnéticas, disquetes y a últimas fechas en discos compactos, pero toda esta información se encuentra dispersa y en algunos casos repetida, generando inconsistencia y pérdida de la misma. En la búsqueda de mejores métodos de almacenamiento se llegó a la creación de las bases de datos, que han revolucionado el proceso de almacenamiento de la información, ya que permiten una rápida localización de la información y consistencia en los datos.

A lo largo del presente trabajo se mostrará el valor que pueden llegar a tener las bases de datos temporales (que de aquí en adelante denominaremos BDT) debido a que el empleo de las mismas hace posible almacenar la información con una propiedad muy importante: *el tiempo*. Al utilizar este tipo de bases de datos se obtiene como resultado un manejo más eficaz de la información.

Se explicará a fondo lo que son las BDT, los diferentes tipos que existen de éstas, así como los elementos de los que se encuentran conformadas, se explicarán los factores que hay que tomar en cuenta al momento de efectuar el diseño del modelo temporal. Las BDT incorporan las bases de un elemento matemático muy importante dentro de su funcionamiento que es el de las series de tiempo, motivo que hace necesaria una explicación de lo que son las series de tiempo y la gama de posibilidades que se presentan al hacer uso de ellas.

Otro punto que se toca en este trabajo es el referente a la toma de decisiones; se explicarán los fundamentos teóricos y se hará referencia a la importancia de la información.

Las BDT como todo, han ido evolucionando a través de los diferentes modelos de bases de datos; dado esto se explicará la forma en la que se ven involucradas dentro del modelo relacional y en el modelo orientado a objetos, dentro de un punto de vista teórico. Así mismo se mostrará que estas bases de datos se encuentran divididas en varios tipos, explicando cada uno de ellos.

Todo lo anterior sirve como fundamento para ver como las BDT, son un mecanismo que proporciona la información que se necesita al requerir datos para ser analizados y así ayudar al proceso de la toma de decisiones.

Se verá la forma en la que el manejador de bases de datos Oracle 8i (versión 8.1.7.0.0) incorpora la teoría de las bases de datos temporales en su arquitectura a través de las series de tiempo. Finalmente, se utilizarán los conceptos de las BDT en una aplicación que responde a una necesidad que presenta el Área de Cursos del Departamento de Servicios de Cómputo (DSC) de la ENEP Acatlán.

El trabajo busca comprobar que el manejo de las bases de datos temporales, permite conocer de una manera ordenada y precisa el comportamiento que presentan los datos conforme transcurre el tiempo, ya que en las empresas e instituciones se requiere tomar diariamente decisiones, también se busca dar a conocer las nuevas tecnologías en el área de las bases de datos.

TESIS CON
FALLA DE ORIGEN

Hipótesis

La hipótesis a comprobar a lo largo de este trabajo será:

El manejo de las bases de datos temporales permite conocer de una manera ordenada y precisa el comportamiento que presentan los datos conforme transcurre el tiempo. Mejorando así el proceso que involucra la recopilación de información necesaria para la toma de decisiones.

Justificación

En las empresas, instituciones o en cualquier sitio en donde diariamente se tienen que tomar decisiones como: la contratación de personal, el despido del mismo, sacar al mercado un nuevo producto, retirar un producto del mercado, en fin cualquier cosa que implique la toma de una decisión requiere que se tenga la información necesaria en el momento preciso, es por ello que se requiere aplicar nuevas estrategias para almacenar la información de una forma más eficiente.

Objetivo

Dar a conocer nuevas tecnologías en el área de las bases de datos, mostrando como éstas, al trabajar de manera conjunta con las matemáticas ayuda a la toma de decisiones.

CAPITULO I

Las Bases de datos temporales

El estudio de las bases de datos temporales se puede decir que dió inicio hace 20 años, pero en la última década se le ha puesto más énfasis debido a que la comunidad científica internacional entró de lleno a su estudio, como ejemplo se puede citar al *Institute of Electrical and Electronics Engineers, Inc. (IEEE)*. La importancia que se le ha dado a últimas fechas, es debido a que se han vislumbrado los beneficios que se logran al modelar la dimensión temporal del mundo real; ésto se vería reflejado en las aplicaciones que se emplean en distintas áreas como: la economía, la banca, clínicas, líneas aéreas, etc.

1.1 Definición de una base de datos temporal

"Una base de datos temporal es aquella que soporta algunos aspectos de tiempo" [OPH98]

La mayoría de las bases de datos que se manejan hoy en día, buscan modelar el estado del mundo real en un punto de tiempo determinado, las bases de datos temporales modelan los estados del mundo real a través del tiempo. Podemos decir que una relación temporal es aquella en la que los datos tienen asignado un tiempo determinado durante el cual son verdaderas. Las BDT manejan un tiempo válido y un tiempo de transacción.

Lo que se ha buscado con las bases de datos temporales es definir un modelo particular temporal, así como su incorporación a un *Database Management System (DBMS)*, lo cual ha causado grandes problemas, porque no es posible representar la noción del tiempo abarcando sus múltiples facetas en un modelo relacional. Ésta es una de las razones por las cuales se ha decidido que los modelos temporales relacionales soporten un modelo discreto y lineal. Cuando se trabaja con un modelo de tiempo, se debe trabajar con una secuencia finita de *chronons*, en términos matemáticos, es isomórfico para una secuencia finita de números naturales. La secuencia *chronons* puede ser pensada como la representación de una partición de la línea de tiempo real de tamaños iguales e indivisibles, por lo tanto los *chronons* son pensados como segmentos de tiempo. Por ejemplo fracciones de segundos o segundos, depende de las necesidades que se tengan para el procesamiento de datos.

Definición 1.1: Un instante de tiempo es un punto cualquiera que ocurre en el tiempo, ese instante tiene un valor único en la línea de tiempo, como se puede observar en la figura 1.1.

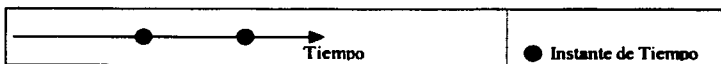


Figura 1.1 Instante de tiempo

Definición 1.2: Un intervalo de tiempo es el tiempo que transcurre entre dos instantes de tiempo dados, los instantes son los que ponen la pauta al intervalo porque le indican en que punto se da inicio y en que punto se pone fin, como se observa en la figura 1.2.

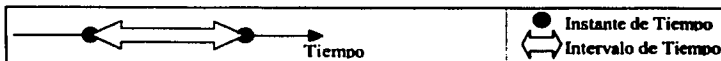


Figura 1.2 Intervalo de tiempo

Por otro lado un intervalo de tiempo se encuentra dividido por una secuencia de *chronons* en donde a su vez un *chronon* representa todos y cada uno de los instantes que ocurren durante el mismo (véase figura 1.3).



Figura 1.3 Representación de los chronons

Una serie es una secuencia finita que se encuentra contenida dentro de dos cotas marcadas por el punto de inicio y el punto de término, pero la serie puede ser vista como un pequeño segmento de una secuencia que posee cotas infinitas que a continuación se muestran como resultado de la combinación de las ecuaciones 1.1, 1.2 y 1.3.

$$Y_t \Big|_{t=-\infty}^{\infty} (\dots, Y_{-2}, Y_{-1}, Y_0, Y_1, Y_2, Y_3, \dots, Y_t, Y_{t+1}, Y_{t+2}, Y_{t+3}, \dots) \quad \dots \quad (1.4)$$

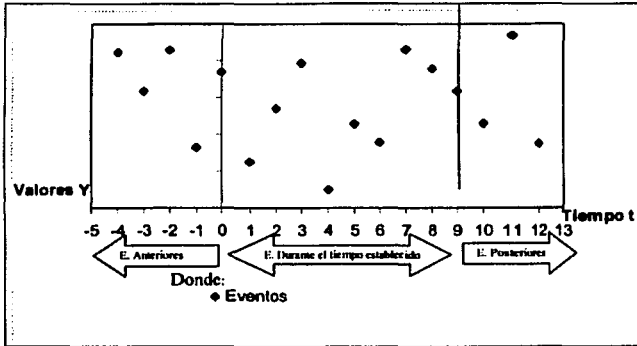


Figura 1.5 Eventos anteriores y posteriores al tiempo definido

En la figura 1.5 se puede observar la secuencia que posee un instante de tiempo en el cual se da origen al intervalo de interés, así como un instante en el cual se le pone fin a dicho intervalo, el subíndice t establece la marca de tiempo en la cual sucede el evento y .

Una serie de tiempo muy importante es el proceso de ruido blanco Gaussiano denotado como:

$$y_t = \varepsilon_t \quad \text{donde} \quad \varepsilon_t \Big|_{t=-\infty}^{\infty}$$

1.2.1 Operadores dentro de las Series de Tiempo

Definición 1.5: Un operador dentro de las series de tiempo es aquel elemento que transforma a una o a un grupo de éstas en nuevas series de tiempo.

Operador de Multiplicación

$$y_t = \beta x_t \quad \text{Donde } \beta \text{ es un escalar.}$$

Operador de Suma

$$y_t = x_t + z_t \quad \text{En este caso se observa como se pueden sumar dos series de tiempo mediante el operador "+"}$$

Operador de Retraso

El operador de retraso (conocido en inglés como "Lag Operator") es muy útil al momento en el cual se manejan fórmulas que involucran a las series de tiempo [HAM94]. Se dice que un valor al que conoceremos como "Y" para un tiempo determinado "t" es igual al valor de x en el tiempo (t-1) representado de la siguiente forma:

$$Y_t = x_{(t-1)} \quad (1.5)$$

Cuando se aplica el operador de retraso a una serie x_t la operación queda de la siguiente forma:

$$Lx_t = x_{t-1} \quad \text{Para cuando } n=1$$

$$L^2x_t = x_{t-2} \quad \text{Para cuando } n=2$$

De lo que podemos decir, que en forma general la ecuación queda como sigue:

$$L^n x_t = x_{t-n}$$

Este operador de retraso es distributivo respecto a la suma, ya que primero se pueden sumar dos series y posteriormente llevar a cabo el uso del operador de retraso como se muestra a continuación:

(x_t, z_t) Se muestran dos series de tiempo

$(x_t + z_t)$ Se aplica el operador de Suma

$(x_{t-1} + z_{t-1})$ Se aplica el operador de retraso

$(Lx_t + Lz_t)$ Otra forma de representar el operador de retraso o bien de forma inversa, primero hacer uso del operador de retraso y al final efectuar la suma de las series.

(x_t, z_t) Se muestran dos series de tiempo

(x_{t-1}, z_{t-1}) Se aplica el operador de retraso

(Lx_t, Lz_t) Otra forma de representar el operador de retraso

$(Lx_t + Lz_t)$ Se aplica el operador de Suma

El operador de retraso y el operador de multiplicación son conmutativos, como se puede ver:

x_t	Serie de tiempo	x_t	Serie de tiempo
βx_t	Al multiplicar por un escalar	x_{t-1}	Al aplicar el valor de retraso
βx_{t-1}	Al aplicar el valor de retraso	Lx_t	Representado de otra forma
βLx_t	Representado de otra forma	βLx_t	Al multiplicar por un escalar

Podemos ver que el operador de retraso sigue las reglas del álgebra [HAM94].

1.2.2 Categorías de las Series de Tiempo

Debido al alcance que presentan las series de tiempo se encuentran divididas en las siguientes categorías [HUR79]:

- Análisis
- Control
- Pronóstico
- Simulación

En la categoría de análisis, se ve cómo una serie de tiempo nos ayuda a comprender cómo los eventos que ocurren en un período de tiempo pueden afectar al resultado final en el problema que originó el estudio.

El control se refiere al caso en el cual la información que es recolectada en un lapso de tiempo definido sirve para diseñar un mecanismo de vigilancia. Este mecanismo servirá a largo plazo para que el problema sea solucionado.

El pronóstico puede definirse como el resultado del estudio de un problema mediante el uso series de tiempo. El problema tiene un conjunto de eventos que ocurren en un lapso de tiempo, ahora bien como estos eventos pueden ser medibles se puede predecir el comportamiento de dichos eventos con características semejantes o iguales en el futuro. Un punto importante que cabe aclarar es que los pronósticos hechos a muy largo plazo pueden caer en grandes errores .

La simulación está ligada con el pronóstico, ya que mediante alteraciones en las características de un evento conocido se pretende obtener el comportamiento de un evento en un punto en el tiempo.

1.2.3 Componentes de una Serie de Tiempo

Una serie de tiempo se conforma por cuatro componentes que son:

- Ciclos
- Componentes Estocásticos
- Estacionalidad o Variación Estacional
- Tendencia

Los ciclos son los componentes que generalmente se pueden generar a través de los años ya que los casos de estudio presentan constantes variaciones y como su nombre lo dice es la repetición de los eventos en un lapso de tiempo determinado.

Los componentes estocásticos *son aquellos movimientos en los datos que no presentan tendencia ni estacionalidad* [HUR79], tales movimientos no son aleatorios sino que existen patrones (existe una dependencia), debido a que la información obtenida en el pasado nos dice lo que puede ocurrir en el futuro.

La estacionalidad se refiere a *la variación sistemática de los datos que se repiten periódicamente* [HUR79], y ayuda a conocer el posible comportamiento de los eventos en un período de tiempo.

La tendencia puede definirse como el crecimiento y decrecimiento en el nivel de los datos, razón por la cual es muy importante medirla ya que en términos matemáticos *es la línea recta o curva que es trazada a través de los datos* [HUR79], luego entonces es la medida que nos da el comportamiento de la serie de tiempo.

1.2.4 Modelos de Series de Tiempo

Dentro del campo de las series de tiempo contamos con varios modelos [MAT95] que se aplicarán dependiendo de la situación en la que nos encontremos; estos modelos son los siguientes:

- Modelos de Nivel
- Modelos de Tendencia
- Modelos Estacionales
- Modelos Estacionales con Tendencia

1.2.4.1 Modelos de Nivel

Un modelo de nivel es aplicable en el momento en el cual los eventos que existen en un periodo de tiempo son constantes hasta cierto punto alrededor de un valor C fijo desconocido [MAT95]. Como se puede observar en la figura 1.6 los valores de la demanda pueden localizarse por arriba o por debajo del valor constante C , a estas variaciones que existen entre la constante C y los valores de los eventos los denominaremos errores aleatorios e_t .

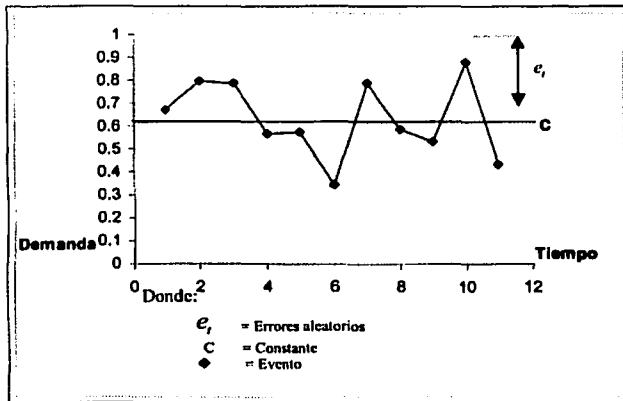


Figura 1.6 Representación de un Modelo de Nivel

1.2.4.2 Modelos de Tendencia

Este modelo se aplica en el momento en el cual se ve que los eventos dados en un periodo de tiempo presentan un patrón creciente o decreciente [MAT95] (modelo lineal), se traza una línea que representa la tendencia que presentan los eventos mencionados (Figura 1.7). En este caso también se presentan los errores aleatorios denotados de la misma forma que en el modelo anterior e_i .

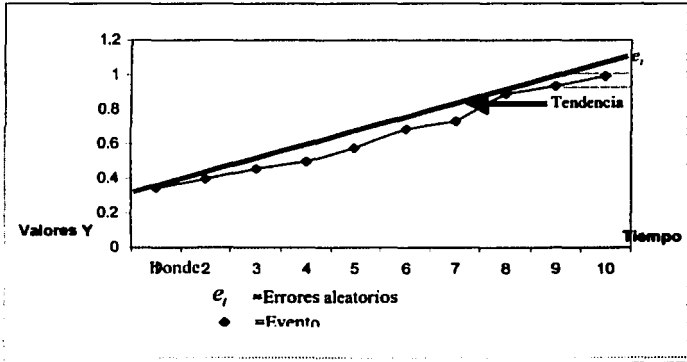


Figura 1.7 Representación de un Modelo de Tendencia Lineal

También se presentan los casos en los cuales los eventos presentan un modelo no lineal, como se puede observar en la figura 1.8.

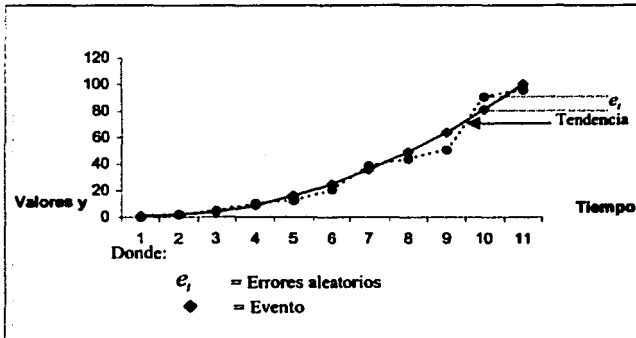


Figura 1.8 Representación de un Modelo de Tendencia No Lineal

1.2.4.3 Modelos Estacionales

Este modelo es aplicable cuando los eventos realizados en un lapso de tiempo presentan semejanza con los eventos efectuados en un lapso de tiempo anterior, es decir, los eventos siguen un patrón determinado[MAT95].

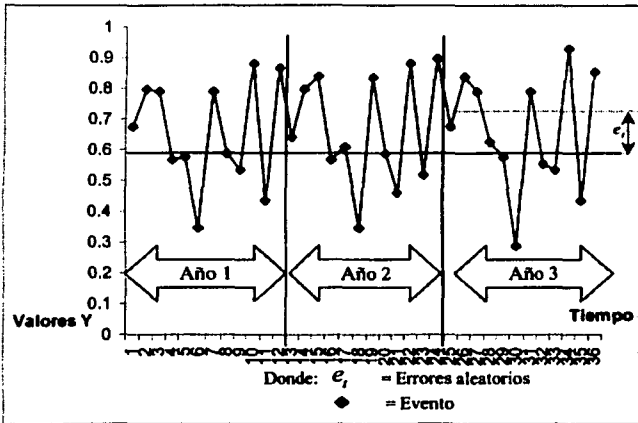


Figura 1.9 Figura 1.6 Representación de un Modelo Estacional

En la figura 1.9 se observan eventos ocurridos a lo largo de tres años en los cuales se puede ver que siguen un patrón de comportamiento, aquí también se maneja el concepto de error aleatorio (e_t).

1.2.4.4 Modelos Estacionales de Tendencia

Los modelos estacionales de tendencia cumplen con características de un modelo de tendencia, debido a que los eventos que se presentan en un periodo de tiempo muestran un patrón creciente o decreciente (modelo lineal), así mismo, presentan características de un modelo estacional en donde los eventos que se realizan en un lapso de tiempo presentan semejanza con eventos realizados en el pasado [MAT95].

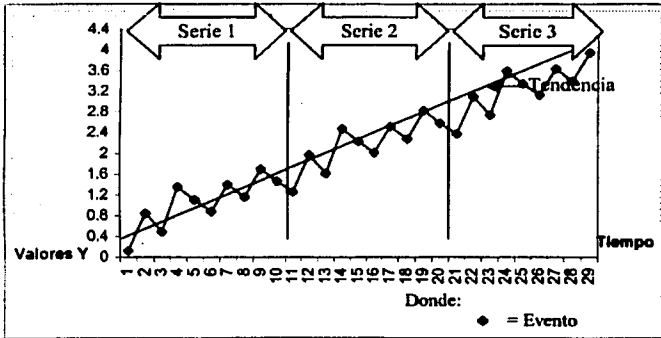


Figura 1.10 Representación de un Modelo Estacional de Tendencia

1.2.4.5 Modelos No Estacionales

Los modelos no estacionales son aquellos en los que los eventos se disparan a un grado tal que es impredecible el momento en el cual van a aparecer, porque pueden existir periodos en los cuales aparezca una gran cantidad de eventos, así como periodos en los cuales éstos sean muy pocos o ninguno. La figura 1.11 muestra periodos en los cuales se puede ver cómo la serie no presenta relación alguna entre los periodos.

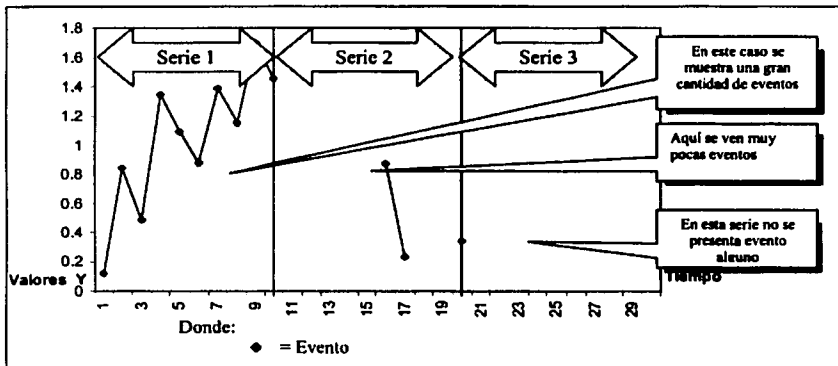


Figura 1.11 Representación de un Modelo No Estacional

1.2.5 Alcance de las Series de Tiempo en la Bases de Datos Temporales

La función principal de las series de tiempo, es llevar a cabo la predicción de eventos que pueden suceder en un futuro basados en el comportamiento de los mismos eventos, que ocurren en un período similar y con varias mediciones de por medio.

Las BDT sólo tocan una pequeña parte de la teoría de las series de tiempo, esto se debe a que la elaboración de los diferentes modelos existentes en las series de tiempo es complejo, debido a que para ello se requiere tomar en cuenta varios factores que no siempre son constantes, estas variaciones pueden ser: fenómenos naturales, económicos, sociales, políticos, etc.

Las BDT utilizan los conceptos básicos de las series de tiempo como son: El manejo de eventos que ocurren en un lapso de tiempo definido por dos puntos en el tiempo (intervalo de tiempo), el momento en el cual ocurre el evento (instantes de tiempo), los *chronons* que son la unidad mínima de tiempo en que ocurren los eventos que se están midiendo.

Hay que aclarar un punto muy importante que es el siguiente: Las BDT no van a realizar los ajustes que requieren las series de tiempo para poder elaborar las predicciones y mucho menos harán las predicciones de los posibles eventos, éstas se limitan a almacenar la información de una manera que permite conocer el comportamiento de los datos en un tiempo definido.

1.2.5 Alcance de las Series de Tiempo en la Bases de Datos Temporales

La función principal de las series de tiempo, es llevar a cabo la predicción de eventos que pueden suceder en un futuro basados en el comportamiento de los mismos eventos, que ocurren en un periodo similar y con varias mediciones de por medio.

Las BDT sólo tocan una pequeña parte de la teoría de las series de tiempo, ésto se debe a que la elaboración de los diferentes modelos existentes en las series de tiempo es complejo, debido a que para ello se requiere tomar en cuenta varios factores que no siempre son constantes, estas variaciones pueden ser: fenómenos naturales, económicos, sociales, políticos, etc.

Las BDT utilizan los conceptos básicos de las series de tiempo como son: El manejo de eventos que ocurren en un lapso de tiempo definido por dos puntos en el tiempo (intervalo de tiempo), el momento en el cual ocurre el evento (instantes de tiempo), los *chronons* que son la unidad mínima de tiempo en que ocurren los eventos que se están midiendo.

Hay que aclarar un punto muy importante que es el siguiente: Las BDT no van a realizar los ajustes que requieren las series de tiempo para poder elaborar las predicciones y mucho menos harán las predicciones de los posibles eventos, éstas se limitan a almacenar la información de una manera que permite conocer el comportamiento de los datos en un tiempo definido.

1.3 Espacio de diseño del modelo temporal

Lo que se busca en un espacio de diseño, es lograr un modelo que cubra las necesidades de una base de datos temporal y ésto se logra mediante la utilización de un conjunto de etapas que conforman el espacio de diseño, es decir, la etapa en la que se define la forma en la que va a funcionar el modelo. El modelo se encuentra dividido en cuatro etapas [OPH98] que son:

- Estructura Temporal
- Representación Temporal
- Orden Temporal
- Historia Temporal

1.3.1 Estructura Temporal

La estructura temporal se refiere a la forma en la que se va a manejar el tiempo, el dominio en el que se van a mover los eventos que se van a manejar, así como la determinancia de los eventos manejados [OPH98].

- Principios Temporales

Los principios que se manejan son el absoluto y el relativo:

Absoluto:	En este caso se tiene conocimiento del lugar donde se localiza un punto en el tiempo. El absoluto maneja dos principios. <ul style="list-style-type: none">Instante Es un punto específico en el tiempo.Intervalo Es un periodo de tiempo entre dos puntos específicos (instantes) que son los límites superior e inferior del intervalo.
Relativo:	En este caso se entiende que es un evento en cualquier punto del tiempo y solamente maneja un principio que es: <ul style="list-style-type: none">Span Longitud de un periodo.

- Dominio Temporal

En el dominio temporal se define una escala para los principios temporales. Los dominios que se manejan pueden ser de dos tipos:

Dominio Discreto:	Comprende al conjunto de los números enteros, y funciona para cualquier principio temporal que se quiera emplear mientras posea un único antecesor y un único predecesor.
Dominio Continuo:	Comprende al conjunto de los números reales. También funciona para cualquier principio temporal que se quiera manejar.

- Determinación Temporal

Aquí encontramos que la determinación se refiere al conocimiento del tiempo de duración de un evento y encontramos dos casos:

Principio Temporal Determinado:	Es el caso en el cual se conoce con precisión el tiempo de duración de un evento
Principio Temporal Indeterminado:	Es el caso en el cual se conoce sólo hasta cierto punto el tiempo de duración de un evento.

Se pueden determinar una serie de combinaciones que cubren lo visto hasta este punto, por ejemplo:

Instantes de Tiempo Continuo: Los instantes continuos de tiempo son puntos que poseen una precisión infinita debido a que no presentan un periodo de indeterminancia.

Intervalos de Tiempo Continuo Determinado: Se encuentran asociados a una actividad que ocurre durante todo el intervalo.

Intervalos de Tiempo Continuo Indeterminado: Se refiere a aquella actividad que ocurre en algún punto del intervalo.

Instantes e Intervalos de Tiempo Discreto: En este caso cada instante de tiempo es al mismo tiempo determinado e indeterminado, estos instantes son semejantes a los intervalos de tiempo continuo. Cada instante de tiempo determinado tiene una Granularidad (G_i) asociada a él. Los instantes de tiempo discreto pueden ser utilizados para formar intervalos de tiempo discreto.

Spans: Los spans de tiempo discretos y continuos determinados, representan información completa sobre la duración del tiempo.

Span Discreto Determinado: Es la suma de distintas granularidades con coeficientes enteros.

Span Continuo Determinado: Es la suma de distintas granularidades con coeficientes reales.

La figura 1.12 busca sintetizar lo referente a las diferentes combinaciones que se pueden presentar.

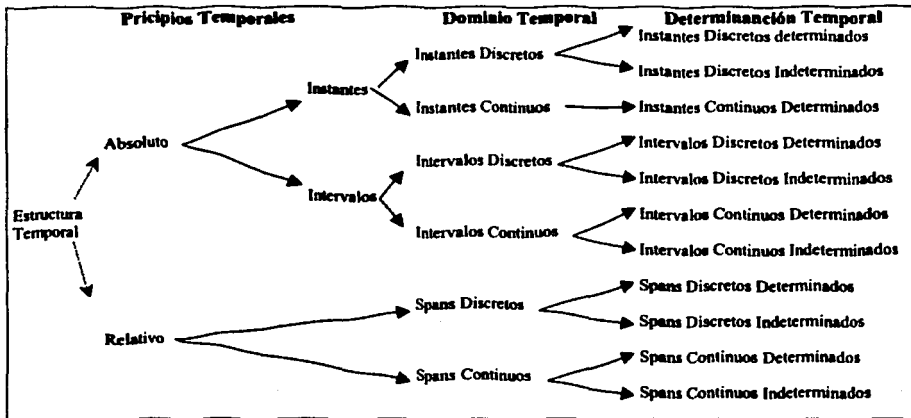


Figura 1.12 Espacio de Diseño de una estructura temporal. (Esta figura se encuentra en el libro [OPH98])

1.3.2 Representación Temporal

La representación temporal es una dimensión que busca plasmar en un esquema la forma en la cual el ser humano mide el tiempo, más conocida como calendarios [OPH98]. Debido a que los calendarios son la forma básica en la que el hombre mide el tiempo, éstos adquieren distintos formatos dependiendo de la actividad que se realice. Ejemplos de calendarios utilizados: el Gregoriano, el Fiscal, los Escolares, etc.

Es de esperarse que un modelo temporal maneje múltiples calendarios ya que las aplicaciones no manejan siempre el mismo calendario, así mismo, los calendarios manejan distintas granularidades. Una propiedad de los calendarios es la definición de funciones para efectuar conversiones entre distintas granularidades.

1.3.3 Orden Temporal

En lo que respecta al orden temporal puede ser clasificado en tres grandes grupos (figura 1.13):

- Orden Lineal: Aquí el tiempo fluye desde el pasado hasta el futuro de una forma continua y ordenada.
- Orden Sublineal: Aquí el tiempo fluye desde el pasado hasta el futuro de una forma continua pero no ordenada ya que pueden ocurrir eventos que inician sin que se acabe el anterior.
- Orden Bifurcado: Aquí el tiempo es lineal, partiendo del pasado hasta que se llega a un punto en que se encuentra una serie de futuros alternativos [OPH98].

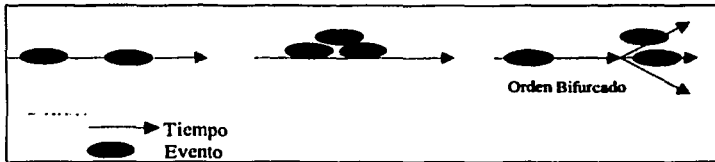


Figura 1.13 Representación del Orden Temporal

1.3.4 Historia Temporal

Se toman en consideración dos tipos de historias en las bases de datos que incorporan el tiempo. El tiempo válido es usado para capturar y el tiempo de transacción modela la actividad de actualización asociada con la relación [OPH98].

Tiempo válido: Es el tiempo en el cual una entidad es válida o efectiva y se encuentra representado de la siguiente manera:

$$D_{VT} = \{cv1, cv2, \dots, cvk\}$$

Tiempo de transacción: Representa el tiempo en el cual una transacción se sitúa en la base de datos. Los tiempos de transacción se encuentran representados de la siguiente manera:

$$D_{TT} = \{c'_1, c'_2, \dots, c'_k\}$$

Un tiempo válido (chronon) será denotado por:

c_v y es miembro de: D_{VT}

Un tiempo de transacción (chronon) será denotado por:

c_i y es miembro de: D_{TT}

1.4 Tipos de bases de datos temporales

Debido a la complejidad que existe para representar en una base de datos el tiempo, tal y como se maneja en el mundo real, se han desarrollado bases de datos que cumplen con algunas de las características de las BDT. Los modelos de bases de datos que se han implementado para soportar el tiempo emplean extractos diferentes de la teoría que sustenta a las BDT, para llevar a cabo la manipulación de los datos en el tiempo, es por ello que como primer punto se explicarán los subtipos de las BDT que existen.

1.4.1 Bases de datos Snapshot

Las bases de datos convencionales pueden ser vistas como bases de datos Snapshot, debido a que la información que se encuentra almacenada en éstas es considerada como actual, ya que este tipo de base de datos no posee características que le permitan asociar el valor del tiempo a los datos, por lo que la información se mueve a partir de un estado de tiempo que es constante "el ahora" [NAS95].

Este tipo de base de datos es considerado como el tradicional, donde los atributos temporales no son soportados salvo la posible excepción en la que se hace uso del tiempo definido por el usuario. Así mismo no permite explotar las capacidades de una base de datos que emplea una semántica temporal. Las bases de datos Snapshot se emplean tanto en los manejadores de bases de datos relacionales, como en los manejadores de bases de datos objeto-relacionales.

Debido a la necesidad que se ha presentado a últimas fechas para que la información sea lo más completa posible, se ha tomado la decisión de incorporar al elemento "tiempo" dentro de los DBMS. El tiempo es un elemento que siempre está incrementándose y de manera continua se encuentra en movimiento en una sola dirección (el futuro).

La percepción del tiempo es el reflejo de la memoria humana, es decir, el ser humano es capaz de memorizar eventos ocurridos en el pasado como hechos y experiencias, pero no es capaz de decir como serán las cosas en el futuro, sólo se pueden realizar predicciones en base a los sucesos ocurridos en el pasado. Es precisamente la posibilidad de realizar predicciones de eventos que ocurran en el futuro, lo que ha motivado a introducir al tiempo como un elemento de las bases de datos.

En el caso de las bases de datos Snapshot que no son capaces de incorporar al tiempo dentro de su estructura, se ha decidido implementar otros modelos. Esto no significa que tal modelo sea caduco y obsoleto ya que el manejo de la información con el elemento tiempo no es requerido para todos los datos que se encuentran en las bases de datos que manejan las compañías, instituciones, empresas, etc; un factor importante que debe ser considerado, es que las bases de datos Snapshot pueden convivir perfectamente con otros modelos que dentro de su estructura si incorporen el tiempo.

1.4.2 Bases de datos Históricas

Las bases de datos históricas son aquellas en las que su funcionamiento está basado en la manipulación que se efectúa del tiempo válido, dado que su labor fundamental es la de preservar la historia de los registros que se encuentran almacenadas dentro de la base de datos. Este tipo de bases de datos puede ser implementado dentro de un modelo relacional, así como, de un modelo objeto-relacional[NAS95].

Este tipo de BDT que incorporan al tiempo (tiempo válido) dentro de su estructura, permiten que de manera automática, los datos actuales se vuelvan históricos en un lapso de tiempo definido por el "database administrator" (DBA). El tiempo en el cual se efectuará el cambio de los datos actuales a datos históricos será determinado de acuerdo a las necesidades de la organización, no existe una regla a seguir ya que todo se efectúa en base a las necesidades que presenta cada organización de manera particular.

La información que se encuentra dentro de las bases de datos históricas puede ser utilizada como respaldo, o como histórico. Una característica importante que presentan las bases de datos históricas, es que son el punto de partida para la generación del llamado "Datawarehousing".

Aunque para este caso ya se ha incorporado al tiempo dentro de la estructura de las bases de datos históricas, no es suficiente para las necesidades que presentan los usuarios al momento de hacer uso de la información. Lo cual no significa que este modelo sea incompleto, únicamente quiere decir que abarca una pequeña parte de lo que realmente son las BDT.

Este tipo de BDT al igual que el modelo anterior, pueden interactuar con otros modelos que en su estructura incorporen o no al tiempo.

1.4.3 Bases de datos Rollback

Las bases de datos Rollback son bases de datos que utilizan el tiempo de transacción para identificar las versiones de los datos, permitiendo a los usuarios efectuar búsquedas de información en un punto específico del pasado, lo cual es posible mediante una marca de tiempo asociada a los datos, y a su vez puede ser empleada para la búsqueda de información de datos ordinarios. Esta marca de tiempo es un atributo extra de los datos, un punto importante en el que se debe hacer hincapié es el orden en que deben tener las marcas de tiempo, ya que debe poseer una seriación con un orden específico de las transacciones que se realizan[LOM92].

Los datos actuales son la última versión de los datos que se encuentran almacenados, es decir, los datos que poseen las marcas de tiempo más recientes. Los datos que dejan de ser actuales no necesariamente deben ser almacenados en el mismo sitio que los datos actuales, a éstos los llamaremos "datos históricos" y ya no pueden ser actualizados, por tal motivo pueden ser almacenados en diversos dispositivos, como los discos ópticos.

En este tipo de bases de datos las marcas de tiempo se llevan a cabo en el momento en el que se efectúa un "commit". Los cambios que efectúan los usuarios no son realizados mientras el usuario que está manipulando la información no ejecute la instrucción antes mencionada, una vez que la persona hace uso de la instrucción, todos los cambios quedan registrados de forma permanente en la base de datos con una marca de tiempo que identifica a esos datos de manera única. Cabe señalar que si el usuario no tiene permiso de realizar alguna operación en los datos, como modificar la información, la base de datos permanecerá como hasta antes de que el usuario modificara la misma[LOM92].

Un término empleado en estas bases de datos es el Time Split B-Tree (TSBTREE)[LOM92], que se utiliza para identificar un índice, que se maneja tanto en datos históricos como en datos actuales y sirve para hacer diferencia entre ambos tipos de datos al momento de hacer transacciones, además permite organizar los datos, de tal modo que se asegura que las versiones de los datos son acumuladas para efectuar una versión válida determinada de los datos usando únicamente los tiempos de inicio[LOM92].

Este tipo de BDT puede ser empleado en sistemas que funcionen bajo el modelo relacional, así como, en los sistemas objeto-relacional, también pueden funcionar con otros modelos que no incorporen en su estructura al tiempo o con aquellos que sí incorporan al tiempo.

1.4.4 Bases de datos Bitemporales

Las bases de datos Bitemporales como su nombre lo indica son aquellas que hacen uso de los dos tiempos que se utilizan en las BDT, es decir, el tiempo de transacción y el tiempo válido. Su estructura se encuentra plasmada en dos árboles, uno de ellos se emplea para indexar el tiempo válido, mientras que el otro se utiliza para el tiempo de transacción. Ambos árboles llegan a un punto en el que comparten sus hojas, de este modo se puede ahorrar espacio al momento de almacenar la información[NAS95].

Se ha reconocido que las BDT requieren de dos dimensiones de tiempo para soportar un sistema de administración de bases de datos, con esto nos referimos al tiempo válido y de transacción, en algunos casos se ha llegado a manejar una tercera dimensión que se refiere básicamente a un tiempo definido por el usuario[BJO97].

- **Tiempo de transacción:** Es aquel tiempo en el que la información fue almacenada en la base de datos, es decir, es el tiempo durante el cual un hecho del mundo real es verdadero en la base de datos.
- **Tiempo válido:** Es la variación de tiempo de modelado real, o bien, es el conjunto de intervalos de tiempo durante los que un suceso del mundo real es verdadero.
- **Tiempo definido por el usuario:** Éste es necesario para información temporal adicional que no es manejada en tiempo válido o tiempo de transacción y que es almacenado en la base de datos.

Se considera a este tipo de BDT como el modelo más completo, porque al hacer uso del tiempo en las dos modalidades, hace que los datos sean identificados de manera única dentro de la base de datos, permitiendo así un mayor control de la información que se encuentra almacenada dentro de la misma [BJO97].

Las BDT de este tipo pueden ser implementadas en DBMS relacionales, así como en las objeto-relacional. También funcionan al lado de los modelos que incorporan al elemento tiempo en su estructura al igual que con los modelos que no lo incorporan.

CAPITULO II

El elemento tiempo en las bases de datos

2.1 El tiempo en los manejadores de bases de datos

El tiempo es un elemento fundamental en las BDT, se introdujo dentro del modelo relacional y adoptó diversos modelos para hacer que funcionaran las bases mencionadas. A lo largo del presente capítulo se hará mención de: los puntos no favorables que ocasionan la poca viabilidad para el manejo de las BDT en un modelo relacional, se verán las bondades que presenta el modelo objeto relacional para la manipulación de las BDT y finalmente se verá como Oracle 8i uno de los DBMS objeto-relacional, emplea las series de tiempo para hacer posible la manipulación adecuada de este tipo de bases de datos.

Se empleará el DBMS Oracle 8i debido a que es uno de los manejadores más completos que existe en el mercado y en su versión 8.1.7.0.0 permite a los usuarios emplear el tiempo como componente de los datos en base a la teoría de las BDT. Otro factor muy importante para el empleo de este manejador es por la disponibilidad del mismo para efectuar pruebas.

2.1.1 El tiempo en el modelo relacional

Este modelo que se encuentra sustentado en la teoría desarrollada por E. F. Codd, donde una base de datos relacional permite la definición de estructuras, operaciones de almacenamiento y recuperación de datos permitiendo que se establezcan reglas para mantener la integridad de la información que se encuentra almacenada. En este modelo los objetos que se manejan se encuentran definidos bajo una estructura denominada "tabla". Las tablas son estructuras compuestas por registros (renglones) que se encuentran divididos en campos (columnas).

Debido a la estructura en la que se mueve el modelo relacional, la representación de las series de tiempo se vuelve difícil, ésto es ocasionado principalmente porque las series de tiempo crecen en forma longitudinal (Figura 2.1), en forma contraria al modelo relacional que crece en forma transversal (incrementa el número de registros) véase figura 2.2. A pesar de que en el modelo relacional las bases de datos presentan datos de tipo fecha, no cuentan con tipos de estructuras que tengan una noción de la variación de los datos en el tiempo, ésta es la razón por la cual se presenta una deficiencia en el manejo de la información en forma de series de tiempo[MAT98].

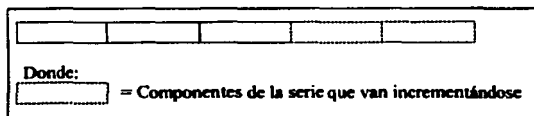


Figura 2.1 Representación del crecimiento de las series de tiempo

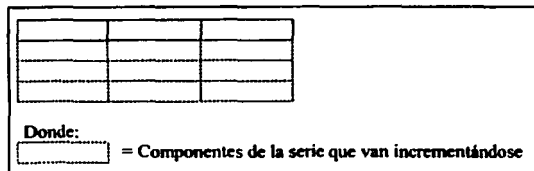


Figura 2.2 Representación del crecimiento del modelo relacional

El mecanismo empleado para el almacenamiento de las series de tiempo, es mediante el uso de archivos planos, los cuales contienen los registros de los datos empleando arreglos para la identificación de una serie. Ésto se vuelve eficaz ya que en el momento en el que se requiere agregar un dato a la serie, éste se agrega al final del arreglo y no mediante un nuevo registro.

Las soluciones propuestas se basan en agregar tiempo al modelo relacional, así como la extensión de los lenguajes de consulta. "El modelo convencional del Doctor Codd, es un modelo Snapshot" [BJO97], debido a que no toma en cuenta la variación del tiempo de la realidad y soporta únicamente un estado de la base de datos, es decir, el estado actual.

El modelo del Doctor Codd, define relaciones en un sentido matemático y teórico, definidos como productos cartesianos. El concepto más pragmático de la relación se encuentra definido por una relación R , que no es sino un conjunto finito de atributos ordenados, A_i .

$$R(A_1, A_2, A_3, \dots, A_n)$$

Definición (2.1) [BJO97]

Se definen mecanismos mediante los cuales se pretende tener un soporte del tiempo dentro del modelo relacional. El primero se refiere a la incorporación de tipos de datos como: *DATE* y *TIME*, el segundo se refiere a la extensión del modelo relacional mediante el manejo de marcas de tiempo, así como el manejo de una semántica temporal. En el primer caso, se encarga de administrar las marcas de tiempo que se encuentran en las tuplas mientras que el otro soporta el atributo de las marcas de tiempo. Ambos acercamientos pueden soportar tanto el tiempo válido como el tiempo de transacción[BJO97].

Hasta este momento se ha hecho mención de mecanismos relacionados con la extensión temporal de los datos y no la intención temporal. La intención temporal está relacionada con el tiempo de transacción donde el esquema actual es el sujeto de cambio, es decir, los esquemas previos e instancias definidas bajo ellos no se ven afectados, únicamente se ven afectados aquellos que son actuales.

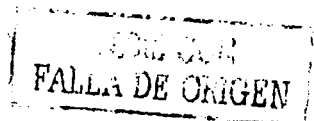
Por otro lado en el esquema del tiempo válido los cambios se ven reflejados tanto en el esquema actual como en el previo y ambos se ven alterados o cambiados. Un simple cambio puede alterar una o múltiples versiones de los esquemas. El beneficio del esquema de tiempo válido es que puede efectuar cambios tanto en esquemas previos como el actual, así como una mejora en la captura de los hechos que aparecen en la realidad.

2.1.1.1 Agregando tipos de datos

El agregar tipos de datos es la alternativa más sencilla que se presentó, ya que lo que se propuso fue extender los tipos de datos básicos que se tenían como: *DATE*, *TIME*, *DATETIME* e *INTERVAL* con algunas operaciones intrínsecas, tanto en SQL-92 y el SQL3 estándar. Otra de las alternativas propuestas para el manejo del tiempo es el SQL /Temporal que es parte del SQL3, éste incluye algunas propuestas para soportar tiempo válido y tiempo de transacción[BJO97].

Muchos de los manejadores de bases de datos relacionales que existen en el mercado soportan uno o más de los tipos de datos temporales simples, pero no soportan al tiempo en un sentido estricto a un punto tal como para decir que es un manejador de bases de datos temporales. El manejo de la temporalidad, la historia, la variación del tiempo y la dependencia del tiempo son manejadas por las aplicaciones, no por el manejador que no tiene construida una semántica para capturar estos aspectos de los datos.

Se han efectuado algunos intentos para incorporar algunas de las variaciones del tiempo dentro de los manejadores de bases de datos relacionales, sin redefinir todo el sistema y lo que se ha obtenido son versiones del sistema muy incompatibles.



2.1.1.2 Agregando una capa temporal

Uno de los primeros acercamientos a lo que se conoce como BDT se efectuó gracias al trabajo de Vassilakis y otras personas, su aportación fue el agregar una serie de capas adicionales que se denominaron "motor temporal", el cual se implementó por primera vez en el DBMS *Ingress*. El objetivo de este motor es desarrollar un soporte de transacción temporal utilizando las características de transacción del manejador *Ingress*: Mediante la extensión del kernel de *Ingress* con un tipo de dato denominado "dateinterval" [BJO97] y la realización de operaciones con el tipo de dato que está basado en un par del tipo de dato "date". El modelo temporal mencionado se encuentra definido por una tupla con marca de tiempo para un tiempo válido, y su semántica se encuentra manejada por el motor temporal.

Una marca de tiempo puede ser de cualquiera de los dos tipos "date" o "dateinterval" [BJO97]. También se generó una álgebra relacional de tiempo válido denominada *VT-AL*, que se encuentra definida para soportar una manipulación de datos eficiente y temporalmente consistente. Los operadores de ésta álgebra son: "fold", "unfold", "punion", "pexcept" y "normalize" [BJO97]. Los primeros dos son los principales, éstos se emplean para mapear relaciones entre intervalos y puntos basados en marcas de tiempo. Cuando "unfold" toma como entrada un intervalo con una relación que posee una marca de tiempo, éste regresa una relación, donde cada tupla de entrada es reemplazada por un conjunto de tuplas de valores equivalentes; cada tupla de este conjunto tiene un punto de la marca de tiempo idéntico al punto dentro del intervalo que posee la marca de tiempo de la tupla original. Mientras que "fold" es el operador inverso al operador antes mencionado. Las tuplas, con marcas de tiempo idénticas o equivalentes, con valores equivalentes son colapsadas dentro de una tupla. El operador "fold" se asegura que todos los intervalos con marcas de tiempo cubran todos los puntos del tiempo definidos por las marcas de tiempo de su relación operacional. Por otro lado "punion" y "pexcept" son redefiniciones temporales del SQL-89. El operador "punion" es un "unlon" ordinario pero el resultado se encuentra normalizado. El "pexcept" es también un "except" ordinario [BJO97].

La novedad que se presenta en este sistema es el soporte integral de las transacciones y la temporalidad de las bases de datos. El manejo de la capa temporal de transacción, utiliza una combinación de asignación de sesiones de usuario y sesiones de sistema, que hacen posible el manejo de las tablas temporales del sistema interno y de este modo se asegura a los objetos, al mismo tiempo la capa permite la recuperación de los mismos.

Un ejemplo adicional de lo que es la capa temporal es el trabajo efectuado por Böhlen y Jensen que se ve reflejado en el "ATSQL" y que es la extensión temporal del SQL-92. A diferencia del trabajo efectuado por Vassilakis el "ATSQL" fue implementado en el DBMS "Oracle" [BJO97].

Por su parte IBM también experimentó con el manejador de bases relacionales "DB2" para "AIX" versión 2 el tipo de dato "date" y un SQL recursivo para "DB2" que juegan un papel importante dentro de la representación temporal de los datos. El acercamiento se encuentra basado en tuplas con marcas de tiempo, tanto de un punto en el tiempo como un intervalo de tiempo, el primero de éstos se encuentra definido como un simple "date" siendo así el atributo de la relación, mientras que el segundo tiene atributos del tipo "date" en un punto de inicio y en un punto de término. Aquí definen dos operadores temporales importantes, uno es el operador "time-slice", que se emplea para acceder a los datos en un punto del tiempo o en un intervalo, éste toma una relación y un valor de tiempo y regresa un conjunto de tuplas válidas para el valor de tiempo especificado. Por otro lado el segundo operador es el operador "time-coalesce", que es utilizado para comprimir el número de tuplas de una relación, éste toma el valor-equivalente de todas las tuplas de la relación y regresa una relación de todas las tuplas de valor-equivalente sin que estén traslapadas ni contengan marcas de tiempo adyacentes.

Cada tupla de salida es el resultado de colapsar el correspondiente conjunto de valor-equivalente de las tuplas de entrada que están traslapadas o con marcas de tiempo adyacentes. Aquí, la marca de tiempo de cada tupla resultante es la unión de las marcas de tiempo de las tuplas de entrada, similar a lo que hace el operador "fold" presentado anteriormente [BJO97].

2.1.1.3 Tupla Timestamping (Registro con marca de tiempo)

La forma más común en la que se introdujo el tiempo al modelo relacional es mediante la tupla timestamping. Dentro de los últimos diez o quince años se han elaborado más de una docena de modelos relacionales temporales y álgebras que han incorporado las tuplas timestamping que pueden soportar tiempo válido, tiempo de transacción o ambas dimensiones de tiempo. Además pueden soportar marcas de tiempo con puntos en el tiempo o bien con intervalos en el tiempo, así como soportar únicamente nociones del pasado o tiempo actual y en algunos casos, nociones de un futuro inmediato donde las marcas de tiempo son definidas en términos de valores explícitos [BJO97].

A continuación se presentan las dos dimensiones o dominios para tiempo-válido D_v , y para tiempo-de-transacción D_t . Éstas se encuentran definidas de la siguiente manera:

Dimensión del tiempo de transacción
 $D_t = \{t_0, t_1, \dots, t_i, \dots, \text{ahora}\}$ Definición (2.2)

Dimensión del tiempo válido
 $D_v = \{t_0, t_1, \dots, t_i, \dots, \text{ahora}, \dots\} \cup \{\infty\}$ Definición (2.3)

Para ambas dimensiones y valores de tiempo, dados en instantes de tiempo o en intervalos de tiempo, se aplican las siguientes ecuaciones:

$t_k = t_0 + k$ y $t_k = t_{k-1} + 1, k \geq 1$ Definición (2.4)

Donde k es un punto específico (instante) en la línea del tiempo.

Las siguientes definiciones muestran los diferentes esquemas bi-temporales, R_i donde la información temporal tiene dominios definidos por las dimensiones de tiempo antes mencionadas.

$R_1 = (A_1, \dots, A_n, T_v, T_t)$ Definición (2.5)

$R_2 = (A_1, \dots, A_n, T_v, T_{tr}, T_t)$ Definición (2.6)

$R_3 = (A_1, \dots, A_n, T_v, T_{tr}, T_t, T_u)$ Definición (2.7)

donde cada A_i es un atributo regular de la relación y $(T_v, T_{tr}, T_u$ y $T_u)$:

T_v = Tiempo válido de inicio.

T_{tr} = Tiempo válido de término.

T_u = Tiempo de transacción de inicio.

T_{tr} = Tiempo de Transacción de término.

La definición del Doctor Codd se extendió de manera informal a $R = (A_1, \dots, A_n | T)$ donde T se encuentra caracterizada por una combinación de las T 's que se vieron anteriormente, la forma en la que entra la información es mediante parejas representadas de la siguiente manera: (T_x, d_x) ó $((T_{xv}, T_{xt}), [d_{xv}, d_{xt}])$, donde el subíndice "x" variará dependiendo de lo que se hable como tiempo válido "v" o tiempo de transacción "t" [BJO97].

2.1.1.4 Modelos basados en eventos

En la definición 2.5 se habla de un evento basado en una relación bi-temporal, en este caso no se toma en cuenta la duración de cuando el objeto fue válido en la realidad, únicamente se toman en cuenta los estados en el que el tiempo $t_i \in D_i$, posiblemente ocurrió algún evento, entonces el hecho se vuelve válido y fue registrado en el tiempo $t_j \in D_j$. Una aplicación puede realizar interpretaciones si se sabe que la duración del evento es siempre el tiempo que transcurre hasta un nuevo cambio que presente el objeto [BJO97].

Modelo Ariav

Éste es un modelo relacional bi-temporal donde las marcas de tiempo se basan en puntos en el tiempo obsérvese (Tabla 2.1) [BJO97].

Nombre	Departamento	Puesto	Tiempo VÁLido	Tiempo de Transacción
Pedro	Ventas	Asistente	ENE1990	FEB1990
Pedro	Ventas	Asistente	MAR1991	DIC1990
Pedro	Ventas	Coordinador	AGO1991	DIC1991
Laura	Ventas	Jefe	ENE1991	ENE1990
Laura	Mercadotecnia	Administrador	DIC1991	FEB1991

TABLA 2.1 Relación bi-temporal de empleados mediante el modelo Ariav

2.1.1.5 Modelos basados en eventos de intervalos

En el caso de la definición 2.6 se puede ver una mezcla de eventos de intervalos basados en el esquema relacional bi-temporal que soporta la semántica actual de la duración de algunos hechos en la realidad. En suma la semántica puede decir algo acerca del periodo de vida (lifespan) de un objeto, es decir, cuándo el objeto fue definido, cuándo desapareció eventualmente y cuándo fue redefinido. El soporte del tiempo de transacción es dado por puntos en el tiempo. Todos los datos son almacenados con su marca de tiempo. La base de datos no captura la semántica del periodo de tiempo cuando el hecho fue actual en la misma, únicamente cuando se volvió actual. Obsérvese (Tabla 2.2) [BJO97].

Modelo Jensen

Nombre	Departamento	Puesto	Tiempo Válido		Tiempo de transacción	Operación
Pedro	Ventas	Asistente	ENE90	SEP90	FEB90	I
Pedro	Ventas	Asistente	ENE90	SEP90	DIC90	B
Pedro	Ventas	Asistente	MAR91	NOV91	DIC90	I
Pedro	Ventas	Asistente	MAR91	NOV91	DIC91	B
Pedro	Ventas	Asistente	MAR91	MAY91	DIC91	I
Pedro	Ventas	Coordinador	AGO91	uc	DIC91	I
Laura	Ventas	Jefe	ENE90	FEB91	ENE90	I
Laura	Mercadotecnia	Administrador	DIC90	DIC93	FEB91	I

TABLA 2.2 Relación bi-temporal de empleados mediante el modelo Jensen

La columna de operación se refiere a la inserción o borrado de registros indicado por una I (inserción) y una B (borrado). Las modificaciones de un objeto son manejadas por un par de peticiones de inserción y borrado y se efectúa de la siguiente manera: primero, la petición de borrado se representa por un nuevo registro, que tiene los mismos atributos y un tiempo válido previo a la petición de inserción del objeto y el valor de la operación es B; como segundo paso, la petición de borrado es seguida por una inserción con el tiempo de transacción idéntico al de la petición de borrado, en este caso el valor de la operación es I.

2.1.1.6 Modelos basados en intervalos

La definición (2.7) muestra un intervalo que se basa en una relación bi-temporal. Ambos intervalos denotan el tiempo de inicio y el tiempo de término. El tiempo de transacción registra el tiempo en el cual la tupla es actual en la base de datos. Hay diferentes interpretaciones de la marca del intervalo de transacción ya que se tiene el manejo de variables. La información de una marca de tiempo $([v_1, v_2], [t_1, t_2])$ del tiempo de transacción y el tiempo válido se define de la siguiente manera:

- Si t_e es una variable (uc) entonces la tupla es actual en la base de datos.
- Si $t_e = t \in D$, entonces la tupla es borrada de manera lógica.
- Si $t_e < v_1$ entonces la tupla es insertada proactivamente.
- Si $t_e > v_2$ entonces la tupla es insertada retroactivamente.
- Si $t_e = v_1$ entonces tenemos una inserción sincronizada.

Modelo Ben-Zvi

El modelo Ben-Zvi fue el primero en introducir diferentes tiempos, denominados tiempo efectivo y tiempo de registro, los cuales son análogos a lo que se ha manejado hasta aquí como tiempo válido y tiempo de transacción respectivamente, obsérvese (Tabla2.3) [BJO97]. Un esquema de relación R se define de la siguiente forma:

$$R_3 = (A_1, \dots, A_n, T_{e1}, T_{e2}, T_{r1}, T_{r2}, T_d) \quad (2.8)$$

donde las A_i 's son los casos previos

T_{e1} y T_{e2} Son el primer par de marcas de tiempo que definen el inicio de tiempo efectivo así como el inicio de tiempo de registro, respectivamente

T_{r1} y T_{r2} Este par de marcas par define el tiempo final efectivo y el tiempo final de registro. Ambos casos pueden estar representados por "-" en caso en el que se desconozca el tiempo

T_d Es utilizado para diferenciar entre cambios y errores. Si T_d llega a tener una marca de tiempo, la información capturada por la tupla es desconocida o bien es un error y después será borrado lógicamente

Nombre	Departamento	Puesto	T_{e1}	T_{e2}	T_{r1}	T_{r2}	T_d
Pedro	Ventas	Ayudante	ENE90	FEB90	SEP90	FEB90	"-"
Pedro	Ventas	Ayudante	MAR91	DEC90	NOV91	DIC90	DEC91
Pedro	Ventas	Ayudante	MAR91	DEC91	MAY91	DIC91	"-"
Pedro	Ventas	Coordinador	AGO91	DEC91	"-"	"-"	"-"
Laura	Ventas	Jefe	ENE90	ENE90	FEB91	ENE90	"-"
Laura	Mercadotecnia	Administrador	DIC90	FEB91	DIC93	FEB91	"-"

TABLA 2.3 Relación bi-temporal de Empleados mediante el modelo Ben-Zvi

En conclusión, el modelo relacional con datos temporales en los sistemas relacionales no presenta la capacidad para crear nuevos tipos de datos, que es lo que mantiene las variaciones de tiempo de los datos. Ésto se ve reflejado en las consultas que se llevan a cabo dentro de la base de datos, porque si las operaciones no pueden realizarse mediante el uso de una selección, una proyección, un join o un producto cartesiano; tales consultas no pueden realizarse en el sistema relacional.

Como se pudo ver, las series de tiempo se pueden implementar en el modelo relacional; pero hay que fijarse que hay puntos en los cuales la información se vuelve redundante y ésto ocasiona un bajo rendimiento, así como inconsistencia en los datos, lo que se puede observar en las tablas resultantes de los diferentes modelos.

Una representación intuitiva de los objetos en el tiempo, sería dejar que los eventos fueran representados por puntos en el tiempo y los hechos por periodos de tiempo o intervalos y procesarlos con funciones de tiempo.

Entonces, se puede decir que: "Un evento toma lo que pasa en un instante de tiempo, mientras que un hecho es tomado como una verdad acerca de algunos aspectos estáticos de la realidad".

2.1.2 Las series de tiempo en el modelo objeto relacional

En el momento en que se piensa en las bases de datos, el modelo objeto relacional es el desarrollado más recientemente. A este modelo se le conoce como "La gran nueva ola". Michael Stonebraker define las bases de datos objeto relacional como aquellas que soportan las bondades del SQL/3 y agrega "*Son de naturaleza relacional porque soportan SQL y tienen una naturaleza objeto relacional porque soportan datos complejos*" [DAT01]

Este modelo cuenta con las características de un manejador de bases de datos, pero además se puede desenvolver en un ambiente orientado a objetos (cabe señalar que no significa que haya dejado de emplear el modelo relacional). Con la llegada de la tecnología orientada a objetos algunos de los problemas que aparecían con el modelo relacional encontraron solución, ésto se debe a que los nuevos sistemas orientados a objetos pueden manejar nuevos tipos de datos que son directamente utilizados por el sistema, ahora es posible efectuar la definición de nuevos tipos de colecciones como lo que son las series de tiempo, que modelan la variación del tiempo en los datos.

Al contar con las características de un sistema orientado a objetos, los datos pueden ser almacenados como objetos permitiendo así su acceso a través de métodos definidos por sus clases. Otra característica que presentan es la herencia, mediante ésta se puede controlar la relación entre objetos similares y el acceso a la información se vuelve más eficiente, por lo que no se necesitan las ligas que se emplean en el modelo relacional, en otras palabras, un objeto puede ser recuperado directamente sin la necesidad de una búsqueda, sólo se requiere de una identificación.

En este tipo de bases de datos se crea una clase denominada series de tiempo y a través de un conjunto de métodos efectúan el almacenamiento y la recuperación de la información. Los sistemas en lugar de almacenar las series en forma tabular (que como se observó anteriormente provoca redundancia), las almacenan en un vector de información.

Al manejar bases de datos dentro del modelo objeto relacional se obtienen las bondades de los objetos, tales como:

- Encapsulación: Un objeto que es leído desde la base de datos tiene el mismo código y datos que tuvo cuando fue almacenado. Los objetos pueden tener partes privadas y partes protegidas, así el objeto puede ser manejado de una manera más confiable.
- Herencia: Las clases que son derivadas de otras clases pueden ser almacenadas con una sola operación. La base de datos debe conocer la herencia de la clase y manejar el almacenamiento del objeto de acuerdo a ésto.

- Polimorfismo:** Cuando un objeto se lee de la base de datos, tiene todo el código y los datos de los miembros que tenía cuando se creó originalmente.
- Identidad de Objeto:** Los sistemas de bases de datos orientadas a objetos, integran la identidad del objeto en la base de datos con la identidad de los objetos en memoria. Si se almacena un objeto, se sabe si corresponde a un objeto en la base de datos y cuando se regresa un objeto se sabe si el objeto ha sido cargado en memoria. En este caso no existe la necesidad de que el programador mantenga la relación entre los objetos de la base de datos y los objetos en memoria.
- Referencias entre objetos:** Los sistemas de bases de datos orientados a objetos pueden resolver automáticamente las direcciones de los apuntadores en los objetos de los programas y representarlos en la base de datos

Como se pudo observar, lo que hace de este modelo lo más indicado para el manejo de las series de tiempo, es que se pueden crear nuevos tipos de datos, así como las bondades que se presentan al trabajar con objetos, evitando el problema que se suscitaba en el modelo relacional en virtud de que en un mismo registro se generaban múltiples tuplas, en las que lo único que presentaba cambios eran las fechas, presentando redundancia en los datos. En este nuevo modelo se evita eso, porque únicamente se hace referencia a la fecha en cada tupla y no se tiene que crear la tupla cada vez que se modifique la fecha.

La desventaja que presenta es el alto costo que presentan los RDBMS que hacen uso del modelo objeto relacional, porque al ser lo último en tecnología sus costos son muy elevados y sólo los grandes corporativos pueden tener acceso a éstos.

2.2 Series de tiempo en un manejador de bases de datos (Oracle 8i)

Una vez que se ha hablado de lo que son las series de tiempo y de las distintas formas en que se pueden encontrar cuando se observa un proceso determinado; ahora hablaremos de cómo maneja las series de tiempo el manejador de base de datos Oracle 8i. En versiones anteriores del manejador Oracle, la parte que corresponde a series de tiempo se vendía como un programa adicional conocido como "Oracle 8 Time Series Cartridge".

Para este manejador de base de datos una serie de tiempo es un conjunto de entradas de datos con marcas de tiempo. Las cuales permiten a los datos estar asociados en un periodo de tiempo determinado. Una serie de tiempo puede ser regular o irregular, esto depende de los calendarios, es decir, si las series se encuentran asociadas a un calendario o no. La información que se verá a continuación fue tomada de "Oracle 8i Time Series" que es la documentación que provee la compañía "Oracle" para el uso del software.

2.2.1 Calendario

Un calendario en el manejador Oracle 8i es la forma en donde se define cómo va a ser representado el tiempo, es decir, la forma en que el ser humano maneja el tiempo. "Un calendario puede ser asociado con una serie de tiempo, sin embargo, un calendario no necesita ser asociado con una serie de tiempo a menos que se requiera realizar alguna de las siguientes funciones" [ORA99]:

Uso de funciones de series de tiempo de Oracle 8i para validar datos de series de tiempo.

Uso de funciones de retraso.

Uso de funciones de balanceo de tiempo.

Se pueden emplear múltiples calendarios como: el Gregoriano que usamos hoy en día, un calendario Fiscal, un calendario escolar, etc. El uso del calendario depende de la actividad que se esté realizando.

En el caso que se manejen múltiples series de tiempo, éstas pueden ser asociadas mediante un mismo calendario (*shared calendar*)

2.2.1.1 Componentes de un calendario

- **Frecuencia:**

La frecuencia especifica la granularidad de la representación del calendario. Los gránulos son las unidades básicas de representación del tiempo y pueden ser de alguno de los siguientes tipos: segundos, minutos, horas, días, semanas, meses, años[ORA99].

El manejador Oracle 8i identifica a las frecuencias asociándoles un número entero, esto lo hace para el manejo de funciones. A continuación se muestra una tabla 2.4 que presenta los códigos y las frecuencias que maneja Oracle.

Frecuencia	Código Entero
Segundo	1
Minuto	2
Hora	3
Día	4
Semana	5
Mes	6
Cuatrimestre	7
Año	8
10-Días	10
Quincena	16
Semestre	18

TAHIA 2.4 Frecuencias Manejadas

Cada frecuencia tiene asociada una precisión, ya que el manejador requiere que las marcas de tiempo que se hagan sean de la precisión de la frecuencia del calendario a la que hacen referencia, para conservar la consistencia de la información, debido a que una marca de tiempo que es inconsistente con respecto a la frecuencia es imprecisa. A continuación muestran las convenciones de precisión obsérvese (Tabla 2.5) [ORA99].

Frecuencia	Convención de precisión	Ejemplo
Segundo	MM-DD-YYYY HH24:MI:SS	09-09-1997 09:09:09
Minuto	MM-DD-YYYY HH24:MI:00	09-09-1997 09:09:00
Hora	MM-DD-YYYY HH24:00:00	09-09-1997 09:00:00
Día	MM-DD-YYYY 00:00:00 (media noche)	09-09-1997 00:00:00
Semana	MM-DD-YYYY 00:00:00 (media noche del jueves anterior)	09-04-1997 00:00:00
Mes	MM-01-YYYY 00:00:00 (media noche del primer día del mes)	09-01-1997 00:00:00
Cuatrimestre	MM-01-YYYY 00:00:00 (media noche del primer día del cuatrimestre)	07-01-1997 00:00:00
Año	01-01-YYYY 00:00:00 (media noche del primer día del año)	01-01-1997 00:00:00
10-Días	MM-DD-YYYY 00:00:00 (media noche de: 1°, 11° o 21° día del mes)	09-01-1997 00:00:00
Quincena	MM-DD-YYYY 00:00:00 (media noche de: 1° y 15° día del mes)	09-01-1997 00:00:00
Semestre	MM-01-YYYY 00:00:00 (media noche del primer día de la mitad del año)	07-01-1997 00:00:00

TABLA 2.5 Precisión de las diferentes frecuencias

- **Patrón**

El patrón especifica un periodo repetitivo de frecuencias, así como, una fecha de anclaje que identifica a una marca de tiempo válida para el primer elemento en el patrón[ORA99].

- **Excepciones**

Las excepciones son marcas de tiempo que no forman parte del patrón del calendario, sin embargo, son de gran trascendencia para la definición del calendario. Existen dos tipos de excepciones: excepción desactivada (*off-exception*) y excepción activada (*on-exception*) [ORA99].

- **Off exception:** Es la marca de tiempo que será excluida del calendario y se representa mediante un (1).
- **On-exception:** Es la marca de tiempo que será incluida en el calendario y se representa mediante un (0).

2.2.2 Tipos de series que utiliza el manejador Oracle 8i

El manejador de base de datos Oracle 8i presenta dos variantes en el manejo de las series de tiempo y se describe a continuación.

- **Series regulares**

Una serie de tiempo regular es aquella que se encuentra asociada a un calendario. En este caso los datos llegan a los intervalos de una manera predecible. En términos matemáticos vistos con anterioridad, corresponden a las series de tiempo estacionales, que ya fueron explicadas[ORA99].

- **Series irregulares**

Una serie de tiempo irregular es aquella que no tiene asociado un calendario. Para este tipo de serie las marcas de tiempo no siguen un patrón determinado, en consecuencia los datos llegan de una forma impredecible, es decir, habrá ocasiones en que los datos lleguen en grandes cantidades a la base de datos o bien ocasiones en que no llegue dato alguno. Visto de forma matemática correspondería a las series de tiempo no estacionales, explicadas con anterioridad[ORA99].

2.2.3 La arquitectura que presentan las series de tiempo en Oracle 8i

Para el manejo de las series de tiempo el manejador Oracle usa una arquitectura con los siguientes elementos[ORA99]:

- **Almacenamiento detallado de los datos de las series de tiempo**

Se almacenan tres tipos diferentes de información: los detalles de la información de las series de tiempo, los calendarios y las series de tiempo *metadata*. En esta sección los datos se almacenan en el orden que establecen las marcas de tiempo, y debido a que muchas de las funciones analíticas al momento de efectuar alguna operación requieren de un acceso a los datos conforme a las marcas de tiempo, en el caso en el cual las marcas de tiempo no se encuentren en orden, las funciones desempeñan un ordenamiento interno antes de procesar las marcas de tiempo[ORA99].

- **Interfaces para las series de tiempo y funciones de balanceo del tiempo.**

Estas interfaces y funciones almacenan los detalles de los datos como datos relacionales, así mismo, se emplea una marca de tiempo por renglón y los calendarios se encuentran almacenados en tablas objeto. Existen dos instancias básicas para estas interfaces y funciones que son[ORA99]:

Instancia basada en Interface: En esta instancia, el primer parámetro de entrada a una función de series de tiempo es una instancia de una serie de tiempo.

Instancia basada en referencia: En este caso, el primer parámetro de entrada a una función de series de tiempo es una referencia a la serie de tiempo. Aquí la interface requiere que se introduzca suficiente información descriptiva para hacer que la función ejecute un SQL dinámico, para que, de este modo, se obtenga una instancia de series de tiempo.

En este punto se pone especial atención en lo que se refiere a la consistencia, de otro modo al momento de emplear las funciones se generan excepciones o los registros que se recuperan de una consulta resultan incorrectos.

• Funciones de calendarios

El manejador facilita la tarea de los usuarios al ofrecerles funciones para la búsqueda o modificación de lo que se encuentra en los calendarios. Dichas funciones se encuentran divididas en dos categorías[ORA99]:

Funciones para los Usuarios Finales: Permite a los desarrolladores de aplicaciones el manejo de las principales funciones de calendario, a continuación se mencionan algunas de ellas:

- **Second:** Crea un calendario con una frecuencia de segundos
- **Minute:** Crea un calendario con una frecuencia de minutos
- **Hour:** Crea un calendario con una frecuencia de horas
- **Day:** Crea un calendario con una frecuencia de días
- **Week:** Crea un calendario con una frecuencia de semanas
- **Ten_Day:** Crea un calendario con una frecuencia de diez días
- **Semi_Monthly:** Crea un calendario con una frecuencia de quincena
- **Month:** Crea un calendario con una frecuencia de meses
- **Quarter:** Crea un calendario con una frecuencia de cuatrimestres
- **Semi_anual:** Crea un calendario con una frecuencia de semestres
- **Year:** Crea un calendario con una frecuencia de años
- **IntersectCals:** Intercepta dos calendarios.
- **UnionCals:** Recupera la Unión de dos calendarios
- **IsValidCal:** Recupera un 1 si un calendario es válido y 0 si un calendario no es válido.
- **ValidateCal:** Valida un calendario. Repara errores hasta donde es posible.

Funciones para el Desarrollador de Productos: Permiten a los desarrolladores modificar o sustituir las capacidades de Oracle 8i mediante la creación de valores agregados.

Funciones de Calendario

- **CombineCals:** Combina dos calendarios, los patrones deben ser idénticos.

Funciones de Excepciones

- **NumOffExceptions:** Recupera el número de excepciones off-exceptions entre dos fechas.
- **NumOnExceptions:** Recupera el número de on-exceptions entre dos fechas.

Funciones de Fecha e Índice

- **IsValidDate:** Determina si el valor dado es válido
- **GetIntervalStart:** Recupera el inicio del intervalo que incluye la marca de tiempo de entrada.

- **Funciones de series de tiempo**

En lo que respecta a estas funciones bajo cualquier tipo de serie, se encuentran clasificadas en tres grupos que son[ORA99]:

Funciones de Extracción:	Estas funciones recuperan uno o más registros de series de tiempo.
Funciones de Recuperación y de Buen Estado:	Éstas únicamente pueden recuperar un registro a la vez.

- **Funciones de balanceo**

Oracle 8i proporciona funciones para balancear la información de las series de tiempo ya sea mediante funciones *scaleup* o bien *scaledown*[ORA99].

- **Funciones scaleup:** También son conocidas como funciones rollup y producen un resumen partiendo de información de granularidad fina.
- **Funciones scaledown:** También conocidas como funciones de distribución. Generan información de granularidad fina a partir de información de granularidad gruesa.

- **Herramientas y procedimientos administrativos**

El manejador proporciona una serie de procedimientos que simplifican la creación de objetos esquema de series de tiempo, a continuación se mencionan algunos de los procedimientos que emplea Oracle 8i[ORA99].

- **Add_Existing_Column**
- **Add_Integer_Column**creation specification
- **Add_Number_Column**
- **Add_Varchar2_Column**
- **Begin_Create_TS_Group**
- **Cancel_Create_TS_Group**
- **Close_Log**

Con lo visto hasta esta sección, se puede decir que en una serie de tiempo cada transacción que se lleva a cabo en la base de datos tiene una marca de tiempo, esta marca identifica a dicha transacción de manera única.

El uso que se les da a las series de tiempo es para procesar datos históricos derivados de transacciones financieras, manipulación de inventarios, etc.

El empleo de las series permite desarrollar operaciones tan simples o complejas como:

- Encontrar el precio más bajo, más alto, o el precio al inicio o final que tengan un patrón de un día específico.
- Calcular volúmenes mensuales que tengan un patrón de un año en específico.
- Derivar el patrón de movimiento promedio en 30 días de un año.

El manejador Oracle 8i nos permite seguir utilizando las tablas que se encuentran en la base de datos, y la creación de calendarios. El manejo de las series de tiempo por parte del manejador permite realizar consultas de la información que se encuentra en la base de datos mediante la llave primaria (la que incluye la marca de tiempo). Las búsquedas a las que nos referimos pueden ser del siguiente tipo:

- La búsqueda de información donde el patrón está conformado por un valor determinado (el valor de una columna), así como una fecha asociada.
- La búsqueda de un elemento que tenga un patrón promedio relacionado con una columna y también acompañado de una fecha.

Permite a los usuarios el manejo de valores que se encuentren dentro de un rango determinado (marcas de tiempo), es decir, un límite inferior y uno superior.

2.2.4 Reglas para la consistencia en las series de tiempo

- Todas las marcas de tiempo deberán ser almacenadas en una secuencia ascendente.
- No deben existir marcas de tiempo duplicadas.
- Todas las marcas de tiempo deben coincidir con la precisión del calendario.
- No deben existir marcas de tiempo más allá de los límites del calendario.
- Todas las marcas de tiempo deben seguir las especificaciones del patrón dado, salvo aquellas que se encuentran dentro de las excepciones (on-exception, off-exception).
- Los datos de las series de tiempo son continuos, es decir, entre las marcas de tiempo que se encuentran en los límites de inicio y fin las marcas existentes deben ser válidas para el calendario dado.
- Al no tener precaución en el manejo de estas reglas de consistencia, ya sea por accidente o a propósito, se puede llegar a destruir la integridad de la información de las series de tiempo. [ORA99].

CAPITULO III

El papel de las bases de datos temporales en la toma de decisiones

El uso de las BDT permite que los datos que se encuentran almacenados en tablas dentro de la base de datos posean una marca de tiempo que los hace únicos, esta marca de tiempo permite que el dato quede ubicado en un determinado lapso de tiempo, lo cual a la postre hará posible un mejor análisis de la información para la toma de decisiones

Las BDT permiten que se lleve a cabo el diseño de un gran número de aplicaciones, donde, se requiere el manejo de estados pasados y futuros de las bases de datos para la toma de decisiones en diversas áreas, como finanzas, seguros, medicina, etc., por lo anterior podemos decir que es difícil encontrar aplicaciones que no involucren la variación de los datos en el tiempo.

La importancia de las BDT se puede ver reflejada en lo que definió Abel Matus Castillejos para obtener su doctorado en ciencias de la computación en la UNAM en 1998 *"El mundo es multidimensional y el tiempo es una de esas dimensiones. Distintos puntos en el tiempo seleccionan diferentes estados del mundo. Una Base de Datos que se ocupa de datos que varían con el tiempo debe hacer lo mismo"*[MAT98].

Un punto muy importante que debe quedar claro es que no importando un gran avance tecnológico, la implementación de nuevos procesos en las bases de datos, o el uso de nuevas aplicaciones que nos brinden los medios más avanzados para la toma de decisiones no servirá de nada si la información que se posee es errónea, lo cual quiere decir que la información que se tiene debe cubrir ciertos requisitos.

3.1 Concepto de información

La información definida desde un punto de vista computacional es el conjunto de datos formalizados que la máquina recibe y con los cuales realiza sus operaciones. Para la toma de decisiones se requiere que la información sea precisa, oportuna, actualizada, completa, no repetida, coherente y adaptada a las necesidades específicas de cada usuario y cada circunstancia.

En ocasiones la necesidad de información es un fin en sí mismo, por lo general responde a una necesidad por conocer el entorno que nos rodea o bien el ambiente en el cual nos desenvolvemos. A medida que se toma conciencia del valor que posee la información en sí misma, se manifiesta la necesidad de impulsar la libre circulación de la información.

El objetivo de la información es dar a conocer los elementos que caracterizan algún evento que ha ocurrido o bien que está ocurriendo al momento de realizar una consulta de los datos.

3.1.1 Cualidades de la información.

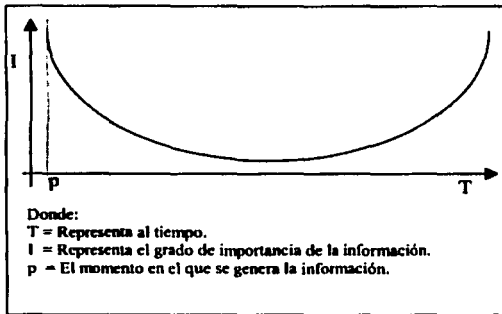
Si la información pierde sus cualidades no puede cumplir su objetivo, por lo tanto ésta se vuelve más nociva que beneficiosa para sus destinatarios. Para evitarlo se exigen un conjunto de cualidades para mantener el valor comunicativo, ya que por el simple hecho de informar se está aportando un conocimiento.

Las cualidades que debe poseer la información y que hacen de ella un recurso fundamental de las organizaciones y de los individuos son:

- Precisión:** Es el porcentaje de información correcta sobre la información total que se posee ya sea un archivo, una base de datos, etc. En este punto es importante aclarar que la información resultante al ejecutar una transacción, va a ser tan precisa como exactos hayan sido los datos de entrada.
- Oportunidad:** Tiempo que ha pasado desde el momento en que se produjo el suceso que generó el dato hasta el momento en que la información se pone a disposición del usuario. En ocasiones se mide en función del instante en el que el dato debería estar disponible.
- Plenitud:** Este punto se refiere al cómo la información debe ser completa para así cumplir su objetivo que es el brindar un conocimiento.
- Significado:** La información debe contener el máximo contenido semántico posible, ya que sin éste no sería verdadera la información. Lo que nos lleva a que debe ser entendible e interesante, lo cual se traduce en no proporcionar a los usuarios grandes masas de información que por su volumen no pueden ser asimiladas.
- Integridad:** Este punto se refiere a la coherencia de la información consigo misma, lo que permitirá obtener resultados concordantes.

3.1.2 Importancia de la información.

La información presenta diferentes grados de importancia conforme transcurre el tiempo, como se puede mostrar en la gráfica 3.1.



Gráfica 3.1 Representación de la importancia de la información

Como se puede observar en la gráfica 3.1, la información tiene su grado máximo en el momento en el que se genera y conforme transcurre el tiempo su valor va disminuyendo; pero después de un periodo de tiempo no determinado, en el que llega a su punto mínimo, esta misma información de manera gradual va recuperando importancia hasta convertirse en un elemento valioso que permitirá la toma de decisiones, esto se debe a que se conocen las consecuencias que se generaron a partir de la información.

3.2 Teoría de decisiones

Una vez que se han establecido ciertas características que debe poseer la información a ser utilizada en la toma de decisiones podemos empezar a hablar de la teoría de decisiones, la cual se encuentra dividida en tres clasificaciones básicas que dependen del grado de certidumbre de las posibles soluciones [REN96].

Toma de decisiones bajo certidumbre:	En este caso se sabe con certeza las consecuencias que se generarán al momento de tomar una decisión.
Toma de decisiones bajo riesgo:	Para este tipo se conoce la probabilidad de las consecuencias que se van a originar por la toma de una determinada solución al problema planteado.
Toma de decisiones bajo incertidumbre:	Para esta clasificación se desconoce la probabilidad de los posibles resultados de los eventos futuros que no se encuentran bajo el control de la persona que toma la decisión.

3.2.1 Criterios para la toma de decisiones

El proceso que se debe llevar a cabo para efectuar la toma de una decisión debe cubrir los criterios que se mencionan a continuación [REN96]:

Definición del problema y los elementos que intervienen en el mismo:	Este punto busca que se fije de forma exacta y precisa el problema que se está presentando. Cabe señalar que éste es el punto más importante y a su vez el más difícil de identificar.
Creación de criterios de decisión:	Las personas encargadas de encontrar la solución del problema deben establecer objetivos específicos y medibles.
Proponer un modelo entre las metas y las variables:	En este punto lo que se busca es representar el problema de estudio en un modelo. Dichos modelos poseen una o más variables dependiendo del grado de complejidad que se tenga.
Identificación y evaluación de alternativas:	Aquí lo que se pretende es encontrar tantas soluciones como sean posibles y de este modo tener una diversidad de opciones que permitan encontrar la solución más adecuada.
Elección de la mejor alternativa :	Ésta es la solución que cubrió de una manera más satisfactoria las metas que se establecieron anteriormente.
Ejecución de la alternativa:	En este punto se efectúa la implementación de la mejor solución, asignando actividades a diferentes áreas, así como una guía para el desarrollo de las mismas.

Los criterios mencionados para la toma de decisiones de cualquier problema se pueden resumir en tres pasos que son los que a continuación se mencionan.

Primero:	Deben existir varias alternativas mediante las cuales se puede resolver el problema en cuestión, de otro modo no habría problema en la decisión.
Segundo:	Debe haber metas u objetivos que se pretenden alcanzar, como pueden ser las ganancias de una empresa o el desarrollo de la misma.
Tercero:	Debe existir un proceso mediante el cual se efectúe un análisis donde las posibles alternativas son evaluadas en función de las metas.

3.3 Como actúan las bases de datos temporales en la toma de decisiones

Las BDT proporcionan la información necesaria para la definición del problema, dado que los datos que se encuentran almacenados de una forma más estructurada, también poseen un alto grado de precisión, al mismo tiempo proveen la información suficiente para la elaboración de alternativas que darán solución al problema.

La información que se obtiene de las BDT se encuentra estructurada de una manera diferente, por lo que su funcionamiento se encuentra considerando un elemento muy importante con el que las personas están familiarizadas, por que lo manejan de manera cotidiana sin percatarse de la importancia que tiene en nuestras vidas, éste es el factor tiempo. El tiempo, que para nosotros los seres humanos es tan natural y tan sencillo de comprender, es un elemento que a costado mucho trabajo plasmar en un sistema; debido a que las máquinas no poseen la noción de lo que es el tiempo lo que significa el manejo del mismo, para el hombre este elemento es el que rige su vida, es por ello que incorporarlo en un sistema es un gran logro.

En el momento en el que las BDT incorporan al tiempo en su funcionamiento de una manera diferente a como se emplea en las bases de datos que se utilizan habitualmente, permiten que los datos almacenados tengan asociado al factor tiempo, es decir, el momento en el que se crea o presenta cambios la información, el instante en el cual se efectuó dicho evento quedará grabado, así mismo, cada uno de los datos quedará identificado de manera única dentro de la base de datos.

Según la teoría de la toma de decisiones al emplear las BDT se obtiene una toma de decisión bajo certidumbre, y así se pueden determinar las consecuencias generadas de la toma de una decisión, basado en el comportamiento que tuvieron los datos en un periodo de tiempo. Cabe señalar que ninguna solución bajo esta definición o cualquier otra es 100% segura en lo que respecta a los resultados que se llegarán a obtener en la realidad, esto se debe a que sólo se puede pronosticar el posible resultado al que se puede llegar al momento de optar por una solución.

Además con las BDT se puede hacer uso de nuevos elementos como los calendarios, que permitirán tener un mayor control de la información, así como nuevas funciones que permiten que la información quede almacenada de manera más estricta, incrementándose así su funcionalidad dentro de la organización.

Al manejar los calendarios dentro de este tipo de bases se pueden establecer las medidas de tiempo más adecuadas a las necesidades de la misma, es decir, un calendario permite dividir la información en diversos periodos de tiempo, así como, establecer fechas importantes, días no laborables o fines de semana en los que determinados datos no pueden ser modificados o la información accedida, todo esto dependiendo de las necesidades de la organización.

Por otro lado se pueden manejar nuevas y eficientes funciones que permiten un análisis más rápido y eficiente de la información. Estas funciones pueden hacer referencia a calendarios, a fechas en las que la información se manipula de un modo diferente a lo cotidiano (excepciones) o para la recuperación de la información en un periodo de tiempo determinado, esto se puede ver reflejado en la tabla 3.2 que se presenta a continuación.

Tipo	Función	Descripción
Funciones relacionadas con Calendarios	CombineCals	Combina dos calendarios. Ambos patrones deben ser idénticos.
Funciones relacionadas con excepciones	NumOffExceptions	Sirve para recuperar el número de off-exceptions entre dos fechas.
	NumOnExceptions	Sirve para recuperar el número de on-exceptions entre dos fechas.
Funciones de relacionadas con fechas e índices	IsValidDate	Determina si la fecha proporcionada es válida.
	OffsetDate	Se recupera la fecha que es k días en el futuro (o k días en el pasado en el caso de k negativa) de la fecha proporcionada.
	GetIntervalStart	Se recupera el inicio del intervalo que incluye la marca de tiempo de entrada.
	GetIntervalEnd	Se recupera el final del intervalo que incluye la marca de tiempo de entrada.
	NumTimestampBetween	Se recupera el número de marcas de tiempo válidas entre dos fechas.
	NumInvalidTimestampBetween	Se recupera el número de marcas de tiempo inválidas entre dos fechas.
	TimestampBetween	Se recuperan las marcas de tiempo válidas entre dos fechas.
	InvalidTimestampBetween	Se recuperan las marcas de tiempo inválidas entre dos fechas.
	SetPrecision	Establece la precisión de las marcas de tiempo que corresponden a la frecuencia del calendario.

Tabla 3.2 Funciones que se encuentran en el manejador Oracle 8i (V 8.1.7)

Existen otras funciones que permiten la obtención de información específica dentro de los intervalos de tiempo, de este modo se ahorra tiempo y se tiene un alto grado de precisión en los datos que se van a analizar para la elaboración de posibles soluciones como se puede observar en la tabla 3.3.

Función	Descripción
FirstN	Obtienen los primeros n elementos en la serie de tiempo.
GetSeries	Obtiene las series de tiempo enteras.
TrimSeries	Obtiene los datos de las series de tiempo entre las fechas proporcionadas.
LastN	Obtiene los últimos n elementos en una serie de tiempo.

Tabla 3.3 Funciones de extracción y recuperación en el manejador Oracle8i(V8.1.7)

Ejemplo del uso de la función FirstN con datos del área de cursos del DSC.

```

SET SERVEROUTPUT ON
ALTER SESSION
SET NLS_DATE_FORMAT = 'DD-MON-RR';
DECLARE
dummyval INTEGER;
BEGIN
SELECT ORDSYS.TimeSeries.Display(
ORDSYS.TimeSeries.FirstN(u.cantidad_cierre, 3), 'Primeros 3 resultados')
INTO dummyval FROM bdsr.tscursos ts, bdsr.tscursos_cal cal
WHERE curso='1AU20902';
END;
    
```

El resultado del código anterior es el siguiente:

```

Session altered.
Primeros 3 resultados :
Calendar Data: Atomic NULL
Series Data:
-----
Date          Value
15-JAN-02    4000
16-JAN-02    4000
17-JAN-02    8600
PL/SQL procedure successfully completed.
    
```



La toma de decisiones requiere de análisis estadísticos y probabilísticos para obtener la mejor alternativa, es decir, la solución que nos proporcione mayores ganancias, menos pérdidas, etc. Ésto depende del problema que se haya suscitado en la organización, por lo general la toma de una decisión está basada en maximizar o minimizar algo. Ahora bien, las funciones de las BDT permiten la manipulación de los datos, de tal modo que se pueden obtener los elementos más comunes para efectuar un análisis estadístico como se puede observar en las tablas 3.4 y 3.5 que muestran los beneficios que se pueden obtener al hacer uso de las funciones de las BDT.

Función	Descripción
TSAvg	Obtiene el valor promedio de una serie de tiempo.
TSCount	Obtiene el número de elementos en una serie de tiempo.
TSMax	Obtiene el valor máximo de una serie de tiempo.
TSMaxN	Especifica el número de los valores más altos en una serie de tiempo.
TSMedian	Obtiene el elemento medio de una serie de tiempo.
TSMin	Obtiene el valor mínimo de una serie de tiempo.
TSMinN	Especifica el número de elementos inferiores en una serie de tiempo.
TSProd	Obtiene el producto de los elementos de una serie de tiempo.
TSSiddev	Obtiene la desviación estándar (raíz cuadrada de la varianza)
TSSum	Suma los elementos de una serie de tiempo
TSVariance	Varianza

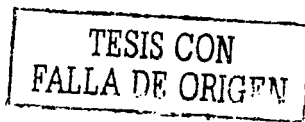
Tabla 3.4 Funciones adicionales en el manejador Oracle8i(V8.1.7)

Ejemplo del uso de la función TSAvg con datos del área de cursos del DSC.

```
SELECT ORDSYS.TimeSeries.TSAvg(cantidad_cierre),
       ORDSYS.TimeSeries.TSVariance(cantidad_cierre),
       ORDSYS.TimeSeries.TSSidDev(cantidad_cierre)
FROM BDSC.ucursos
WHERE curso='IEXB1002';
SQL> @TSAV(1.a4)
```

El resultado del código anterior es el siguiente:

```
ORDSYS.TIMESERIES.TSAVG(CANTIDAD_CIERRE)
-----
ORDSYS.TIMESERIES.TSVARIANCE(CANTIDAD_CIERRE)
-----
ORDSYS.TIMESERIES.TSSTDDEV(CANTIDAD_CIERRE)
-----
          3780
          8057000
          2838.48551
```



Función	Descripción
Mavg	La función Moving average recupera de las series de tiempo el promedio de los valores de cada marca de tiempo sucesiva en un intervalo especificado.
Msum	Moving sum La función Moving Sum recupera la suma de los valores de cada marca de tiempo sucesiva de intervalo de tiempo específico.

Tabla 3.5 Funciones adicionales en el manejador Oracle®(V8.1.7)

Ejemplo del uso de la función TSAvg con datos del Área de Cursos del DSC.

```

SELECT to_char(totamp) totamp, cantidad_cierre
FROM BDCS.tscursos ts,
TABLE (CAST (ORDSYS.TimeSeries.ExtractTable(
ORDSYS.TimeSeries.Mavg(ts.cantidad_cierre,
to_date( '17-JAN-02' , 'DD-MON-YY' ),
to_date( '18-JAN-02' , 'DD-MON-YY' ),10) )
AS ORDSYS.ORDTimeTab)) t
WHERE ts.curso= '1EXB1002' ;
    
```

El resultado del código anterior es el siguiente:

```

SQL> @MAVG.sql

TSTAMP          VALUE
-----
01/17/2002 00:00:00
01/18/2002 00:00:00
    
```

Además de las funciones anteriores, también se cuenta con una serie de procedimientos que permiten que la administración de una BDT sea menos compleja, la tabla 3.6 presenta algunas de los procedimientos que pueden ser usados por los administradores para mantener en buenas condiciones las tablas que utilizan las series de tiempo.

Procedimientos	Descripción
Begin_Create_TS_Group	Inicio del contexto para crear un grupo de series de tiempo (El esquema de los objetos para las series de tiempo).
Cancel_Create_TS_Group	Cancela la creación de un grupo de series de tiempo, es decir, cancela el contexto iniciado por el procedimiento Begin_Create_TS_Group.
Close_Log	Cierra el archivo log que ha sido abierto por el procedimiento Open_Log.
Display_Attributes	Muestra información del esquema de la serie de tiempo que ha sido creada.
Drop_TS_Group	Borra la definición de las series de tiempo y las vistas que se encuentran asociadas. Sin embargo las tablas subyacentes(tablas calendario, tablas de detalles de los datos, etc.) son borradas.
Drop_TS_Group_All	Borra la definición de la serie de tiempo, así como, todas las tablas, vistas, índices, constraints, y triggers asociadas a la misma.
End_Create_TS_Group	Cierra el contexto que se estableció por el procedimiento Begin_Create_TS_Group y crea todos los objetos del esquema.
Get Flat Attributes	Recupera los atributos de una serie de tiempo plana.
Get_Object_Attributes	Recupera los atributos de un modelo-objeto de una serie de tiempo.
Get_Status	Checa si la creación de la secuencia de una serie de tiempo está en progreso.
Open_Log	Abre un archivo log que contendrá el DDL con las sentencias generadas por los procedimientos de las herramientas de administración de las series de tiempo.
Set Flat Attributes	Establece los atributos de una serie de tiempo plana.
Set_Object_Attributes	Establece los atributos de un modelo-objeto de una serie de tiempo.

Tabla 3.6 Procedimientos de administración para el manejo de series de tiempo en el manejador Oracle®(V8.1.7)

El ejemplo que a continuación se muestra, es parte del código que se empleó para la creación del esquema que cubrirá las necesidades del área de cursos, en éste se puede ver como se emplean las funciones: Drop_Ts_Group_All, Begin_Create_TS_Group, Set_Flat_Attributes, End_Create_TS_Group, y Cancel_Create_Ts_Group.

```
DECLARE
BEGIN
ORDSYS.TSTools.Drop_Ts_Group_All('scursos');
exception
when others then
raise;
END;
/

DECLARE
BEGIN
ORDSYS.TSTools.Begin_Create_Ts_Group('scursos','Ylat');
ORDSYS.TSTools.Set_Flat_Attributes('aname_colname'=>'curso','aname_length'=>10);
ORDSYS.TSTools.Add_Number_Column('cantidad_sperna',7,2);
ORDSYS.TSTools.End_Create_Ts_Group;
exception
when others then
begin
ORDSYS.TSTools.Cancel_Create_Ts_Group;
raise;
end;
/
```

El resultado del código anterior es la creación de un grupo denominado cursos. El RDBMS nos indica que se ha realizado tal tarea mostrando el siguiente mensaje:

PL/SQL procedure successfully completed.

3.4 Factores a analizar para la implementación de una base de datos temporal

Para convertir un sistema de bases de datos convencional a un sistema temporal, primero hay que revisar las dependencias funcionales, dado que juegan un papel principal dentro de una base de datos convencional, así como el chequeo de nuestro sistema, lo que ayuda a que cuando se lleva a cabo el diseño de una base de datos no se ignoren puntos de tiempo válido y de tiempo de transacción, entonces tenemos diferentes atributos en el mismo esquema inicial, debido a que los requerimientos son diferentes cuando se habla de un soporte temporal. Algo que es importante recalcar es que no por cambiar a un esquema de este tipo es necesario cambiar todos los atributos, porque no es necesario.

A continuación se tiene un pequeño resumen de las tareas que deben llevar a cabo el diseñador de la base de datos. El diseñador da inicio con un conjunto de conceptos atemporales para posteriormente anotar este esquema con propiedades temporales.

1. Identificar tipos de entidad y representarlos con atributos sustitutos. Los objetos que se encuentran en el mundo real (o entidades) esos atributos de la base de datos descritos son representados con atributos sustitutos.
2. Determinar el soporte temporal requerido. Para cada atributo se debe indicar el soporte temporal requerido para los atributos. Grabar las interacciones entre el tiempo válido y el tiempo de transacción implicados por las especificaciones temporales.
3. Descripción de precisiones. Para cada atributo time-varying (variante de tiempo), indica un conjunto de aplicaciones granulares.
4. Descripción de lifespans. Para cada esquema de relación se describen los lifespan de los atributos.
5. Especificación de dependencias funcionales temporales en el esquema. Ésto incluye la identificación de las llaves primarias.

CAPITULO IV

Aplicación de una base de datos temporal

La toma de decisiones se efectúa en todo momento y a cualquier nivel pero el presente capítulo se enfoca a un problema en particular que ocurre en el Departamento de Servicios de Cómputo (DSC) de la ENEP Acatlán.

A lo largo del presente capítulo se pondrá énfasis en una propuesta para solucionar el problema que le atañe al área de cursos del DSC. El área a la cuál se hace referencia se encarga de planificar, organizar e impartir los cursos de computación que se brindan al público en general por parte de la UNAM campus Acatlán.

4.1 Planteamiento del Problema

El problema se refiere al almacenamiento de información que lleva a cabo el área de cursos. El área mencionada guarda su información mediante un sistema de archivos, que es obsoleto y ocasiona redundancia en los datos almacenados, provocando a la larga inconsistencia de la información. Lo cual es grave porque al momento de evaluar estos datos para la toma de decisiones se presentan problemas como: Datos incorrectos, datos no actualizados, datos repetidos, etc. lo que ocasiona que las decisiones que se toman no sean las más adecuadas.

4.2 Propuesta

La solución que se propone para poner fin a este problema es la implementación de una base de datos que cubra las necesidades del área de cursos, pero que al mismo tiempo forme parte del DSC, ya que ésta área no es independiente sino que es parte de un departamento. Un punto importante es que no sólo se trata de diseñar e implementar una base de datos que cubra las necesidades del área antes citada, sino, ir un paso más allá, elaborar una base de datos que les permita almacenar la información de una forma más organizada, lo que les permitirá efectuar análisis más detallados de lo que está sucediendo dentro del área. Este punto es donde se va a poner en práctica toda la teoría que se ha explicado en capítulos anteriores, es decir, la implementación de los conceptos de las BDT.

4.3 Alcance

Lo que se pretende con esta propuesta es cubrir las necesidades del área, y al mismo tiempo brindarles una herramienta, que haga posible que los datos almacenados sean lo más explícitos posible, para que de este modo se puedan tomar mejores decisiones; por ejemplo se podrán realizar estadísticas y análisis de lo sucedido en todos y cada uno de los cursos de años anteriores, de temporadas anteriores, etc.

4.4 Desarrollo

Como primer punto, se analizó la forma en la que se debía estructurar la base de datos del DSC de una forma general, debido a que la base de datos que posee éste es poco funcional y es necesario rediseñarla para que cubra de un mejor modo las necesidades del propio departamento. Una vez realizado el análisis se procede con el estudio más a fondo de las necesidades que presenta el área de cursos dado que el problema principal es proveer al área de un medio para almacenar su información permitiéndole así un mejor desempeño en su actividad.

El área de cursos presenta las siguientes requerimientos de forma general. Tener los datos de los participantes que se inscriben a sus cursos, así como del personal que se encarga de impartir los mismos, además necesita guardar información específica de los diversos cursos que imparte como: el nombre, el costo, conocimientos previos para poder tomar el curso, duración, horario. También necesita tener localizadas las diferentes aulas donde se dan los cursos, así como las capacidades de las mismas.

Una vez que se localizaron las necesidades que presenta el DSC de forma global, así como, las necesidades que presenta el área de cursos se elaboró el siguiente esquema ^{NOTA} (ver figura 4.1).

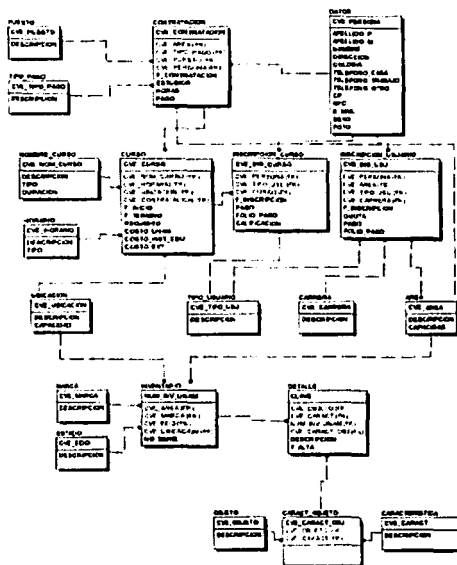


Figura 4.1 Estructura de la base de datos del DSC

TESIS CON
FALLA DE ORIGEN

NOTA Como el presente trabajo está enfocado al área de cursos se pondrá énfasis en esta, no obstante si se desea información más detallada se puede consultar el Apéndice B (Diccionario de datos) donde se encuentra información más explícita de la base de datos que se elaboró para darle solución al problema presentado.

Para cubrir las necesidades del área antes mencionada se requiere del manejo de las siguientes tablas; cabe mencionar que estas tablas pueden contener información que involucre a otras partes del centro de cómputo ya que esta base no es exclusiva del área, sino que cubre necesidades de todo el departamento:

- Datos
- Contratación
- Curso
- Horario
- Inscripción al Curso
- Nombre del Curso
- Ubicación
- Tipo usuario

Como prueba del funcionamiento de las BDT se pretende cubrir una necesidad del área mediante el uso de las series de tiempo; dicho problema consiste en determinar la cantidad de dinero y el número de alumnos que se tienen al inicio y término de un día. Al encontrar la solución del problema se podrán generar reportes económicos y administrativos más detallados. En la tabla 4.1 se puede apreciar de forma gráfica un ejemplo del problema citado.

Nombre del curso	Fecha	Cantidad al inicio del día	Cantidad al fin del día	Cantidad alumnos al inicio del día	Cantidad de alumnos al final del día
Word	01-ENE-2002	\$2300.00	\$5200.00	3	6

Tabla 4.1 Esquema de la tabla que se necesita para el área de curso

Una vez que se ha llegado a este punto, el siguiente paso es la elaboración del esquema que se requiere para el buen funcionamiento de las base de datos temporales.

4.4.1 Esquema Temporal de la base de datos del Área de Cursos del Centro de Cómputo

El esquema temporal que cubre las necesidades del área de cursos de acuerdo a la teoría vista en el capítulo I es el siguiente:

- Estructura Temporal

- Principio Temporal

- Absoluto:

- Esto se debe a que se tiene conocimiento del lugar donde se localiza un punto en el tiempo y trabaja bajo el principio de:

- Intervalo: En donde se conoce la fecha de inicio y fin del calendario conforme a la definición que dice *"Es un periodo de tiempo entre dos puntos específicos (instantes) que son los límites superior e inferior del intervalo, en este caso se conoce el principio y fin del intervalo"*.

- Dominio Temporal

- Dominio Discreto: Como se va a trabajar con unidades enteras (los días). Que el dominio comprenda al conjunto de los números es la razón por la cual se establece la pertenencia a este dominio.

- Determinación Temporal

- Principio Temporal Determinado: La definición dice: *"Es el caso en el cual se conoce con precisión el tiempo de duración de un evento"*, cumple con esta característica debido a que cada evento ocurre a diario, es decir, su duración es de 24 hrs, por lo tanto se conoce el tiempo de duración del evento.

- Representación Temporal

- Se refiere a la medición del tiempo a través de calendarios. En este caso se elaboraron dos calendarios *"CursoInv2002"* y *"CursoPrim2002"* que corresponde a los requerimientos.

- Orden Temporal

- El orden que presenta este esquema es lineal ya que el tiempo fluye desde el pasado hasta el futuro de una forma continua y ordenada.

- Historial Temporal

- En este caso se maneja tanto el tiempo válido como el tiempo de transacción, porque el primero es empleado para la captura de datos, mientras que el de transacción modela la actividad de actualización asociada con la relación.

Este esquema de base de datos temporal corresponde a una base de datos Bitemporal, como se observó en el punto anterior maneja tanto tiempo válido como tiempo de transacción.

4.4.2 Elaboración de los elementos de series de tiempo que requiere el Área de Cursos

Como resultado del análisis efectuado con anterioridad se conoce lo que se necesita para llevar a cabo la construcción del modelo temporal. Cabe señalar que en este punto se debe tener creada la base de datos y los componentes necesarios para la manipulación de las series de tiempo. En el Apéndice A se pueden ver los elementos que se instalaron para la utilización de las series de tiempo.

Debido a que los datos que se van a analizar corresponden a dos convocatorias que se efectuaron en el Centro de Cómputo durante el año 2002 y que de aquí en adelante denominaremos "*convocatoria de invierno*" y "*convocatoria de primavera*", se requiere la creación de dos calendarios uno para cada una.

En el script que se presenta a continuación se verá la serie de instrucciones que se emplearon para la construcción de los objetos que harán posible la manipulación de los datos que estén asociados directamente al tiempo, dicho script está basado en información de [ORA99]:

```

set echo on;
connect hdsej/*****;
set serveroutput on;
ALTER SESSION SET NLS_DATE_FORMAT = 'MM/DD/YYYY';

DECLARE
-- El siguiente código sirve para eliminar objetos creados con anterioridad.
BEGIN
ORDSYS.TSTools.Drop_Ts_Group_All('scursos');
exception
when others then
raise;
END;
/

DECLARE
-- Creación del grupo "cursos" como el esquema de series de tiempo
-- que será utilizado por las herramientas de administración
BEGIN
ORDSYS.TSTools.Begin_Create_Ts_Group('scursos','flat');
ORDSYS.TSTools.Set_Flat_Atributos('sname_colname => 'curso',sname_length => 10);

ORDSYS.TSTools.Add_Number_Column('cantidad_apertura',7,2);
ORDSYS.TSTools.Add_Number_Column('cantidad_cierre',7,2);

ORDSYS.TSTools.Add_Integer_Column('inscritos_inicio');
ORDSYS.TSTools.Add_Integer_Column('inscritos_final');
ORDSYS.TSTools.End_Create_Ts_Group;
-- En el momento en el que se ejecuta la línea que contiene la instrucción End_Create_Ts_Group
-- se crean automáticamente varios objetos que forman parte del esquema como:
-- TSCURSOS: Es un objeto de tipo vista que se emplea para el uso de las funciones de las series de tiempo.
-- TSCURSOS_TAB: Es la tabla detalle de los datos.
-- TSCURSOS_MAP: Es la tabla de mapeo.
-- TSCURSOS_CAL: Es la tabla para las definiciones de calendario
exception
when others then
begin
ORDSYS.TSTools.Cancel_Create_Ts_Group;
raise;
end;
END;
/

```

--Se establece "curso" como el elemento
--que servirá para la utilización de las
--funciones
--Se definen columnas numéricas para el
--manejo de cantidades de dinero con dos
--máximos de 7 cifras donde se permiten dos
--decimales
--Se definen columnas de tipo entero

TESIS CON
FALLA DE ORIGEN

--El siguiente código es para la creación de las definiciones de los calendarios que se van a emplear.

```

INSERT INTO tcursos_cal
VALUES(
ORDSYS.ORDTCalendar(
0,
'CursoInv2002',
4,
ORDSYS.ORDTPattern(ORDSYS.ORDTPatternBits(1,1,1,1,1,1)),
(to_date('01-DEC-2001','DD-MON-YYYY')),
to_date('01-DEC-2001','DD-MON-YYYY'),
to_date('28-FEB-2002','DD-MON-YYYY'),
ORDSYS.ORDTExceptions(to_date('12-DEC-2001','DD-MON-YYYY'),
to_date('17-DEC-2001','DD-MON-YYYY'),
to_date('18-DEC-2001','DD-MON-YYYY'),
to_date('19-DEC-2001','DD-MON-YYYY'),
to_date('20-DEC-2001','DD-MON-YYYY'),
to_date('21-DEC-2001','DD-MON-YYYY'),
to_date('22-DEC-2001','DD-MON-YYYY'),
to_date('23-DEC-2001','DD-MON-YYYY'),
to_date('24-DEC-2001','DD-MON-YYYY'),
to_date('25-DEC-2001','DD-MON-YYYY'),
to_date('26-DEC-2001','DD-MON-YYYY'),
to_date('27-DEC-2001','DD-MON-YYYY'),
to_date('28-DEC-2001','DD-MON-YYYY'),
to_date('29-DEC-2001','DD-MON-YYYY'),
to_date('30-DEC-2001','DD-MON-YYYY'),
to_date('31-DEC-2001','DD-MON-YYYY'),
to_date('01-JAN-2002','DD-MON-YYYY'),
to_date('02-JAN-2002','DD-MON-YYYY'),
to_date('03-JAN-2002','DD-MON-YYYY'),
to_date('04-JAN-2002','DD-MON-YYYY'),
to_date('05-JAN-2002','DD-MON-YYYY'),
to_date('06-JAN-2002','DD-MON-YYYY'),
to_date('05-FEB-2002','DD-MON-YYYY'),
NULL));

```

commit;

```

INSERT INTO tcursos_cal
VALUES(
ORDSYS.ORDTCalendar(
1,
'CursoPrim2002',
4,
ORDSYS.ORDTPattern(ORDSYS.ORDTPatternBits(1,1,1,1,1,1)),

```

```

(to_date('01-MAR-2002','MM-DD-YYYY')),
to_date('01-MAR-2002','MM-DD-YYYY'),
to_date('31-MAY-2002','MM-DD-YYYY'),
ORDSYS.ORDTExceptions(to_date('03-21-2002','MM-DD-YYYY'),
to_date('03-15-2002','MM-DD-YY'),
to_date('03-26-2002','MM-DD-YY'),
to_date('03-27-2002','MM-DD-YY'),
to_date('03-28-2002','MM-DD-YY'),
to_date('03-29-2002','MM-DD-YY'),
to_date('03-30-2002','MM-DD-YY'),
to_date('03-31-2002','MM-DD-YY'),
to_date('05-01-2002','MM-DD-YY'),
to_date('05-10-2002','MM-DD-YY'),
to_date('05-15-2002','MM-DD-YY'),
NULL));

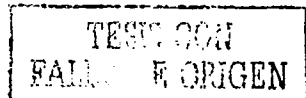
```

commit;

- Calendario de los cursos de Inverno del año 2002.
- Se indica que el calendario se maneja por días.
- Se indica el patrón de días que se van a manejar (en este caso es de lunes a domingo)
- Fecha en que empieza el año calendario
- patrón
- Fecha de Inicio del Calendario
- Fecha en que caduca el calendario
- Días que se excluyen del calendario

- Este valor indica que no hay fechas extras
- que deban incluirse en el calendario

- Calendario de los cursos de Primavera del año 2002.
- Se indica que el calendario se maneja por días.
- Se indica el patrón de días que se van a manejar (en este caso es de lunes a domingo)
- Fecha en que empieza a funcionar el calendario.
- Fecha de inicio del calendario
- fecha en que caduca el calendario
- Días que se excluyen del calendario



- Este valor indica que no hay fechas extras
- que deban incluirse en el calendario

--El código que se presenta a continuación es para que el manejador pruebe que
--el calendario de invierno no presente errores de definición. --

```
DECLARE
  tsCal ordsys.ordicalendar;
  dummyVal integer;
  validFlag integer;
BEGIN
  select value(cal) into tsCal
  from bdsr.tscursos_cal cal
  where cal name = 'CursoInv2002';
  select ordsys.timeseries.display(tsCal) into dummyVal from dual;
  dbms_output.new_line;
  validFlag := ORDSYS.CALENDAR.IsValidCal(tsCal);
  if (validFlag = 1) then
    dbms_output.put_line('CursoInv2002 calendar is valid.');
```

```
  else
    dbms_output.put_line('CursoInv2002 calendar is NOT valid.');
```

```
  dbms_output.put_line('Use ValidateCal to determine inconsistency.');
```

```
  end if;
END;
```

--El código que se presenta a continuación es para que el manejador pruebe que
--el calendario de primavera no presente errores de definición.

```
DECLARE
  tsCall ordsys.ordicalendar;
  dummyVal1 integer;
  validFlag1 integer;
BEGIN
  select value(cal) into tsCall
  from bdsr.tscursos_cal cal
  where cal name = 'CursoPrim2002';
  select ordsys.timeseries.display(tsCall) into dummyVal1 from dual;
  dbms_output.new_line;
  validFlag1 := ORDSYS.CALENDAR.IsValidCal(tsCall);
  if (validFlag1 = 1) then
    dbms_output.put_line('CursoPrim2002 calendar is valid.');
```

```
  else
    dbms_output.put_line('CursoPrim2002 calendar is NOT valid.');
```

```
  dbms_output.put_line('Use ValidateCal to determine inconsistency.');
```

```
  end if;
END;
```

TESIS CON
FALLA DE ORIGEN

A continuación se muestran los resultados obtenidos al correr el script anterior.

```

SQL> @@tcursos.sql
SQL> set echo on;
SQL> connect hdsc***;
Connected.
SQL> set serveroutput on;
SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'MM/DD/YYYY';
Session altered.
SQL> DECLARE
2
3 BEGIN
4
5   ORDSYS.TSTools.Drop_Ts_Group_All('tcursos');
6
7   exception
8   when others then
9     raise;
10
11 END;
12 /
PL/SQL procedure successfully completed.
SQL> DECLARE
3 BEGIN
13   ORDSYS.TSTools.Begin_Create_Ts_Group('tcursos','IAT');
14   ORDSYS.TSTools.Set_Flat_Attributes(sname_colname => 'curso',sname_length => 10);
15   ORDSYS.TSTools.Add_Number_Column('cantidad_apertura',7,2);
16   ORDSYS.TSTools.Add_Number_Column('cantidad_ciencia',7,2);
17   ORDSYS.TSTools.Add_Integer_Column('inscritos_inicio');
18   ORDSYS.TSTools.Add_Integer_Column('inscritos_final');
19   ORDSYS.TSTools.End_Create_Ts_Group;
20
21   exception
22   when others then
23     begin
24       ORDSYS.TSTools.Cancel_Create_Ts_Group;
25     raise;
26   end;
27
28 END;
29 /
PL/SQL procedure successfully completed.
SQL> INSERT INTO tcursos_cal
3 VALUES
4 (
5   'Curso Inv 2002',
6   4,
7   ORDSYS.ORDTPattern(ORDSYS.ORDTPatternBits(1,1,1,1,1,1,1)),
8   (to_date('01-DEC-2001','DD-MON-YYYY'))),
9   (to_date('01-DEC-2001','DD-MON-YYYY')),
10  (to_date('28-FEB-2002','DD-MON-YYYY')),
11  ORDSYS.ORDTExceptions(to_date('12-DEC-2001','DD-MON-YYYY')),
12  (to_date('17-DEC-2001','DD-MON-YYYY')),
13  (to_date('18-DEC-2001','DD-MON-YYYY')),
14  (to_date('19-DEC-2001','DD-MON-YYYY')),
15  (to_date('20-DEC-2001','DD-MON-YYYY')),
16  (to_date('21-DEC-2001','DD-MON-YYYY')),
17  (to_date('22-DEC-2001','DD-MON-YYYY')),
18  (to_date('23-DEC-2001','DD-MON-YYYY')),
19  (to_date('24-DEC-2001','DD-MON-YYYY')),
20  (to_date('25-DEC-2001','DD-MON-YYYY')),
21  (to_date('26-DEC-2001','DD-MON-YYYY')),
22  (to_date('27-DEC-2001','DD-MON-YYYY')),
23  (to_date('28-DEC-2001','DD-MON-YYYY')),
24  (to_date('29-DEC-2001','DD-MON-YYYY')),
25  (to_date('30-DEC-2001','DD-MON-YYYY')),
26  (to_date('31-DEC-2001','DD-MON-YYYY')),
27  (to_date('01-JAN-2002','DD-MON-YYYY')),
28  (to_date('02-JAN-2002','DD-MON-YYYY')),
29  (to_date('03-JAN-2002','DD-MON-YYYY')),
30  (to_date('04-JAN-2002','DD-MON-YYYY')),
31  (to_date('05-JAN-2002','DD-MON-YYYY')),
32  (to_date('06-JAN-2002','DD-MON-YYYY')),
33  (to_date('05-FEB-2002','DD-MON-YYYY')),

```

TESIS CON
FALLA DE ORIGEN

```

34 NULL);
1 row created.
SQL> commit;
Commit complete.

SQL> INSERT INTO tucursos_cal
2 VALUES(
3 ORDSYS.ORDTCalendar(
4 1,
5 'CursoPrim2002',
6 4,
7 ORDSYS.ORDTPattern(ORDSYS.ORDTPatternBits(1,1,1,1,1,1),
8 (to_date('03-01-2002','MM-DD-YYYY'))),
9 to_date('03-01-2002','MM-DD-YYYY'),
10 to_date('05-31-2002','MM-DD-YYYY'),
11 ORDSYS.ORDTEExceptions(to_date('03-21-2002','MM-DD-YYYY'),
12 to_date('03-25-2002','MM-DD-YY'),
13 to_date('03-26-2002','MM-DD-YY'),
14 to_date('03-27-2002','MM-DD-YY'),
15 to_date('03-28-2002','MM-DD-YY'),
16 to_date('03-29-2002','MM-DD-YY'),
17 to_date('03-30-2002','MM-DD-YY'),
18 to_date('03-31-2002','MM-DD-YY'),
19 to_date('05-01-2002','MM-DD-YY'),
20 to_date('05-10-2002','MM-DD-YY'),
21 to_date('05-15-2002','MM-DD-YY')),
22 NULL));
1 row created.
SQL> commit;
Commit complete.

SQL> DECLARE
2   tsCal ordsys.ordtcalendar;
3   dummyVal integer;
4   validFlag integer;
5 BEGIN
6   select value(cal) into tsCal
7   from bdsc.tucursos_cal
8   where cal.name = 'CursoInv2002';
9   select ordsys.timeSeries.display(tsCal) into dummyVal from dual;
10  dbms_output.put_line('');
11  validFlag := ORDSYS.CALENDAR.IsValidCal(tsCal);
12
13  if (validFlag = 1) then
14    dbms_output.put_line('CursoInv2002 calendar is valid. ');
15  else
16    dbms_output.put_line('CursoInv2002 calendar is NOT valid. ');
17  end if;
18
19  dbms_output.put_line('Use ValidateCal to determine inconsistency. ');
20
21 END;
22 /
Calendar Name = CursoInv2002
Frequency = 4 (day)
MinDate = 12/01/2001
MaxDate = 03/28/2002
pattern:
1,1,1,1,1,1
patternAnchor = 12/01/2001
onExceptions : Atomic NULL
offExceptions :
12/12/2001 12/17/2001 12/18/2001
12/19/2001 12/20/2001 12/21/2001
12/22/2001 12/23/2001 12/24/2001
12/25/2001 12/26/2001 12/27/2001
12/28/2001 12/29/2001 12/30/2001
12/31/2001 01/01/2002 01/02/2002
01/03/2002 01/04/2002 01/05/2002
01/06/2002 02/05/2002
CursoInv2002 calendar is valid.
PL/SQL procedure successfully completed.

```

TESIS CON
FALLA DE ORIGEN

```

SQL> DECLARE:
2      tsCall ordsys.ordcalendar;
3      dummyVal1 integer;
4      validFlag1 integer;
5  BEGIN
6      select value(cal) into tsCall
7      from bdsc.tscursor_cal cal
8      where cal.name = 'CursoPrim2002';
9  select ordsys.timeSeries.display(tsCall) into dummyVal1 from dual;
10 dbms_output.new_line;
11
12 validFlag1 := ORDSYS.ValidateCal(tsCall);
13
14 if (validFlag1 = 1) then
15     dbms_output.put_line('CursoPrim2002 calendar is valid');
16 else
17     dbms_output.put_line('CursoPrim2002 calendar is NOT valid');
18     dbms_output.put_line('Use ValidateCal to determine inconsistency.');

```

TESIS CON
FALLA DE ORIGEN

Una vez realizados los pasos anteriores se han creado nuevos objetos que contienen elementos necesarios para poder trabajar con las series de tiempo, las nuevas tablas que se generan son las que se muestran en la tabla 4.2:

Nombre de la Tabla	Tip
TSCURSOS	VIEW
TSCURSOS_CAL	TABLE
TSCURSOS_MAP	TABLE
TSCURSOS_RVW	VIEW
TSCURSOS_TAB	TABLE

Tabla 4.2 Elementos que hacen posible la manipulación de una base de datos temporal

Las tablas que se crearon presentan las siguientes características:

TSCURSOS Name	Null?	Type
CURSO	NOT NULL	VARCHAR2(10)
CANTIDAD_APERTURA		ORDSYS.ORDTNUMSERIESIOTREF
CANTIDAD_CIERRE		ORDSYS.ORDTNUMSERIESIOTREF
INSCRITOS_INICIO		ORDSYS.ORDTNUMSERIESIOTREF
INSCRITOS_FINAL		ORDSYS.ORDTNUMSERIESIOTREF
TSCURSOS_CAL Name	Null?	Type
CALTYPE		NUMBER(38)
NAME	NOT NULL	VARCHAR2(256)
FREQUENCY		NUMBER(38)
PATTERN		ORDSYS.ORDTPATTERN
MINDATE		DATE
MAXDATE		DATE
OFFEXCEPTIONS		ORDSYS.ORDTEXCEPTIONS
ONEXCEPTIONS		ORDSYS.ORDTEXCEPTIONS
TSCURSOS_MAP Name	Null?	Type
CALTYPE		NUMBER(38)
NAME	NOT NULL	VARCHAR2(256)
FREQUENCY		NUMBER(38)
PATTERN		ORDSYS.ORDTPATTERN
MINDATE		DATE
MAXDATE		DATE
OFFEXCEPTIONS		ORDSYS.ORDTEXCEPTIONS
ONEXCEPTIONS		ORDSYS.ORDTEXCEPTIONS
TSCURSOS_RVW Name	Null?	Type
CURSO	NOT NULL	VARCHAR2(10)
ISTAMP	NOT NULL	DATE
CANTIDAD_APERTURA		NUMBER(7,2)
CANTIDAD_CIERRE		NUMBER(7,2)
INSCRITOS_INICIO		NUMBER
INSCRITOS_FINAL		NUMBER
TSCURSOS_TAB Name	Null?	Type
CURSO	NOT NULL	VARCHAR2(10)
ISTAMP	NOT NULL	DATE
CANTIDAD_APERTURA		NUMBER(7,2)
CANTIDAD_CIERRE		NUMBER(7,2)
INSCRITOS_INICIO		NUMBER
INSCRITOS_FINAL		NUMBER

TESIS CON
FALLA DE ORIGEN

Otra de las bondades que se puede explotar con las series de tiempo es la manipulación de calendarios, es decir, una vez que se encuentran creados, se pueden efectuar operaciones como: Intersección de calendarios, unión de calendarios y combinación de calendarios. La finalidad de estas operaciones es la manipulación al máximo de los calendarios que se han creado.

Como ejemplo veremos el script de cómo se combinaron los dos calendarios creados para el área de cursos para trabajar con ellos de manera conjunta.

```

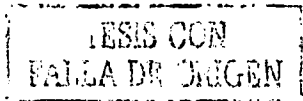
set echo on;
set serveroutput on;
ALTER SESSION SET NLS_DATE_FORMAT = 'MM/DD/YYYY';
DECLARE
  tsCal1 ORDSYS.ORDTCalendar;
  tsCal2 ORDSYS.ORDTCalendar;
  resultCal ORDSYS.ORDTCalendar;
  equalFlag INTEGER;
  dummyVal INTEGER;
BEGIN
  SELECT value(cal) INTO tsCal1
  FROM BDCSCURSORS_cal cal
  WHERE cal name = 'CursoInv2002';
  SELECT value(cal) INTO tsCal2
  FROM BDCSCURSORS_cal cal
  WHERE cal name = 'CursoPrim2002';
  SELECT ORDSYS.TimeSeries.Display(tsCal1) INTO dummyVal FROM dual;
  SELECT ORDSYS.TimeSeries.Display(tsCal2) INTO dummyVal FROM dual;
  resultCal = ORDSYS.Calendar.CombineCals(tsCal1, tsCal2, equalFlag);
  SELECT ORDSYS.TimeSeries.Display(resultCal, 'result of CombineCals')
  INTO dummyVal
  FROM dual;
  DBMS_OUTPUT.PUT_LINE('equalFlag = ' || equalFlag);
  resultCal = ORDSYS.Calendar.IntersectCals(tsCal1, tsCal2);
  --Se muestran los calendarios 1 y 2.
  --Se combinan mediante la función CombineCals
  --Se buscan los puntos de intersección de los mismos mediante la
  --función IntersectCals
  SELECT ORDSYS.TimeSeries.Display(resultCal, 'result of IntersectCals'
  )
  INTO dummyVal
  FROM dual;
END;
/

```

```

Calendar Name = CursoInv2002
Frequency = 4 (day)
MinDate = 12/01/2001
MaxDate = 03/28/2002
pathBits:
1,1,1,1,1,1
pathAnchor = 12/01/2001
onExceptions : Atomic NULL
offExceptions :
12/12/2001 12/17/2001 12/18/2001
12/19/2001 12/20/2001 12/21/2001
12/22/2001 12/23/2001 12/24/2001
12/25/2001 12/26/2001 12/27/2001
12/28/2001 12/29/2001 12/30/2001
12/31/2001 01/01/2002 01/02/2002
01/03/2002 01/04/2002 01/05/2002
01/06/2002 02/05/2002

```



```
Calendar Name = CursoPrim2002
Frequency = 4 (day)
MinDate = 03/01/2002
MaxDate = 05/31/2002
pathbits
1,1,1,1,1,1,1
patAnchor = 03/01/2002
onExceptions : Atomic NULL
offExceptions :
03/21/2002 03/25/2002 03/26/2002
03/27/2002 03/28/2002 03/29/2002
03/30/2002 03/31/2002 05/01/2002
05/10/2002 05/15/2002
result of CombineCals :
Frequency = 4 (day)
MinDate = 03/01/2002
MaxDate = 03/28/2002
pathbits
1,1,1,1,1,1,1
patAnchor = 12/01/2001
onExceptions :
offExceptions :
03/27/2002 03/25/2002 03/26/2002
03/27/2002 03/28/2002
equalFlag = 0
result of IntersectCals :
NULL Calendar
```

PI/SQL procedure successfully completed.

LONG CON
FALLA DE CONJUN

4.5 Pruebas de funcionamiento de la base de datos temporal del Área de Cursos

Una vez que la información estuvo dentro de las tablas, se efectuaron pruebas con las funciones dadas en el capítulo tres para demostrar el funcionamiento de las series de tiempo; ya que dichas funciones proporcionan la información de una manera diferente y esto da como resultado información que simplificará la labor de recopilación de datos para la toma de decisiones.

A continuación se muestra la consulta y los resultados obtenidos de:

Se desea saber la cantidad de dinero que se obtuvo al final de cada uno de los días de inscripción por concepto del curso de Autocad 2D sabatino de 3:00PM a 8:00PM de la temporada de invierno.

```
SELECT to_char(tstamp) tstamp, value
FROM tcursos ta,
TABLE(CAST(ORDSYS.TimeSeries.ExtractTable(ta.cantidad_cierre) AS ORDSYS.ORDTNumTab)) t
WHERE ta.curso='TAU20902';
```

TSTAMP	VALUE
01/14/2002	2600
01/15/2002	6800
01/16/2002	7800
01/17/2002	8800
01/18/2002	14700

El siguiente caso que se presenta es el resultado de una búsqueda similar a la anterior, lo único que cambia es la temporada, en este caso los datos corresponden a la temporada de primavera.

```
SELECT to_char(tstamp) tstamp, value
FROM tcursos ta,
TABLE(CAST(ORDSYS.TimeSeries.ExtractTable(ta.cantidad_cierre) AS ORDSYS.ORDTNumTab)) t
WHERE ta.curso='PAU20902';
```

TSTAMP	VALUE
04/15/2002	3000
04/17/2002	6000
04/18/2002	9300
04/19/2002	9300

Al efectuar una comparación de los datos obtenidos podemos ver que las ganancias que se obtienen del curso disminuyeron con respecto al periodo anterior, también se puede ver de forma detallada los cierres de cada día en los que se inscribieron los participantes. Se puede ver que en el último día de inscripciones en la temporada de invierno acudió el grueso de los participantes a realizar su inscripción. Esto nos lleva al análisis de las posibles causas que hicieron que se presentará una disminución considerable en las ganancias y afirmar que las bases de datos temporales proporcionan información más detallada de los datos conforme transcurre el tiempo. Además en este caso en particular permite tener un mayor control de lo que se recauda diariamente por concepto de cursos.

Como resultado de un análisis de los datos que se tienen en una BDT se pueden tomar diferentes decisiones como: cambiar al curso de horario, modificar el precio, hacer una mayor promoción de los cursos que se imparten, cambiar al instructor, eliminar el curso, ampliar el nombre del curso, disminuir el nombre del curso, etc.

TESIS CON
FALLA DE ORIGEN

Se desea saber la fecha en que se inscribió la primera persona al curso de Autocad 2D sabatino de la temporada de invierno, así como, la cantidad de dinero que se recabó al final de ese día. Para resolver el cuestionamiento se hará uso de la función "First" de las series de tiempo.

```
SQL> @First.sql;
SQL> SET ECHO ON;
SQL> SET SERVEROUTPUT ON;
SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MONTH-YYYY HH24:MI:SS';
Session altered.
SQL> DECLARE dummyval INTEGER;
2 BEGIN SELECT ORDSYS.TimeSeries.Display( ORDSYS.TimeSeries.First(ts.cantidad_cierre), 'Primer Resultado')
3 INTO dummyval
4 FROM BDESC tscurso ts
5 WHERE ts.curso= 'IAU20902';
6 END;
7 /
```

Primer Resultado :
Timestamp : 14-JANUARY -2002 00:00:00
Value : 2600

PL/SQL procedure successfully completed.

A continuación se presenta un caso similar al anterior lo que varía es la temporada.

```
SQL> @First.sql;
SQL> SET ECHO ON;
SQL> SET SERVEROUTPUT ON;
SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MONTH-YYYY HH24:MI:SS';
Session altered.
SQL> DECLARE dummyval INTEGER;
2 BEGIN SELECT ORDSYS.TimeSeries.Display( ORDSYS.TimeSeries.First(ts.cantidad_cierre), 'Primer Resultado')
3 INTO dummyval
4 FROM BDESC tscurso ts
5 WHERE ts.curso= 'PAU20902';
6 END;
7 /
```

Primer Resultado :
Timestamp : 15-APRIL -2002 00:00:00
Value : 3000

PL/SQL procedure successfully completed.

En este caso se puede observar que en el primer día de inscripciones se recopiló una cantidad de dinero muy similar en ambas temporadas, por lo que podemos decir que es un curso que no presenta muchas variaciones en los inicios de la inscripción.

TESIS CON
FALLA DE ORIGEN

Se desea saber qué cantidad de dinero se recaudó en los primeros tres días de inscripción del curso: Diseño de Páginas Web Básico dominical de la temporada de invierno 2002. Para resolver el cuestionamiento se hará uso de la función "FirstN" de las series de tiempo.

```
SQL> @t.FirstN.sql
SQL> SET ECHO ON;
SQL> SET SERVEROUTPUT ON;
SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MONTH-YYYY HH24:MI:SS';
```

Session altered.

```
SQL> DECLARE dummyval INTEGER;
2 BEGIN SELECT ORDSYS.TimeSeries.Display( ORDSYS.TimeSeries.FirstN(ts.cantidad_cierre, 3), 'Primeros 3 Resultados')
3 INTO dummyval
4 FROM BDSC.tscursos ts
5 WHERE ts.curso='IDWB1002';
6 END;
7 /
```

Primeros 3 Resultados :

Calendar Data: Atomic NULL

Series Data:

Date	Value
14-JANUARY -2002 00:00:00	3800
15-JANUARY -2002 00:00:00	8100
16-JANUARY -2002 00:00:00	10200

PL/SQL procedure successfully completed.

A continuación se muestra cómo cambiaron los datos para la temporada de primavera 2002 del mismo curso anterior.

```
SQL> @FirstN.sql
SQL> SET ECHO ON;
SQL> SET SERVEROUTPUT ON;
SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MONTH-YYYY HH24:MI:SS';
```

Session altered.

```
SQL> DECLARE dummyval INTEGER;
2 BEGIN SELECT ORDSYS.TimeSeries.Display( ORDSYS.TimeSeries.FirstN(ts.cantidad_cierre, 3), 'Primeros 3 Resultados')
3 INTO dummyval
4 FROM BDSC.tscursos ts
5 WHERE ts.curso='PAU20902';
6 END;
7 /
```

Primeros 3 Resultados :

Calendar Data: Atomic NULL

Series Data:

Date	Value
15-APRIL -2002 00:00:00	3000
17-APRIL -2002 00:00:00	6000
18-APRIL -2002 00:00:00	9300

PL/SQL procedure successfully completed.

Se puede observar el alcance que se tiene al hacer uso de las bases de datos temporales y los beneficios que se obtienen al hacer uso de funciones como la que se vió. En este caso no existe una gran diferencia en lo que se obtuvo hasta el tercer día en ambas temporadas.

Se desea saber la fecha en que se inscribió la última persona al curso de AutoCAD 2D sabatino de la temporada de invierno 2002, la cantidad total de dinero que se recopiló de ese curso. Para contestar la interrogante se hará uso de la función "Last" de las series de tiempo.

```
SQL> @Last.sql
SQL> SET ECHO ON;
SQL> SET SERVEROUTPUT ON;
SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MONTH-YYYY HH24:MI:SS';
```

Session altered.

```
SQL> DECLARE dummyval INTEGER;
2 BEGIN SELECT ORDSYS.TimeSeries.Display( ORDSYS.TimeSeries.Last(t.cantidad_cierre, 'Primeros Resultados')
3 INTO dummyval)
4 FROM BDSC.tscursos ts
5 WHERE ts.cursor= 'IAU20902';
6 END;
7 /
```

Primeros Resultados :
Timestamp : 18-JANUARY -2002 00:00:00
Value : 14700

PL/SQL procedure successfully completed.

Los datos que presentan a continuación corresponden a la pregunta anterior para el caso de la temporada primavera 2002.

```
SQL> @Last.sql
SQL> SET ECHO ON;
SQL> SET SERVEROUTPUT ON;
SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MONTH-YYYY HH24:MI:SS';
```

Session altered.

```
SQL> DECLARE dummyval INTEGER;
2 BEGIN SELECT ORDSYS.TimeSeries.Display( ORDSYS.TimeSeries.Last(t.cantidad_cierre, 'Primeros Resultados')
3 INTO dummyval)
4 FROM BDSC.tscursos ts
5 WHERE ts.cursor= 'PAU20902';
6 END;
7 /
```

Primeros Resultados :
Timestamp : 19-APRIL -2002 00:00:00
Value : 9300

PL/SQL procedure successfully completed.

A simple vista se puede ver que hubo una baja considerable en la cantidad obtenida en el curso que se brindó en la temporada primavera 2002 con respecto a su similar de la temporada invierno 2002.



Se desea saber lo sucedido en los últimos tres días de inscripción en lo que se refiere al carácter económico del curso de Autocad 2D sabatino de la temporada de invierno 2002. Para resolver el problema en cuestión se hará uso de la función "LastN" de las series de tiempo.

```
SQL> @l lastN.sql
SQL> SET ECHO ON;
SQL> SET SERVEROUTPUT ON;
SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MONTH-YYYY HH24:MI:SS';
```

Session altered.

```
SQL> DECLARE dummyval INTEGER;
2 BEGIN SELECT ORDSYS.TimeSeries.Display(ORDSYS.TimeSeries.LastN(ts.cantidad_cierre, 3), 'Primeros 3 Resultados')
3 INTO dummyval
4 FROM BDSC.tscursos ts
5 WHERE ts.curso='IAU20902';
6 END;
7 /
```

Primeros 3 Resultados:
Calendar Data: Atomic NULL
Series Data:

Date	Value
16-JANUARY -2002 00:00:00	7800
17-JANUARY -2002 00:00:00	8800
18-JANUARY -2002 00:00:00	14700

PL/SQL procedure successfully completed.

A continuación se presentan datos correspondientes a la temporada primavera 2002 para la interrogante anterior.

```
SQL> @l LastN.sql
SQL> SET ECHO ON;
SQL> SET SERVEROUTPUT ON;
SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MONTH-YYYY HH24:MI:SS';
```

Session altered.

```
SQL> DECLARE dummyval INTEGER;
2 BEGIN SELECT ORDSYS.TimeSeries.Display(ORDSYS.TimeSeries.LastN(ts.cantidad_cierre, 3), 'Primeros 3 Resultados')
3 INTO dummyval
4 FROM BDSC.tscursos ts
5 WHERE ts.curso='PAU20902';
6 END;
7 /
```

Primeros 3 Resultados:
Calendar Data: Atomic NULL
Series Data:

Date	Value
17-APRIL -2002 00:00:00	6000
18-APRIL -2002 00:00:00	9300
19-APRIL -2002 00:00:00	9300

PL/SQL procedure successfully completed.

Lo que se puede observar en este caso es la baja en las percepciones económicas con respecto a la temporada anterior y cómo la mayoría de lo recopilado se obtuvo en los primeros días, de lo que surge la siguiente pregunta. ¿Por qué al final no hubo demanda por el curso en la temporada de primavera? o lo que podría ser lo mismo visto desde otra perspectiva es ¿Por qué acudió más gente a inscribirse al final en el curso que se brindó en invierno?. Las BDT al almacenar la información con el elemento tiempo permiten un mejor análisis de la información, de otro modo sólo se podría tener el dato de la cantidad que se recopiló al final de las inscripciones de cada caso. Es por ello que se afirma que las BDT son una herramienta para la toma de decisiones que nos proporcionan la información de una manera más detallada.

Se desea conocer el día en que se inscribió la cuarta persona al curso de Introducción a la Computación y Windows 98 y lo único que recuerda es que se inscribió el penúltimo día. Este es un dato que se requiere para la elaboración de su factura. Para resolver el problema anterior se hace uso de la función "ExtractDate" de las series de tiempo.

```
SQL> @Extractdate.sql
SQL> SET ECHO ON;
SQL> SET SERVEROUTPUT ON;
SQL> SELECT to_char(ORDSYS.TimeSeries.ExtractDate(ORDSYS.TimeSeries.GetNthElement(cantidad_apertura, 4)), 'DD-MONTH-YYYY
HH24.MISS')
  2 FROM BDSC.tscursos ts
  3 WHERE curso = '1C'W0302';

TO_CHAR(ORDSYS.TIMESERIES.
-----
17-JANUARY -2002 00:00:00
```

A continuación se presenta un caso similar al anterior presentado en la temporada primavera 2002, la diferencia radica en que el día que se conoce es el segundo.

```
SQL> @Fextractdate.sql
SQL> SET ECHO ON;
SQL> SET SERVEROUTPUT ON;
SQL> SELECT to_char(ORDSYS.TimeSeries.ExtractDate(ORDSYS.TimeSeries.GetNthElement(cantidad_apertura, 2)), 'DD-MONTH-YYYY
HH24.MISS')
  2 FROM BDSC.tscursos ts
  3 WHERE curso = '1C'W0302';

TO_CHAR(ORDSYS.TIMESERIES.
-----
10-APRIL -2002 00:00:00
```

Se desea conocer la situación económica al final de cada uno de los diferentes días de inscripción al curso semanal de Introducción a la Computación y Windows 98 de la temporada invierno 2002. La consulta de datos se elaborará mediante el uso de la función "ExtractTable" de las series de tiempo.

```
SQL> @Extracttable.sql
SQL> SET ECHO ON;
SQL> SET SERVEROUTPUT ON;
SQL> SELECT * FROM the (SELECT CAST(ORDSYS.TimeSeries.ExtractTable(ts.cantidad_cierre) as ORDSYS.ORDTNumTab)
  2 FROM BDSC.tscursos ts
  3 WHERE curso = '1C'W0302);

TSTAMP          VALUE
-----
14-JANUARY -2002 00:00:00    1600
15-JANUARY -2002 00:00:00    4700
16-JANUARY -2002 00:00:00    5700
17-JANUARY -2002 00:00:00   11300
18-JANUARY -2002 00:00:00   11300
```

A continuación se muestra lo que aconteció en el curso semanal de Introducción a la Computación y Windows 98 de la temporada primavera 2002.

```
SQL> @Extracttable.sql
SQL> SET ECHO ON;
SQL> SET SERVEROUTPUT ON;
SQL> SELECT * FROM the (SELECT CAST(ORDSYS.TimeSeries.ExtractTable(ts.cantidad_cierre) as ORDSYS.ORDTNumTab)
  2 FROM BDSC.tscursos ts
  3 WHERE curso = '1C'W0302');

TSTAMP          VALUE
-----
08-APRIL -2002 00:00:00     3800
10-APRIL -2002 00:00:00     7000
11-APRIL -2002 00:00:00    10600
12-APRIL -2002 00:00:00    13000
```

**TESIS CON
FALLA DE ORIGEN**

Se desea conocer la situación económica al final del segundo día de inscripciones al curso semanal de Introducción a la Computación y Windows 98 de 11:00AM a 1:00PM de la temporada invierno 2002. La consulta de datos se elaborará mediante el uso de la función "ExtractValue" de las series de tiempo.

```
SQL> @ExtractVal.sql
SQL> SET ECHO ON;
SQL> SET SERVEROUTPUT ON;
SQL> SELECT ORDSYS.TimeSeries.ExtractValue( ORDSYS.TimeSeries.GetNthElement(cantidad_cierre, 2))
  2 FROM BDSC.tcourses
  3 WHERE curso = 'ICW0302';

ORDSYS.TIMESERIES.EXTRACTVALUE(ORDSYS.TIMESERIES.GETNTHELEMENT(CANTIDAD_CIERRE,2)
-----
4700
```

A continuación se muestra lo que aconteció para la misma consulta en la temporada primavera 2002.

```
SQL> @ExtractVal.sql
SQL> SET ECHO ON;
SQL> SET SERVEROUTPUT ON;
SQL> SELECT ORDSYS.TimeSeries.ExtractValue( ORDSYS.TimeSeries.GetNthElement(cantidad_cierre, 2))
  2 FROM BDSC.tcourses
  3 WHERE curso = 'ICW0302';

ORDSYS.TIMESERIES.EXTRACTVALUE(ORDSYS.TIMESERIES.GETNTHELEMENT(CANTIDAD_CIERRE,2)
-----
7000
```

Se desea saber el valor máximo que se alcanzó a recaudar en el curso sabatino de Mantenimiento Preventivo de PC's. La consulta de datos se elaborará mediante el uso de la función "TSMMax" de las series de tiempo.

```
SQL> @TsMax.sql
SQL> SET ECHO ON;
SQL> SET SERVEROUTPUT ON;
SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MONTH-YYYY HH24:MI:SS';
Session altered.

SQL> SELECT ORDSYS.TimeSeries.TSMMax(cantidad_cierre,
  2 to_date('14-JAN-2002 00:00:00','DD-MON-YYYY HH24:MI:SS'),
  3 to_date('18-JAN-2002 23:59:59','DD-MON-YYYY HH24:MI:SS'))
  4 TSMMax
  5 FROM BDSC.tcourses
  6 WHERE curso = 'IMAN0902';

TSMAX
-----
13100
```

A continuación se muestra lo que sucedió en el caso de la temporada primavera 2002 para el mismo curso de Mantenimiento Preventivo de PC's.

```
SQL> @TsMax.sql
SQL> SET ECHO ON;
SQL> SET SERVEROUTPUT ON;
SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MONTH-YYYY HH24:MI:SS';
Session altered.

SQL> SELECT ORDSYS.TimeSeries.TSMMax(cantidad_cierre,
  2 to_date('14-APR-2002 00:00:00','DD-MON-YYYY HH24:MI:SS'),
  3 to_date('19-APR-2002 23:59:59','DD-MON-YYYY HH24:MI:SS'))
  4 TSMMax
  5 FROM BDSC.tcourses
  6 WHERE curso = 'PMAN0902';

TSMAX
-----
8700
```

Se desea conocer las cuatro cantidades máximas recaudadas del curso sabatino de Mantenimiento Preventivo de PC's. La consulta de datos se elaborará mediante el uso de la función "TSMaXN" de las series de tiempo.

```
SQL> @@TSMaXN.sql
SQL> SET ECHO ON;
SQL> SET SERVEROUTPUT ON;
SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MONTH-YYYY HH24:MI:SS';
Session altered.

SQL> SELECT * FROM THE(SELECT CAST( ORDSYS.TimeSeries.TSMaXN(cantidad_cierre, 4,
2 to_date('14-APR-2002 00:00:00','DD-MON-YYYY HH24:MI:SS'),
3 to_date('19-APR-2002 23:59:59','DD-MON-YYYY HH24:MI:SS')) as ORDSYS.ORDTNumTab)
4 FROM BPS.C:scursos
5 WHERE curso = 'PMAN0902');
```

TSTAMP	VALUE
19-APRIL -2002 00:00:00	8700
18-APRIL -2002 00:00:00	5800
16-APRIL -2002 00:00:00	4200
15-APRIL -2002 00:00:00	2900

Ahora se verá lo que sucedió con la consulta similar a la anterior para el caso de la temporada de primavera.

```
SQL> @@TSMaXN.sql
SQL> SET ECHO ON;
SQL> SET SERVEROUTPUT ON;
SQL> ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MONTH-YYYY HH24:MI:SS';
Session altered.

SQL> SELECT * FROM THE(SELECT CAST( ORDSYS.TimeSeries.TSMaXN(cantidad_cierre, 4,
2 to_date('14-JAN-2002 00:00:00','DD-MON-YYYY HH24:MI:SS'),
3 to_date('18-JAN-2002 23:59:59','DD-MON-YYYY HH24:MI:SS')) as ORDSYS.ORDTNumTab)
4 FROM BPS.C:scursos
5 WHERE curso = 'TMAN0902');
```

TSTAMP	VALUE
18-JANUARY -2002 00:00:00	13100
17-JANUARY -2002 00:00:00	11800
16-JANUARY -2002 00:00:00	5500
15-JANUARY -2002 00:00:00	3400

En el ejemplo siguiente se mostrarán las funciones estadísticas de las series de tiempo como: porcentaje, varianza y desviación estándar. Se desea conocer el promedio, la varianza y la desviación estándar de lo acontecido en el curso dominical de Diseño de Páginas Web Básico. La consulta de datos se elaborará mediante el uso de las funciones "TSAvg", "TSVariance" y "TSStdDev" de las series de tiempo.

```
SELECT ORDSYS.TimeSeries.TSAvg(cantidad_cierre),
ORDSYS.TimeSeries.TSVariance(cantidad_cierre),
ORDSYS.TimeSeries.TSStdDev(cantidad_cierre)
FROM scursos
WHERE curso='DWB1002';
```

```
ORDSYS.TIMESERIES.TSAVG(CANTIDAD_CIERRE)
ORDSYS.TIMESERIES.TSVARIANCE(CANTIDAD_CIERRE)
ORDSYS.TIMESERIES.TSSDDEV(CANTIDAD_CIERRE)
```

```
10660
28800000
5367,30845
```

Conclusiones

Con base en los resultados obtenidos en las diferentes pruebas podemos decir que la hipótesis planteada resultó verdadera. Ésto se debe a que las BDT al almacenar los datos con el tiempo como un elemento principal, se proporciona la información de un modo tal que se puede ver el comportamiento de la misma en la línea del tiempo, permitiendo así la elaboración de modelos que ayuden a la toma de decisiones, también se pudo observar el gran alcance que se logra con la utilización de las diferentes funciones que proporciona el manejador Oracle Bi en su paquete de series de tiempo.

Así mismo se observó que el paquete que proporciona el manejador de base de datos denominado "Oracle Bi Time Series" cumple con la teoría que presentan las bases de datos temporales.

El área de aplicación de las BDT es inmensa, por lo que se pueden plantear soluciones a problemas que anteriormente no tenían solución o en el mejor de los casos se tenía una solución a medias, debido a la falta de una estructura que soportará datos más complejos. Esta mejoría se logra mediante el manejo del modelo orientado a objetos.

La seguridad es mejorada al hacer uso de las BDT, ya que presenta restricciones en la introducción de los datos, por ejemplo, no permite que se incluyan datos con fecha anterior a la del último registro que se encuentra almacenado.

El manejo de este tipo de bases de datos permite elaborar calendarios que le queden a la medida a las organizaciones. Lo cual es muy importante, porque las organizaciones se mueven con diferentes fechas a lo largo del año dependiendo de sus necesidades, estas ventajas se adquieren al trabajar con periodos de tiempo.

Otra ventaja que también se observa al manipular las BDT es la reducción de inconsistencia en los datos, dado que se presenta una mejoría en el almacenamiento de información teniendo al tiempo como elemento adicional en los datos, dicho de otro modo, los datos tienen una marca de tiempo que los identifica de manera única dentro de la base de datos.

Se mejora la legibilidad de los datos, al presentar un comportamiento con el transcurrir del tiempo que es de gran ayuda al momento de la elaboración de alternativas para la toma de decisiones. Un ejemplo es la representación de los datos que se tienen almacenados del área de cursos, porque no es fácil entender cómo un mismo curso con el mismo número de alumnos tenga ganancias para la UNAM totalmente diferentes y ésto se debe a que el costo de los cursos varía dependiendo de la persona que toma el curso. Estas diferencias se deben a que la UNAM efectúa descuentos a estudiantes de la misma institución o a estudiantes de otras instituciones educativas, de igual modo proporciona becas a alumnos y trabajadores que pertenecen sus sindicatos (STUNAM, AAPAUNAM). En si la cantidad que se recaba de un curso no depende únicamente de la cantidad de alumnos que estén inscritos, sino del tipo de éstos y mediante el uso de las BDT queda más clara esta relación.

Una vez que se emplean las series de tiempo en las bases de datos se reduce el uso de aplicaciones realizadas en los diversos compiladores para realizar el trabajo de agrupar y manipular los datos en intervalos de tiempo ya que el manejador posee la capacidad de administrar el tiempo.

Con la propuesta del presente trabajo se pudieron observar las bondades que se obtienen al momento de trabajar con las BDT, esto se debe a que al trabajar con las series de tiempo dentro de las bases de datos se obtienen los beneficios que proporcionan las funciones y procedimientos, que permiten que los datos sean obtenidos de una manera más ordenada y eficiente, lo cual es importante, en virtud de lo que respecta a la toma de decisiones se puede obtener la información de una manera más rápida y con un alto grado de precisión. Dado que los datos al contar con marcas de tiempo, provocan que el personal que toma decisiones puede observar como cambia la información conforme transcurre el tiempo, ayudándolo así a obtener las soluciones más óptimas.

El Área de Cursos del DSC cuenta con personal que posee conocimientos matemáticos y computacionales los que les permitirán una mejor explotación de la herramienta que el presente trabajo propone y por ende las posibles soluciones a problemas que se susciten y que impliquen la toma de decisiones tendrán un mejor sustento y hará que la decisión que se tome sea la que brinda mayores beneficios.

Debo decir que al trabajar con las BDT pude mejorar mis habilidades en el área de bases de datos, así como tener un mayor conocimiento de las actividades que se desempeñan en el lugar en donde laboré es por ello que el presente trabajo benefició al área de cursos, al departamento y mejoró mis habilidades.

Glosario

AAPAUNAM	Asociación Autónoma del Personal Académico de la UNAM.
BDT	Abreviación de Bases de Datos Temporales
Chronon	Un chronon es la unidad mínima en que se puede descomponer un intervalo de tiempo.
Datawarehousing	Son aquellas bases de datos que almacenan información de manera multidimensional, en la que encontramos una gran cantidad de datos históricos que permiten el análisis a gran detalle de los datos que ahí se encuentran para diferentes actividades.
DB2	Es el nombre del manejador de bases de datos manufacturado por IBM y que se encuentra en el mercado.
DBA	Database Administrator. Administrador de la base de datos
DBMS	Database Management System, en español sería: Sistema manejador de bases de datos
DDL	Data Definition Language cuya traducción al español es: Lenguaje de definición de datos.
Determinancia	Se refiere al conocimiento del tiempo de duración de un evento.
DSC	Departamento de Servicios de Cómputo
IEEE	Institute of Electrical and Electronics Engineers, cuya traducción es: Instituto de Ingenieros Eléctricos y Electrónicos
Forma Normal Temporal	Un par (R,F) de un esquema de relación temporal R y un conjunto de dependencias temporales funcionales asociado F y es una forma temporal Boyce-Codd (TBCNF):
Gránulos	Es un número determinado de chronons consecutivos que pueden ser agrupados en segmentos de gran tamaño.
History	Una historia es una representación temporal de un "objeto del mundo real" de una base de datos. Dependiendo del objeto podemos tener atributos, historias, historias de relaciones, historia de esquema, historias de transacción, etc.
INGRESS	Es el nombre de un manejador de bases de datos que se encuentra en el mercado.
Intervalo bitemporal	Un intervalo bitemporal es una región que cuenta con lados paralelos, en dos espacios que son tiempo válido y de tiempo de transacción.. Un intervalo bitemporal puede ser representado con un conjunto no vacío de chronons bitemporales.

Bases de datos temporales, una herramienta para la toma de decisiones

Lag Operator	Es un operador de retraso que se emplea dentro de las series de tiempo y sirve para manjar el tiempo en sucesos que se encuentran antes del punto de partida de la serie.
Lifespan	El lifespan de un objeto de base de datos es el tiempo durante el cual se encuentra definido dentro de la misma base. El lifespan de tiempo válido de un objeto de una base de datos se refiere al tiempo en el que el objeto correspondiente existe en el modelo real
ORACLE	Es el nombre de un manejador de base de datos que se encuentra en el mercado.
Periodo de Tiempo	Un periodo de tiempo se encuentra definido por el tiempo transcurrido entre dos instantes. También puede ser representado por un conjunto de gránulos continuos.
RDBMS	Es la abreviación en inglés de Relational Database Management System, que en español significa Sistema Manejador de Base de Datos Relacional
Relación Bitemporal	Una relación bitemporal es una relación con un sistema que soporta tiempo válido y tiempo de transacción esta relación hereda sus propiedades de relaciones de tiempo válido y de relaciones de tiempo de transacción. No existen restricciones de cómo ambas relaciones de tiempo pueden ser incorporadas en las tuplas.
Relación Temporal Predictiva	Una esquema de relación temporal incluye el último tiempo válido para cada hecho de cualquier relación y es válido en el futuro. El concepto puede ser aplicado a instancias de una relación temporal, tuplas individuales, y actualizaciones.
Scrip	Porción de código que contiene una o varias sentencias que hacen posible la ejecución de líneas de código con un solo llamado.
Secuencia de Tiempo	Una secuencia de tiempo (del inglés TS, time sequence) es una secuencia ordenada por pares de tiempo $\langle v, t \rangle$ donde v es un dato objeto arbitral y t es un conjunto de gránulos de instancias futuras o pasadas. Una secuencia de tiempo es identificada por constante en el tiempo.
Span	Un span es un lapso de tiempo. Este lapso es una cantidad de tiempo con longitud conocida, pero no especifica el instante de tiempo de inicio ni tampoco el tiempo de término.
STUNAM	Sindicato de Trabajadores de la UNAM.

Tiempo Absoluto	El modificador absoluto indica que un tiempo válido específico está dado por una granularidad timestamp que es asociado con el hecho o suceso.
Tiempo de Transacción	El tiempo de transacción de una base de datos es el tiempo durante el cual los datos son actuales y pueden ser sustituidos. En consecuencia los tiempos de transacción no son por lo general instantes de tiempo pero tienen duración.
Tiempo Válido	El tiempo válido es un hecho en el tiempo, es decir, es un valor verdadero en la realidad modelada. Un hecho puede tener asociados un número no determinado de instantes e intervalos.
Timestamp	Un timestamp es un valor de tiempo asociado con algunos objetos, por ejemplo un valor atributo o una tupla.
Tipo de Dato temporal	El usuario define el tipo de dato temporal como una representación de tiempo especial, diseñada para cubrir las necesidades específicas del usuario.
TSBTREE	Time Split B-TREE. Se refiere al manejo de un índice que manipula dos árboles, uno de tiempo de transacción y uno de tiempo válido.
Tuplas	Es otra manera de denominar a los registros en las tablas de las bases de datos.
UNAM	Universidad Nacional Autónoma de México.
Variable Span	Un span es una variable que su duración depende del contexto y puede ser desde fracciones de segundo hasta años.

Apéndice A Elementos necesarios para la utilización de las series de tiempo

Una vez que efectuado el diseño y la implementación de la base de datos es necesario efectuar algunas modificaciones en el manejador de la base de datos Oracle 8i, estas modificaciones consisten en la instalación del cartucho que hace posible el manejo de las series de tiempo para el manejador, ésto se lleva a cabo siguiendo los siguientes pasos:

El paquete que permite el manejo de las series de tiempo se denomina "Oracle 8i Time Series". Es posible que se haya instalado en el momento de la instalación del manejador de la base de datos Oracle 8i, de no ser así se requiere que el manejador de base de datos Oracle 8i tenga instalado PL/SQL, una vez revisado ésto se puede proceder con la instalación del paquete mencionado. El procedimiento de instalación del paquete es muy sencillo y consiste en conectarse al manejador como usuario con permisos de instalación (por lo general es mediante el usuario oracle) se inserta el disco de instalación del manejador y se selecciona el elemento que se desea agregar. Una vez instalado el paquete se debe ubicar el lugar donde se encuentra almacenado el mismo.

El RDBMS(Relational Data Base Management System) Oracle 8i del DSC se encuentra instalado en el servidor Orion que trabaja en una plataforma UNIX bajo el sistema operativo Solaris 8. Una vez que se conoce la plataforma en la que se encuentra trabajando el RDBMS se puede ubicar la ruta en la que se encuentra el paquete que permite la utilización de las funciones de las series de tiempo, la ruta es la siguiente:

```
SORACLE_HOME/ord/ts/admin.
```

```
ORACLE_HOME corresponde a la siguiente ruta: /opt/home/oracle/app/oracle/product/8.1.7/
```

Una vez que se encuentra localizado el paquete de las series de tiempo, se debe correr un script para instalar los componentes necesarios para poner en operación las diferentes funciones que nos permiten efectuar las series de tiempo.

Antes de realizar la instalación del paquete "Oracle 8i Time Series" se debe correr el siguiente script que es necesario para el buen funcionamiento de las series de tiempo.

```
SVRMGRL> @ ORACLE_HOME/rdbms/admin/utrfid.sql
```

A continuación se procede con la instalación (cabe señalar que ésto lo debe realizar el administrador de la base de datos ya que se requiere de información que sólo él posee) siguiendo las siguientes sentencias:

```
SVRMGRL> connect sys/ <PASSWORD> as sysdba
SVRMGRL> @ ORACLE_HOME/ord/ts/admin/ordinst.sql
```

El script que se muestra a continuación permite la instalación de tipos de datos, paquetes y tablas metadata

```
SVRMGRL> @ ORACLE_HOME/ord/ts/admin/tainst.sql
```

Una vez que se han ejecutado los scripts antes mencionados se deben generar sinónimos públicos para los paquetes de las series de tiempo, ésto se efectúa para facilitarle las cosas a las personas que hagan uso de las diferentes herramientas que brinda el paquete de las series de tiempo. Para la elaboración de dichos sinónimos también encontramos un script que realiza tal operación.

```
SVRMGRL> @ ORACLE_HOME/ord/ts/admin/ordtsyn.sql
```

El script anterior hace posible el manejo de los siguientes sinónimos:

```
CREATE PUBLIC SYNONYM TimeSeries FOR ORDSYS.TimeSeries;
CREATE PUBLIC SYNONYM Calendar FOR ORDSYS.Calendar;
CREATE PUBLIC SYNONYM TSTools FOR ORDSYS.TSTools;
CREATE PUBLIC SYNONYM TimeScale FOR ORDSYS.TimeScale;
```

Apéndice B

Diccionario de datos

Nombre **Autor**
 HDSC Antonio Benito Patricio

Reporte

Diagram Name **HDSC**
Autor Antonio Benito Patricio
Company UNAM DSC
Version 1.0
Copyright UNAM
Copyright Year 2002

Modelo

Nombre del modelo **BDSCBI**
Plataforma de la base de datos Oracle 8i

BDSCBI

Nombre de la tabla	Tipo de la tabla	Reporte de Tablas		# Col
		Llaves primarias		
AREA	Independiente	CVE_AREA		3
CARACT_OBJETO	Dependiente	CVE_CARACT_OBJ,CVE_OBJETO,CVE_CARACT		3
CARACTERÍSTICA	Independiente	CVE_CARACT		2
CARRERA	Independiente	CVE_CARRERA		2
CONTRATACIÓN	Independiente	CVE_CONTRATACION		9
CURSO	Independiente	CVE_CURSO		11
DATOS	Independiente	CVE_PERSONA		14
DETALLE	Independiente	CLAVE		7
ESTADO	Independiente	CVE_EDO		2
HORARIO	Independiente	CVE_HORARIO		3
INSCRIPCIÓN_CURSO	Independiente	CVE_INS_CURSO		8
INSCRIPCIÓN_USUARIO	Independiente	CVE_INS_USU		9
INVENTARIO	Independiente	NUM_INV_UNAM		6
MARCA	Independiente	CVE_MARCA		2
NOMBRE_CURSO	Independiente	CVE_NOM_CURSO		4
OBJETO	Independiente	CVE_OBJETO		2
PUESTO	Independiente	CVE_PUESTO		2
TIPO_PAGO	Independiente	CVE_TIPO_PAGO		2
TIPO_USUARIO	Independiente	CVE_TIPO_USU		2
UBICACIÓN	Independiente	CVE_UBICACION		3

TESIS CON
 FALLA DE ORIGEN

Bases de datos temporales, una herramienta para la toma de decisiones

AREA

Nombre de la tabla AREA
Llaves primarias CVE_AREA
Definición Esta entidad almacena las diferentes áreas que se encuentran en el departamento de servicios de cómputo

Columnas

Nombre de la columna	Tipo de dato	Null	Definición
CVE_AREA	NUMBER(3,0)	N	Es el número mediante el cual están representadas las áreas
DESCRIPCIÓN	VARCHAR2(50)	N	Contiene el nombre completo del área representada
CAPACIDAD	NUMBER(4,0)	N	Contiene el número máximo de usuarios que soporta el servidor del área

Llaves foráneas

Tabla padre	Tabla hijo
AREA	CONTRATACIÓN
AREA	INVENTARIO
AREA	INSCRIPCION_USUARIO

Código DDL

```
CREATE TABLA AREA(  
  CVE_AREA NUMBER(3,0) NOT NULL,  
  DESCRIPCION VARCHAR2(50) NOT NULL,  
  CAPACIDAD NUMBER(4,0) NOT NULL,  
  CONSTRAINT PK3 PRIMARYKEY(CVE_AREA);
```

TESIS CON
FALLA DE ORIGEN

CARACT_OBJETO

Nombre de la tabla **CARACT_OBJETO**
 Claves primarias **CVE_CARACT_OBJ,CVE_OBJETO,CVE_CARACT**
 Definición Esta entidad contiene todas las relaciones válidas entre los objetos y sus características

Columnas

Nombre de la columna	Tipo de dato	Null	Definición
CVE_CARACT_OBJ	NUMBER(4,0)	N	Es el número mediante el cual se identifica la característica de un objeto en particular.
CVE_OBJETO	NUMBER(4,0)	N	Es la clave mediante la cual se identifica a un objeto de forma particular.
CVE_CARACT	NUMBER(6,0)	N	Es la clave con la que se identifica la característica de un objeto.

Claves foráneas

Tabla padre	Tabla hija
OBJETO	CARACT_OBJETO
CARACTERISTICA	CARACT_OBJETO
CARACT_OBJETO	DETALLE

Código DDL

```
CREATE TABLA CARACT_OBJETO(
    CVE_CARACT_OBJ NUMBER(4,0) NOT NULL,
    CVE_OBJETO NUMBER(4,0) NOT NULL,
    CVE_CARACT NUMBER(6,0) NOT NULL,
    CONSTRAINT PK18 PRIMARYKEY(CVE_CARACT_OBJ,CVE_OBJETO,CVE_CARACT);
```

**TESIS CON
FALLA DE ORIGEN**

CARACTERISTICA

Nombre de la tabla CARACTERISTICA
Llaves primarias CVE_CARACT
Definición Esta entidad contienen todas y cada una de las características que pueden presentar cada uno de los objetos.

Columnas

Nombre de la columna	Tipo de dato	Null	Definición
CVE_CARACT	NUMBER(6, 0)	N	Es un número que va a identificar las diferentes características que poseen los objetos.
DESCRIPCION	VARCHAR2(60)	N	Es la definición de la característica que presenta un objeto en particular.

Llaves foráneas

Tabla padre CARACTERISTICA **Tabla hijo** CARACT_OBJETO

Código DDL.

```
CREATE TABLA CARACTERISTICA(
  CVE_CARACT NUMBER(6,0) NOT NULL,
  DESCRIPCION VARCHAR2(60) NOT NULL,
  CONSTRAINT PK12 PRIMARYKEY(CVE_CARACT);
```

TESIS CON
 SELLA DE ORIGEN

CARRERA

Nombre de la tabla

CARRERA

Claves primarias

CVE_CARRERA

Definición

Esta entidad es un catálogo de todas las carreras que se imparten en la ENEP Acatlán.

Columnas

Nombre de la columna

Tipo de dato

Null

Definición

CVE_CARRERA

NUMBER(3,0)

N

Es el número que

DESCRIPCION

VARCHAR2(50)

N

identifica a las carreras.
Este campo contiene el
nombre completo de todas
y cada una de las carreras

Claves foráneas

Tabla padre

Tabla hijo

CARRERA

INSCRIPCION_USUARIO

Código DDL

```
CREATE TABLE CARRERA(  
  CVE_CARRERA NUMBER(3,0) NOT NULL,  
  DESCRIPCION VARCHAR2(50) NOT NULL,  
  CONSTRAINT PK2 PRIMARYKEY(CVE_CARRERA);
```

Bases de datos temporales, una herramienta para la toma de decisiones

CONTRATACION

Nombre de la tabla **CONTRATACION**
Claves primarias **CVE_CONTRATACION**
Definición En esta entidad se van a encontrar almacenados datos del personal que se encuentra trabajando dentro del Centro de Computo.

Columnas		Null	Definición
Nombre de la columna	Tipo de dato		
CVE_CONTRATACION	NUMBER(9,0)	N	En este atributo se encuentra almacenado un número que identificará de manera única cada contratación que se realiza.
CVE AREA	NUMBER(3,0)	N	Esta es una llave foránea que hace referencia al área en la cual se desempeña cada persona contratada.
CVE TIPO_PAGO	NUMBER(3,0)	N	Esta es una llave foránea que hace referencia a un catálogo en el cual se encuentran los diferentes tipos de pago que existen.
CVE_PUESTO	NUMBER(3,0)	N	Esta es una llave foránea que hace referencia a un catálogo que posee los diferentes puestos que existen dentro del Centro de Computo.
CVE_PERSONA	NUMBER(11,0)	N	Este atributo es una llave foránea que hace referencia a los datos personales de cada trabajador del Centro de Computo.
F_CONTRATACION	DATE	N	Este atributo posee la fecha en la cual se contrata a la persona.
ESTUDIOS	VARCHAR2(50)	N	Este atributo posee el grado de estudios que posee la persona que trabaja dentro del Centro de Computo.
HORAS	VARCHAR2(10)	Y	Este atributo es para tener el registro de las horas por las que se encuentra contratada la persona.
PAGO	NUMBER(6,2)	N	Aquí se encontrará almacenado el pago que se le dará al trabajador por su servicio.

Llaves foráneas

Tabla padre	Tabla hijo
AREA	CONTRATACION
TIPO_PAGO	CONTRATACION
PUESTO	CONTRATACION
DATOS	CONTRATACION
CONTRATACION	CURSO

Código DDL

```
CREATE TABLA CONTRATACION
  CVE_CONTRATACION NUMBER(9,0) NOT NULL,
  CVE AREA NUMBER(3,0) NOT NULL,
  CVE TIPO_PAGO NUMBER(3,0) NOT NULL,
  CVE PUESTO NUMBER(3,0) NOT NULL,
  CVE PERSONA NUMBER(11,0) NOT NULL,
  F_CONTRATACION DATE NOT NULL,
  ESTUDIOS VARCHAR2(50) NOT NULL,
  HORAS VARCHAR2(10),
  PAGO NUMBER(6,2) NOT NULL,
  CONSTRAINT PK23 PRIMARYKEY(CVE_CONTRATACION);
```

TESIS CON FALLA DE ORIGEN

Bases de datos temporales, una herramienta para la toma de decisiones

CURSO

Nombre de la tabla CURSO
Llaves primarias CVE_CURSO
Definición En esta entidad se van a encontrar almacenadas las características de los diversos cursos que se imparten en el Centro de Cómputo

Columnas

Nombre de la columna CVE_CURSO	Tipo de dato VARCHAR2(9)	Null	Definición
CVE_NOM_CURSO	NUMBER(3,0)	N	Es el número mediante el cual quedan representados todos y cada uno de los diversos cursos que se imparten dentro del Centro de Cómputo.
CVE_HORARIO	NUMBER(3,0)	N	Es el número que representa los nombres de todos y cada uno de los cursos impartidos.
CVE_UBICACION	NUMBER(4,0)	N	Almacena una clave que representa la hora en la cual se impartirá el curso.
CVE_CONTRATACION	NUMBER(9,0)	N	Es un número que representa el lugar en el cual se impartirá el curso.
F_INICIO	DATE	N	Contiene la clave que hace relación a la persona que impartirá el curso.
F_TERMINO	DATE	N	Es la fecha en la cual inicia el curso.
REQUISITO	VARCHAR2(30)	N	Es la fecha en que termina el curso.
COSTO_UNAM	NUMBER(6,2)	N	Es el conjunto de conocimientos que se necesitan para poder tomar el curso.
COSTO_INST_EDU	NUMBER(6,2)	N	Es el precio que debe pagar la comunidad de la universidad por el curso.
COSTO_EXT	NUMBER(6,2)	N	Es el precio que deben pagar los estudiantes de otras instituciones educativas por el curso.
		N	Es el precio que debe pagar cualquier persona que no sea estudiante.

Llaves foráneas

Tabla padre	Tabla hijo
CONTRATACION	CURSO
NOMBRE_CURSO	CURSO
HORARIO	CURSO
UBICACION	CURSO
CURSO	INSCRIPCION_CURSO

Código DDL

```

CREATE TABLA CURSOX
  CVE_CURSO VARCHAR2(9) NOT NULL,
  CVE_NOM_CURSO NUMBER(3,0) NOT NULL,
  CVE_HORARIO NUMBER(3,0) NOT NULL,
  CVE_UBICACION NUMBER(4,0) NOT NULL,
  CVE_CONTRATACION NUMBER(9,0) NOT NULL,
  F_INICIO DATE NOT NULL,
  F_TERMINO DATE NOT NULL,
  REQUISITO VARCHAR2(30) NOT NULL,
  COSTO_UNAM NUMBER(6,2) NOT NULL,
  COSTO_INST_EDU NUMBER(6,2) NOT NULL,
  COSTO_EXT NUMBER(6,2) NOT NULL,
  CONSTRAINT PK4 PRIMARYKEY(CVE_CURSO);
    
```

**TESIS CON
FALLA DE ORIGEN**

DATOS

Nombre de la tabla	DATOS			
Llaves primarias	CVE_PERSONA			
Definición	Esta entidad va a contener todos los datos personales de las personas que se encuentran involucradas de una u otra manera dentro del Centro de Cómputo.			
Columnas				
Nombre de la columna	Tipo de dato	Null	Definición	
CVE_PERSONA	NUMBER(11, 0)	N	Es el número mediante el cual se identifica de manera única a todas y cada una de las personas que están involucradas en el Centro de Cómputo.	
APELLIDO_P	VARCHAR2(25)	N	Almacena el apellido paterno de la persona.	
APELLIDO_M	VARCHAR2(25)	N	Almacena el apellido materno de la persona.	
NOMBRE	VARCHAR2(25)	N	Almacena el o los nombre(s) de la persona.	
DIRECCION	VARCHAR2(30)	N	Contiene la dirección de la persona.	
COLONIA	VARCHAR2(30)	N	Almacena la colonia donde habita la persona.	
TELEFONO_CASA	NUMBER(15, 0)	Y	Este atributo almacena el número telefónico de la casa de la persona.	
TELEFONO_TRABAJO	NUMBER(15, 0)	Y	Almacena el número telefónico del trabajo de la persona.	
TELEFONO_OTRO	NUMBER(15, 0)	Y	Este atributo almacena el número celular o radio localizador de la persona.	
CP	NUMBER(6, 0)	Y	Contiene el código postal del lugar donde habita la persona.	
RFC	VARCHAR2(20)	Y	Este atributo almacena el registro federal de contribuyentes de la persona.	
E_MAIL	VARCHAR2(50)	Y	Almacena el correo electrónico de la persona.	
SEXO	VARCHAR2(2)	N	Contiene el género de la persona.	
FOTO	LONG RAW	Y	Contiene el retrato de la persona.	

Llaves foráneas

Tabla padre	Tabla hijo
DATOS	INSCRIPCION_CURSO
DATOS	INSCRIPCION_USUARIO
DATOS	CONTRATACION

Código DDL

```

CREATE TABLA DATOS;
CVE_PERSONA NUMBER(11,0) NOT NULL,
APELLIDO_P VARCHAR2(25) NOT NULL,
APELLIDO_M VARCHAR2(25) NOT NULL,
NOMBRE VARCHAR2(25) NOT NULL,
DIRECCION VARCHAR2(30) NOT NULL,
COLONIA VARCHAR2(30) NOT NULL,
TELEFONO_CASA NUMBER(15,0),
TELEFONO_TRABAJO NUMBER(15,0),
TELEFONO_OTRO NUMBER(15,0),
CP NUMBER(6,0) DEFAULT 00000,
RFC VARCHAR2(20),
E_MAIL VARCHAR2(50),
SEXO VARCHAR2(2) NOT NULL,
FOTO LONG RAW,
CONSTRAINT PK17 PRIMARYKEY(CVE_PERSONA);
    
```

TESIS CON
FALLA DE ORIGEN

DETALLE

Nombre de la tabla DETALLE
Clave primaria CLAVE
Definición Esta entidad almacena las características particulares de todos y cada uno de los objetos que se encuentran inventariados y que se encuentran bajo custodia del Centro de Cómputo.

Columnas

Nombre de la columna CLAVE	Tipo de dato NUMBER(6,0)	Nul N	Definición
NUM_INV_UNAM	NUMBER(10,0)	N	Este atributo contiene un número que identifica de manera única a todos y cada uno de los detalles de los objetos que se encuentran en el Centro de Cómputo
CVE_CARACT_OBJ	NUMBER(4,0)	N	Este atributo es una llave foránea que hace referencia al número de inventario de la UNAM que contienen todos y cada uno de los objetos.
DESCRIPCIÓN	VARCHAR2(75)	N	Es una llave foránea que contiene el número de una característica en particular que posee un objeto determinado.
F_ALTA	DATE	N	En este atributo se almacenan los detalles particulares de cada uno de las características.
		N	En este atributo se almacena la fecha en la cual los objetos quedan bajo la custodia del Centro de Cómputo.

Claves foráneas

Tabla padre CARACT_OBJETO INVENTARIO	Tabla hijo DETALLE DETALLE
---	---

Código DDL

```
CREATE TABLA DETALLE(
CLAVE NUMBER(6,0) NOT NULL,
CVE OBJETO NUMBER(4,0) NOT NULL,
CVE_CARACT NUMBER(6,0) NOT NULL,
NUM_INV_UNAM NUMBER(10,0) NOT NULL,
CVE_CARACT_OBJ NUMBER(4,0) NOT NULL,
DESCRIPCION VARCHAR2(75) NOT NULL,
F_ALTA DATE NOT NULL,
CONSTRAINT PK19 PRIMARYKEY(CLAVE);
```

**TESIS CON
FALLA DE ORIGEN**

Bases de datos temporales, una herramienta para la toma de decisiones

ESTADO

Nombre de la tabla ESTADO
Llaves primarias CVE_EDO
Definición Este es un catálogo que contiene los diferentes estados que puede poseer un objeto mientras se encuentra dentro del Centro de Cómputo.

Columnas

Nombre de la columna	Tipo de dato	Null	Definición
CVE_EDO	NUMBER(3, 0)	N	Este atributo posee un número único que identifica el tipo de estado en el que se encuentra un objeto.
DESCRIPCION	VARCHAR2(30)	N	Este atributo contiene la descripción del estado al cual hace referencia la clave del estado.

Llaves foráneas

Tabla padre ESTADO
Tabla hija INVENTARIO

Código DDL

```
CREATE TABLA ESTADO  
CVE_EDO NUMBER(3, 0) NOT NULL,  
DESCRIPCION VARCHAR2(30) NOT NULL,  
CONSTRAINT PK20 PRIMARYKEY(CVE_EDO);
```

ANÁLISIS CON
VALOR DE ORIGEN

HORARIO

Nombre de la tabla HORARIO
 Llaves primarias CVE_HORARIO
 Definición Dentro de esta entidad se encuentran almacenados todos y cada uno de los diferentes horarios en los que se imparten cursos.

Columnas

Nombre de la columna CVE_HORARIO	Tipo de dato NUMBER(3,0)	Null N	Definición En este campo se almacena un número que identifica de manera única a los horarios de cursos.
DESCRIPCION	VARCHAR2(50)	N	En este campo se almacena el periodo que abarca un horario determinado.
TIPO	VARCHAR2(50)	N	Se almacenan datos que se refieren a que días de la semana le competen a un horario.

Llaves foráneas

Tabla padre HORARIO	Tabla hijo CURSO
Código ODL	

```

CREATE TABLA HORARIO
  CVE_HORARIO NUMBER(3,0) NOT NULL,
  DESCRIPCION VARCHAR2(50) NOT NULL,
  TIPO VARCHAR2(50) NOT NULL,
  CONSTRAINT PK24 PRIMARYKEY(CVE_HORARIO);
    
```

**TESIS CON
 FALLA DE ORIGEN**

INSCRIPCION_CURSO

Nombre de la tabla INSCRIPCION_CURSO
 Llaves primarias CVE_INS_CURSO
 Definición Dentro de esta entidad se encontrarán almacenados los diferentes asistentes a los cursos que hayan quedado inscritos.

Columnas

Nombre de la columna	Tipo de dato	Null	Definición
CVE_INS_CURSO	NUMBER(9, 0)	N	Este atributo almacena la clave de cada uno de los asistentes a cursos.
CVE_PERSONA	NUMBER(11, 0)	N	Este atributo es una llave foránea que hace referencia a los datos personales del asistente.
CVE_TIPO_USU	NUMBER(3, 0)	N	Este atributo es una llave foránea en la cual se hace referencia al tipo de usuario que toma el curso.
CVE_CURSO	VARCHAR2(9)	N	Este atributo es una llave foránea en la cual se almacena el tipo de curso que se va a tomar.
F_INSCRIPCION	DATE	N	En este atributo se almacena la fecha de inscripción de la persona que toma el curso.
PAGO	NUMBER(6, 2)	N	En este atributo queda almacenado la cantidad que el asistente paga por concepto del curso.
FOLIO_PAGO	NUMBER(10, 0)	Y	En este atributo se almacena el folio del recibo de pago que se le da al participante en la caja.
CALIFICACION	NUMBER(3, 0)	N	En este atributo se almacena la calificación obtenida por el participante.

Llaves foráneas

Tabla padre	Tabla hijo
DATOS	INSCRIPCION_CURSO
CURSO	INSCRIPCION_CURSO
TIPO_USUARIO	INSCRIPCION_CURSO

Código DDL.

```
CREATE TABLA INSCRIPCION_CURSO(
  CVE_INS_CURSO NUMBER(9,0) NOT NULL,
  CVE_PERSONA NUMBER(11,0) NOT NULL,
  CVE_TIPO_USU NUMBER(3,0) NOT NULL,
  CVE_CURSO VARCHAR2(9) NOT NULL,
  F_INSCRIPCION DATE NOT NULL,
  PAGO NUMBER(6,2) NOT NULL,
  FOLIO_PAGO NUMBER(10,0),
  CALIFICACION NUMBER(3,0) NOT NULL,
  CONSTRAINT PK21 PRIMARYKEY(CVE_INS_CURSO);
```

TESIS CON
FALLA DE ORIGEN

INSCRIPCION_USUARIO

Nombre de la tabla INSCRIPCION_USUARIO
 Llaves primarias CVE_INS_USU
 Definición Dentro de esta entidad se encontrarán almacenados los diferentes usuarios que utilizan el equipo a lo largo de un semestre.

Columnas

Nombre de la columna	Tipo de dato	Null	Definición
CVE_INS_USU	NUMBER(9,0)	N	Este atributo almacena la clave de cada uno de los usuarios del Centro de Cómputo.
CVE_PERSONA	NUMBER(11,0)	N	Este atributo es una llave foránea que hace referencia a los datos personales del asistente.
CVE_AREA	NUMBER(3,0)	N	Este atributo es una llave foránea que hace referencia al área a la que el usuario tendrá acceso.
CVE_TIPO_USU	NUMBER(3,0)	N	Este atributo es una llave foránea que hace referencia al tipo de usuario que hará uso del equipo.
CVE_CARRERA	NUMBER(3,0)	N	Este atributo es una llave foránea que hace referencia a un catálogo donde se encuentran todas las carreras.
F_INSCRIPCION	DATE	N	En este atributo se almacenará la fecha de inscripción de la persona que usa el equipo durante un semestre.
CUOTA	NUMBER(3,0)	N	Este atributo contiene la cantidad de horas a las que tiene derecho un usuario a usar un determinado equipo.
PAGO	NUMBER(6,2)	N	En este atributo queda almacenada la cantidad que el usuario paga por concepto del uso del equipo.
FOLIO_PAGO	NUMBER(10,0)	Y	En este atributo se almacena el folio del talón de pago que se le da al usuario en la caja.

Llaves foráneas

Tabla padre	Tabla hijo
DATOS	INSCRIPCION_USUARIO
TIPO_USUARIO	INSCRIPCION_USUARIO
AREA	INSCRIPCION_USUARIO
CARRERA	INSCRIPCION_USUARIO

Código DDL

```
CREATE TABLE INSCRIPCION_USUARIO(
  CVE_INS_USU NUMBER(9,0) NOT NULL,
  CVE_PERSONA NUMBER(11,0) NOT NULL,
  CVE_AREA NUMBER(3,0) NOT NULL,
  CVE_TIPO_USU NUMBER(3,0) NOT NULL,
  CVE_CARRERA NUMBER(3,0) NOT NULL,
  F_INSCRIPCION DATE NOT NULL,
  CUOTA NUMBER(3,0) NOT NULL,
  PAGO NUMBER(6,2) NOT NULL,
  FOLIO_PAGO NUMBER(10,0),
  CONSTRAINT PK22 PRIMARYKEY(CVE_INS_USU);
```

TESIS CON
 FOLIO DE PAGOS

INVENTARIO

Nombre de la tabla
Llaves primarias
Definición

INVENTARIO
NUM_INV_UNAM
En esta entidad se encuentran almacenadas todas las características en común que poseen todos y cada uno de los objetos inventariados.

Columnas

Nombre de la columna	Tipo de dato	Null	Definición
NUM_INV_UNAM	NUMBER(10, 0)	N	En este atributo se almacenan los números de inventario que tienen los objetos.
CVE_AREA	NUMBER(3, 0)	N	Este atributo es una llave foránea que hace referencia al área a la que pertenecen los objetos.
CVE_MARCA	NUMBER(4, 0)	N	Este atributo es una llave foránea que hace referencia a la marca que poseen los objetos.
CVE_EDO	NUMBER(3, 0)	N	Este atributo es una llave foránea que hace referencia al estado actual de los objetos.
CVE_UBICACION	NUMBER(4, 0)	N	Este atributo contiene el lugar en donde se encuentran los objetos físicamente.
NO_SERIE	NUMBER(38, 0)	N	En este atributo se almacenan los números de serie de todos y cada uno de los objetos.

Llaves foráneas

Tabla padre	Tabla hijo
MARCA	INVENTARIO
ESTADO	INVENTARIO
UBICACIÓN	INVENTARIO
AREA	INVENTARIO
INVENTARIO	DETALLE

Código DDL

```
CREATE TABLA INVENTARIO(
  NUM_INV_UNAM NUMBER(10,0) NOT NULL,
  CVE_AREA NUMBER(3,0) NOT NULL,
  CVE_MARCA NUMBER(4,0) NOT NULL,
  CVE_EDO NUMBER(3,0) NOT NULL,
  CVE_UBICACION NUMBER(4,0) NOT NULL,
  NO_SERIE NUMBER(38,0) NOT NULL,
  CONSTRAINT PK10 PRIMARYKEY(NUM_INV_UNAM);
```

TESIS CON
FALLA DE ORIGEN

Bases de datos temporales, una herramienta para la toma de decisiones

MARCA

Nombre de la tabla MARCA
Claves primarias CVE_MARCA
Definición Dentro de esta entidad se encuentran almacenadas las diferentes marcas de los objetos que se encuentran dentro del Centro de Cómputo.

Columnas

Nombre de la columna	Tipo de dato	Null	Definición
CVE_MARCA	NUMBER(4,0)	N	Este atributo es un número único mediante el cual queda representada una marca.
DESCRIPCION	VARCHAR2(30)	N	En este atributo se encuentra almacenado el nombre completo de una marca.

Claves foráneas

Tabla padre MARCA
Tabla hijo INVENTARIO

Código DDL

```
CREATE TABLA MARCA(  
  CVE_MARCA NUMBER(4,0) NOT NULL,  
  DESCRIPCION VARCHAR2(30) NOT NULL,  
  CONSTRAINT PK11 PRIMARYKEY(CVE_MARCA);
```

TESIS CON
FALLA DE ORIGEN

NOMBRE_CURSO

Nombre de la tabla NOMBRE_CURSO
 Llaves primarias CVE_NOM_CURSO
 Definición En esta entidad se encuentran almacenados los nombres de todos y cada uno de los cursos identificados por una clave.

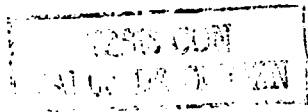
Columnas

Nombre de la columna	Tipo de dato	Null	Definición
CVE_NOM_CURSO	NUMBER(3, 0)	N	Este atributo almacena la clave de cada uno de los cursos que se imparten dentro del Centro de Cómputo.
DESCRIPCIÓN	VARCHAR2(60)	N	Este atributo almacena el nombre completo de los cursos que se dan en el Centro de Cómputo.
TIPO	VARCHAR2(40)	N	Este atributo contiene la categoría a la que pertenece el curso.
DURACIÓN	NUMBER(3, 0)	N	Este atributo almacena la cantidad de horas que dura el curso.

Llaves foráneas

Tabla padre NOMBRE_CURSO Tabla hijo CURSO
 Código DDI.

```
CREATE TABLA NOMBRE_CURSO
  CVE_NOM_CURSO NUMBER(3,0) NOT NULL,
  DESCRIPCIÓN VARCHAR2(60) NOT NULL,
  TIPO VARCHAR2(40) NOT NULL,
  DURACIÓN NUMBER(3, 0) NOT NULL,
  CONSTRAINT PK7 PRIMARYKEY(CVE_NOM_CURSO);
```



OBJETO

Nombre de la tabla OBJETO
Llaves primarias CVE_OBJETO
Definición En esta entidad es un catálogo que almacena todos y cada uno de los diferentes objetos ubicados dentro del Centro de Cómputo

Columnas

Nombre de la columna	Tipo de dato	Null	Definición
CVE_OBJETO	NUMBER(4, 0)	N	Este atributo es un número único mediante el cual queda representado un objeto.
DESCRIPCION	VARCHAR2(40)	N	Este atributo almacena el nombre completo de todos y cada uno de los objetos existentes en el Centro de Cómputo.

Llaves foráneas

Tabla padre	Tabla hijo
OBJETO	CARACT_OBJETO

Código DDL

```
CREATE TABLA OBJETO
  CVE_OBJETO NUMBER(4, 0) NOT NULL,
  DESCRIPCION VARCHAR2(40) NOT NULL,
  CONSTRAINT PK16 PRIMARYKEY(CVE_OBJETO);
```

TESE CO 1
 FALLA DE ORIGEN

PUESTO

Nombre de la tabla PUESTO
 Llaves primarias CVE_PUESTO
 Definición Esta entidad es un catálogo que almacena todos los posibles cargos que puede llegar a ocupar el personal que labora en el Centro de Cómputo.

Columnas

Nombre de la columna	Tipo de dato	Null	Definición
CVE_PUESTO	NUMBER(3,0)	N	Este atributo es un número único que representa un puesto del Departamento de Servicios de Cómputo.
DESCRIPCION	VARCHAR2(50)	Y	Este atributo contiene el nombre completo de un puesto que existe en la estructura jerárquica del Centro de Cómputo.

Llaves foráneas

Tabla padre	Tabla hijo
PUESTO	CONTRATACIÓN

Código DDL

```
CREATE TABLA PUESTO(
    CVE_PUESTO NUMBER(3,0) NOT NULL,
    DESCRIPCION VARCHAR2(50),
    CONSTRAINT PK26 PRIMARYKEY(CVE_PUESTO);
```

TESIS CON FALLA DE ORIGEN

TIPO_PAGO

Nombre de la tabla

TIPO_PAGO

Llaves primarias

CVE_TIPO_PAGO

Definición

Esta entidad es un catálogo que contiene los diferentes tipos de pago que emplea el Centro de Cómputo para saldar sus cuentas con los diferentes trabajadores.

Columnas

Nombre de la columna

Tipo de dato

Null

Definición

CVE_TIPO_PAGO

NUMBER(3,0)

N

Este atributo es un número único mediante el cual quedará representada la forma de pago que realiza el Centro de Cómputo con sus empleados.

DESCRIPCIÓN

VARCHAR2(40)

Y

Este atributo contiene el nombre completo de la forma en la cual se le paga a los empleados por sus servicios.

Llaves foráneas

Tabla padre

Tabla hijo

TIPO_PAGO

CONTRATACIÓN

Código DDL

```
CREATE TABLA TIPO_PAGO
  CVE_TIPO_PAGO NUMBER(3,0) NOT NULL,
  DESCRIPCIÓN VARCHAR2(40),
  CONSTRAINT PK25 PRIMARYKEY(CVE_TIPO_PAGO);
```

TESIS CON
FALLA DE ORIGEN

TIPO_USUARIO

Nombre de la tabla

TIPO_USUARIO

Claves primarias

CVE_TIPO_USU

Definición

Esta entidad es un catálogo en la cual se encuentran las diferentes clasificaciones de las personas que hacen uso del equipo con que cuenta el Centro de Cómputo.

Columnas

Nombre de la columna

Tipo de dato

Null

Definición

CVE_TIPO_USU

NUMBER(3,0)

N

Este atributo es un número único mediante el cual quedará representada una clasificación de usuario.

DESCRIPCION

VARCHAR2(40)

N

Este atributo contiene la clasificación de los usuarios que hacen uso del equipo.

Claves foráneas

Tabla padre

Tabla hijo

TIPO_USUARIO

INSCRIPCIÓN_CURSO

TIPO_USUARIO

INSCRIPCIÓN_USUARIO

Código DDL

```
CREATE TABLE TIPO_USUARIO(
  CVE_TIPO_USU NUMBER(3,0) NOT NULL,
  DESCRIPCION VARCHAR2(40) NOT NULL,
  CONSTRAINT PK6 PRIMARYKEY(CVE_TIPO_USU);
```

TESIS CON
VALLA DE ORIGEN

UBICACION

Nombre de la tabla UBICACION
 Claves primarias CVE_UBICACION
 Definición En esta entidad se encuentran almacenados los datos correspondientes a las diferentes áreas en que se encuentra dividido físicamente el Departamento de Servicios de Computo.

Columnas

Nombre de la columna	Tipo de dato	Null	Definición
CVE_UBICACION	NUMBER(4,0)	N	Este campo almacena un número único que representa al área.
DESCRIPCION	VARCHAR2(50)	N	Este campo contiene el nombre con el que se denomina al área física.
CAPACIDAD	NUMBER(3,0)	Y	Este campo almacena el número de computadores que se encuentran en el área física.

Claves foráneas

Tabla padre	Tabla hijo
UBICACION	INVENTARIO
UBICACION	CURSO

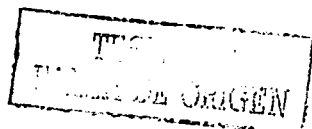
Código DDL

```
CREATE TABLE UBICACION(
    CVE_UBICACION NUMBER(4,0) NOT NULL,
    DESCRIPCION VARCHAR2(50) NOT NULL,
    CAPACIDAD NUMBER(3,0),
    CONSTRAINT PK27 PRIMARYKEY(CVE_UBICACION);
```

TESIS CON
 FALLA DE ORIGEN

Referencias Bibliográficas

- [AIX93] ADORACIÓN, de Miguel Castaño/ Piattini, Mario Gerardo. Concepción y Diseño de Bases de Datos: Del modelo E/R al Modelo Relacional, Ed. Addison -Wesley Iberoamericana. Estados Unidos 1993. 989p. ISBN: 0-201-64497-5
- [BJO97] BJORN, Skjellaug. Temporal Data: Time and Relational Databases, Research report 246, Ed. Institutt for informatikk, Universitetet i Oslo, Noruega, 1997. 45p. ISBN 82-7368-161-0.
- [DAT01] DATE, Chris J. Sistemas Mecedores de Bases de Datos, Ed Prentice Hall, Séptima Edición, México 2001. 960p. ISBN : 968-444-419-2
- [HAM94] HAMILTON, James D. Time Series Analysis, Ed. Princeton University Press, Estados Unidos, 1994. 799p.
- [HAR98] HARVEY, Andrew C. Times Series Models, Segunda Edición, Gran Bretaña, 1998. 308p. ISBN 0-262-08224-1.
- [HUR79] HURLEY, Dennis Phee. Introducción a las Series de Tiempo (Serie Matemática en la Provincia), Ed. UNAM Comunicaciones Internas, México 1979. 34p.
- [LOM92] LOMET, David, Salzber Betty. Rollback Databases, Ed DIGITAL-Cambridge Research Laboratories, Estados Unidos. 37p.
- [MAT95] MATHUR, Kamlesh. Investigación de Operaciones. El Arte de la Toma de Decisiones. Ed. Prentice Hall Hispanoamericana, México 1995. 977p. ISBN 968-880-698-6
- [MAT98] MATUS, Abel Castillejos. Las Series de Tiempo en las Tecnologías de los Sistemas de Bases de Datos y los Sistemas de Minería de Datos . Tesis para obtener el Grado de Doctorado en Ciencias de la Computación. México 1998. 21 p.
- [NAS95] NASCIMENTO, Mario A. Indexing Bitemporal Databases Via Trees with Shared Leaves - The SLT Approach, Department of Computer Science and Engineering Southern Methodist University, Estados Unidos. 26p.
- [OPH98] OPIER, Etzion, /Sushil Jajodia, Suryanarayana Sripada. Temporal Databases: Research and Practice, Ed. Springer-Verlag Berlin Heidelberg, Alemania, 1998. 428p ISBN 302-9743
- [ORA99] Oracle corporation. Oracle 8i Time Series Ed. 1999, 476p.
- [REN96] RENDER, Barry, Tr. Juan Puron Mier y Teran. Principios de Administración de Operaciones. Ed. Pearson Education, México 1996. 624p. ISBN 0-205-15644-4
- [RIC97] LEVIN, Richard I/ KIRKPATRICK, Charles A. Quantitative Approches to Management, Ed. Mc Graw Hill, Cuarta edición, Estados Unidos. 465p. ISBN 0-07-037423-6
- [SIL.98] SILBERSHATZ, Abraham/ KORTH, Henry F. Fundamentos de Bases de datos, Ed. Mc Graw Hill, Tercera Edición, España 1998. 641p. ISBN:84-481-20-21-3



- [ULI.97] ULLMAN, Jeffrey D. A First course in Database Systems, Ed. Prentice Hall, Estados Unidos 1997, 470p. ISBN 0-13-861337-0
- [WEI94] WEI, William W. 'S. Time Series Analysis Univariate and Multivariate Methods, Ed. Addison Wesley Publishing Company, Estados Unidos 1994, 478p. ISBN 0-201-15911-2.

<http://www.ifi.uio.no/~bjornsk/>
<http://greco.dit.upm.es/~tomas/cursos/bd/18dist/index.htm>
<http://sd-cenidet.com.mx/>
http://aluna.udca.edu.co/~enorena/guja_de_estudio/distribuidos.html
<http://serpiente.dgsca.unam.mx/rectoria/hm/mexico.html>
<http://computer.org/proceedings/wpdfrs/8096/80960149abs.htm>
<http://computer.org/ikde/ik1995/k0513abs.htm>
<http://computer.org/proceedings/time/0756/07560191abs.htm>
<http://computer.org/proceedings/dexa/0281/02810475abs.htm>
<http://computer.org/proceedings/dexa/8147/81470474abs.htm>
<http://computer.org/ikde/ik1997/k0464abs.htm>
<http://computer.org/proceedings/time/0756/07560059abs.htm>
<http://computer.org/Proceedings/hicss/0493/04932/04932007abs.htm>
<http://computer.org/ikde/ik1999/k0036abs.htm>
<http://computer.org/ikde/ik1998/k0021abs.htm>
<http://computer.org/proceedings/dexa/0281/02810441abs.htm>
<http://computer.org/Proceedings/secc/0296/02960170abs.htm>
<http://www.ai.es.uni-dortmund.de/EVENTS/ICA12001-L.TSD/>
<http://citeseer.nj.nec.com/slivinskas99query.html>
<http://citeseer.nj.nec.com/eachedpage/332684/2>
<http://citeseer.nj.nec.com/eachedpage/332684/1>
<http://www.es.auc.dk/~esj/>
<http://www.es.auc.dk/~esj/glossary/index.html>

TESIS CON
FALLA DE ORIGEN